



digital

**VAX/VMS
Operator's Guide**

Order No. AA-D025A-TE

VAX11



August 1978

This manual contains descriptions of operating procedures and commands that a system operator uses to keep the VAX/VMS operating system running smoothly. This manual also contains the messages produced by the Operator's Communication Process (OPCOM).

VAX/VMS Operator's Guide

Order No. AA-D025A-TE

16

SUPERSESSION/UPDATE INFORMATION:	This is a new document for this release.
OPERATING SYSTEM AND VERSION:	VAX/VMS V01
SOFTWARE VERSION:	VAX/VMS V01

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

digital equipment corporation • maynard, massachusetts

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

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

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

Copyright © 1978 by Digital Equipment Corporation

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECTape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10
VAX	VMS	SBI
DECnet	IAS	

CONTENTS

		Page
PREFACE		vi
CHAPTER 1	INTRODUCTION	1-1
1.1	DUTIES OF THE OPERATOR	1-1
1.2	OPERATOR PRIVILEGES	1-2
1.3	THE OPERATOR'S TERMINAL	1-3
CHAPTER 2	VAX/VMS OPERATING PROCEDURES	2-1
2.1	HANDLING FILES AND VOLUMES	2-2
2.1.1	Operator Assistance in Handling Disks and Single-Volume Tapes	2-4
2.1.2	Operator Assistance in Handling Multivolume Tapes	2-6
2.1.3	Operator Assistance in Creating a Multivolume Tape Set	2-7
2.1.4	Mounting Public Disks	2-9
2.1.5	Backing Up the System Disk to Disk	2-10
2.1.6	Backing Up the System Disk to Tape	2-14
2.1.7	Restoring of the System Disk from Tape	2-16
2.1.8	Backing Up a Nonsystem Disk to Disk	2-18
2.1.9	Selective Back-up of Files	2-21
2.1.10	Backing Up a Floppy Diskette to Disk	2-23
2.1.11	Restoring a Floppy Diskette from Disk	2-25
2.2	CONTROLLING BATCH AND PRINT QUEUES	2-28
2.2.1	Merging Line Printer Queues	2-29
2.2.2	Preventing Loss of Data When Line Printer Runs out of Paper	2-30
2.2.3	Terminating a Batch Job	2-31
2.2.4	Terminating a Print Job	2-32
2.3	TENDING THE CARD READER	2-33
2.3.1	Operator Assistance in Processing a Card Deck	2-35
2.4	MONITORING SYSTEM STATUS AND ACTIVITY	2-36
2.4.1	Printing the Error Log File	2-38
2.4.2	Printing the Operator's Log File	2-40
2.5	SHUTTING DOWN AND RESTARTING THE VAX/VMS OPERATING SYSTEM	2-41
2.5.1	Orderly Shutdown of the System	2-43
2.5.2	Emergency Shutdown of the System	2-46
2.5.3	Restarting the System	2-47
CHAPTER 3	COMMANDS USED BY OPERATORS	3-1
3.1	SUMMARY OF COMMANDS USED BY OPERATORS	3-1
3.2	ASSIGN/MERGE	3-4
3.3	ASSIGN/QUEUE	3-5
3.4	DEASSIGN/QUEUE	3-7
3.5	DELETE/QUEUE	3-8

CONTENTS (Cont.)

	Page	
3.6	INITIALIZE/QUEUE	3-9
3.7	MCR BAD	3-12
3.8	MCR DSC	3-14
3.9	MCR VFY	3-18
3.10	REPLY	3-21
3.11	RUN SYS\$SYSTEM:INSTALL	3-27
3.12	RUN SYS\$SYSTEM:SYE	3-31
3.13	SET	3-33
3.13.1	SET ACCOUNTING	3-34
3.13.2	SET DEVICE	3-36
3.13.3	SET LOGINS	3-38
3.13.4	SET PRINTER	3-39
3.13.5	SET PROTECTION/DEVICE	3-41
3.13.6	SET TERMINAL/PERMANENT	3-44
3.13.7	SET UIC	3-51
3.14	SHOW TERMINAL/PERMANENT	3-52
3.15	START/QUEUE	3-53
3.16	STOP/ABORT	3-56
3.17	STOP/ENTRY	3-57
3.18	STOP/QUEUE	3-58
3.19	STOP/REQUEUE	3-60
CHAPTER 4	THE OPERATOR'S LOG FILE	4-1
4.1	OVERVIEW OF THE OPERATOR'S LOG FILE	4-1
4.2	INITIALIZATION MESSAGES	4-1
4.3	DEVICE STATUS MESSAGES	4-2
4.4	TERMINAL ENABLE AND DISABLE MESSAGES	4-3
4.5	TIME STAMP MESSAGES	4-4
4.6	USER REQUEST AND OPERATOR REPLY MESSAGES	4-4
APPENDIX A	QUALIFIERS APPENDED TO THE MCR VFY COMMAND	A-1
A.1	VALIDITY CHECK	A-1
A.1.1	File Error Reporting	A-1
A.1.2	Files Marked-for-Delete	A-3
A.1.3	Deletion of Multiply-Allocated Blocks	A-3
A.1.4	Elimination of Free Blocks	A-4
A.1.5	Recovering Lost Blocks	A-4
A.2	FILE QUALIFIERS	A-4
A.2.1	Delete Qualifier (/DE)	A-4
A.2.2	Free Qualifier (/FR)	A-4
A.2.3	List Qualifier (/LI)	A-5
A.2.4	Lost Qualifier (/LO)	A-5
A.2.5	Read Check Qualifier (/RC)	A-5
A.2.6	Rebuild Qualifier (/RE)	A-6
A.2.7	Update Qualifier (/UP)	A-6
A.2.8	Validity Check (Default)	A-7
APPENDIX B	ERROR MESSAGES PRODUCED BY THE BAD BLOCK UTILITY (BAD)	B-1
VAX/VMS OPERATOR'S NOTES		
INDEX		Index-1

CONTENTS (Cont.)

Page

FIGURES

FIGURE	4-1	The Operator's Log File (OPERATOR.LOG)	4-2
	A-1	VFY Listing Sample Using the /LI Qualifier	A-5

TABLES

TABLE	1-1	Operator Privileges	1-2
	2-1	Error Causes and Remedies	2-2
	2-2	Error Alarm Causes and Remedies	2-34
	3-1	Commands Used by Operators	3-1
	3-2	Output Qualifiers	3-15
	3-3	Input Qualifiers	3-16
	3-4	DSC-Supported Devices	3-17
	3-5	File Qualifiers	3-19
	3-6	Keywords Appended to the /DISABLE and /ENABLE Qualifiers	3-35
	3-7	Default Characteristics for Terminals	3-45

PREFACE

MANUAL OBJECTIVES

The VAX/VMS Operator's Guide contains information that is useful in the day-to-day operation of the VAX/VMS operating system. Its objectives are twofold:

1. To give the reader an understanding of the duties of the VAX/VMS system operator
2. To gather in one volume the procedures, commands, and system messages that pertain specifically to the operation of the VAX/VMS system

INTENDED AUDIENCE

This guide is intended for operators of the VAX/VMS system, for persons who want to become operators of the system, and for persons who must periodically perform an operator's functions on the system.

STRUCTURE OF THIS DOCUMENT

This operator's guide consists of four chapters, as follows:

- Chapter 1 briefly describes the duties of the system operator and the relationship between the system and the operator.
- Chapter 2 presents the commonly used operating procedures.
- Chapter 3 describes the commands commonly used by operators. The commands in this chapter are not included in the VAX/VMS Command Language User's Guide.
- Chapter 4 summarizes the messages that are related to system operations.
- In addition, a section has been included in the back of this manual for operator's notes. There, the operator can write down any procedure not documented in Chapter 2.

ASSOCIATED DOCUMENTS

The VAX-11 Information Directory provides a complete list of all VAX-11 documents, including a brief description of each.

For general background information about the VAX/VMS system, see the VAX/VMS Primer and VAX/VMS Summary Description.

The following VAX/VMS documents may also be useful:

- VAX/VMS Command Language User's Guide
- VAX-11/RSX-11M User's Guide
- VAX/VMS Release Notes
- VAX-11 Software Installation Guide
- VAX/VMS System Manager's Guide
- VAX/VMS System Messages and Recovery Procedures Manual
- VAX-11 Disk Save and Compress User's Guide

For hardware operating instructions, refer to the VAX-11/780 Hardware User's Guide (Order Number EK-UG780-UG-PRE).

CONVENTIONS USED IN THIS DOCUMENT

CTRL/x	The expression "CTRL/x" indicates that the key labeled CTRL and another key should be pressed simultaneously. For example, CTRL/Y, CTRL/C, CTRL/Z.
\$ INITIALIZE/QUEUE \$_Queue: LPA0:	In all examples, the output lines and prompting characters printed or displayed by the system are in black ink; all user-entered commands are in red ink.
\$ START/QUEUE LPB0: . : .	Vertical ellipses indicate that not all the data the system would display is shown.
Keyword,...	Horizontal ellipses indicate that additional parameters, values, or information can be entered.
[queue-name]	Square brackets mean that the enclosed expression is optional. (Square brackets are, however, required syntax in the SET UIC command, the SET DEFAULT command, and the CREATE/DIRECTORY command.)

CHAPTER 1
INTRODUCTION

For the most part, the VAX/VMS system runs without operator intervention. However, in many installations the system may function more smoothly when attended by one or more operators.

1.1 DUTIES OF THE OPERATOR

The VAX/VMS system operator is responsible for keeping the system running smoothly and providing the best possible service to system users. Typically, therefore, the system operator performs the following tasks:

- Initializing and mounting system volumes
- Backing up critical public files and volumes
- Carrying out user requests
- Sending messages to specified users
- Broadcasting messages to all users
- Controlling batch processing
- Controlling printer queues
- Tending line printers
- Tending card readers
- Physically mounting magnetic tapes and disks at the request of the users who own them
- Monitoring the system and noting emergencies and responding to them
- Printing copies of the operator's log file and the error log file

Chapters 2 and 3 of this guide describe the procedures and commands used in performing these tasks.

To carry out their tasks, operators of the VAX/VMS system may have to:

1. Establish working relationships with other users of the system
2. Interact with the VAX/VMS operating system

INTRODUCTION

3. Interact directly with the VAX-11/780 hardware on which the operating system runs

The relationships of the VAX/VMS operator with users and their programs are based on two-way communication: messages pass between the operator and other users. A record of these messages is displayed or printed on the operator's terminal; the messages are also entered in the operator's log file for later reference.

The interactions between the operator and the VAX/VMS operating system are based on the operator's ability to use the entire set of VAX/VMS commands and on the operator's understanding of the system messages displayed or printed on the operator's terminal. The system messages are entered in the operator's log file along with the record of the messages that pass between the operator and other users.

The direct interactions between the operator and the VAX-11/780 hardware are based mainly on the operator's ability to operate and maintain the peripheral devices that the system supports. For detailed instructions on operating and maintaining the VAX-11/780 peripheral devices, see the VAX-11/780 Hardware User's Guide.

1.2 OPERATOR PRIVILEGES

In many ways, operators of the VAX/VMS system are no different from other users of the system. However, operators are granted privileges to perform functions that are denied to most other users. These privileges always include the privilege of performing operator functions (the OPER privilege). Table 1-1 summarizes the privileges an operator needs to perform the procedures and commands documented in this manual. The system manager is responsible for granting these and other privileges to the operator.

Table 1-1
Operator Privileges

Privilege	Service
CMKRNL	To execute Change Mode to Kernel system service
GROUP	To issue process control system services within a group
GRPNAM	To insert group logical names in the name table
LOG_IO	To issue logical I/O requests
OPER	To execute operator functions
PHY_IO	To issue physical I/O requests
PRMCEB	To create or delete permanent common event flag clusters
PRMMBX	To create permanent mailboxes
SYSNAM	To insert system logical names in the name table
TMPMBX	To create temporary mailboxes
VOLPRO	To override volume protection

INTRODUCTION

The VAX/VMS System Manager's Guide fully describes these privileges and how to set up accounts for the operators.

1.3 THE OPERATOR'S TERMINAL

Operators of the VAX/VMS system can perform many of their duties from ordinary users' terminals. However, one of the chief operator's functions, communicating with other users, must be performed at terminals that have been defined as operators' terminals. Operators themselves can set up operators' terminals by use of the privileged command `REPLY/ENABLE` (described in Chapter 3).

Still other operating functions can be performed only from the VAX-11/780 system console. These functions include booting the system, halting the system, and examining physical memory.

CHAPTER 2
VAX/VMS OPERATING PROCEDURES

This chapter describes the procedures commonly used by operators of the VAX/VMS operating system. The procedures documented cover the following subjects:

- Handling files and volumes
- Controlling batch queues and print queues
- Tending the card reader
- Monitoring system status and activity
- Shutting down and restarting the VAX/VMS operating system

All these procedures are skeleton models and should be tailored by the system manager or the system operator to fit the needs of the installation. All the commands used in the procedures are described in Chapter 3 or in the VAX/VMS Command Language User's Guide.

VAX/VMS OPERATING PROCEDURES

2.1 HANDLING FILES AND VOLUMES

At some installations, the attendance of a system operator is required for users to gain access to or to create files. Tape and disk volumes must be properly mounted on devices and the files contained on these volumes must be regularly backed up. The procedures described on the following pages explain how to mount and back up volumes.

Mounting Volumes: Mounting a volume involves placing the volume on the specified device and performing the necessary start-up procedure. On disk drives, the start-up procedure requires pressing the START or RUN button; on tape drives, it requires pressing the LOAD button.

Generally, when the operator receives a request to mount a volume on a device, the request is sent by a user. However, sometimes the system sends a request to mount a volume. This occurs when a user is accessing a tape and the tape reaches end-of-tape. If the tape is a new tape, the operator must initialize it by use of the /INITIALIZE keyword. For example:

```
$ REPLY/TO=107 "PAD009/INITIALIZE"
```

The system performs a protection and expiration check before initializing the volume. If, however, the tape has been processed by a verifying machine, the operator must specify the /BLANK keyword to prevent the processing of protection information on the tape. For example:

```
$ REPLY/TO=1560 "KKG015/BLANK"
```

The system initializes the tape and ignores the protection information already there.

After the operator has mounted the next volume, a message may be displayed on the operator's terminal indicating that the volume is not properly mounted. Table 2-1 lists the probable causes for this condition and the operator action required to remedy the situation.

Table 2-1
Error Causes and Remedies

Cause	Corrective Action
The operator mounted the wrong volume on the device	Remove the volume and replace it with the correct one
The ONLINE (for tapes) or the READY (for disks) indicator is not lit	Press the LOAD button Press the LOAD button followed by the ONLINE button Press the RUN or START button
The WRITE-LOCK indicator is illuminated, but the user wants to write to tape	Insert the write-ring into the tape Push the WRITE-LOCK switch to the OFF position

(continued on next page)

VAX/VMS OPERATING PROCEDURES

Table 2-1 (Cont.)
Error Causes and Remedies

Cause	Corrective Action
<p>The operator did not initialize the tape volume</p> <p>The operator mounted a tape processed by a verifying machine, but did not specify the /BLANK keyword</p>	<p>Reissue the REPLY command including the /INITIALIZE keyword</p> <p>Reissue the REPLY command including the /BLANK keyword</p>

Backing Up Volumes: Backing up a volume means copying the contents of the volume to another medium, for example, another disk or tape. The operator backs up volumes as a precautionary measure to prevent the loss or destruction of valuable information.

Three commands enable the operator to copy data from one volume to another: the Disk Save and Compress (DSC) command, the COPY command, and the File Transfer (FLX) command. The operator specifies one of these commands depending upon the type of back-up to be performed.

The operator uses the DSC command to back up entire volumes; this command automatically copies all the blocks allocated to active files onto the back-up, or scratch volume. The DSC command has two forms: DSC1 and DSC2. The DSC1 command copies files contained on Files-11 Structure Level 1 volumes and the DSC2 command copies files contained on Files-11 Structure Level 2 volumes.

There is also an entirely self-contained stand-alone version of the DSC2 command. This version enables operators with one-disk systems to back up and restore their system disks. The procedures for backing up and restoring volumes on a one disk system are documented in Sections 2.1.5 and 2.1.6.

For more information on the DSC command, see the VAX-11 Disk Save and Compress User's Guide.

The operator uses the COPY command to copy selected files from one volume onto another volume. This command is fully described in the VAX/VMS Command Language User's Guide.

Distributed with the VAX/VMS software is a command procedure file which copies and restores the contents of RT-11 formatted floppy diskettes. This command file uses the FLX command to perform the copy operation. Invoking and using this command file is described in Sections 2.1.10 and 2.1.11.

Typically, the system manager is responsible for setting up a schedule for backing up volumes. This schedule depends on the frequency at which the volumes are updated and the importance of the files contained on these volumes. Then the system operator is responsible for putting this schedule into effect.

2.1.1 Operator Assistance in Handling Disks and Single-Volume Tapes

This procedure describes how to mount the specified disk or tape on a device. When a user requests a specific disk or tape to be mounted on a device, the following type of message appears on the operator's terminal:

```
OPCOM, hh:mm:ss.cc, process-name ACCNT=[blank] REPLY-ID=identification-number
OPCOM, terminal-name:, "mount-message"
```

The process name identifies the user. This name is either the name under which the user logged into the system or the name of the terminal at which the user is logged in. The process name is the name of a terminal when the user logs into the system on a second terminal before logging out on the first terminal.

Procedure

1. Locate the volume and place it on the specified device.
2. Ready the device by pressing either the LOAD button if it is a magnetic tape drive, or the RUN/STOP button or the START/STOP switch if it is a disk drive.
3. Issue the following command indicating that the request has been fulfilled:

```
$ REPLY/TO=identification-number ["message-text"]
```

If the user did not request a specific device, the operator should respond with a message indicating where the tape or disk has been mounted. For example:

```
$ REPLY/TO=18 "MYVOL MOUNTED ON DMA3:"
```

For more information on the REPLY command, see Chapter 3.

4. If the user requests an unattainable disk or tape, or unavailable drive, the operator can respond using the following command:

```
$ REPLY/ABORT=identification-number ["message-text"]
```

5. If the operator needs time to locate the disk or tape the user has asked for, the operator can respond with:

```
$ REPLY/PENDING=identification-number ["message-text"]
```

Examples

1.


```
OPCOM, 12:54:57.07, REYNOLDS ACCNT= REPLY-ID=29
OPCOM, _TTF1:, "MOUNT SYSFILE ON DBA1:"

$ REPLY/TO=29 "SYSFILE MOUNTED ON DBA1:"
$
OPCOM, 01:03:43.90, REQUEST COMPLETED, OPERATOR=_OPA0:, ID=29
```

In the above example, OPCOM alerts the operator to place the disk volume named SYSFILE on DBA1. The operator locates the volume, places it on the device, readies the device, then informs the user that the volume is ready for use.

VAX/VMS OPERATING PROCEDURES

```
2.      OPCOM, 09:45:23.74, LEBB      ACCNT=      REPLY-ID=4
        OPCOM, _TTF3:, "MOUNT TESTER ON MTA0:"

$ REPLY/PENDING=4 "PLEASE WAIT A FEW MINUTES"
$ REPLY/TO=4 "MTA0: IS DOWN -- TESTER MOUNTED ON MTA2:"
$
        OPCOM, 10:02:34.12, REQUEST COMPLETED, OPERATOR=_OPA0:, ID=4
```

In this example, OPCOM notifies the operator that a user wants the magnetic tape named TESTER placed on MTA0. The operator issues a REPLY/PENDING command to indicate that the task can be done, but not immediately. Later, the operator mounts the magnetic tape on MTA2 and notifies the user that the tape volume is on MTA2.

2.1.2 Operator Assistance in Handling Multivolume Tapes

This procedure describes how to mount tapes, previously initialized, in a multivolume tape set.

When a user is in the process of reading or writing to a tape and the tape reaches end-of-tape, the system suspends processing and sends a request to the operator to mount the next tape. The user does not see this message and may not realize that another tape is needed to complete the read or write operation.

The message, sent by the system, appears on the system console and all terminals designated to receive tape messages, as follows:

```
OPCOM, hh:mm:ss.cc, process-name ACCNT=[blank] REPLY-ID=identification-number
OPCOM, MOUNT RELATIVE VOLUME x (label-name) ON device-name:
```

The process name identifies the user. This name is either the name under which the user logged into the system or the name of the terminal at which the user is logged in. The process name is the name of a terminal when the user logs into a second terminal before logging out on the first terminal.

The variable, *x*, indicates the volume number. The name within the parentheses indicates the volume label of the tape.

Procedure

1. Locate the tape, place it on the specified device, and ready the device by pressing the LOAD button. If the drive does not come online automatically, press the ONLINE button.
2. Enter the following command, which indicates that the request has been fulfilled:

```
$ REPLY/TO=identification-number ["message-text"]
```

See Chapter 3 for more information on the REPLY command.

Example

1.


```
OPCOM, 14:24:07.53, TURNER ACCNT= REPLY-ID=4
OPCOM, MOUNT RELATIVE VOLUME 2 (GMB002) ON MTAL:

$ REPLY/TO=4
$
OPCOM, 14:31:43.71, REQUEST COMPLETED, OPERATOR=_OPA0:, ID=4
```

In this example, a user is reading or writing to the first tape in the volume set GMB002. The first OPCOM message indicates that the user needs the second volume of the tape set. After mounting it, the operator issues the REPLY command to indicate that the tape has been mounted and the system can resume processing. The second OPCOM message confirms that the operator's message has been sent to the facility handling tapes.

2.1.3 Operator Assistance in Creating a Multivolume Tape Set

This procedure describes how to initialize additional tapes in a multivolume tape set.

When a user is in the process of writing to a tape and the tape reaches end-of-tape, the system suspends processing and sends the operator a request to mount the next tape. The user does not see this message and may not realize that another tape is required to complete the write operation. The message displayed on the system console and all terminals designated to receive tape messages is as follows:

```
OPCOM, hh:mm:ss.cc, process-name ACCNT=[blank] REPLY-ID=identification-number
OPCOM, MOUNT RELATIVE VOLUME x () ON device-name:
```

The process name identifies the user. This name is either the name under which the user logged into the system or the name of the terminal at which the user is logged in. The process name is the name of a terminal when the user logs into a second terminal before logging out on the first terminal.

The variable, x, indicates the relative volume number. The empty parentheses (that is, a left parenthesis followed directly by a right parenthesis) indicate that the volume label is not known. Usually, this message indicates that a new tape volume must be initialized. Normally, when a user needs another tape and the tape has already been initialized, the volume label is included between the parentheses.

Procedure

1. If the user did not previously tell the operator that an additional tape may be needed and what the volume label of the tape should be, the operator should try to contact the user and obtain the following information:
 - a. Does the user really want another tape?
 - b. If the user wants another tape, what is the volume label with which the tape should be initialized?
2. Enter the following command if the user does not want another tape or if the user cannot be contacted:

```
§ REPLY/ABORT=identification-number ["message-text"]
```

NOTE

Steps 3, 4, and 5 below are ignored if the operator aborts the user request.

Aborting the request also aborts the user's program.

See Chapter 3 for more information on the REPLY command.

3. Write a visual label on the tape volume specifying the volume label and the relative volume number.
4. Place the tape on the specified drive and ready the device by pressing the LOAD button. If the drive does not come online automatically, press the ONLINE button.

VAX/VMS OPERATING PROCEDURES

5. Issue the following command, which initializes the tape and informs the system that the request has been fulfilled:

```
$ REPLY/TO=identification-number "volume-label/INITIALIZE"
```

Examples

1. OPCOM, 14:16:24.98, HAUSER ACCNT= REPLY-ID=6
OPCOM, MOUNT RELATIVE VOLUME 2 () ON MTAL:

\$ REPLY/TO=6 "GMB004/INITIALIZE"
\$
OPCOM, 14:23:36.10, REQUEST COMPLETED, OPERATOR=_OPA0:,ID=6

In this example, the OPCOM message notifies the operator that a user needs a second tape. After mounting the tape, the operator issues the REPLY command to indicate the tape is mounted. The volume label GMB004 is assigned to the tape by means of the /INITIALIZE keyword. The second OPCOM message confirms that the request has been sent to the tape handling facility.

2. OPCOM, 12:56:03.76, RYAN ACCNT= REPLY-ID=15
OPCOM, MOUNT RELATIVE VOLUME 3 () ON MTAL:

\$ REPLY/ABORT=15
\$
OPCOM, 13:03:12.42, REQUEST ABORTED, OPERATOR=_OPA0:,ID=15

The first OPCOM message indicates that the user needs another tape and the tape must be initialized. Upon notifying the user, the operator discovers that the user does not want another tape. Hence, the operator cancels the request using the REPLY/ABORT command.

VAX/VMS OPERATING PROCEDURES

2.1.4 Mounting Public Disks

This procedure describes how to mount a disk as a public volume.

Public volumes are volumes that contain more than one users' files. These volumes must be designated as shareable at the time they are mounted so they can be accessed by all system users.

Procedure

1. Allocate a drive by issuing the following command:

```
$ ALLOCATE device-name:
```
2. Locate the volume and place it on the allocated drive.
3. Ready the device by pressing the STOP/START switch or the RUN/STOP button.
4. Issue the MOUNT/SYSTEM command to allow all users access to the data on the device:

```
$ MOUNT/SYSTEM device-name: volume-label
```

NOTE

To use the /SYSTEM qualifier, the operator needs the SYSNAM privilege.

The volume label should be affixed to the outside of the volume.

5. Optionally, inform all users on the system that the volume is now available for use:

```
$ REPLY/USERS/BELL "message-text"
```

Example

1.

```
$ ALLOCATE DBB2:
  _DBB2: ALLOCATED
$ MOUNT/SYSTEM DBB2: USERFILES
%MOUNT-I-MOUNTED, USERFILES mounted on _DBB2:
$ REPLY/USERS/BELL "PUBLIC VOLUME MOUNTED ON DBB2:"
```

The ALLOCATE COMMAND requests the allocation of DBB2; the response from ALLOCATE indicates that DBB2 was successfully allocated. The MOUNT/SYSTEM command makes the disk mounted on DBB2 available to all users. The MOUNT command response indicates that the disk was successfully mounted. The REPLY/USERS/BELL command broadcasts a message to all users on the system that the disk mounted on DBB2 can be accessed.

2.1.5 Backing Up the System Disk to Disk

This procedure describes how the Disk Save and Compress utility copies the contents of the system disk to another disk after the system has been booted. If the system is a one-disk system, the operator must follow the procedure described in Section 2.1.6. This procedure copies the contents of the system disk to magnetic tape.

When backing up the system disk, the operator must be the only user logged into the system. This ensures that all the files that can be closed are closed and that the copy operation can be performed successfully.

To perform this procedure, the operator needs read access to all the files on the disk.

Procedure

1. Inform all users to log out of the system by issuing the following command:

```
$ REPLY/ALL/BELL "message-text"
```

This message should include when and why to log out.

2. Set the login count to 0 by typing:

```
$ SET LOGINS/INTERACTIVE=0
```

3. Check, after the prescribed time, that all the users have logged out by typing the command:

```
$ SHOW SYSTEM
```

The only processes that should be running are: NULL, SWAPPER, ERRFMT, OPERATOR, JOB_CONTROL, and the process that performs the copy operation. All other processes should be aborted by typing:

```
$ STOP/IDENTIFICATION=process-id
```

4. Allocate a disk drive by typing:

```
$ ALLOCATE device-name:
```

5. Place a scratch disk in the allocated drive and ready that device by pressing the START/STOP switch or RUN/STOP button.

6. Mount the scratch disk by typing:

```
$ MOUNT/FOREIGN device-name:
```

7. Run the Bad Block (BAD) utility on the scratch disk by entering the following command:

```
$ MCR BAD device-name:
```

NOTE

Make certain that the disk to be analyzed by BAD is the scratch disk and not the system disk. When BAD analyzes a disk, it destroys all the previous data recorded on the disk.

VAX/VMS OPERATING PROCEDURES

BAD analyzes the integrity of the blocks on the disk and displays, upon completion, the number of faulty blocks found.

8. Copy the files on the system disk to the scratch disk by typing the following command.

```
$ MCR DSC2 output-device-name:/VE = input-device-name:
```

NOTE

Make certain that the output device is the scratch disk and not the source disk. When DSC copies data to the output device, it destroys all the previous data recorded on that disk.

DSC copies the files to the scratch disk, then compares the contents of both disks.

9. Run DMP if DSC returns any verification messages by entering the following command:

```
$ MCR DMP SYS$OUTPUT = device-name:/FI:m:n/HD
```

NOTE

The variables m and n refer to the file identification number indicated in DSC's verification message.

This command displays the file name associated with the file identification number. If the file name is ERRLOG.SYS, ACCOUNTNG.DAT, JBCSYSQUE.DAT, OPERATOR.LOG, PAGEFILE.SYS, or SWAPFILE.SYS, the verification message is legitimate and usually can be ignored; if the file name is any other file, make sure all the users are logged out and perform the back-up procedure again.

10. Dismount and deallocate the scratch disk by typing:

```
$ DISMOUNT device-name:  
$ DEALLOCATE device-name:
```

11. Remove the scratch disk from the drive and affix a visual label to it specifying the volume label and current date.

12. Return the login count to its original state by typing:

```
$ SET LOGINS/INTERACTIVE=n
```

13. Inform all users that the system is back to normal by typing:

```
$ REPLY/ALL/BELL "message-text"
```

VAX/VMS OPERATING PROCEDURES

Example

```

1. $ REPLY/ALL/BELL "PLEASE LOG OUT -- SYSTEM DISK WILL BE BACKED UP
_OPA0:;SYSTEM 22:49:52.89
"PLEASE LOG OUT -- SYSTEM DISK WILL BE BACKED UP

$ REPLY/ALL/BELL "IN 5 MIN."

_OPA0:;SYSTEM 22:49:52.89
"IN 5 MIN."

$ SET LOGINS/INTERACTIVE=0
Login quotas - Interactive limit=0, Current interactive value=15
$ SHOW SYSTEM
  VAX/VMS      Processes on      18-JUL-1978 22:50:15.54
  Pid         Process Name      UIC   State Pri Dir. I/O   CPU   Page flts Ph.Mem
00010000     NULL                000,000 COM   0   0 00:10:14.03 0 0
00010001     SWAPPER              000,000 HIB   16   0 00:00:00.41 0 0
0001003B     ERRFMT                001,006 HIB   7   5 00:00:00.17 26 29
0001003C     OPERATOR              001,004 LEP   10   2 00:00:00.08 29 33
0001003D     JOB_CONTROL           001,004 HIB   13  74 00:00:00.83 31 61
0001003E     DMA0ACP               001,003 HIB   9  103 00:00:01.14 30 78
0002003F     SYSTEM                001,004 CUR   5   4 00:00:00.45 119 92

$ ALLOCATE DMA1:
  _DMA1: ALLOCATED
$ MOUNT/FOREIGN DMA1:
%MOUNT-I-MOUNTED,                mounted on _DMA1:
$ MCR BAD DMI:
BAD -- TOTAL NO. OF BAD BLOCKS = 3.
$ MCR DSC2 DMA1:/VE = DMAU:
DSC -- 45 STARTING VERIFY PASS

DSC -- *WARNING* 42 VERIFICATION ERROR ON DMI:
FILE ID 000503,000001,000000 VBN 000000,000001

$ MCR DMP SYSS$OUTPUT = DMA0:/FI:503:1/HD

Dump of DM0:[SYSERR]. - FILE ID 503,001,00
                                     File Header
ERRLOG.SYS;1 (503,001) 2./2. 08-JUN-78 09:55
.
.
.

$ DISMOUNT DMA1:
$ DEALLOCATE DMA1:
$ SET LOGINS/INTERACTIVE=25
$ REPLY/ALL/BELL "DMA0: BACK-UP IS COMPLETE"

_OPA0:;SYSTEM 23:07.33.09
"DMA0: BACK-UP IS COMPLETE"

```

In this example, the operator performs the preliminary steps of ensuring that no users are logged into the system and that BAD has been run on the scratch disk.

The operator then uses the MCR DSC2 command to copy the contents of the system disk to the scratch disk. The /VE qualifier requests that after the copy operation is complete, the contents of both disks be checked against each other to verify that the copy operation was successful. During the verification pass, DSC finds a file that was not properly

VAX/VMS OPERATING PROCEDURES

copied. To determine the file name, the operator issues the MCR DMP command; the parameter SYS\$OUTPUT requests that the information be displayed on the operator's terminal. The file name associated with the file identification number 503:001 is ERRLOG.SYS. Because this file never closes, its contents are always changing, making it impossible to create an exact copy of it. Consequently, the verification message can be ignored.

To complete the back-up procedure, the operator dismounts and deallocates the scratch disk and informs all users that the system disk is operational again.

VAX/VMS OPERATING PROCEDURES

2.1.6 Backing Up the System Disk to Tape

This procedure uses the stand-alone version of the Disk Save and Compress utility to copy the contents of the system disk onto magnetic tapes for retention. To restore the contents of the tape to disk, the operator uses the procedure described in Section 2.1.7.

To perform this procedure, the operator needs read access to all files on the system disk.

This procedure must be performed from the system console.

Procedure

1. Run the SHUTDOWN procedure as described in Section 2.5.1.
2. Write-protect the system disk by turning the WRITE-PROTECT switch, on the disk drive, to the ON position.
3. Place a magnetic tape on an available drive and ready the device. If there is more than one tape drive, mount and ready additional magnetic tapes on each available drive.
4. Load the diagnostic floppy diskette labeled ZZ-ESZCC into the console floppy drive.
5. Turn the AUTO-RESTART switch on the top front of the processor cabinet to the ON position.
6. Turn the DC ON/OFF switch on the LSI-11 control panel to the OFF position. The LSI-11 control panel is located on the bottom left side of the VAX-11/780 CPU cabinet.
7. Return the DC ON/OFF switch to the ON position. This initiates a system restart as indicated by the following console message:

```
CPU HALTED,SOMM CLEAR,STEP=NONE,CLOCK=NORM
RAD=HEX,ADD=PHYS,DAT=LONG,FILL=00,REL=00000000
INIT SEQ DONE
HALTED AT 00000000
(RELOADING WCS)
LOAD DONE, 00003200 BYTES LOADED
VER: PCS=01 WCS=03-10 FPLA=03 CON=PX03-08
(AUTO-RESTART)
CPU HALTED
INIT SEQ DONE
```

The console then loads the stand-alone system, which takes approximately 1.5 minutes. At the completion of loading, the following message is displayed on the system console:

```
LOAD DONE, 00020400 BYTES LOADED
VAX/VMS DSC2, A.1 30-MAY-78
DSC>
```

VAX/VMS OPERATING PROCEDURES

8. A copy of the VAX-11 Disk Save and Compress utility for structure level 2 disks (DSC2 utility) is now running. Type the DSC command appropriate for the system disk and the kind of back up needed. The following commands are examples of different types of back-ups:

```
DSC> MTA0:/RW/DENS=1600/VE = DBA0:
```

The above command copies the contents of the disk mounted on DBA0 to the tape mounted on MTA0. The /RW qualifier causes the output tape to be rewound before the copy operation begins. The /DENS=1600 qualifier requests that the data be copied to the tape at 1600 bits per inch. The /VE qualifier requests that the contents of the tape be compared with the contents of the disk to make certain the copy operation was successful.

```
DSC> MTA0:,MTA1:/RW/VE = DBA0:
```

In this example, DSC copies the contents of the disk mounted on DBA0 to the output devices MTA0 and MTA1. When DSC finishes copying data to the first tape, MTA0, it overlaps writing data to the second tape with rewinding the first tape, thus quickening the copy operation. Because no density at which the data should be copied to the tape is indicated, DSC copies the data at 800 bits per inch. The /VE qualifier requests that the contents of the tape be compared with the contents of the disk to make certain the copy operation was successful.

9. See the VAX-11 Disk Save and Compress User's Guide if DSC returns any verification messages.
10. Label each magnetic tape with the following information: the volume label of the disk backed up, the current date, the density at which the data was recorded, and the relative volume number.

VAX/VMS OPERATING PROCEDURES

2.1.7 Restoring of the System Disk from Tape

This procedure describes how to copy the contents of a magnetic tape to disk. To perform this procedure, the operator must enter the commands from the system console.

Procedure

1. Shut down the system by following the procedure documented in Section 2.5.1.
2. Place a scratch disk in an available drive and ready the device by pressing the RUN/STOP button or START/STOP switch.
3. Place the magnetic tape which contains the restoration data on tape drive 0. If the restoration material is stored on more than one tape, place the additional tapes on additional tape drives.
4. Load the diagnostic floppy diskette labeled ZZ-ESZCC into the console floppy drive.
5. Turn the AUTO-RESTART switch located on the top front of the processor cabinet to the ON position.
6. Turn the DC ON/OFF switch on the LSI-11 control panel to the OFF position. The LSI-11 control panel is located on the bottom left side of the VAX-11/780 CPU cabinet.
7. Return the DC ON/OFF switch to the ON position. This initiates a system restart as indicated by the following console message:

```
CPU HALTED,SOMM CLEAR,STEP=NONE,CLOCK=NORM
RAD=HEX,ADD=PHYS.DAT=LONG,FILL=00,REL=0000000
INIT SEQ DONE
(RELOADING WCS)
LOAD DONE, 00003200 BYTES LOADED
VER: PCS=01 WCS=03-10 FPLA=03 CON=PX03-08
(AUTO-RESTART)
CPU HALTED
INIT SEQ DONE
```

The console then loads the stand-alone system, which takes approximately 1.5 minutes. At the completion of loading, the following message is displayed on the system console:

```
LOAD DONE, 00020400 BYTES LOADED
VAX/VMS DSC2, VERSION A.1 30-MAY-1978
DSC>
```

8. A copy of the VAX-11 Disk Save and Compress utility for structure level 2 volumes (DSC2 utility) is now running. Type the DSC command appropriate for the restorations system disk and the kind of restoration desired. The following commands are examples of different types of restorations:

```
DSC> DBA0:/VE =MTA0:/RW
```

This command copies the contents of the tape mounted on MTA0 to the disk mounted on DBA0. The /RW qualifier causes the

VAX/VMS OPERATING PROCEDURES

tape to be rewound before the copy operation is started and the /VE qualifier requests that the contents of the tape and disk be compared to make certain the copy operation was successful.

```
DSC> DBA0:/VE = MTA0:,MTA1:/RW
```

In this example, DSC copies the contents of the tapes mounted on MTA0 and MTA1 to the disk mounted on DBA0. First, DSC copies the contents of the tape mounted on MTA0. When all the data on the tape has been copied, DSC begins to copy the contents of the second tape. The /RW qualifier causes the first tape to be rewound prior to starting the copy operation. The /VE qualifier requests that DSC performs a verification pass after each volume reaches end-of-tape. When the verify pass reaches end-of-tape, the copy operation continues.

9. See the VAX-11 Disk Save and Compress User's Guide if DSC returns a verification message.
10. Label the disk with the following information: the name of the tape the restoration was performed from and the current date.

2.1.8 Backing Up a Nonsystem Disk to Disk

This procedure describes how to copy the contents of a nonsystem disk to another disk. A nonsystem disk is a public disk, which has been mounted by use of the /SYSTEM qualifier.

This procedure is performed online. The operator need only write-lock the device which restrains system users from changing the data on the disk. They can, however, still read data.

Procedure

1. Issue the following command to warn all users that the specified disk will be write-locked so its contents can be copied to another disk:

\$ REPLY/ALL/BELL "message-text"

This message should include the name of the source disk being write-locked and when the write-lock will occur.

2. After the prescribed time, write-lock the source disk by pressing the WRITE-PROTECT switch to the ON position. This switch is located on the front panel of the disk drive.
3. Allocate a drive for the scratch disk by typing:

\$ ALLOCATE device-name:

4. Place a scratch disk in the allocated drive and ready the device by pressing the RUN/STOP button or the START/STOP switch.
5. Mount the scratch disk by issuing the following command:

\$ MOUNT/FOREIGN device-name:

6. Issue the following command to find out the structure level of the source disk:

\$ DIRECTORY device-name:[0,0]

If the directory listing is alphabetized, it is a structure level 2 volume; otherwise, it is structure level 1.

7. Run the appropriate VAX-11 Disk Save and Compress utility by typing one of the following commands:

a. \$ MCR DSC1 output-device-name:/VE = input-device-name:

b. \$ MCR DSC2 output-device-name:/VE = input-device-name:

NOTE

Make certain that the output device is the scratch disk and not the source disk. When DSC copies data to the output device, it destroys all the previous data recorded on that disk.

8. See to the VAX-11 Disk Save and Compress User's Guide if DSC returns any verification errors.

VAX/VMS OPERATING PROCEDURES

9. Dismount and deallocate the scratch disk by typing:


```

$ DISMOUNT device-name:
$ DEALLOCATE device-name:
      
```
10. Remove the scratch disk from the drive and affix a label to it, specifying the volume label and current date.
11. Write-enable the source disk, by pressing the WRITE-PROTECT switch to the OFF position.
12. Inform all users that the source disk is no longer write-locked by issuing the following command:


```

$ REPLY/ALL/BELL "message-text"
      
```

Example

1. \$ REPLY/ALL/BELL "DMA0: WILL BE WRITE-LOCKED IN 5 MIN FOR BACK-UP."

```

_OPAO: ,SYSTEM 19:36:11.70
"DMA0: WILL BE WRITE-LOCKED IN 5 MIN FOR BACK-UP."

$ ALLOCATE DM:
 DMA1: ALLOCATED
$ MOUNT/FOREIGN DMA1:
%MOUNT-I-MOUNTED, mounted on _DMA1:
$ DIRECTORY DMA0:[0,0]
      
```

```

DIRECTORY DM0:[0,0]
20-JUL-78 19:45
      
```

```

000000.DIR;1 2. c 20-JUL-78 19:45
001001.DIR;1 4. c 20-JUL-78 19:45
001002.DIR;1 1. c 20-JUL-78 19:45
001004.DIR;1 1. c 20-JUL-78 19:45
001005.DIR;1 4. c 20-JUL-78 19:45
001006.DIR;1 1. c 20-JUL-78 19:45
001007.DIR;1 9. c 20-JUL-78 19:45
001010.DIR;1 1. c 20-JUL-78 19:45
001020.DIR;1 1. c 20-JUL-78 19:45
001040.DIR;1 7. c 20-JUL-78 19:45
BACKUP.SYS;1 0. c 20-JUL-78 19:45
BADBLK.SYS;1 22. c 20-JUL-78 19:45
BADLOG.SYS;1 0. c 20-JUL-78 19:45
BITMAP.SYS;1 5. c 20-JUL-78 19:45
CONTIN.SYS;1 0. c 20-JUL-78 19:45
CORIMG.SYS;1 0. c 20-JUL-78 19:45
INDEXF.SYS;1 349. c 20-JUL-78 19:45
SYSERR.DIR;1 1. c 20-JUL-78 19:45
SYSEXE.DIR;1 7. c 20-JUL-78 19:45
SYSHLP.DIR;1 4. c 20-JUL-78 19:45
SYSLIB.DIR;1 4. c 20-JUL-78 19:45
SYSMAN.DIR;1 1. c 20-JUL-78 19:45
SYSMGR.DIR;1 1. c 20-JUL-78 19:45
SYSTST.DIR;1 9. c 20-JUL-78 19:45
SYSUPD.DIR;1 1. c 20-JUL-78 19:45
VOLSET.DIR;1 0. c 20-JUL-78 19:45
      
```

TOTAL OF 436./1132. BLOCKS IN 26. FILES

VAX/VMS OPERATING PROCEDURES

```
$ MCR DSC2 DMA1:/VE = DMA0:
DSC -- 45 STARTING VERIFY PASS
$ DISMOUNT DMA1:
$ DEALLOCATE DMA1:
$ REPLY/ALL/BELL "DMA0: NO LONGER WRITE-LOCKED"
```

In this example, the operator informs all system users that DMA0 will be write-locked for back-up purposes. After write-locking the disk, the operator performs the necessary steps to mount and ready the scratch disk.

The operator types the DIRECTORY command to determine the structure level of the source disk. In this case, the directory listing is in alphabetical order, which indicates that the volume is a structure level 2 disk.

The operator then performs the copy operation by using the MCR DSC2 command. The qualifier /VE requests that the contents of both disks be checked against each other to ensure that the copy operation was successful. DSC informs the operator when the verify pass begins. At completion, the dollar sign prompt (\$) appears indicating that the copy operation was successful.

The operator then dismounts and deallocates the scratch disk, write-enables DMA0, and informs all users the DMA0 can be written to.

2.1.9 Selective Back-up of Files

This procedure describes how to copy selected files from one disk to another. Generally, if files must be backed up regularly, the operator should create a command procedure that contains the required back-up commands. Then, the operator need only specify the execute command (@) before the command procedure file specification and the system will begin processing the commands in the command procedure.

Procedure

1. Allocate a drive in which the back-up volume is placed by typing:

```
$ ALLOCATE device-name:
```

2. Place the back-up volume in the allocated drive and ready that device by pressing either the RUN/STOP button or the START/STOP switch.
3. Mount the back-up volume by typing:

```
$ MOUNT device-name: volume-label
```

NOTE

If the back-up volume has not been previously initialized, it must be initialized before it is mounted and processed by the Bad Block (BAD) utility after it is mounted.

4. Allocate a drive in which the source disk is placed by typing:

```
$ ALLOCATE device-name:
```

5. Place the source disk in the allocated drive and ready that device by pressing the RUN/STOP button or the START/STOP switch.
6. Mount the source disk by typing:

```
$ MOUNT device-name: volume-label
```

7. Create directories on the back-up volume by typing:

```
$ CREATE/DIRECTORY device-name:[directory]
```

Generally, the operator creates a separate directory for each user who has files to be backed up.

8. Copy the files from the source disk to the back-up volume by typing the following command:

```
$ COPY input-device:[directory]filename.type.ver output-device:[directory]
```

This command is repeated for each file to be copied. If a user has several files of the same file name, type, or version, the operator can use wild cards (*) to indicate them.

VAX/VMS OPERATING PROCEDURES

9. Dismount and deallocate the back-up volume by typing:


```

$ DISMOUNT device-name:
$ DEALLOCATE device-name:
      
```
10. Remove the back-up volume from the device and affix a label to it specifying the volume label and current date.
11. Dismount and deallocate the source disk by typing:


```

$ DISMOUNT device-name:
$ DEALLOCATE device-name:
      
```
12. Remove the source disk from the device.

Example

```

1. $ ALLOCATE DMA0:
   _DMA0: ALLOCATED
$ MOUNT DMA0: MYVOLUME
%MOUNT-I-MOUNTED, MYVOLUME           mounted on _DMA0:
$ ALLOCATE DMAL:
   _DMAL: ALLOCATED
$ MOUNT DMAL: DATCOM
%MOUNT-I-MOUNTED, DATCOM             mounted on _DMAL:
$ CREATE/DIRECTORY DMA0:[RMSTST]
$ CREATE/DIRECTORY DMA0:[FORMAB]
$ CREATE/DIRECTORY DMA0:[SYSTEM]
$ COPY DMAL:[RMSTST]*.*              DMA0:[RMSTST]
$ COPY DMAL:[FORMAB]*.COM            DMA0:[FORMAB]
$ COPY DMAL:[SYSTEM]DUNGEON.*        DMA0:[SYSTEM]
$ DISMOUNT DMA0:
$ DEALLOCATE DMA0:
$ DISMOUNT DMAL:
$ DEALLOCATE DMAL:
      
```

In this example, the operator copies files from three directories on DMAL to three newly created directories on DMA0. After performing the necessary steps to mount and ready the source and back-up volumes, the operator creates three directories [RMSTST], [FORMAB], and [SYSTEM] on the back-up volume. By using the COPY command, the operator copies:

- All the files in the [RMSTST] directory on the source disk to the [RMSTST] directory on the back-up volume
- All the files of the file type COM in the [FORMAB] directory on the source disk to the [FORMAB] directory on the back-up volume
- All the files with the file name DUNGEON in the [SYSTEM] directory on the source disk to the [SYSTEM] directory on the back-up volume.

Upon completion of the copy operation, the operator dismounts and deallocates the source and back-up volumes.

2.1.10 Backing Up a Floppy Diskette to Disk

This procedure describes how to copy the contents of a floppy diskette to disk for retention.

Procedure

1. Create an empty directory on a back-up volume using the following command:

```
$ CREATE/DIRECTORY device-name:[directory]
```

2. Insert, into the console device, the floppy diskette to be backed up. If the floppy diskette is already inserted and mounted, the operator must dismount it by typing:

```
$ DISMOUNT console-device:
```

3. Enter the following command:

```
$ @[SYSUPD]CONSCOPY
```

This command executes the command procedure CONSCOPY, which resides in the directory [SYSUPD]. This command procedure steps the operator through the back-up procedure.

4. Supply the information requested in the following questions:

```
WHAT IS THE FILES-11 DEVICE AND DIRECTORY (DDCU:[DIRECTORY])
WHAT IS THE NAME OF THE CONSOLE DEVICE (DDCU:):
```

The first question asks for the device and directory to which the floppy diskette will be backed up; the second question asks for the name of the device in which the floppy diskette is inserted.

5. Answer "Y" to the following question:

```
IS THE CONSOLE MEDIUM TO BE SAVED (Y/N):
```

6. Follow the directions specified in the following instruction:

```
MOUNT THE MEDIA TO BE COPIED ON DEVICE "device-name", TYPE <CR> WHEN READY
```

The MOUNT command then indicates whether the console medium has been successfully mounted. If so, the copy operation begins. This operation takes approximately five minutes to transcribe the contents of the floppy into the directory. At completion, control is returned to the command interpreter.

VAX/VMS OPERATING PROCEDURES

Example

```
1. $ CREATE/DIRECTORY DBAL:[HOMER.CONSOLE]
    $ DISMOUNT DXA1:
    $ @[SYSUPD] CONSCOPY
    !
    ! Save or restore the contents of the console device from a Files-11 medium
    !
    WHAT IS THE FILES-11 DEVICE AND DIRECTORY (DDCU:[DIRECTORY]): DBAL:[HOMER.CONSOLE]
    WHAT IS THE NAME OF THE CONSOLE DEVICE (DDCU:): DXA1:
    IS THE CONSOLE MEDIUM TO BE SAVED (Y/N): Y
    MOUNT THE MEDIA TO BE COPIED ON DEVICE DXA1:, TYPE <CR> WHEN READY:
    %MOUNT-I-MOUNTED,          mounted on _DXA1:
    !
    ! Backup of the console device is now complete. Remove the media
    ! from the device.
    !
    $
```

To copy the floppy diskette to disk, the operator 1) creates the subdirectory [CONSOLE] on DBAL, which will store the contents of the floppy, and 2) dismounts the floppy diskette. The operator then executes the command procedure CONSCOPY to perform the copy operation. This procedure requests the following information:

- The device and directory names to which the contents of the floppy diskette are copied; the operator responds with DBAL:[HOMER.CONSOLE]
- The device in which the floppy diskette is inserted; the operator responds with DXA1
- Whether the console medium (that is, the floppy diskette) is to be saved; the operator responds with Y (yes)
- Whether the floppy diskette is inserted in the console device; the operator responds by pressing the RETURN key to indicate that the floppy diskette is mounted

The CONSCOPY command procedure then performs the copy operation and informs the operator when it is finished.

2.1.11 Restoring a Floppy Diskette from Disk

This procedure describes how to restore the contents of a floppy diskette from a disk.

Procedure

1. Locate the directory which contains the data to be restored to a floppy diskette.
2. Insert, into the console device, a scratch floppy diskette. If a scratch diskette is already inserted and mounted, the operator must dismount it by typing:

\$ DISMOUNT console-device:

3. Enter the following command:

\$ @[SYSUPD]CONSCOPY

This command executes the command procedure CONSCOPY, which resides in the directory [SYSUPD]. This command procedure will step the operator through the restoration procedure.

4. Supply the information requested in the following questions:

WHAT IS THE FILES-11 DEVICE AND DIRECTORY (DDCU:[DIRECTORY]):
WHAT IS THE NAME OF THE CONSOLE DEVICE (DDCU:)

The first question asks for the device and directory which contains the material to be restored on a floppy; the second question asks for the name of the device in which the floppy is inserted.

5. Answer "N" to the question:

IS THE CONSOLE MEDIUM TO BE SAVED (Y/N):

6. Answer the following question by supplying the name of the bootable system image:

WHAT IS THE NAME OF THE BOOTABLE IMAGE:

NOTE

Generally, the name of this file is CONSOL.SYS.

7. Follow the directions specified in the following instruction:

MOUNT A SCRATCH MEDIUM ON DEVICE "device-name:", TYPE <CR> WHEN READY:

The MOUNT command then indicates whether the scratch medium has been successfully mounted. If so, the restoration procedure begins. It takes approximately five minutes to transcribe the contents of the directory to the floppy diskette. At completion, the contents of the diskette directory are printed on the terminal.

VAX/VMS OPERATING PROCEDURES

Example

```

1.  $ DISMOUNT DXA1:
    $ @[SYSUPD]CONSCOPY
    !
    ! Save or restore the contents of the console device from a Files-11 medium
    !
    WHAT IS THE FILES-11 DEVICE AND DIRECTORY (DDCU:[DIRECTORY]): DB1:[CONSOL]
    WHAT IS THE NAME OF THE CONSOLE DEVICE (DDCU:): DXA1:
    IS THE CONSOLE MEDIUM TO BE SAVED (Y/N): N
    !
    ! If no bootable file is to be written on the console medium type only a <CR>
    ! in response to the next query.
    !
    !
    ! For bootable floppies supplied by DEC, the name of this file is CONSOL.SYS.
    !
    WHAT IS THE NAME OF THE BOOTABLE IMAGE: CONSOL.SYS
    MOUNT A SCRATCH MEDIUM ON DEVICE DXA1:, TYPE <CR> WHEN READY:
    %MOUNT-I-MOUNTED                mounted on _DXA1:

    DIRECTORY          CX0:
    13-JUL-78

    17APR7.VB0         0. 13-JUL-78
    ESZAB.VB0          0. 13-JUL-78
    CONSOL.SYS         31. 13-JUL-78
    WCSMON.SYS         15. 13-JUL-78
    WCSSRV.BIN         10. 13-JUL-78
    PCS .PAT           97. 13-JUL-78
    FILEA .PAT         7. 13-JUL-78
    WCS115.PAT         25. 13-JUL-78
    WCSMON.HLP         6. 13-JUL-78
    CONSOL.HLP         10. 13-JUL-78
    ABBREV.HLP         4. 13-JUL-78
    ERROR.HLP          9. 13-JUL-78
    REMOTE.HLP         3. 13-JUL-78
    BOOT .HLP          7. 13-JUL-78
    VMB .EXE           11. 13-JUL-78
    DM0BOO.CMD         2. 13-JUL-78
    DM1BOO.CMD         2. 13-JUL-78
    DM2BOO.CMD         2. 13-JUL-78
    DM3BOO.CMD         2. 13-JUL-78
    DMOGEN             2. 13-JUL-78
    DM1GEN             2. 13-JUL-78
    DM2GEN             2. 13-JUL-78
    DM3GEN             2. 13-JUL-78
    DMOXDT             2. 13-JUL-78
    DM1XDT             2. 13-JUL-78
    DB0BOO.CMD         2. 13-JUL-78
    DB1BOO.CMD         2. 13-JUL-78
    DB2BOO.CMD         2. 13-JUL-78
    DB3BOO.CMD         2. 13-JUL-78
    DB4BOO.CMD         2. 13-JUL-78
    DB5BOO.CMD         2. 13-JUL-78
    DB6BOO.CMD         2. 13-JUL-78
    DB7BOO.CMD         2. 13-JUL-78
    DB0GEN             2. 13-JUL-78
    DB1GEN             2. 13-JUL-78
    DB2GEN             2. 13-JUL-78
    DB3GEN             2. 13-JUL-78
    DB4GEN             2. 13-JUL-78
    DB5GEN             2. 13-JUL-78
    DB6GEN             2. 13-JUL-78
  
```


VAX/VMS OPERATING PROCEDURES

```
DB7GEN      2. 13-JUL-78
DB0DXT      2. 13-JUL-78
DBAXDT      2. 13-JUL-78
DBABOO.CMD  2. 13-JUL-78
DBAGEN      2. 13-JUL-78
DBBBOO.CMD  2. 13-JUL-78
DEFBOO.CMD  2. 13-JUL-78
DBBXDT      2. 13-JUL-78
DBBGGEN     2. 13-JUL-78
RESTAR.CMD  2. 13-JUL-78
< UNUSED > 175.
```

175. FREE BLOCKS

TOTAL OF 305. BLOCKS IN 50. FILES

```
!
! Restoration of the console device is complete. Remove the media
! from the device.
!
$
```

In this example, a scratch diskette is already mounted in the console device when the operator begins the restoration operation. The operator dismounts the floppy diskette by typing the DISMOUNT command and begins the restoration operation by typing the command @[SYSUPD]CONSCOPY.

The CONSCOPY command procedure requests the following information:

- The device and directory names which contain the data to be restored onto a floppy diskette; the operator responds with DBA1:[CONSOL]
- Whether the console medium (that is, floppy diskette) is to be saved; the operator responds with a N (no)
- The name of the bootable system image; the operator responds with CONSOL.SYS
- Whether the floppy diskette is inserted in the console device; the operator responds by pressing the RETURN key to indicate that the floppy is mounted

The CONSCOPY command procedure then performs the restoration operation and at completion, displays the directory listing of the disk.

2.2 CONTROLLING BATCH AND PRINT QUEUES

Although the system manager is responsible for creating batch queues and print queues, the operator must be able to control these queues so that the operating system can run smoothly. The procedures described in this section tell how to control existing batch and print queues. This section does not, however, contain a procedure for creating additional queues. This topic is fully described in the VAX/VMS System Manager's Guide.

In brief, a queue is a list in which jobs are placed to wait their turn to be executed. Unless a job is explicitly assigned a priority, by use of the SET QUEUE command, the jobs are executed on a "first come, first served" basis.

Print jobs are placed in print queues by means of the PRINT command and batch jobs are placed in batch queues by means of the SUBMIT command. Print queues can be any one of the following:

- Physical queues -- queues associated with (that is, named for) a specific line printer.
- Generic queues -- queues from which files can be printed out on any available line printer that has correctly matching characteristic.
- Logical queues -- queues that are not associated with a line printer. To obtain printed output from a logical queue, the operator must explicitly assign that queue to a printer. The ASSIGN/QUEUE command, described in Chapter 3, is used for this purpose.

To control print queues and batch queues, the operator must understand and be able to use the commands that regulate queueing. These commands allow the operator to manipulate the queues and the jobs contained within them. These commands are described in detail in Chapter 3 of this guide.

In addition, the operator should also be familiar with two other VAX/VMS commands:

- SHOW QUEUE -- displays information about a file (or files) entered in batch or print queues
- SET QUEUE -- changes the attributes of a file (or files) entered in batch or print queues

These commands are described fully in the VAX/VMS Command Language User's Guide.

2.2.1 Merging Line Printer Queues

When a problem occurs with a line printer, the queue associated with that printer should be rerouted to another printer. This procedure describes how to reroute a queue without losing any of the jobs in that queue.

Procedure

1. Stop the queue associated with the malfunctioning printer, by issuing the following command:

```
$ STOP/QUEUE/NEXT queue-name[:]
```

This command inhibits further dequeuing; however, the current job is permitted to complete.

2. Terminate the printing of the current job and requeue that job at the end of the queue by entering the following command:

```
$ STOP/REQUEUE queue-name[:]
```

3. Take the device offline by pressing the ON/OFF LINE switch, located on the top panel of the line printer, to the OFF position. The indicator light above the switch goes off when the device is offline.
4. Reroute the jobs queued to the malfunctioning printer to another printer by entering the following command:

```
$ ASSIGN/MERGE device-name[:] queue-name[:]
```

The operator should check that characteristics of the new printer are appropriate for the new jobs.

5. Optionally, delete the queue associated with the malfunctioning printer by typing:

```
$ DELETE/QUEUE queue-name[:]
```

Example

1. \$ STOP/QUEUE/NEXT LPB0:
\$ STOP/REQUEUE LPB0:
\$ ASSIGN/MERGE LPA0: LPB0:
\$ DELETE/QUEUE LPB0:

The STOP/QUEUE/NEXT command stops dequeuing jobs in the queue associated with printer LPB0 after the current job completes printing. The STOP/REQUEUE command stops the job currently printing and places it at the end of the queue. Then, the ASSIGN/MERGE command removes the jobs in the queue LPB0 and places them in the queue LPA0. The queue LPB0 can now be deleted using the DELETE/QUEUE command.

2.2.2 Preventing Loss of Data When Line Printer Runs out of Paper

This procedure describes how to prevent loss of data while the paper is changed in the line printer.

When the printer runs out of paper, OPCOM prints the following message on the operator's terminal indicating that the device is not ready:

```
OPCOM, hh:mm:ss.cc, DEVICE OFFLINE, device-name[:]
```

Procedure

1. Suspend the current queue operation by issuing the following command:

```
$ STOP/QUEUE queue-name[:]
```

2. Load a new box of paper in the printer and return the printer to the online condition.
3. Resume printing by entering the command:

```
$ START/QUEUE/optional-qualifier queue-name[:]
```

Note that the operator can optionally append an additional qualifier to the above command that will adjust the characteristics of the queue. These qualifiers are:

```
/BACKSPACE[=n]    Backspaces n pages before resuming the
                  printing. Currently, n is limited to one
                  page.
```

```
/TOP_OF_FILE      Starting at the beginning of the job, print
                  the job that was being printed when the
                  paper ran out.
```

Example

1. OPCOM, 22:08:43.40, DEVICE OFFLINE, LPA0:

```
$ STOP/QUEUE LPA0:
$ START/QUEUE/TOP_OF_FILE LPA0:
```

In the above example, OPCOM notifies the operator that line printer LPA0 went offline at 22:08:43.40. The operator stops the queue associated with the printer, loads a new box of paper into the printer, then returns the printer to the online condition. The operator can then resume printing by entering the START/QUEUE command. The /TOP_OF_FILE qualifier indicates that the job currently printing will be restarted at the beginning of the file.

VAX/VMS OPERATING PROCEDURES

2.2.3 Terminating a Batch Job

This procedure describes how to terminate execution of a batch job. This is a procedure that should be used only if the system manager or the owner of a job requests that the job be terminated.

Procedure

1. Type the following command to determine the queue name and entry number of the batch job that is to be terminated:

```
$ SHOW QUEUE/BATCH/ALL
```

2. Delete the bad batch job by issuing the following command:

```
$ STOP/ENTRY=entry-number queue-name[:]
```

Example

1. \$ SHOW QUEUE/BATCH/ALL

```
* Batch Queue * "SYS$BATCH" Jobum=6, Inipriv=4, Swap
Current Job 376  DEBUG      DBGBUILD Pri=4, 24-JUL-1978 11:09
Current Job 315  JEROME     BEGINBLD Pri=4, 24-JUL-1978 10:05
Current Job 313  SYSTEM     DELETELO Pri=4, 24-JUL-1978 09:57
Current Job 312  SYSTEM     LNK32    Pri=4, 24-JUL-1978 09:57
Current Job 207  LANGLEY    FORPROG  Pri=4, 24-JUL-1978 09:24
```

```
$ STOP/ENTRY=207 SYS$BATCH
```

In this example, a user has requested that job 207 be deleted. The operator types the SHOW QUEUE/BATCH/ALL command to determine the queue in which the job has been entered. The display shows that job 207 is in the SYS\$BATCH queue. The operator then aborts the job by typing the STOP/ENTRY command.

2.2.4 Terminating a Print Job

This procedure describes how to terminate a job that is currently printing on a line printer. This is a procedure that should be used only if the system manager or the owner of a job requests that the job be terminated, or if the operator observes garbled output being printed on the line printer.

Procedure

1. Enter the following command:

```
$ STOP/ABORT printer-name[:]
```

Terminating a job while it is currently being printed is a one-step procedure. The above command terminates the current job and begins printing the next job in the queue without a pause or stop.

Example

1. \$ STOP/ABORT LPA0:

This command terminates the printing of the current job on LPA0. The next job in the queue is immediately dequeued for printing.

2.3 TENDING THE CARD READER

When the operator is given a card deck to process through the card reader, the operator should check:

1. If the deck contains a batch job, that the first cards in the deck contain JOB and PASSWORD commands and that the last card contains an EOJ command or is an EOF card
2. If the deck is a data deck, that the user has correctly allocated the card reader, and that the deck contains an EOF card
3. That the card reader is set to the correct translation mode

Processing a Batch Job: When a batch job is submitted through the card reader, the first two cards in the card deck must contain the JOB and PASSWORD commands. Without these cards, the system cannot execute the job. If the operator is given a card deck without these cards, the operator should return the deck so the user can insert the necessary cards.

The last card in the deck must contain the EOJ command or be an EOF card (12-11-0-1-6-7-8-9 overpunch). If this card is missing, the operator can type one on the card punch and insert it at the end of the deck.

Processing a Data Deck: When a user wants to process a data deck, that is, a deck that contains only data and does not contain JOB and PASSWORD commands, the operator should make sure that the user has allocated the card reader. If the card reader is not allocated, the system tries to submit the deck as a batch job and subsequently flushes the deck through the system.

The EOF card (12-11-0-1-6-7-8-9 overpunch) must be the last card in the deck. This card signals the end of the deck. Without this card, the system continually waits for more cards. If the card deck lacks an EOF card, the operator can type one on a card punch and insert it at the end of the deck.

Card Reader Translation Modes: For the system to read input properly, the card reader must be set to the correct translation mode. The translation mode depends on the type of card punch on which the cards were punched. VAX/VMS supports 026 and 029 card punches. (For more information on the translation modes, see the VAX/VMS I/O User's Guide.) However, unless the card deck contains, as the first card, a mode translation card, or unless the operator knows the mode in which the cards were punched, the operator has little way of knowing the mode to which the card reader should be set. The operator should then explicitly set the translation mode of the card reader by use of the SET CARD READER command. This command is fully described in the VAX/VMS Command Language User's Guide.

Correcting Physically Defective Cards: However, even when the card deck contains all the required cards, the card reader may not be able to process the deck. This is generally due to a physical defect on one of the cards.

If the deck contains a faulty card, one of the error indicators, located on the front panel of the card reader, lights up when the card

VAX/VMS OPERATING PROCEDURES

is read. Operator intervention is then required to correct the situation. Table 2-2 describes the error indicators, the probable cause of their illumination, and the operator action required to correct the situation.

Table 2-2
Error Alarm Causes and Remedies

Error Indicator	Possible Cause	Corrective Action
READ CHECK	Card edges torn Punch in 0 or 81st column	Remove the faulty card from the output stacker, duplicate the card, place it in the input hopper and press the RESET button If READ CHECK occurs for all cards, it indicates a malfunction in the card reader read logic
PICK CHECK	Damage to leading edge Torn webs Cards stapled together	Remove the cards from the input hopper, duplicate the faulty card, place the cards back in the input hopper and press the RESET button If there is no evidence of card damage, check for excessive warpage of the card deck and/or a build-up of ink glaze on the picker face
STACK CHECK	Jam in the card track Badly mutilated card	Correct the jam and/or remove the mutilated card from the output stacker, duplicate the card, place it in the input hopper and press the RESET button
HOPPER CHECK	Input hopper empty Output stacker full	Load the input hopper Unload the output stacker

VAX/VMS OPERATING PROCEDURES

2.3.1 Operator Assistance in Processing a Card Deck

This procedure describes how to load and process a card deck from a card reader.

Procedure

1. Remove the card weight from the input hopper and place the cards, face down with column 1 to the left, in the hopper. Make sure the first card to be read is at the bottom of the hopper.

NOTE

Do not pack the input hopper so full that the air from the blower cannot rifle the cards. If the cards are packed too tightly, the vacuum picker cannot operate properly.

2. Press the RESET button; this causes the HOPPER CHECK error indicator and the STOP light to extinguish and the cards to be read.

NOTE

If the card deck is too large to fit in the input hopper, the excess cards can be loaded while the reader is operating if tension is maintained on the front portion of the deck.

3. Remove the cards from the output stacker when the HOPPER CHECK error indicator and the STOP light are lit.
4. See Table 2-2 in Section 2.3 for recovery action if the cards are not read properly upon pressing the RESET button.

2.4 MONITORING SYSTEM STATUS AND ACTIVITY

Periodically, the operator must examine the error log file (ERRLOG.SYS) and the operator's log file (OPERATOR.LOG). These files gather and maintain information on system errors and events.

Handling the Error Log File: The error log facility consists of three parts:

- A set of executive routines that detect errors and events and write relevant information into error log buffers in memory
- A program called ERRFMT that periodically empties the error log buffers, transforms the descriptions of the errors into standard formats, and stores the formatted information in a file on the system disk
- A program called SYE that generates readable reports from the information formatted by ERRFMT

The executive routines and the ERRFMT program operate continuously without operator intervention. The routines fill the error log buffers in memory with raw data on every detected error and event. When one of the available buffers becomes full, or when a time allotment expires, ERRFMT writes the information into the ERRLOG.SYS file. Then, the report generator SYE can use ERRLOG.SYS as its input file and create an error log report as its output file.

However, while SYE is accessing ERRLOG.SYS, ERRFMT cannot write any error information into it. Therefore, if SYE is accessing the highest version of ERRLOG.SYS when ERRFMT needs to log an error, ERRFMT creates a new version of the file. The new version picks up logging errors where the previous version left off. All versions of the ERRLOG.SYS file remain on the system disk (SYS\$SYSTEM) until the operator explicitly manipulates them in some way.

Generating error log reports is simpler and more efficient if the operator has fewer log files to deal with. The system manager or operator can take steps to minimize or control the number of versions created. The procedure for obtaining a printed copy of the error log file includes creating a copy of the current ERRLOG.SYS file and using that copy as input to SYE. In this way, SYE does not close the file to the ERRFMT program.

Handling the Operator's Log File: The operator's log file contains reports concerning the status of the devices currently attached to the system, time stamps that serve as indicators that the system is still running, and user requests and operator replies.

Unlike the error log file, the operator's log file is already printable text; no intermediate program need be invoked to translate it into readable output.

For more information about the operator's log file, see Chapter 4.

Maintaining the Error Log File and the Operator's Log File: Either the system manager or the operator is responsible for devising a plan for maintaining the versions of the error log file and the operator's log file.

VAX/VMS OPERATING PROCEDURES

One way to maintain these files is to rename the highest version of ERRLOG.SYS file and the OPERATOR.LOG file on a daily basis. For example, an operator could rename the files every morning at nine A.M. To free space on the system disk, the operator could then back up the old versions of the files onto a different volume and purge the files on the system disk. Caution should be taken, however, to ensure that the error log files are not deleted inadvertently. The information contained in the error log files is essential for efficient maintenance of VAX/VMS operating system.

2.4.1 Printing the Error Log File

This procedure describes how to create an error log report and how to obtain a copy of it. For a complete description of the RUN SYS\$SYSTEM:SYE command see Chapter 3.

Procedure

1. Set the default disk to the system disk and the default directory [SYSERR] by typing the following commands:

```
$ SET DEFAULT SYS$SYSTEM
$ SET DEFAULT SYS$DISK:[SYSERR]
```

2. Examine the [SYSERR] directory to see what versions of the ERRLOG.SYS file are on disk by typing:

```
$ DIRECTORY ERRLOG.SYS;*
```

3. Rename all the versions of the ERRLOG.SYS file to ERRLOG.OLD by issuing the command:

```
$ RENAME ERRLOG.SYS;* ERRLOG.OLD;*/NEW_VERSION
```

4. Invoke the SYE utility by typing the command:

```
$ RUN SYS$SYSTEM:SYE
```

5. To create an error log report, see the description of the RUN SYS\$SYSTEM:SYE command in Chapter 3. However, enter the following file name in response to the input file prompt (input file:):

```
ERRLOG.OLD
```

This is the file created in step 3 of this procedure. By default, SYE uses the highest version of the ERRLOG.OLD file.

6. Obtain a copy of the error log report by entering the following command:

```
$ PRINT filename
```

The file name is the name of the file entered in response to the output file prompt (output file:)

Example

1. \$ SET DEFAULT SYS\$SYSTEM
\$ SET DEFAULT SYS\$DISK:[SYSERR]
\$ DIRECTORY ERRLOG.SYS

```
DIRECTORY DB22:[SYSERR]
18-JUL-78 15:13
```

```
ERRLOG.SYS;1      14.      18-JUL-78      13:48
```

```
TOTAL OF 14./18. BLOCKS IN 1. FILE
```

VAX/VMS OPERATING PROCEDURES

```

$ RENAME ERRLOG.SYS;* ERRLOG.OLD;*/NEW_VERSION
$ RUN SYS$SYSTEM:SYE
SYE x0.6-0
_input file:      [[1,6]ERRLOG.SYS]      ?ERRLOG.OLD
_output file:    [SYS$OUTPUT]           ?ERRLOG.DAT
_options:        [ROLL-UP]              ?R
_device name:    [<all>]                 ?<CR>
_after date:     [17-NOV-1858]          ?<CR>
                                           17-NOV=1858  00:00:00.00
_before date:    [31-DEC-9999]          ?<CR>
                                           31-DEC-9999  23:59:59.99

Successful completion
$ PRINT ERRLOG.DAT

```

The SET DEFAULT commands set the operator's default disk and directory to DEB2:[SYSERR]. The DIRECTORY command lists all the ERRLOG.SYS files contained in the [SYSERR] directory. In this example, [SYSERR] contains only one version of ERRLOG.SYS. The RENAME command renames ERRLOG.SYS to ERRLOG.OLD; the /NEW_VERSION qualifier requests that ERRLOG.OLD be assigned a new version number if a file of this name already exists.

The operator then invokes the SYE utility by typing RUN SYS\$SYSTEM:SYE. SYE prompts for the following six parameters:

- The name of the file to be manipulated; the operator responds with ERRLOG.OLD
- The name of the file that is to contain the error log report; the operator responds with ERRLOG.DAT
- The type of report that SYE should generate: the operator responds with R, which indicates the ROLL UP report
- The devices on which SYE should report errors; the operator responds by pressing the RETURN key, which requests SYE to report the occurrence of errors on all devices
- The time from which SYE should record errors; the operator responds by pressing the RETURN key, which requests SYE to report the occurrence of errors after November 17, 1858
- The time to which SYE should record errors; the operator responds by pressing the RETURN key, which requests SYE to report the occurrence of errors until December 31, 9999

SYE creates the error log report and stores it in the ERRLOG.DAT file. The operator obtains a hard copy of this report by using the PRINT command.

2.4.2 Printing the Operator's Log File

This procedure describes how to produce a printed copy of the current version of the operator's log file (OPERATOR.LOG), which should be given to the system manager for retention. This procedure should be performed periodically.

Procedure

1. Close the current log file and open a new one by entering the following command:

```
$ REPLY/LOG
```

2. Set the default to the system disk by typing:

```
$ SET DEFAULT SYS$SYSTEM
```

3. Examine the directory [SYSMGR] to see what versions of the operator's log file are on disk using the command:

```
$ DIRECTORY [SYSMGR]OPERATOR.LOG;*
```

4. Obtain a printed copy of the operator's log file by issuing the following command:

```
$ PRINT [SYSMGR]OPERATOR.LOG;n
```

Note

The version number, n, must be 1 less than the highest version of this file.

Example

1. \$ REPLY/LOG
OPCOM, 16-JUL-1978 12:29:24.52, LOGFILE
INITIALIZED, OPERATOR=_TTA2:

```
$ SET DEFAULT SYS$SYSTEM
$ DIRECTORY [SYSMGR]OPERATOR.LOG;*
```

```
DIRECTORY DB22:[SYSMGR]
16-JUL-78 12:50
```

```
OPERATOR.LOG;202 1.      16-JUL-78 12:29
OPERATOR.LOG;201 1.      16-JUL-78 12:21
OPERATOR.LOG;200 5.      15-JUL-78 17:02
OPERATOR.LOG;199 1.      15-JUL-78 14:05
```

```
TOTAL OF 8./186.  BLOCKS IN 4.  FILES
```

```
$ PRINT [SYSMGR]OPERATOR.LOG;201
```

The REPLY/LOG command closes the current log file and opens a new one; the response from OPCOM verifies that a new log file has been opened. The SET DEFAULT command sets the operator's default disk to the system disk, thus enabling the operator to examine the files contained in the directory [SYSMGR]. The operator then requests that the second most recent version of the file OPERATOR.LOG be printed by use of the PRINT command.

VAX/VMS OPERATING PROCEDURES

2.5 SHUTTING DOWN AND RESTARTING THE VAX/VMS OPERATING SYSTEM

Occasionally, the system operator may need to shut down and restart the VAX/VMS operating system to ensure that it runs properly. Reasons for halting the system include danger of power loss, the need to back-up the system disk, a hardware or software problem, or simply no need to use the computer system. The operator restarts the system once the problem is solved or when there is a need to use the system again.

Shutting Down the Operating System: This section describes two procedures by which the operator can shut down the system:

1. An Orderly Shutdown of the System
2. An Emergency Shutdown of the System

The first procedure is a command file distributed with the VAX/VMS software. This command file, named SHUTDOWN, resides in the SYS\$SYSTEM directory. Once invoked, SHUTDOWN performs certain housekeeping functions that ensure a smooth shutdown of the system. These housekeeping functions include disabling future logins, stopping the batch and device queues, dismounting mounted volumes, and stopping user processes. This file also invokes a site-specific command procedure named SYSHUTDOWN.COM that is tailored by the system manager to the needs of the specific installation. This file is furnished without contents in the distribution kit.

If the operating system cannot be shut down by means of the SHUTDOWN command file, the operator must perform the second procedure, An Emergency Shut Down of the System. This procedure shuts down the system immediately without performing any of the housekeeping functions; hence, data may be lost. For this reason, this procedure should only be used if the first procedure fails.

Restarting the Operating System: Restarting the system means loading the operating system into memory and performing the necessary housekeeping functions for the system to run smoothly. Generally, when the system crashes, it automatically restarts itself. However, sometimes operator intervention is required to restart the system. This usually occurs after the operator has halted the operating system by one of the methods described above.

When the operating system is booted, the command file STARTUP is automatically executed. This start-up file contains commands for performing site-independent operations that must be performed if the system is to run properly. These operations include assigning system-wide logical names, installing executable images as known images and creating permanent global sections. This file also invokes a site-specific command procedure file named SYSTARTUP.COM in which the system manager places site-specific initialization commands. This file is furnished without contents in the distribution kit.

Sometimes, the operating system does not boot after the operator has entered the restarting command. This can be caused by either a software or hardware malfunction. A read error on a floppy diskette or a machine check error indicate a hardware malfunction; DIGITAL field service should be contacted.

VAX/VMS OPERATING PROCEDURES

If, however, the operating system is loaded into memory, but the STARTUP command file is not executed, a software malfunction has probably occurred. The operator can perform one or more of the following actions to correct the situation:

1. Repeat the start-up procedure
2. Start the system from an alternative drive or disk pack
3. Restart the LSI-11 console by pressing the BREAK key, followed by typing 177300G in response to the @ prompt

2.5.1 Orderly Shutdown of the System

This procedure describes how to shut down the system in an orderly fashion. This procedure is contained in a command file and can be modified at the discretion of the system manager or operator.

Procedure

1. Type the following command to begin the shutdown procedure:

```
$ @SYS$SYSTEM:SHUTDOWN
```

2. Enter a number in response to the following question:

```
HOW MANY MINUTES UNTIL SHUTDOWN?
```

3. Give the reason for shutting down the system in response to the following prompt:

```
REASON?
```

The execution of the shutdown procedure continues and the following events occur and are printed on the terminal:

- a. All future nonoperator logins are disabled.
- b. The batch and device queues are stopped when there are 4 minutes left until system shutdown.
- c. A message that requests users to log out is broadcasted to all users on the system. This message is broadcasted at decreasing time intervals.
- d. The site-specific command procedure SYSHUTDWN.COM is invoked. This command procedure contains commands inserted by the system manager to tailor the shutdown procedure to the needs of the installation.
- e. All mounted volumes are dismounted.
- f. All user processes are stopped; the following processes are not stopped: NULL, SWAPPER, JOB_CONTROL, OPCOM, ERRFMT, and the process running the SHUTDOWN command procedure.
- g. The operator's log file is closed.
- h. The program SYS\$SYSTEM:OPCCRASH is invoked to shut down the system. This program performs a bug check and dumps the contents of the registers, the kernel/interrupt stacks, and the executive stack and the process privileges.
- i. The system is halted.

VAX/VMS OPERATING PROCEDURES

Example

1. \$ @SYS\$SYSTEM:SHUTDOWN

SYSTEM SHUTDOWN COMMAND PROCEDURE.

27-JUN-1978 08:27:08
HOW MANY MINUTES UNTIL SHUTDOWN?: 5
REASON?: WEEKLY SYSTEM DISK BACKUP
Login quotas - Interactive limit=64, Current interactive value=4

NON-OPERATOR LOGINS ARE DISABLED.

STOP RUNNING QUEUES.

OPA0:,SYSTEM 08:27:54.94
"SYSTEM SHUTDOWN IN 4 MINUTES. LOGINS ARE DISABLED, PLEASE LOGOFF

OPA0:,SYSTEM 08:27:58.87
"WEEKLY SYSTEM DISK BACKUP

OPA0:,SYSTEM 08:30:01.37
"SYSTEM SHUTDOWN IN 2 MINUTES. LOGINS ARE DISABLED, PLEASE LOGOFF

OPA0:,SYSTEM 08:30:05.30
"WEEKLY SYSTEM DISK BACKUP

OPA0:,SYSTEM 08:31:07.82
"SYSTEM SHUTDOWN IN 1 MINUTE. LOGINS ARE DISABLED, PLEASE LOGOFF

OPA0:,SYSTEM 08:31:11.73
"WEEKLY SYSTEM DISK BACKUP

SYSTEM SHUTDOWN IMMINENT.

INVOKE INSTALLATION DEPENDENT SHUTDOWN PROCEDURE

DISMOUNT ALL MOUNTED VOLUMES

STOP ALL "USER" PROCESSES

Opcom, 08:31:28.94, SYSTEM Accnt=
Opcom, _OPA0:, "OPERATOR REQUESTED SHUTDOWN"

Opcom, 27-JUN-1978 08:31:32.95, logfile closed, operator=_OPA0:

**** FATAL BUG CHECK OPERATOR, Operator requested system crash

VAX/VMS OPERATING PROCEDURES

CURRENT PROCESS = SYSTEM

R0 = 00000024
R1 = 00000000
R2 = 00000004
R3 = 7FFE9ABE
R4 = 800A5FD0
R5 = 7FFEE484
R6 = 31000513
R7 = 7FFEF99C
R8 = 7FFEF900
R9 = 7FFEF98C
R10 = 7FFEF9A4
R11 = 7FFE7810
AP = 00000000
FP = 7FFEEDD0
SP = 7FFEEDC8
PC = 00000218
PSL = 00C00000

KERNEL/INTERRUPT STACK

7FFEEDD0 00000000
7FFEEDD4 00000000
7FFEEDD8 7FFDE178
7FFEEDDC 7FFEEDE4
7FFEEDE0 800120FA
7FFEEDE4 00000000
7FFEEDE8 00000000
7FFEEDEC 7FFDE178
7FFEEDF0 7FFDE160
7FFEEDF4 80015EB7
7FFEEDF8 80000096
7FFEEDFC 03C00000

EXEC STACK

PROCESS PRIVILEGES

800ABA00 FFFFFFFF
800ABA04 FFFFFFFF

IMAGE NAME = DMA0:[10,40]OPCCRASH.EXE;8

HALT INST EXECUTED
HALTED AT 800131CD

>>>

In this example, the operator requests that the system be shut down in 5 minutes to allow a system disk back-up. After indicating when and why the system will be shut down, the housekeeping functions that ensure a clean shutdown are performed and the system is halted at 800131CD.

2.5.2 Emergency Shutdown of the System

This procedure describes how to halt the system immediately without performing any of the housekeeping functions that ensure a smooth shutdown. Generally, the operator shuts down the system by following the procedure described on the previous pages. If, however, that procedure fails, the operator must perform an emergency shutdown.

To perform this procedure, the operator must enter the commands from the system console.

Procedure

1. Enter the following command to force an immediate shutdown of the system:

```
$ RUN SYS$SYSTEM:OPCCRASH
```

Note

Step 2 is ignored if the RUN SYS\$SYSTEM:OPCCRASH command is successful.

2. Enter the following two LSI-11 console commands if the system does not respond to the above command:

```
CTRL/P
>>>HALT
```

The CTRL/P command places the system console under the control of the console program instead of the VAX/VMS operating system. The HALT command halts the system completely.

3. Observe the following information typed on the system console:

```
HALTED AT "address"
```

The console program reports that the system is halted by typing the system's program counter at the time of the halt.

Example

1. CTRL/P
>>>HALT

```
HALTED AT 800172C0
```

In this example, the operator types CTRL/P followed by HALT, and the system stops at 800172C0.

VAX/VMS OPERATING PROCEDURES

2.5.3 Restarting the System

The procedure describes how to restart the VAX/VMS operating system after it has been shut down and consequently needs to be explicitly booted. To perform this procedure, the operator must enter the commands from the system console.

Procedure

1. Enter the following command to boot the system:

```
>>>BOOT [device-name:]
```
2. Observe the bootstrap information printed on the system console.

Example

1. >>>BOOT DM0:

```
CPU HALTED  
INIT SEQ DONE  
G 0000000E 00000200  
LOAD DONE 00001600 BYTES LOADED  
  
VAX/VMS BASE LEVEL 6.0 02-AUG-1978 08:25  
  
OPCOM, 02-AUG-1978 14:36:04.89, LOGFILE INITIALIZED, OPERATOR=_OPA0:
```

In this example, after the operator enters the BOOT command, the operating system is loaded into memory and the STARTUP file is executed. Upon completion, the system is initialized and ready to accommodate system users.

CHAPTER 3

COMMANDS USED BY OPERATORS

This chapter contains descriptions of some of the commands used by operators of the VAX/VMS system. The use of most (but not all) of the commands described in the chapter is restricted to users who have the operator privilege (OPER); usually, only system operators and the system manager have this privilege. Table 3-1 lists the commands described in this chapter.

For information about using the commands that are not described in this chapter and about running utility programs, see the following sources:

- The VAX/VMS Command Language User's Guide, which contains both a general introduction to the use of VAX/VMS commands and detailed descriptions of the VAX/VMS commands that are not described in this chapter.
- The VAX-11 Disk Save and Compress User's Guide, which contains a detailed description for running the DSC utility.

3.1 SUMMARY OF COMMANDS USED BY OPERATORS

Table 3-1 briefly describes the commands used by operators. The sections, following the table provide more detailed descriptions, with examples, of the commands.

Table 3-1
Commands Used by Operators

Command	Function
ASSIGN/MERGE	To remove the jobs from one queue and place them in another queue
ASSIGN/QUEUE	To assign a logical queue to a device
DEASSIGN/QUEUE	To deassign a queue from a device
DELETE/QUEUE	To delete batch queues and output queues

(continued on next page)

COMMANDS USED BY OPERATORS

Table 3-1 (Cont.)
Commands Used by Operators

Command	Function
INITIALIZE/QUEUE	To create batch queues and output queues
MCR BAD	To locate and count the bad blocks contained on Files-11 disks
MCR DSC1	To transfer files contained on Files-11 Structure Level 1 disks to tapes or disks for back-up and storage
MCR DSC2	To transfer files contained on Files-11 Structure Level 2 disks to tapes or disks for back-up and storage
MCR VFY1	To check the readability and validity of Files-11 Structure Level 1 disks
MCR VFY2	To check the readability and validity of Files-11 Structure Level 2 disks
REPLY	To communicate with system users, to selectively enable and disable operator status, and to examine the log file
RUN SYS\$SYSTEM:INSTALL	To install known images
RUN SYS\$SYSTEM:SYE	To create an error log report from a binary formatted file
SET ACCOUNTING	To selectively enable or disable the recording of particular kinds of accounting information
SET DEVICE	To establish the spooling and error logging status on a specified device
SET LOGINS	To establish the maximum number of users able to log into the system
SET PRINTER	To establish the characteristics of a specified line printer
SET PROTECTION/DEVICE	To establish the protection for a shareable, non-file-structured device

(continued on next page)

COMMANDS USED BY OPERATORS

Table 3-1 (Cont.)
Commands Used by Operators

Command	Function
SET TERMINAL/PERMANENT	To modify the permanent characteristics of a specified terminal
SET UIC	To establish a new user identification code as the process UIC
SHOW TERMINAL/PERMANENT	To display the current permanent characteristics of a specified terminal
START/QUEUE	To start batch queues and output queues
STOP/ABORT	To stop the printing of a job that is currently being printed
STOP/ENTRY	To stop the execution of a batch job that is currently running
STOP/QUEUE	To suspend batch queues and output queues
STOP/REQUEUE	To stop the printing of a job that is currently being printed and requeue that job at the end of the queue

ASSIGN/MERGE

3.2 ASSIGN/MERGE

Removes jobs from one queue and places them in another queue. The original queue is then eligible for deletion.

Format

<pre> ASSIGN/MERGE device-name[:] queue-name[:] <u>Command Qualifiers</u> None. </pre>

Prompts

Device: device-name[:]

Log_Name: queue-name[:]

Command Parameters

device-name[:]

Specifies the name of the line printer or terminal queue to which the files will be directed.

queue-name[:]

Specifies the name of the queue that is going to be reassigned to a new line printer or terminal.

Description

The ASSIGN/MERGE command merges one queue with another queue. However, before the merge operation can take effect, the source queue (that is, the queue to be merged) must be stopped by issuing the STOP/QUEUE/NEXT command. The target queue (that is, the queue that will accommodate the source queue) need not be stopped.

This command is particularly useful when an output device malfunctions; issue ASSIGN/MERGE to reroute the queue to a different output device. However, to perform the merge operation without losing or disrupting any user jobs, the operator should follow the procedure described in Section 2.2.1.

Example

1. \$ STOP/QUEUE/NEXT LPB0:
\$ ASSIGN/MERGE LPA0: LPB0:

The STOP/QUEUE/NEXT command stops the queue LPB0. The ASSIGN/MERGE command removes the jobs from the LPB0 queue and places them in the queue LPA0.

ASSIGN/QUEUE

3.3 ASSIGN/QUEUE

Assigns, or redirects, a logical queue to a line printer or to a terminal.

Format

<pre> ASSIGN/QUEUE device-name[:] logical-queue-name[:] <u>Command Qualifiers</u> None. </pre>
--

Prompts

Device: device-name[:]

Log_Name: logical-queue-name[:]

Command Parameters

device-name[:]

Specifies the physical name of a line printer or terminal that is to be associated with a logical queue.

logical-queue-name[:]

Specifies the name of the logical queue that is to be assigned to a line printer or terminal (a logical queue is a queue whose name does not match a physical device).

Description

Jobs in a logical queue cannot be printed unless the queue has been assigned to a physical device, that is, a line printer or a terminal. The ASSIGN/QUEUE command allows the operator to make this assignment. However, to print the jobs, the operator must then start the queue using the START/QUEUE command.

The ASSIGN/QUEUE command is not used when a physical queue (that is, a queue implicitly assigned to a physical device) is initialized and started.

Examples

```

1. $ INITIALIZE/QUEUE/FLAG/GENERIC TEST_QUEUE
   $ INITIALIZE/QUEUE/FLAG/GENERIC LPA0:
   $ ASSIGN/QUEUE LPA0: TEST_QUEUE
   $ START/QUEUE LPA0:
   $ START/QUEUE TEST_QUEUE
                
```

The INITIALIZE commands create the logical queue TEST_QUEUE and the physical queue LPA0. Both these queues are initialized with generic printing enabled and a header page preceding each job. However, because the logical queue is not associated with a printer, jobs entered in that queue do not have a device on which they can be printed. The ASSIGN/QUEUE command associates the logical queue TEST_QUEUE with the physical device LPA0; hence, printing can begin when the queues are started with the START/QUEUE commands.

COMMANDS USED BY OPERATORS

2. \$ INITIALIZE/QUEUE/GENERIC LPB0:
\$ START/QUEUE LPB0:

In this example, the ASSIGN/QUEUE command is not needed because a logical queue is not being initialized. The physical queue LPB0 is the name of a line printer; therefore, for printing to begin, the queue need only be started by use of the START/QUEUE command.

DEASSIGN/QUEUE

3.4 DEASSIGN/QUEUE

Deassigns a logical queue from a line printer or terminal.

Format

```
DEASSIGN/QUEUE logical-queue-name[:]
```

Command Qualifiers

None.

Prompts

Log_Name: logical-queue-name[:]

Command Parameters

logical-queue-name[:]

Specifies the name of a logical queue that was previously assigned to a line printer or to a terminal (a logical queue is a queue whose name does not match a physical device).

Description

This command takes effect only if the specified queue is idle. That is, the queue must be stopped by the STOP/QUEUE/NEXT command before the DEASSIGN/QUEUE command is issued.

Once the DEASSIGN/QUEUE command has been issued, the jobs in the logical queue remain in a hold state until the queue is deleted or reassigned to another physical device.

Example

```
1. $ ASSIGN/QUEUE LPA0: ASTER
    .
    .
    .
    $ STOP/QUEUE/NEXT ASTER
    $ DEASSIGN/QUEUE ASTER
```

The ASSIGN/QUEUE command associates the logical queue ASTER with the physical device LPA0. Later, the operator may decide to disassociate the queue ASTER from the device LPA0. First, the operator stops the logical queue using the STOP/QUEUE/NEXT command. Then, the queue can be deassigned with the DEASSIGN/QUEUE command.

DELETE/QUEUE

3.5 DELETE/QUEUE

Deletes a specified batch or output queue; hence, deletes all the jobs contained in that queue.

Format

```
DELETE/QUEUE queue-name[:]  
  
  Command Qualifiers  
  
  None.
```

Prompts

File: queue-name[:]

Command Parameters

queue-name[:]
 Specifies the name of the queue to be deleted.

Description

This command takes effect only if the specified queue is in an idle state; that is, the queue must be stopped by the STOP/QUEUE/NEXT command before a DELETE/QUEUE command is issued.

Example

```
1. $ INITIALIZE/QUEUE/FLAG LPA0:  
   $ START/QUEUE LPA0:  
     .  
     .  
     .  
   $ STOP/QUEUE/NEXT LPA0:  
   $ DELETE/QUEUE LPA0:
```

In the preceding sequence of commands the printer queue LPA0 is initialized and started. Later, the operator may decide to delete that queue. First, the operator stops the queue by using the STOP/QUEUE/NEXT; then, the operator deletes the queue by issuing the DELETE/QUEUE command.

INITIALIZE/QUEUE

3.6 INITIALIZE/QUEUE

Creates or initializes a batch queue or an output queue.

Format

INITIALIZE/QUEUE queue-name[:]	
<u>Command Qualifiers</u>	<u>Defaults</u>
/BATCH	/NOBURST
/[NO]BURST	/NODISABLE_SWAPPING
/[NO]DISABLE_SWAPPING	/ENABLE_GENERIC_PRINTING
/[NO]ENABLE_GENERIC_PRINTING	/NOFLAG
/[NO]FLAG	
/FORMS TYPE=n	
/[NO]GENERIC	/NOGENERIC
/JOB_LIMIT=val	/JOB_LIMIT=1
/PRIORITY=val	/PRIORITY=4
/TERMINAL	

Prompts

Queue: queue-name[:]

Command Parameters

queue-name[:]

Specifies the name of a physical device, the name of a generic queue, or a logical queue name.

Description

The INITIALIZE/QUEUE command creates batch and print queues and assigns them names and attributes. However, to print or run the jobs placed in the queues, the operator must issue the START/QUEUE command. All the options that can be specified in the INITIALIZE/QUEUE command can also be specified in the START/QUEUE command.

When the operator creates a batch queue, the qualifier /BATCH is required.

Command Qualifiers

/BATCH

Indicates that this is a batch queue.

/BURST

/NOBURST

Controls printing of a burst header page for each print job.

Use the /BURST qualifier to print a header page over the paper perforations; hence, the page header is visible from the side of a stack of paper. A burst header simplifies separating listings.

The default qualifier is /NOBURST.

COMMANDS USED BY OPERATORS

`/DISABLE_SWAPPING`

`/NODISABLE_SWAPPING`

Controls whether batch jobs executed from the queue can be swapped in and out of memory.

The default qualifier is `/NODISABLE_SWAPPING`.

`/ENABLE_GENERIC_PRINTING`

`/NOENABLE_GENERIC_PRINTING`

Controls whether files queued to a generic queue can be placed in a physical queue (that is, a device-specific queue) for printing.

By default, generic printing from a queue associated with a physical device is enabled.

`/FLAG`

`/NOFLAG`

Controls whether a header page is printed at the beginning of the first file in each print job.

The default qualifier is `/NOFLAG`.

`/FORMS_TYPE=n`

Specifies the forms type for the specified queue. Once a queue has been assigned a forms type, all jobs in that queue that do not conform to that forms type are placed in a holding state.

The forms type should be specified using a numeric value or an alphanumeric code. Codes for forms type are installation defined.

`/GENERIC`

`/NOGENERIC`

Controls whether files placed in a queue can be printed on any available line printer that has correctly matching characteristics.

If the qualifier `/GENERIC` is not specified, the queue is by default a nongeneric queue.

`/JOB_LIMIT=val`

Indicates the number of batch jobs that can be executed concurrently from the queue. The default value of the job limit is 1.

`/PRIORITY=val`

Indicates the priority at which jobs are initiated from the queue. The default value of the priority of a queue is 4.

`/TERMINAL`

Indicates that this is a terminal queue.

COMMANDS USED BY OPERATORS

Examples

1. \$ INITIALIZE/QUEUE/BATCH/JOB_LIMIT=4 SYS\$BATCH
\$ START/QUEUE SYS\$BATCH
\$ INITIALIZE/QUEUE/FLAG LPA0:
\$ START/QUEUE LPA0:

The preceding sequence of commands initializes and starts the batch queue SYS\$BATCH and the line printer queue LPA0. The /JOB_LIMIT=4 qualifier allows up to four jobs to be initiated concurrently from the batch queue. The /FLAG qualifier causes a header page to precede each job in the printer queue.

2. \$ INITIALIZE/QUEUE TESTER
\$ INITIALIZE/QUEUE/FLAG LPA0:
\$ ASSIGN/QUEUE LPA0: TESTER
\$ START/QUEUE TESTER
\$ START/QUEUE LPA0:

The preceding procedure causes all jobs queued to the logical queue TESTER to be printed on the physical device LPA0. First, the INITIALIZE commands initialize the logical queue TESTER and the physical queue LPA0. The physical queue LPA0 is enabled to print a header page at the beginning of the first file in each job. Next, the ASSIGN/QUEUE command associates the logical queue TESTER with the physical queue LPA0. After the START commands are issued, all jobs queued to TESTER or LPA0 will be printed on LPA0.

MCR BAD

3.7 MCR BAD

Determines the number and location of bad blocks on Files-11 disks.

Format

<p>MCR BAD device-name:</p> <p><u>Command Qualifiers</u></p> <p>None.</p>

Prompts

None.

Command Parameters

device-name:

Specifies a device containing a volume on which BAD will be run. The device name consists of a two-character device name, followed by a one- or two-digit number that specifies the unit code and a device name terminator (:).

Alphabetic controller designators in the device name are not supported.

Description

The BAD command analyzes the condition of the blocks on Files-11 disks. BAD writes a test pattern onto each block, reads the blocks into a buffer in storage, then compares the data in the buffer with the data that was written to the blocks. If the data does not compare exactly, the block is bad and cannot reliably store data.

When BAD locates a bad block, it records the address of the bad block in a separate buffer for reference. When all bad blocks on the disk are located, the addresses and extents of the bad blocks are recorded on the last good block of the disk. The disk must have at least one good block for BAD to execute properly.

The operator must mount the disk to be analyzed with the /FOREIGN qualifier. This qualifier indicates that the disk volume is in a format not recognized by the VAX/VMS operating system. (See the MOUNT command in the VAX/VMS Command Language User's Guide.)

On system disks, block zero usually contains the bootstrap record. However, the bootstrap record is not required by VAX/VMS and thus, if BAD prints a message warning the operator not to use the disk as a system disk, the message can be ignored.

While BAD analyzes a disk, that disk must not be accessed by any other programs.

COMMANDS USED BY OPERATORS

BAD stores up to 102 entries for addresses and extents of bad blocks on all disks except RK06 and RK07 disks. If more than 102 entries are recorded for a disk, BAD exits with an error message. For RK06 and RK07 disks, the maximum number of bad block extents that can be written is 126.

Example

```
1. $ ALLOCATE DBA2:
    DBA2: ALLOCATED
$ MOUNT/FOREIGN DBA2:
%MOUNT-I-MOUNTED      mounted on _DBA2:
$ MCR BAD DB2:
BAD -- TOTAL NO. OF BAD BLOCKS = 2.
```

The ALLOCATE command requests the allocation of a specific disk drive, DBA2. The response from the ALLOCATE command indicates that the device was successfully allocated. The MOUNT/FOREIGN command mounts the disk volume as a foreign disk. The MOUNT command response indicates that DBA2 was successfully mounted. The MCR BAD command analyzes each block on the disk volume and records the bad blocks. After BAD has tested all the blocks, it indicates that the number of bad blocks on DBA2 is 2.

MCR DSC

3.8 MCR DSC

Copies files contained on Files-11 Structure Level 1 and Files-11 Structure Level 2 disks to tapes or disks for back-up and storage, and reallocates and consolidates the disk area used for data storage.

Files-11 Structure Level 1 Format

<p>MCR DSC1 command string</p> <p><u>Command Qualifiers</u></p> <p>None.</p>
--

Files-11 Structure Level 2 Format

<p>MCR DSC2 command-string</p> <p><u>Command Qualifiers</u></p> <p>None.</p>
--

Prompts

If the operator does not enter the command string on the initial command line, DSC prompts for it (DSC>), and continues prompting until CTRL/Z is typed.

If the operator enters the command string on the initial command line, DSC executes the command, then returns control to the command interpreter.

Command Parameters

command-string

Specifies a single DSC command that controls how DSC transcribes the contents of a disk to a tape or disk.

The operator enters DSC commands in the following format:

```
outdev:[filelabel][/qualifier]=indev:[filelabel][/qualifier]
```

outdev:

Lists the physical device(s) to which data is transferred.

The format for outdev: is:

AAAn:

AAA ASCII characters specifying the device name followed by the alphabetic controller designator

n A 1-digit (octal) unit number

COMMANDS USED BY OPERATORS

: The device name terminator

When more than one tape drive is requested, the device symbols must appear in a comma-separated list.

[filelabel]

Is an optional 1- to 12-character alphanumeric name identifying the tape file created when files are transferred from disk to a set of magnetic tape volumes. If the file label is not specified when a tape file is created from disk, the disk volume label is used as the identifier.

/qualifier

Specifies one or more optional output qualifiers, as listed in Table 3-2.

Table 3-2
Output Qualifiers

Qualifier	Format	Meaning
Append -	/AP	Writes a file to the first specified volume of a tape set that contains only entire DSC-created files
Bad Block	/BAD = { MAN NOAUTO AUTO	Allows manual entry of bad block data to supplement the output disk's bad block file; ignores the bad block descriptor on the disk; uses manually entered bad blocks exclusively
Compare	/CMP	Compares the contents of the input device with the contents of the output device
Density	/DENS=1600	Creates magnetic tapes at 1600 bits per inch density
Nonmerge	/NMG	Copies files from disks without consolidating all the extents of a file
Rewind	/RW	Rewinds output tape volumes before DSC starts writing to them
Verify	/VE	Performs a copy operation, then compares the contents of both devices (input and output)

indev:

Lists the physical device(s) from which data is transferred. The format for indev: is:

AAAN:

AAA ASCII characters specifying the device abbreviation followed by the alphabetic controller designator

n A 1-digit (octal) unit number

COMMANDS USED BY OPERATORS

: The device name terminator

When more than one tape drive is requested, the device symbols must appear in a comma-separated list.

[filelabel]

Is an optional 1- to 12-character alphanumeric name identifying a DSC-created tape file to be transferred to the specified disk. If the file label is not specified, DSC copies onto the disk medium the first tape file it locates on the first tape volume specified.

[/qualifier]

Specifies one or more optional input qualifiers, as listed in Table 3-3.

Table 3-3
Input Qualifiers

Qualifier	Format	Meaning
Nonmerge	/NMG	Copies files from one disk to a second disk without consolidating all the extents of a file
Rewind	/RW	Rewinds first tape volume before the copy operation begins

Description

The DSC1 command transfers data from a Files-11 Structure Level 1 disk to another disk or set of magnetic tape volumes; the DSC2 command transfers data from a Files-11 Structure Level 2 disk to another disk or set of magnetic tape volumes. Both commands perform the same copy operation -- they copy only the blocks allocated to active files.

DSC's operation is a cycle, starting with data on a disk and concluding with that data, in compressed form, on another disk or set of tapes. The operation can be performed using one of the following methods:

- A direct disk-to-disk transfer
- A disk-to-tape transfer followed by a tape-to-disk transfer

DSC does not support data transfer from tape to tape.

For a complete description of the DSC utility, see the VAX-11 Disk Save and Compress User's Guide.

Table 3-4 lists the devices that can be used with DSC. Transfers occur between devices of the same type. Transfers from a smaller disk to a larger disk are allowed; however, transfers from a larger disk to a smaller one are not.

COMMANDS USED BY OPERATORS

Table 3-4
DSC-Supported Devices

Abbreviation	Device	Class
DM	RK06 or RK07 cartridge disk	Block-structured
DB	RP06 or RP05 disk pack	Block-structured
DR	RM03 disk pack	Block-structured
DX	RX01 console floppy diskette	Block-structured
MT	TE16 9-track magnetic tape	Tape

All volumes, except input disks, that DSC uses must be mounted with the /FOREIGN qualifier. Input disks can be mounted as either structured or foreign disks.

Examples

1. DSC> **MTA0:BACK_FILE/VE=DBA1:**

DSC writes the contents of the disk pack DBA1 to a single tape drive MTA0. After copying the data files from DBA1 to the tape file named BACK_FILE, DSC compares the contents of both devices to make certain the copy operation was successful.

2. DSC> **MTA1: ,MTA2:SYSFILE/DENS=1600=DBA0:**

In the above example, DSC writes the contents of the disk pack DBA0 to two TE16 tape drives, MTA1 and MTA2. The resulting tape volume set, recorded at a density of 1600 bits per inch, contains all the data files from DBA0 in the tape file named SYSFILE.

MCR VFY

3.9 MCR VFY

Checks the readability and validity of Files-11 Structure Level 1 and Files-11 Structure Level 2 volumes.

Files-11 Structure Level 1 Format

```
MCR VFY1 [listfile,scratchdev:=] indev:/qualifier
```

Command Qualifiers

None.

Files-11 Structure Level 2 Format

```
MCR VFY2 [listfile,scratchdev:=] indev:/qualifier
```

Command Qualifiers

None.

Prompts

If the operator does not enter the command string on the initial command line, VFY prompts for it (VFY>). and continues prompting until CTRL/Z is typed.

If the operator enters the command string on the initial command line, VFY executes the command, then returns control to the command interpreter.

Command Parameters

listfile

Specifies the output listing file in the following format:

```
dev:[ufd]filename.filetype;version
```

If this parameter is omitted, the output listing file is displayed on the terminal.

cratchdev:

Specifies the device on which the scratch file produced by VFY is to be written. This parameter has the following format:

```
dev:
```

If this parameter is omitted, SYS\$DISK: is automatically inserted.

VFY uses the scratch file during the verification scan and the lost file scan. The scratch file is created but not entered in a directory; hence, it is transparent to the operator. The scratch file is automatically deleted upon termination of the VFY program.

COMMANDS USED BY OPERATORS

If the system disk is faulty, this parameter should be used to force the scratch file to another device. The scratch file should always be assigned to another volume. The scratch file is not used for the /FR and /LI qualifiers.

indev

Indicates the volume to be verified. This parameter has the following format:

dev:

If this parameter is omitted, SYS\$DISK: is automatically inserted.

/qualifier

Specifies the function to be performed. If no qualifier is indicated, VFY validates the structure of the volume mounted on the specified device. Table 3-5 summarizes the qualifiers; see Appendix A for a complete description of each qualifier.

Table 3-5
File Qualifiers

Qualifier	Format	Meaning
Delete	/DE	Resets marked-for-deletion indicators
Free	/FR	Indicates the number of available blocks on the volume, the number of used blocks on the volume, and the total number of blocks on the volume
List	/LI	Lists the entire index file by file identification
Lost	/LO	Scans the entire file structure for files that are not in any directory
Read Check	/RC[:n]	Checks readability of every block of every file on the entire volume
Rebuild	/RE	Recovers blocks that appear to be allocated but are not contained in a file
Update	/UP	Allocates blocks that appear to be available but also have been allocated to a file
Validity Check	DEFAULT	Validates the structure of the volume mounted on specified device

COMMANDS USED BY OPERATORS

Description

By default, VFY validates the structure of Files-11 Structure Level 1 and Files-11 Structure Level 2 volumes. When the operator specifies a particular qualifier, VFY executes the function of that qualifier.

Only one qualifier can be specified for a command; if more than one qualifier is supplied, an error results.

No other activity should occur on the volume while VFY is executing. In particular, activities that create new files, extend existing files, or delete files should not be attempted while VFY is executing a function.

VFY must not be aborted while a /UP, /RE, or /DE qualifier is being processed. Aborting VFY while it is in the process of modifying the storage allocation or index files may seriously endanger the integrity of that volume.

Examples

1. \$ MCR VFY2 DMA2:/LO

0. LOST FILES WERE FOUND

The /LO qualifier scans the file structure of the volume mounted on DMA2 looking for files that are not entered in any directories. In this example, no files were found (that is, no files were lost).

2. \$ MCR VFY2 DMA0:/FR

DM0: HAS 2768. BLOCKS FREE, 24358. BLOCKS USED OUT OF 27126.

The /FR qualifier indicates that there are 2768 blocks available for use on the volume mounted on DMA0 and that 24358 blocks have been used out of 27126 blocks.

REPLY

3.10 REPLY

The REPLY command:

- Displays a message at a user's terminal
- Responds to user requests
- Designates a terminal to be an operator's terminal
- Restores a terminal to nonoperator status
- Closes the current operator log file and opens a new one

Format

```

REPLY      "message-text"

  Command Qualifiers

  /ABORT=identification-number
  /ALL
  /BELL
  /DISABLE [= (keyword [,...])]
  /ENABLE [= (keyword [,...])]
  /[NO]LOG
  /PENDING=identification-number
  /STATUS
  /TERMINAL=(terminal-name [,...])
  /TO=identification-number
  /USER
    
```

Prompts

None.

Command Parameters

message-text

Indicates the message to be displayed at the specified user's terminal(s).

The maximum length of a message is 128 characters; if this limit is exceeded, an error results.

When a message consists of more than one word, the message should be enclosed in double quotation marks.

Description

The REPLY command is the tool used by the operator to communicate with system users, to create and inhibit operator status on terminals attached to the system, and to control the operator's log file.

COMMANDS USED BY OPERATORS

Responding to User Requests: When a user issues the REQUEST command, the operator responds using one of the following REPLY commands:

- REPLY/ABORT=identification-number "message-text"
- REPLY/PENDING=identification-number "message-text"
- REPLY/TO=identification-number "message-text"

The /ABORT qualifier indicates that the user request has been canceled. The /PENDING qualifier places the user in a waiting state until the request can be fulfilled or aborted. The /TO qualifier indicates that the user request has been fulfilled.

When a user issues the REQUEST/REPLY command, the message appears on the system console as follows:

```
OPCOM, hh:mm:ss:cc user-name ACCNT=account REPLY-ID=identification-number
OPCOM, terminal-name:, "message-text"
```

The user cannot enter any commands until the operator responds using the /ABORT or /TO qualifier or the user aborts the request. If the operator does not respond and the user does not abort the request, the request is repeated at five minute intervals on the operator's terminal until the operator replies.

Broadcasting Messages to Terminals: To contact one or more system users, the operator issues one of the following REPLY commands:

- REPLY/ALL
- REPLY/TERMINAL=(terminal-name[,...])
- REPLY/USER

The /ALL qualifier sends a message to all terminals that are attached to the VAX/VMS system and are turned on. The /TERMINAL qualifier sends a message to one or more specified terminals on the system. The /USER qualifier sends a message to all terminals at which system users are logged in. Generally, to broadcast an important message, such as a message that indicates a system shutdown, the operator specifies the /ALL qualifier.

Enabling and Disabling Operator Status on a Terminal: Any terminal, except a remote terminal, attached to the VAX/VMS operating system can be established as an operator's terminal. The operator simply enters the command REPLY/ENABLE on the designated terminal, and that terminal can then respond to user requests and monitor device status.

When the operator enters the REPLY/ENABLE command, OPCOM confirms that the terminal has been enabled by displaying the following message:

```
OPCOM, dd-mmm-yyyy hh:mm:ss.cc, TERMINAL ENABLED, OPERATOR=operator-name

OPCOM, hh:mm:ss.cc, OPERATOR TERMINAL STATUS, operator-name
CENTRAL, PRINTER, TAPES, DISKS, DEVICES, CARDS, NETWORK.
OPER (1,2,3,4,5,6,7,8,9,10,11,12)
```

COMMANDS USED BY OPERATORS

When the operator enters the `REPLY/DISABLE` command, OPCOM confirms that the terminal is no longer an operator terminal by displaying the following message:

```
OPCOM, hh:mm:ss.cc, TERMINAL DISABLED, OPERATOR=operator-name
```

To grant partial operator status on a particular terminal, the operator includes one or more keywords after the `/ENABLE` qualifier. For example, to establish a terminal as an operator terminal that receives messages pertaining to mounting and dismounting tapes, the operator types:

```
$ REPLY/ENABLE=TAPES
```

```
OPCOM, 1-AUG-1978 18:40:28.37, TERMINAL ENABLED, OPERATOR=_TTE3:
$
OPCOM, 18:40:31.47, OPERATOR TERMINAL STATUS, _TTE3:
TAPES,
```

OPCOM confirms that the terminal has operator status for tape messages.

To discontinue communication with a particular facility, the operator includes one or more keywords after the `/DISABLE` qualifier. For example, to inhibit an operator terminal from receiving messages pertaining to mounting and dismounting disks, the operator types the following command:

```
$ REPLY/DISABLE=DISKS
```

```
OPCOM, 18:44:04.48, OPERATOR TERMINAL STATUS, _TTE5:
CENTRAL, PRINTER, TAPES, DEVICES, CARDS, NETWORK,
OPER(1,2,3,4,5,6,7,8,9,10,11,12)
```

Note that OPCOM lists the partial operator status still assigned to the terminal.

When an operator disables operator status on all terminals, including the system console, OPCOM records all subsequent user requests in the operator's log file and responds to these requests with the following message:

```
%OPCOM-S-OPRNOTIF, operator notified, waiting. hh.mm.ss.cc
%OPCOM-S-OPREPLY, no operator response text
%OPCOM-S-NOOPERATOR, no operator coverage
```

To determine the operator status of a particular terminal, the operator types the following command:

```
$ REPLY/STATUS
```

Closing the Log File: To close the current log file and open a new one, the operator issues the `REPLY/LOG` command. All subsequent messages are recorded in the new log file for future reference. To close the current log file without opening a new one, the operator issues the `REPLY/NOLOG` command. All subsequent messages are not recorded until the operator issues the `REPLY/LOG` command.

For more information about the operator's log file, see Chapter 4. To examine the contents of the log file, use the procedure documented in Chapter 2.

COMMANDS USED BY OPERATORS

Command Qualifiers

/ABORT=identification-number
Sends a message to the user indicated by the identification number and aborts the request.

/ALL
Broadcasts a message to all terminals attached to the system.

/BELL
Rings a bell at each terminal that receives the message. This qualifier should be specified with any one of the following qualifiers:

/ALL
/TERMINAL=(terminal-name [,...])
/USER

/DISABLE[(keyword [,...])]
Restores a terminal at which the command is issued or whose name is specified to normal status (that is, nonoperator status). This qualifier can impose a limit on the types of messages displayed on an operator's terminal, by specifying one or more of the following keywords:

<u>Keyword</u>	<u>Meaning</u>
CENTRAL	Inhibits receiving messages sent to the operator identified as the central system operator
DEVICES	Inhibits receiving messages pertaining to mounting disks and tape volumes
DISKS	Inhibits receiving messages pertaining to mounting and dismounting disk volumes
OPER1 THROUGH OPER12	Inhibits receiving messages sent to operators identified as OPER1, OPER2,... OPER12
PRINTER	Inhibits receiving messages pertaining to printer requests
TAPES	Inhibits receiving messages pertaining to mounting and dismounting tape volumes

If no keywords are specified, all operator characteristics are disabled. If more than one keyword is specified, separate them with commas and enclose the entire list in parentheses.

COMMANDS USED BY OPERATORS

`/ENABLE=(keyword [,...])`

Designates the terminal at which the command is issued to be an operator's terminal. To designate a terminal to receive messages from a particular facility, specify one or more of the following keywords:

<u>Keyword</u>	<u>Meaning</u>
CENTRAL	Designates a terminal to receive messages sent to the central system operator
DEVICES	Designates a terminal to receive messages pertaining to mounting disks and tape volumes
DISKS	Designates a terminal to receive messages pertaining to mounting and dismounting disk volumes
OPER1 THROUGH OPER12	Designates a terminal to receive messages sent to an operator identified as either OPER1, OPER2, ... OPER12
PRINTER	Designates a terminal to receive messages pertaining to printer requests
TAPES	Designates a terminal to receive messages pertaining to mounting and dismounting tape volumes

If no keywords are specified, all operator characteristics are prescribed. If more than one keyword is specified, they should be separated with commas and the entire list should be enclosed within parentheses.

`/LOG`

`/NOLOG`

The `/LOG` qualifier closes the current log file and opens a new one. The operator can examine the contents of the previous log file.

The `/NOLOG` qualifier closes the current log file but does not open a new one. Hence, no messages are recorded until the operator opens another log file.

`/PENDING=identification-number`

Sends a message to the user indicated by the identification number and places the request in a wait state.

`/STATUS`

Reports the current operator status and all outstanding user requests for the terminal at which this command was entered.

`/TERMINAL=(terminal-name[,...])`

Broadcasts a message to one or more specified terminals.

`/TO=identification-number`

Sends a message to the user indicated by the identification number and completes the request.

`/USER`

Broadcasts a message to all terminals at which users are logged into the system.

COMMANDS USED BY OPERATORS

Examples

1. \$ **REPLY/ALL "SYSTEM GOING DOWN FOR BACK-UP. PLEASE LOG OFF."**

The REPLY/ALL command broadcasts a message to all terminals on the system.

2. \$ **REPLY/ENABLE=DISKS**
OPCOM, 14-JUN-1978 10:21:56.04, TERMINAL ENABLED, OPERATOR=_TTA3:
\$
OPCOM, 10:22:01.03, OPERATOR TERMINAL STATUS, _TTA3:
DISKS,

The REPLY/ENABLE command designates the terminal at which this command is issued (terminal TTA3) to be an operator's terminal that receives messages pertaining to mounting and dismounting disks. The OPCOM message confirms that terminal TTA3 is established as an operator's terminal.

3. OPCOM, 13:01:43.22, RALPH ACCT=DOCUMENT REPLY-ID=57
OPCOM, TTA1:, "PLEASE MOUNT MYFILE ON DBA3:"
\$ **REPLY/PENDING=57 "YOU'LL HAVE TO WAIT-THERE ARE SEVERAL REQUESTS BEFORE YOUR REQUEST"**
\$ **REPLY/TO=57**
OPCOM, 13:11:14.55, REQUEST COMPLETED, OPERATOR TTA4: ID=57

The OPCOM message indicates that a user wants the operator to place the disk volume with the visual label MYFILE on the disk drive DBA3 and ready the device. The REPLY/PENDING command indicates that the operator can perform the task but not immediately. The /PENDING qualifier places the user's terminal in a wait state; that is, the user cannot enter commands at the terminal. The operator later mounts the specified disk on the specified drive, and sends a message to the user indicating the request has been fulfilled. In this example, no message accompanies the REPLY/TO command; hence, the MOUNT command sends a standard message indicating that the task has been performed.

4. \$ **REPLY/STATUS**
OPCOM, 15:25:43.73, OPERATOR TERMINAL STATUS, _TTE3:
CENTRAL, PRINTER, TAPES, DISKS, DEVICES, CARDS, NETWORK
OPER(1,2,3,4,5,6,7,8,9,10,11,12)

The REPLY/STATUS command requests the operator terminal status for TTE3. The response from OPCOM indicates that TTE3 is enabled to receive messages from all devices.

RUN SYS\$SYSTEM:INSTALL

3.11 RUN SYS\$SYSTEM:INSTALL

Installs executable images as known images, creates permanent global sections by installing shareable images as known images, and deletes known images from the system's list of known images.

Format

```

RUN SYS$SYSTEM:INSTALL<RET>
* file-spec option,...<RET>
.
.
.
* <CTRL/Z> or <CTRL/Y> or <CTRL/C>
    
```

Prompts

After the command line RUN SYS\$SYSTEM:INSTALL is entered in response to the system prompt, the INSTALL utility program prompts (with an asterisk) for the file specification of an image file that is to be installed as a known image. This file specification must be followed by one or more of the options described below. When a file specification and the option or options associated with it are entered, INSTALL responds with another prompt for a file specification. Images that are activated less frequently should be installed first; images that are activated more frequently should be installed last. After the last file specification and its options have been entered, the execution of INSTALL can be terminated by entering <CTRL/Z> or <CTRL/Y> or <CTRL/C> in response to the prompt.

Program Parameters

file-spec

Specifies 1) the image file that is to be installed as a known image, 2) a known image file that is to be replaced by the latest version of that file, or 3) a known image file that is to be deleted from the system's list of known images.

INSTALL accepts a full file specification or simply a file name. If the latter is entered in response to a program prompt, the following defaults are applied to form a full file specification:

- Logical device and directory name: SYS\$SYSTEM
- File type: EXE
- File version: highest

COMMANDS USED BY OPERATORS

Description

The INSTALL utility program is used in installing images as known images. Known images are images that can have the following characteristics that set them apart from other images:

- They can be shared by all users of the system
- They can be quickly accessible with a minimum of disk overhead
- They can require privileges and yet be executed by users who normally have few if any privileges

To run the INSTALL utility, a user needs the CMKRNL privilege to execute the Change Mode to Kernel system service, the PRMGBL privilege to create permanent global sections, and the SYSGBL privilege to create system global sections. Usually, the system manager or the system operator runs this program.

Commonly, images are installed from the system device, the volume on which the operating system resides. Images can, however, be installed from other volumes.

Thus, a system may have more than one known file list. Each known file list contains the names of known image files that have the same device, directory, and file type. Thus all known image files with the device and directory name SYS\$SYSTEM and the file type EXE are included in one known file list. All known image files with the device and directory name SYS\$LIBRARY and the file type EXE are included in a second known file list.

At start-up time, the number of known file lists that can be established on a system can be specified by use of the Known File List Count parameter, which is described in the VAX-11 Software Installation Guide.

If images are installed as shared images from a volume other than the system device, this volume cannot be dismounted until the shared images are deleted from the appropriate known file list (by use of the /DELETE option). The option /LIST is used to determine which images must be deleted. Because the deletion of a shared image from the known file list cannot be completed until the last process using that image exits from it, the dismounting of a volume containing shared images may be delayed for the same reason.

The system's lists of known images must be restored each time the system is restarted after it crashes or is shut down. Thus, a command to run the INSTALL program is usually included in a system start-up command procedure.

Refer to the VAX/VMS System Manager's Guide for more on known images and permanent global sections.

COMMANDS USED BY OPERATORS

Options

/DELETE

Removes the specified file from the system's list of known images. Note that when a shared image is deleted from the known file list, its associated global sections become unavailable and its associated open file is closed only after the last process using that image exits from the image.

/FULL

Displays statistics about the system resources being used by the image files contained in the system's known file lists.

/GLOBAL

Displays statistics about the system resources being used by permanent and temporary global sections.

/HEADER_RESIDENT

Causes the header of the specified image file to remain permanently resident in memory.

This option is specified for frequently used files such as BACKTRANS.EXE (which is the back translator of DCL commands to MCR commands). Keeping the header of an image file permanently in memory saves one access to disk whenever the file is referred to, at a cost of less than one page of paged dynamic memory.

/LIST

Displays the names and brief descriptions of the image files contained in the system's known file lists. This option is often used with the option /FULL or /GLOBAL or both.

/OPEN

Causes the specified image file to remain permanently open; this option is specified for frequently used utilities, such as SOS and SHOW. The cost is about 160 (decimal) bytes of nonpaged dynamic memory.

If a file is installed /SHARED, /HEADER_RESIDENT, or /WRITABLE, it is open by default. If a file from a volume other than the system device is installed /PRIVILEGED, it is open by default.

A program file can be installed without being open; in that case, it is opened each time the program is activated.

/PRIVILEGED[=(privilege-name,...)]

Allows images that need special privileges to be executed to be run by users who do not normally have these privileges. LOGINOUT.EXE and DISMOUNT are two such images.

The privileges that an image needs should be included in the optional list of privilege names. Privilege names are documented in the VAX/VMS System Manager's Guide. If the optional list of privilege names is omitted, all privileges are given to the specified image.

/REPLACE

Replaces the specified image file in the system's list of known images with the latest version of that file; all previously specified characteristics of the image file are retained.

COMMANDS USED BY OPERATORS

/SHARED

Allows more than one user to share a single copy of the read-only segments of a specified executable image file and creates a permanent global section from a specified shareable image file.

This option can be specified for native-mode executable image files (such as PRTSMB.EXE, the executable image of the print symbiont) and for native-mode shareable image files (such as SYS\$LIBRARY:VMSRTL.EXE, the shareable image of the VAX-11 Common Run-Time Procedure Library).

In addition, this option can be specified for some compatibility-mode utilities, commons, and libraries built by the RSX-11M Task Builder.

/WRITABLE

Allows more than one user to share the read/write data areas of the specified shareable image file; the /SHARED option must be specified with the /WRITABLE option.

This option can be specified for native-mode FORTRAN programs that have common regions. Such programs must be linked /SHAREABLE. These common regions are accessible to any user program that maps into them, that is, to any user program that is linked with the specified shareable image.

In addition, this option can be specified for some compatibility-mode tasks built with read/write common areas by the RSX-11M Task Builder.

Examples

1. \$ **RUN SYS\$SYSTEM:INSTALL**
* **BACKTRANS/OPEN/HEADER_RESIDENT/SHARED**

In this example, the executable image of the back translator of DCL commands to MCR commands is installed as a known image file that is permanently open, whose header is permanently resident in memory, and whose read-only segments can be used by more than one user at a time. Note that by default the device and directory name SYS\$SYSTEM and the file type EXE are applied to the file name BACKTRANS.

2. \$ **RUN SYS\$SYSTEM:INSTALL**
* **SYS\$LIBRARY:VMSRTL/OPEN/SHARED**

In this example, the shareable image of the VAX-11 Common Run-Time Procedure Library is installed as a set of permanent global sections. Note that SYS\$LIBRARY, the name of the device and directory of this shareable image file, must be specified.

RUN SY\$SYSTEM:SYE

3.12 RUN SY\$SYSTEM:SYE

Creates an error log report from a file containing error data in binary format.

Format

```
$ RUN SY$SYSTEM:SYE
```

Prompts

Input File: input-file-spec
 Output File: output-file-spec
 Option: report-option
 Device Name: device-name[:]
 After: date time
 Before: date time

Program Parameters

input-file-spec

Specifies an error log file created by the ERRFMT process. By default, SYE uses the highest version of the [SYSERR]ERRLOG.SYSfile, which resides on the system disk.

When the user specifies an input file, SYE defaults any omitted fields to the defaults for FORTRAN unit 2, that is, the highest version of SY\$DISK:[default-directory]FOR001.DAT.

output-file-spec

Specifies the file to contain the error log report. By default, SYE sends the output to SY\$OUTPUT, which is usually the user's terminal.

If the user specifies an output file, SYE defaults any omitted fields to the defaults for FORTRAN unit 2, that is, the highest version of SY\$DISK:[default-directory]FOR002.DAT.

report-option

One of the following report options:

```
R Roll up
C Cryptic
B Brief
S Standard
U Unknown
```

By default, SYE uses the Roll up (R) option when none other is specified.

The five report options are explained below in the Description section.

device-name[:]

Instructs SYE to report only errors encountered by the specified device type or device unit. By default, SYE reports errors on all devices.

COMMANDS USED BY OPERATORS

date time

Instructs SYE to report on errors occurring after a specified date and time (After: absolute time) and/or before a specified date and time (Before: absolute time).

By default, SYE reports errors after November 17, 1858 and before December 31, 9999.

Description

The types of reports that SYE can generate are as follows:

- Roll up - A roll up report is a summary of errors. For each erring device or system component covered in the report, the report totals the number of hardware and software errors. The summary does not include any information about individual errors.
- Brief - A brief report includes a descriptive, but brief, entry for each error covered in the report. Each entry describes, at a minimum, the type of error, the device or component that caused the error, the error's sequence number, and when the error was logged.
- Cryptic - A cryptic report applies to device errors only. For each device included in the report, the report shows the contents of the device registers every time an error occurred. The device register contents are shown without explanation.
- Standard - A standard report contains an entry for every error; each entry includes all the information gathered about the error. In addition, each item of information has a corresponding English explanation.

A standard report, which is 80 columns wide, is suitable for displaying at a terminal.

- Unknown - An unknown error report documents unknown, invalid, and undefined errors. Such errors include errors on devices not recognized by SYE. In addition, an error can be classified as unknown when the system information gathered to describe the error has been corrupted in some way.

See Chapter 2 for information on how to obtain copies of the error log file.

SET

3.13 SET

Allows the operator to establish or change the characteristics of a process, job, or device.

Format

SET option
<u>Options</u>
ACCOUNTING
DEVICE
LOGINS
PRINTER
PROTECTION/DEVICE
TERMINAL/PERMANENT
UIC

Prompts

What: option

Description

A description of each of the SET command options follows.

SET ACCOUNTING

3.13.1 SET ACCOUNTING

Establishes the status of various activities in the account file.

Format

<pre> SET ACCOUNTING <u>Command Qualifiers</u> /DISABLE[=(keyword,...)] /ENABLE[=(keyword,...)] /NEW_FILE </pre>
--

Prompts

None.

Command Parameters

None.

Description

The SET ACCOUNTING command enables or disables the logging of various activities in the accounting file. SET ACCOUNTING can also close the current accounting file and open a new one with a version number incremented by one.

When accounting is enabled, five types of accounting records can be entered into this file:

- Batch job termination
- Interactive job termination
- Login failures
- Print job completion
- Process completion

For more information about the accounting file, see the VAX/VMS System Manager's Guide and the VAX/VMS System Services Reference Manual.

Command Qualifiers

/DISABLE[=(keyword,...)]

Disables the logging of the specified activities in the account file. Table 3-6 lists and describes the activities that can be disabled.

/ENABLE[=(keyword,...)]

Enables the logging of the specified activities in the account file. Table 3-6 lists and describes the activities that can be enabled.

COMMANDS USED BY OPERATORS

/NEW_FILE

Closes the current accounting file called ACCOUNTNG.DAT, and opens a new version of that file.

Table 3-6
Keywords Appended to the /DISABLE and /ENABLE Qualifiers

Keyword	Meaning
BATCH	All batch jobs are recorded at job termination
INTERACTIVE	All interactive jobs are recorded at job termination
LOGIN_FAILURE	All login failures are recorded
PRINT	All print jobs are recorded
PROCESS	All processes are recorded

Examples

1. \$ SET ACCOUNTING/ENABLE=(BATCH,INTERACTIVE)

The SET ACCOUNTING/ENABLE command requests that all batch and interactive jobs be recorded in the accounting file at job termination.

2. \$ SET ACCOUNTING/NEW_FILE

This command closes the current accounting file and creates a new version of it.

SET DEVICE

3.13.2 SET DEVICE

Establishes a line printer or terminal as a spooled device; or establishes the status of the error log facility.

Format

```

SET DEVICE device-name[:]

  Command Qualifiers

/[NO] SPOOLED [= (queue-name[:], intermediate-disk-name[:])]
/[NO] ERROR_LOGGING
    
```

Prompts

Device: device-name[:]

Command Parameters

device-name[:]

Specifies the name of the device whose spooling or error logging status is to be changed. The device must be a line printer or a terminal if its spooling status is to change; or a disk, magnetic tape, or DMC-11 if its error logging status is to change.

Description

When the /SPOOLED qualifier is used, program output is written onto an intermediate disk rather than directly to a terminal or line printer.

Once a device has been set spooled to a disk, the intermediate device cannot be dismounted until it is set nospooled.

When the /ERROR_LOGGING qualifier is used, all error messages reported on the device on which error logging is enabled, are recorded in the error log file.

Command Qualifiers

/SPOOLED [= (queue-name[:], intermediate-disk-name[:])]

/NOSPOOLED [= (queue-name[:], intermediate-disk-name[:])]

Controls whether files will be spooled to an intermediate disk.

The queue name indicates the print queue to which a file is queued. If the queue name is not supplied, it defaults to SYS\$PRINT.

The intermediate disk name identifies the disk to which the spooled files are written. If the intermediate disk name is not supplied, it defaults to SYS\$DISK.

The intermediate disk must be mounted before files can be written onto it.

No channels can be assigned to a device once its spooling characteristics are changed.

COMMANDS USED BY OPERATORS

`/ERROR_LOGGING`

`/NOERROR_LOGGING`

Controls whether device errors will be logged in the error log file.

Error logging can currently be enabled for disks, magnetic tapes, and the DMC-11.

Examples

1. `$ SET DEVICE/SPOOLED=(LPA0:) LPA0:`

In this command, the printer queue LPA0 is spooled to an intermediate disk before the jobs in that queue are printed. Because no intermediate disk was specified, the intermediate disk defaults to SYS\$DISK.

2. `$ SET DEVICE/ERROR_LOGGING DBB2:`

The SET DEVICE command requests that all device errors reported on DBB2 be logged in the error log file.

SET LOGINS

3.13.3 SET LOGINS

Controls the number of users that can log into the system.

Format

```

SET LOGINS
  Command Qualifiers
  /INTERACTIVE[=n]
    
```

Prompts

None.

Command Parameters

None.

Description

The SET LOGINS command limits the number of users who can gain access to the system.

The SET LOGINS command is not retroactive; that is, all users logged into the system before the operator issues the SET LOGINS command are not affected by it. However, once the limit set by the operator is achieved, no more users can log in to the system until someone logs out.

If the operator does not specify a value for n, the SET LOGINS command displays the following information:

Login quotas - Interactive limit=m, Current interactive value=n

Command Qualifiers

/INTERACTIVE=n

Sets the number of interactive users allowed to gain access to the system.

Examples

1. \$ SET LOGINS/INTERACTIVE=5

The SET LOGINS command specifies that only five interactive users can be logged in to the system.

2. \$ SET LOGINS/INTERACTIVE

Login quotas - Interactive limit=15, Current interactive value=4

In this example, the SET LOGINS command indicates that the maximum number of users allowed on the system is fifteen and that the number of users currently logged in is four.

SET PRINTER

3.13.4 SET PRINTER

Establishes the characteristics of the specified line printer.

Format

SET PRINTER printer-name[:]	
<u>Command Qualifiers</u>	<u>Defaults</u>
/[NO]CR	/NOCR
/[NO]FF	/NOFF
/LA11	
/LA180	
/[NO]LOWER	/NOLOWER
/LP11	
/PAGE=n	/PAGE=64
/UNKNOWN	
/[NO]UPPER	/UPPER
/WIDTH=n	/WIDTH=132

Prompts

Device: printer-name[:]

Command Parameters

printer-name[:]

Specifies the name of the printer that will have its characteristics set or modified.

Command Qualifiers

/CR

/NOCR

Controls whether the printer outputs a carriage return; this characteristic is set for printers on which line feeds imply carriage returns.

The default qualifier is /NOCR.

/FF

/NOFF

Controls whether the printer performs a form feed.

The default qualifier is /FF.

/LA11

Sets the printer type to that of an LA11 line printer.

/LA180

Sets the printer type to that of an LA180 line printer.

COMMANDS USED BY OPERATORS

/LOWER
/NOLOWER

Controls whether the printer prints lowercase and uppercase alphabetic letters.

When the operator specifies the /NOLOWER qualifier, all alphabetic characters are translated to uppercase.

The /NOLOWER qualifier is complementary to the /UPPER qualifier that is, the /LOWER qualifier is equivalent to the /NOUPPER qualifier.

The default qualifier is /NOLOWER.

/LP11

Sets the printer type to that of an LP11 line printer.

/PAGE=n

Establishes the maximum number of lines that the line printer prints on an output page; the number of lines can range between 0 and 255 (decimal).

The default value is 64 lines per page.

/UNKNOWN

Sets the printer to unknown printer type.

/UPPER
/NOUPPER

Controls whether the printer translates all lowercase characters to uppercase.

When the /UPPER qualifier is specified, the printer performs a conversion from lowercase to uppercase characters. The /NOUPPER qualifier is complementary to the /LOWER qualifier.

The default qualifier is /UPPER, which is equivalent to the /NOLOWER qualifier.

/WIDTH=n

Establishes the number of characters to be printed per output line, where the width, n, can range between 0 and 65535 (decimal).

The default value is 132 characters per line.

Examples

1. \$ SET PRINTER/PAGE=100/WIDTH=80 LPA0:

The SET PRINTER command establishes the size of an output page as 100 lines and the width of a line as 80 characters.

2. \$ SET PRINTER/LA11 LPB0:

The SET PRINTER command establishes the line printer LPB0 as an LA11 printer and sets the default characteristics for that printer type.

3. \$ SET PRINTER/LOWER LPA0:

The SET PRINTER command requests that lowercase printing be enabled on line printer LPA0.

SET PROTECTION/DEVICE

3.13.5 SET PROTECTION/DEVICE

Establishes the protection to be applied to the specified shareable, non-file-structured device. The protection for the device thus limits the type of access available to system users.

Format

```
SET PROTECTION[=code]/DEVICE device-name[:]  
  
    Command Qualifiers  
  
    /OWNER_UIC=uic
```

Prompts

File: device-name[:]

Command Parameters

code

Establishes the protection code for a device. The protection code defines the user and type of access available for the user. The code should be specified according to the syntax rules described below under "How to Specify Protection Codes."

If the operator does not specify a protection code, no protection checks are made and all users are allowed to gain access to the device.

device-name[:]

Specifies the name of the device whose protection is to be set or modified. The device must be a shareable, non-file-structured device.

Description

The command qualifier /DEVICE must be specified to set or to change the protection applied to the device.

How to Specify Protection Codes: Devices can be accessed by users in four categories:

- System -- all users who have group numbers of 0 through 10 octal (generally, system managers, system programmers, and operators)
- Owner -- the UIC of the person who issued the SET PROTECTION command and therefore owns the device
- Group -- all users who have the same group number in their UICs as the owner of the device
- World -- all users who do not fall into the categories above

COMMANDS USED BY OPERATORS

Each of these categories of users can be allowed or denied one of the following types of access:

- Read -- the right to issue read requests to the device
- Write -- the right to issue write requests to the device
- Logical I/O -- the right to issue logical I/O requests to the device
- Physical I/O -- the right to issue physical I/O requests to the device

Any combination of access types can be specified for any category of user.

When the operator specifies a user access code, the code must be abbreviated to one character.

The user access codes and user categories can be specified in any order. If the operator omits a user access code for a user category, that category of user is denied that type of access. If the operator omits a user category, the user category is allowed all access.

To specify a protection code, the operator separates each user category from access type with a colon. To specify more than one user category, the operator separates the categories with commas and encloses the entire code within parentheses. For example:

```
SET PROTECTION=(SYS:RWLP,GR:R,W:)/DEVICE
```

This protection code allows:

- All types of access to users with a group number of 0 through 10 octal
- Group members read access only
- Users in the world category no access
- The device's owner all types of access

Command Qualifiers

/OWNER_UIC=uc

Requests that the identification code be assigned ownership of the device for the purpose of access checks. The default owner is the UIC of the process issuing the SET PROTECTION command.

Specify the UIC in the format:

[group,member]

In this format, group is an octal number from 0 through 377 representing the user's group number; and member is an octal number from 0 through 377 representing the user's member number.

The brackets are required.

COMMANDS USED BY OPERATORS

Examples

1. \$ SET PROTECTION=(SYS:RWLP,OWN:RWLP,G:,W:)/DEVICE LAA0:

The SET PROTECTION command requests that the protection for device LAA0 be set to allow all types of access to system processes and processes with the UIC of the current process and to deny access to anyone else.

2. \$ SET PROTECTION=(SYS:,OWN:RWPL,G:,W:)/DEVICE/OWNER_UIC=[103,4] LAB0:

This SET PROTECTION command requests that the protection for device LAB0 be set to allow all types of access to processes with a UIC of [103,4] and to deny access to anyone else.

SET TERMINAL/PERMANENT

3.13.6 SET TERMINAL /PERMANENT

Allows the operator to modify the permanent characteristics of the specified terminal.

Format

SET TERMINAL/PERMANENT [device-name:]	
<u>Command Qualifiers</u>	<u>Defaults</u>
/[NO] BROADCAST	/BROADCAST
/[NO] CRFILL[=formula]	/NOCR FILL
/[NO] ECHO	/ECHO
/[NO] EIGHTBIT	/NOEIGHTBIT
/[NO] ESCAPE	/NOESCAPE
/[NO] HARDCOPY	/HARDCOPY
/[NO] HOSTSYNC	/NOHOSTSYNC
/[NO] INTERACTIVE	/INTERACTIVE
/LA36	
/[NO] LFFILL[=formula]	/NOLFFILL
/[NO] LOCAL	/LOCAL
/[NO] LOWERCASE	/LOWERCASE
/PAGE=N	
/[NO] PASSALL	/NOPASSALL
/[NO] REMOTE	/NOREMOTE
/[NO] SCOPE	/NOSCOPE
/SPEED=rate	
/[NO] TAB	/NOTAB
/[NO] TTSYNC	/TTSYNC
/[NO] TYPEAHEAD	/TYPEAHEAD
/UNKNOWN	
/[NO] UPPERCASE	/UPPERCASE
/VT05	
/VT52	
/VT55	
/WIDTH=n	
/[NO] WRAP	/WRAP

Prompts

None.

Command Parameters

device-name:

Specifies the name of the terminal whose permanent characteristics are to be changed.

If the operator does not include a device name, the qualifiers change the characteristics of the terminal at which the command was issued.

Description

The SET TERMINAL/PERMANENT command allows the operator to change the permanent characteristics of the specified terminal. These characteristics override the system default characteristics, which are defined at system generation.

COMMANDS USED BY OPERATORS

The characteristics defined by use of the SET TERMINAL/PERMANENT command remain in effect until the system is halted; at this time, the permanent characteristics revert to the characteristics defined at system generation.

This command is generally used when a user requests the speed at which the terminal sends or receives data be changed.

To use the SET TERMINAL/PERMANENT command, the operator needs PHY_IO or LOG_IO privilege.

Table 3-7 shows terminal defaults.

Table 3-7
Default Characteristics for Terminals

Terminal Characteristics	Terminal Name and Characteristics ¹			
	UNKNOWN /UNKNOWN	LA36 /LA36	VT05 /VT05	VT5x /VT52 /VT55
WIDTH	*	132	72	80
HEIGHT	*	16	16	16
SPEED	*	*	*	*
BROADCAST	*	*	*	*
CRFILL	*	no	no	no
ECHO	*	yes	yes	yes
EIGHTBIT	*	no	no	no
ESCAPE	*	*	no	*
HOLDSCREEN	*	no	no	*
HOSTSYNC	*	no	no	yes
LFFILL	*	no	4	no
LOWERCASE	*	yes	no	yes
PASSALL	*	no	no	no
REMOTE	*	*	*	*
TYPEAHEAD	*	yes	yes	yes
TAB	*	no	no	yes
TTSYNC	*	yes	yes	yes
WRAP	*	yes	yes	yes

1. An asterisk indicates that the current setting is not changed by the qualifier.

COMMANDS USED BY OPERATORS

Command Qualifiers

`/BROADCAST`

`/NOBROADCAST`

Controls whether the terminal can receive messages broadcasted by the system operator. By default, a terminal receives all messages the system operator or another privileged user sends.

Use the `/NOBROADCAST` qualifier on a terminal that has been designated as a noninteractive terminal.

`/CRFILL=formula`

`/NOCRFILL`

Specifies whether the system generates fill characters following a carriage return on the terminal.

The formula is a number in the range of 0 through 9 indicating the number of null fill characters required to ensure that the carriage return completes successfully before the next meaningful character is sent. This qualifier may be needed if the terminal is a non-DIGITAL terminal.

`/ECHO`

`/NOECHO`

Controls whether the terminal echoes, or displays, the input characters that it receives.

When the `/NOECHO` qualifier is specified, the terminal displays only the data that a system or user application program writes to it.

The operator must have `PHY_IO` privilege to use the `/NOECHO` qualifier.

`/EIGHTBIT`

`/NOEIGHTBIT`

Indicates whether the terminal uses an 8-bit character code.

The `/NOEIGHTBIT` qualifier is the default; the terminal interprets characters using 7-bit ASCII code.

`/ESCAPE`

`/NOESCAPE`

Indicates whether the terminal generates valid escape sequences that will be interpreted by an applications program controlling the terminal.

If the operator specifies the `/ESCAPE` qualifier, the terminal checks the escape sequences for syntax before passing them to the program. For more information on escape sequences, see the VAX/VMS I/O User's Guide.

`/HARDCOPY`

`/NOHARDCOPY`

Indicates whether the terminal prints hardcopy output, as opposed to a video terminal.

This qualifier is complementary to the `/SCOPE` qualifier.

`/HOLDSCREEN`

`/NOHOLDSCREEN`

Enables and disables the operation of the `SCROLL` key on VT5x video terminals.

COMMANDS USED BY OPERATORS

/HOSTSYNC
/NOHOSTSYNC

Controls whether the system can synchronize the flow of input from the terminal.

When the operator specifies the /HOSTSYNC qualifier, the system generates CTRL/S and CTRL/Q to enable or disable the reception of input. When the type-ahead buffer is full, the system sends CTRL/S to temporarily stop input; when the buffer is empty, the system sends CTRL/Q so that more input can be entered.

/INTERACTIVE
/NOINTERACTIVE

Indicates that the terminal is in use as an interactive terminal.

This qualifier is complementary to the /PASSALL qualifier, that is, the /INTERACTIVE qualifier is equivalent to the /NOPASSALL qualifier.

/LA36

Indicates that the terminal is an LA36 terminal. When the operator specifies this qualifier, the default terminal characteristics for LA36 terminals are set. These settings are summarized under "/LA36" in Table 3-7.

/LFFILL=formula
/NOLFFILL

Specifies whether the system must generate fill characters following a line feed on the terminal.

The formula is a number in the range of 0 through 9 indicating the number of null fill characters required to ensure that the line feed completes successfully before the next meaningful character is read. This qualifier may be needed if the terminal is a non-DIGITAL terminal.

/LOCAL
/NOLOCAL

Controls, for terminals attached to dial-up lines, whether the terminal can be accessed from a remote location.

This qualifier is complementary to the /REMOTE qualifier, that is, the /NOLOCAL qualifier is equivalent to the /REMOTE qualifier.

/LOWERCASE
/NOLOWERCASE

Indicates whether the terminal has uppercase and lowercase characters.

If the operator specifies the /NOLOWERCASE qualifier, all alphabetic characters are translated to uppercase.

This qualifier is complementary to the /UPPERCASE qualifier, that is, the /LOWERCASE qualifier is equivalent to the /NOUPPERCASE qualifier.

COMMANDS USED BY OPERATORS

`/PAGE=n`

Specifies the page length of the terminal. For hardcopy terminals, the page size, *n*, equals the number of print lines between perforations on the paper. When the terminal reads a form feed character, it advances the paper to the next perforation.

The operator specifies values of 0 through 255 for the page size. The default page size is a system option.

`/PASSALL`

`/NOPASSALL`

Controls whether the system interprets special characters or passes all data to an application program as binary data.

A terminal operating with the `/PASSALL` qualifier set does not expand tab characters to blanks, fill carriage return or line feed characters, or recognize control characters.

The operator must have `PHY_IO` privilege to use the `/PASSALL` qualifier.

`/REMOTE`

`/NOREMOTE`

Specifies, when set, that the terminal is accessed from a remote location.

The system sets a terminal to remote mode when the user dials in and resets the terminal to non remote, or local, mode when the user hangs up.

`/SCOPE`

`/NOSCOPE`

Indicates whether the terminal is a video terminal.

This qualifier is complementary to the `/HARDCOPY` qualifier, that is, the `/NOSCOPE` qualifier is equivalent to the `/HARDCOPY` qualifier.

`/SPEED=rate`

Specifies the rate at which the terminal sends and receives data.

To set the input and output baud rates to the same speed, the operator can specify a single value. To send and receive data at different rates, the operator specifies the rate in the format (n,m) . The values *n* and *m* indicate the input and output rates, respectively.

The valid values for input and output baud rates are:

50	300	2400
75	600	3600
110	1200	4800
134	1800	7200
150	2000	9600

The default transmission rates are system options.

COMMANDS USED BY OPERATORS

/TAB

/NOTAB

Controls how the terminal handles tab characters. By default, the system expands all tab characters to blanks, assuming tab stops at 8-character intervals.

Use the /TAB qualifier to inhibit the system from converting tabs to blanks.

/TTSYNC

/NOTTSYNC

Controls whether the terminal synchronizes output by responding to CTRL/S and CTRL/Q.

The /TTSYNC qualifier is the default; the system stops sending output when the operator presses CTRL/S and resumes output when the operator presses CTRL/Q.

/TYPEAHEAD

/NOTYPEAHEAD

Controls whether the terminal accepts unsolicited input (that is, input that is typed when there is no outstanding read).

When the operator specifies the /NOTYPEAHEAD qualifier, the terminal is dedicated, and will only accept input when a program issues a read to it.

Use this qualifier to ensure that a specific terminal remains dedicated to a particular application.

/UNKNOWN

Indicates that the terminal is of an unknown terminal type. When the operator specifies this qualifier, the default terminal characteristics for a terminal of an unknown type are set. For a summary of the settings, see Table 3-7.

/UPPERCASE

/NOUPPERCASE

Specifies whether the terminal translates all lowercase letters to uppercase.

This qualifier is complementary to the /LOWERCASE qualifier, that is, the /UPPERCASE qualifier is equivalent to the /NOLOWERCASE qualifier.

/VT05

Indicates that the terminal is a VT05 terminal. When the operator specifies this qualifier, the default terminal characteristics for VT05 terminals are set. For a summary of the settings, see Table 3-7.

/VT52

Indicates that the terminal is a VT52 terminal. When the operator specifies this qualifier, the default terminal characteristics for VT52 terminals are set. These settings are summarized under "VT5x" in Table 3-7.

/VT55

Indicates that the terminal is a VT55 terminal. When the operator specifies this qualifier, the default terminal characteristics for VT55 terminals are set. These settings are summarized under "VT5x" in Table 3-7.

COMMANDS USED BY OPERATORS

/WIDTH=n

Specifies the number of characters on each input or output line. The width, n, must be in the range of 0 through 255.

If the /WRAP qualifier is in effect, the terminal generates a carriage return/line feed when a line reaches the specified width.

/WRAP

/NOWRAP

Controls whether the terminal generates a carriage return/line feed when it reaches the right margin. The right margin is determined by the setting of the terminal width.

If the operator specifies the /NOWRAP qualifier, the terminal does not generate a carriage return/line feed when it reaches the right margin, but continues to accept input at the last physical character position on the terminal line.

Examples

1. \$ SET TERMINAL/PERMANENT/NOBROADCAST

This command inhibits the display of messages sent by the system operator on the terminal at which this command was entered.

2. \$ SET TERMINAL/PERMANENT/NOTYPEAHEAD/PAGE=50 TTA4:

This command establishes the following permanent characteristics for the terminal TTA4:

- a. Inhibits the terminal from accepting input when it is not in a read state
- b. Indicates that the page length of the terminal is 50 lines

3. \$ SET TERMINAL/PERMANENT/SPEED=(9600,9600) TTAL:

This command causes terminal TTAL to send and receive data at 9600 baud.

SET UIC

3.13.7 SET UIC

Establishes a new user identification code as the default.

Format

<pre> SET UIC uic <u>Command Qualifiers</u> None.</pre>

Prompts

File: uic

Command Parameters

uic

Consists of the group number and member number. The format for the UIC is:

[group,member]

In this format, group is an octal number from 0 through 377 representing the user's group number; and member is an octal number from 0 through 377 representing the user's member number.

The square brackets are required.

Description

The operator uses the SET UIC command to gain access to a restricted file; that is, a file contained in a directory whose protection restricts access to the owner of that directory.

Using this command requires change-mode-to-kernel (CMKRNL) privilege.

By default, the SET UIC command changes the default directory name to the UIC supplied. Hence, to set the default directory to a code different than the UIC, the operator must issue the SET DEFAULT command after the SET UIC command.

Examples

1. \$ SET UIC [370,10]

This command establishes the operator's UIC as [370,10]. The operator can now read or modify any files contained in that directory.

2. \$ SET UIC [214,4]
\$ SET DEFAULT [ANDERS]

The SET UIC sets the operator's UIC to [214,4]; the SET DEFAULT command sets the default directory name to [ANDERS].

SHOW TERMINAL/PERMANENT

3.14 SHOW TERMINAL/PERMANENT

Displays the current permanent characteristics of the specified terminal.

Format

```
SHOW TERMINAL/PERMANENT [device-name:]
```

Command Qualifiers

None.

Prompts

None.

Command Parameters

device-name:

Specifies the name of the terminal whose permanent characteristics are to be displayed. If, however, the terminal is allocated, the system cannot display the characteristics.

If the operator does not include a device name, the characteristics of the terminal at which the command was issued are displayed.

Description

When the operator issues the SHOW TERMINAL/PERMANENT command, the status of each of the permanent settable characteristics is displayed. To use the SHOW TERMINAL/PERMANENT command, the operator needs PHY_IO or LOG_IO privilege.

Example

1. \$ **SHOW TERMINAL/PERMANENT**

```
TTf1: /VT52, WIDTH=80, PAGE=16, OWNER=SELF
SPEED=(9600.9600), CRFILL=0, LFFILL=0, NOPARITY
INTERACTIVE, ECHO, TYPEAHEAD, NOESCAPE, NOHOSTSYNC, TTSYNC,
LOWERCASE, TAB, WRAP, SCOPE, LOCAL, NOHOLDSCREEN,
NOEIGHTBIT, BROADCAST, NOREADSYNC
```

This SHOW TERMINAL/PERMANENT command displays the permanent settable characteristics for terminal TTf1.

START/QUEUE

3.15 START/QUEUE

Starts or restarts a specified queue.

Format

<u>Command Qualifiers</u>	<u>Defaults</u>
/BACKSPACE=n	/BACKSPACE=1
/BATCH	
/[NO]BURST	/NOBURST
/[NO]DISABLE_SWAPPING	/NODISABLE_SWAPPING
/[NO]ENABLE_GENERIC_PRINTING	/ENABLE_GENERIC_PRINTING
/[NO]FLAG	/NOFLAG
/FORMS_TYPE=n	
/[NO]GENERIC	/NOGENERIC
/[NO]NEXT	/NONEXT
/JOB_LIMIT=val	/JOB_LIMIT=1
/PRIORITY=val	/PRIORITY=4
/TERMINAL	
/TOP_OF_FILE	

Prompts

Queue: queue-name[:]

Command Parameter

queue-name[:]

Specifies the name of the queue to be started or restarted.

Description

Before a queue can be started, it must be initialized using the INITIALIZE/QUEUE command. However, once initialized, the queue can be started and stopped as necessary.

Command Qualifiers

/BACKSPACE[:n]

Backspaces n pages then resumes printing. Currently n is limited to 1. This qualifier is used in restarting a queue.

/BATCH

Indicates that this is a batch queue.

/BURST

/NOBURST

Controls whether a burst header page is printed for each print job.

Use the /BURST qualifier to print a header page over the paper perforations; hence, the page header is visible from the side of a stack of paper. Using this qualifier simplifies separating listings.

The default qualifier is /NOBURST.

COMMANDS USED BY OPERATORS

`/DISABLE_SWAPPING`

`/NODISABLE_SWAPPING`

Controls whether batch jobs executed from the queue can be swapped in and out of memory.

The default qualifier is `/NODISABLE_SWAPPING`.

`/ENABLE_GENERIC_PRINTING`

`/NOENABLE_GENERIC_PRINTING`

Controls whether files queued to a generic queue can be placed in a physical queue (that is, a device-specific queue) for printing.

By default, generic printing from a queue associated with a physical device is enabled.

`/FLAG`

`/NOFLAG`

Controls whether a header page is printed at the beginning of the first file in each print job.

The default qualifier is `/NOFLAG`.

`/FORMS_TYPE=n`

Specifies the forms type for the specified queue. Once a queue has been assigned a forms type, all jobs in that queue that do not conform to that forms type are placed in a holding state.

Specify the forms type using a numeric value or an alphanumeric code. Codes for forms type are installation defined.

`/GENERIC`

`/NOGENERIC`

Controls whether files placed in a queue can be printed out on any available line printer that has correctly matching characteristics.

If the qualifier `/GENERIC` is not used, the queue is by default a nongeneric queue.

`/JOB_LIMIT=val`

Indicates the number of batch jobs that can be executed concurrently from the queue. The default value of the job limit is 1.

`/NEXT`

`/NONEXT`

Controls whether the current job will be printed when the queue is restarted. The default is the `/NONEXT` qualifier.

This qualifier is used in restarting queues.

`/PRIORITY=val`

Indicates the priority at which jobs are initiated from the queue. The default value of the priority of a queue is 4.

`/TERMINAL`

Indicates that this is a terminal queue.

`/TOP_OF_FILE`

Prints, starting from the beginning of the job, the job that was being printed when the queue was stopped. This qualifier is used in restarting a queue.

COMMANDS USED BY OPERATORS

Examples

1. \$ **START/QUEUE/BATCH SYS\$BATCH**

This command starts the batch queue named SYS\$BATCH. The /BATCH qualifier indicates that this is a batch queue.

2. \$ **STOP/QUEUE LPA0:**
\$ **START/QUEUE/TOP_OF_FILE LPA0:**

The STOP/QUEUE command suspends operation of the physical queue LPA0. Then, the START/QUEUE/TOP_OF_FILE command resumes operation; however, the job currently printing is started again from the beginning.

3. \$ **INITIALIZE/QUEUE LPA0:**
\$ **START/QUEUE/NOFLAG LPA0:**

The INITIALIZE/QUEUE command initializes the queue named LPA0. The START/QUEUE command then starts the queue. The /FLAG qualifier requests that a header page precede each job in the queue.

STOP/ABORT

3.16 STOP/ABORT

The STOP/ABORT command aborts a job that is currently being printed.

Format

```
STOP/ABORT printer-name[:]  
  
Command Qualifiers  
  
None.
```

Prompts

Device: printer-name[:]

Command Parameters

printer-name[:]
Specifies the name of the printer queue in which the job was entered.

Description

The operator uses this command only to abort the printing of jobs entered in the system output queues, that is, the line printer or terminal queues.

When the operator issues STOP/ABORT, the job currently being printed is terminated, and the next job in the queue is dequeued.

Example

1. \$ **STOP/ABORT LPAL:**

This command aborts the job currently printing on line printer LPAL.

STOP/ENTRY

3.17 STOP/ENTRY

Deletes an entry from a batch queue.

Format

```
STOP/ENTRY=entry-number    queue-name[:]
```

Command Qualifiers

None.

Prompts

Queue: queue-name[:]

Command Parameters

entry-number

Specifies the entry number of the job to be deleted from the batch queue.

queue-name[:]

Specifies the name of the queue in which the job was entered.

Description

This command is used to terminate the execution of a batch job while it is running. This command cannot, however, delete a job while it is waiting to be executed.

Example

1. \$ STOP/ENTRY=230 SYS\$BATCH

The STOP/ENTRY command deletes the job associated with the entry number 230 in the batch queue SYS\$BATCH.

STOP/QUEUE

3.18 STOP/QUEUE

Suspends or controls operations of a specified output or batch queue.

Format

STOP/QUEUE queue-name[:]	
<u>Command Qualifiers</u>	<u>Default</u>
/ABORT	
/[NO]NEXT	/NONEXT

Prompts

QUEUE: queue-name[:]

Command Parameter

queue-name[:]

Specifies the name of the queue to be stopped.

Description

When the operator issues this command without any qualifiers, the current queue operation is placed in a paused state.

To delete a queue, the operator must first stop the queue using this command with the /NEXT qualifier appended to it.

Command Qualifiers

/ABORT

Aborts a job that is currently being printed; the next job in the queue is then started without a pause or stop. The STOP/QUEUE/ABORT command is equivalent to the STOP/ABORT command.

/NEXT

/NONEXT

Inhibits further dequeuing from the specified queue. When the /NEXT qualifier is specified, the current job is allowed to complete before the queue is stopped. When the /NONEXT qualifier is specified, the queue is stopped immediately. The default qualifier is /NONEXT.

Examples

1. \$ STOP/QUEUE/ABORT LPA0:

This command aborts the job currently printing on LPA0. The next job in the queue is then started.

2. \$ STOP/QUEUE/NEXT LPB0:

The STOP/QUEUE/NEXT command inhibits further printing of jobs in the queue LPB0; however, the current job is permitted to complete.

COMMANDS USED BY OPERATORS

3. OPCOM, 01:15:34.28, DEVICE OFFLINE, LPA0:

\$ STOP/QUEUE LPA0:

The OPCOM message indicates that line printer LPA0 is not in the ready state; that is, the line printer is malfunctioning. The operator issues the STOP/QUEUE command to suspend the queue associated with the malfunctioning printer and to find out what is wrong.

STOP/REQUEUE

3.19 STOP/REQUEUE

Stops the printing of the job currently being printed and places that job at the end of the output queue.

Format

```
STOP/REQUEUE queue-name[:]
```

Command Qualifiers

None.

Prompts

Queue: queue-name[:]

Command Parameters

queue-name[:]

Specifies the name of the queue to be stopped.

Description

When the operator requeues a job, that job is placed at the end of the queue with its priority level lowered to 1. The next job in the queue is immediately dequeued for printing.

This command is useful when the line printer runs out of paper while it is printing a job; or when a large job of low priority is currently printing and one or more other jobs in the queue must be printed immediately.

Example

1. \$ **STOP/REQUEUE LPB0:**

This command suspends the current print operation on LPB0, places the job that was currently printing at the end of the queue, and resumes the print operation.

CHAPTER 4

THE OPERATOR'S LOG FILE

The operator's log file records the occurrence of events that are of interest to the system managers and operators. The following list summarizes all the messages recorded in the operator's log file:

- Initialization of the operator's log file
- Status changes for devices attached to the system
- Terminals enabled and disabled
- Time stamps
- User requests and operator replies

These messages are produced by the Operator's Communication Process (OPCOM) and are preceded by the label Opcom. These messages are explained in detail in the following sections.

4.1 OVERVIEW OF THE OPERATOR'S LOG FILE

The operator's log file (OPERATOR.LOG) resides on the system disk in the [SYSMGR] directory. This file is in ASCII format and can be printed when needed.

When the system is shut down for whatever reason, then rebooted, the operator's log file is purged; only the two most recent versions of the log file are saved. Hence, if the manager wants to maintain a complete record of this file, the operator should print a copy of the log file regularly. See Chapter 2 for information on how to obtain copies of the operator's log file.

Figure 4-1 shows some of the typical messages that appear in the operator's log file.

4.2 INITIALIZATION MESSAGES

When the operator issues the command REPLY/LOG, the current log file is closed and a new version of that file is created and opened. All subsequent OPCOM messages are recorded in this new log file.

THE OPERATOR'S LOG FILE

When a new log file is created, the first message recorded in it is an initialization message. This line appears in the following format:

```
Opcom, dd-mmm-yyyy, hh:mm:ss,cc, LOGFILE INITIALIZED, OPERATOR=operator-name
```

The initialization line tells the operator when and by whom the log file was created.

```
Opcom, 9-JUL-1978 18:52:10.61, Logfile initialized, operator=_OPA2:
Opcom, 9-JUL-1978 19:22:13.85, Logfile time stamp
Opcom, 9-JUL-1978 19:40:37.95, Terminal enabled, operator=_TTA1:
Opcom, 9-JUL-1978 19:40:50.31, Terminal disabled, operator=_TTA1:
Opcom, 9-JUL-1978 20:12:13.78, Logfile time stamp
Opcom, 9-JUL-1978 20:47:13.78, Logfile time stamp
Opcom, 9-JUL-1978 21:17:26.84, Terminal enabled operator=_TTA2:
Opcom, 21:18:42.77, SWORDS Accnt=DOCUMENT
Opcom, _TTA1:, "ARE YOU READY?"

Opcom, 21:22:31.65, PARDON Accnt=DOCUMENT Reply-ID=0
Opcom, _TTA1:, "ARE YOU READY?"

Opcom, 06:58:57.70, Device offline, LPA0:
Opcom, 06:59:29.70, Device offline, LPA0:
Opcom, 07:00:01.70, Device offline, LPA0:
Opcom, 07:00:33.70, Device offline, LPA0:
Opcom, 07:01:05.70, Device offline, LPA0:

Opcom, 11:31:19.70, Device offline, LPA0:
Opcom, 11:31:51.70, Device offline, LPA0:
Opcom, 14-JUL-1978 13:59:30.27, Terminal enabled, operator=_TTC3:
Opcom, 13:59:41.88, SPARLP Accnt=VMS
Opcom, TTC3:, "TEST

Opcom, 15:26:42.73, Device offline, CRA0:
```

Figure 4-1 The Operator's Log File (OPERATOR.LOG)

4.3 DEVICE STATUS MESSAGES

The VAX/VMS I/O drivers send messages to OPCOM concerning device status changes. A change occurs and is recorded when a device goes offline. Once offline, the OPCOM message is written into the operator's log file at 30-second intervals until the device is explicitly returned to online status.

The device status message appears in the operator's log file in the following format:

```
Opcom, hh:mm:ss,cc, DEVICE OFFLINE, device-name
```

The devices for which this message can appear are card readers, line printers, magnetic tapes and networks. See the DECnet-VAX User's Guide for more information.

THE OPERATOR'S LOG FILE

4.4 TERMINAL ENABLE AND DISABLE MESSAGES

The operator designates a terminal as an operator's terminal by issuing the command REPLY/ENABLE at the desired terminal. The operator program confirms the request by displaying the following message at the operator's terminal and in the operator's log file:

```
Opcom, dd-mmm-yyyy hh:mm:ss,cc, Terminal enabled, operator=operator-name
```

This message tells the operator which terminal has been established as an operator's terminal and when it was established.

If a terminal has been designated as an operator's terminal for a particular function, OPCOM displays which function it is. For example, if the operator issues the command REPLY/ENABLE=TAPES, OPCOM displays the following message:

```
Opcom, 14-aug-1978 10:25:35/74, terminal enabled, operator=_TTEl:
```

```
$
```

```
Opcom, 10:25:38.82, operator terminal status, _TTEl:  
TAPES,
```

OPCOM confirms that the terminal is established as an operator's terminal and indicates that the terminal can only receive and respond to requests concerning mounting and dismounting of tapes.

The terminal at which a REPLY/ENABLE command has been issued can be restored to normal, that is, nonoperator, status by the command REPLY/DISABLE. The operator program confirms that the terminal is no longer an operator terminal by displaying the following message at the operator's terminal and in the operator's log file:

```
Opcom, dd-mmm-yyyy hh:mm:ss,cc, Terminal disabled, operator=operator-name
```

This message tells the operator which terminal has been restored to nonoperator status and when the transition occurred.

If, however, a terminal is designated as an operator's terminal and only partial operator status is taken away from it, OPCOM displays a status message. This message tells the operator which requests the terminal can still receive and respond to. This message is displayed at the operator's terminal and in the operator's log file in the following format:

```
Opcom, hh:mm:ss,cc, Operator terminal status, operator=operator-name  
status-report
```

For example, if an operator designates a terminal as an operator's terminal that receives messages concerning magnetic tapes, disks, and operator 1. Later, the operator relinquishes the terminal's ability to receive messages concerning tapes. When the operator issues the REPLY/DISABLE=TAPES command, OPCOM returns the following message:

```
Opcom, 09:23:45.32, Operator terminal status, operator=_TTA3:  
DISKS, OPER(1)
```

This message tells the operator that terminal TTA3 still receives and can respond to messages about disks and operator 1.

5.12

THE OPERATOR'S LOG FILE

4.5 TIME STAMP MESSAGES

A time stamp message is recorded at 30-minute intervals in the operator's log file. This message serves as an indicator as to when a system crash occurred. But, because the message only appears every 30 minutes, it can only approximate when the crash occurs.

A time stamp message appears in the log file in the following format:

```
Opcom, dd-mmm-yyyy hh:mm:ss.cc, logfile time stamp
```

4.6 USER REQUEST AND OPERATOR REPLY MESSAGES

To talk to the operator the user issues the REQUEST command specifying either the /REPLY or /TO qualifier.

If the user issues a REQUEST/REPLY command, that request is recorded in the operator's log file in the following format:

```
Opcom, hh:mm:ss.cc, user-name Acct=DOCUMENT Reply-ID=identification-number  
Opcom, terminal-name, "message-text"
```

This message tells the operator who sent the message, the time the message was sent, the identification number assigned to the message, and the message itself.

If the user issues a REQUEST/TO command, the request is recorded in the operator's log file in the same format as a REQUEST/REPLY but without the identification number.

When the operator responds to a user's request, specifying either the /ABORT qualifier or the /TO qualifier, the response is recorded in the operator's log file in the following format:

```
Opcom, hh:mm:ss.cc, operator-response, operator=operator-name
```

This message indicates how the operator responded to the user's request, when the operator responded, and which operator responded.

When the operator responds to a user request using the /PENDING qualifier, the response is not recorded in the operator's log file because the request has not yet been completed (that is, the request has not been fulfilled or aborted).

When a user sends a REQUEST/REPLY and the operator has disabled all terminals as operator's terminals, OPCOM records all subsequent user requests in the log file in the format shown above, but returns a message to the users indicating that there is no operator coverage at this time.

All other OPCOM responses to REPLY commands, except responses involving the REPLY/ENABLE, REPLY/DISABLE, and REPLY/LOG commands, are not logged in the operator's log file.

APPENDIX A

QUALIFIERS APPENDED TO THE MCR VFY COMMAND

This appendix presents a complete description of all the qualifiers that can be appended to the command string for the VERIFY command. Section A.1 describes the validity check which is the default qualifier. Section A.2 describes the remaining file qualifiers.

See Table 3-5 in Chapter 3 for the summarized list of qualifiers.

A.1 VALIDITY CHECK

The validity check (no qualifier) validates the structure of the volume mounted on the specified device. This feature entails reading all the file headers in the index file and ensuring that all the disk blocks referenced in the map area of each file header are marked as allocated in the bit map (that is, allocated to that file).

Rules for running the validity check:

1. The volume to be checked must be mounted as a Files-11 structured volume.
2. The volume may be write-protected if:
 - a. It is not the system volume; or
 - b. The required scratch file is directed to another file-structured volume.

When the validity check is completed, a listing of the results is printed. This output is described in Section A.1.1, below.

A.1.1 File Error Reporting

After the volume has been verified, and the normal output messages have been printed, error conditions are reported. All errors for a given file are preceded by a file identification line that identifies the file in error. This line is formatted as follows:

```
FILE ID nn,mm filename.filetype;version OWNER [g,m]
```

nn,mm is the file identification number assigned to the file by the system at file-creation time.

filename is the filename.

.filetype is the filetype (that is, OBJ for object file).

QUALIFIERS APPENDED TO THE MCR VFY COMMAND

;version is the version number of the file.

[g,m] is the UIC that owns the file.

This file identification line is followed by one or more of the following messages:

I/O ERROR READING FILE HEADER-ERROR CODE -32

Failed to read the file header for the specified file ID.

BAD FILE HEADER

Software checks on the validity of the file header indicate that the header has been corrupted.

MUTLIPLIE ALLOCATION n,m

The specified (double precision) logical block number is allocated to more than one file. If this error occurs, a second pass, which is automatically taken, will indicate all files that share each multiply-allocated block. The second pass is taken after all file headers have been checked (see Section A.1.3).

BLOCK IS MARKED FREE n,m

The specified logical block number is allocated to the indicated file but is not marked as allocated in the storage allocation map (see Section A.1.4).

BAD BLOCK NUMBER n,m

The specified block number was found in the header for this file but is illegal for the device (out of range). This indicates a corrupted file header.

FILE IS, MARKED FOR DELETE

A system failure occurred while the specified file was being deleted. The deletion was not completed and the file header still exists (see Section A.1.2).

HEADER MAP OUT OF SYNC

An error in the header map area which also indicates a corrupted file header.

The last error message for the file is followed by a summary line for that file, as follows:

SUMMARY: MULT=nn, FREE=nn, BAD=nn.

MULT is the number of multiple block allocations.

FREE is the number of blocks marked free that should have been allocated.

BAD is the number of bad retrieval pointers in the file header.

If the output for VFY is directed to a terminal device, and the operator does not wish to see all the error messages for a given file, entering CTRL/O terminates the listing of all further error messages for that file, that is, all messages but the summary line.

QUALIFIERS APPENDED TO THE MCR VFY COMMAND

A.1.2 Files Marked-for-Delete

If a file has been marked-for-delete but the deletion process was not completed, the operator can either restore the file or delete the file to recover the space it was occupying. This situation occurs only when the system crashes during file processing.

- Restoring a File

To restore a file marked-for-delete, mount the disk volume using the MOUNT command with the /UNLOCK qualifier specified. For example:

```
$ MCR VFY2 DMAL:/UNLOCK
```

Then, run VFY specifying the /DE qualifier to reset the marked-for-delete indicators in file headers. Once the delete indicator has been reset, run VFY specifying the /LO qualifier to scan the entire file structure.

The deletion process may have proceeded partially and a portion at the end of the file may be missing. This condition can be detected by a directory listing obtained with the PIP/FULL command.

- Deleting a File

Files that are marked-for-delete can be deleted directly with PIP, once their File ID has been obtained via a validity check. The File ID appears as the first entry in the file identification line which precedes each list of file errors (see Section A.1.1). The following example illustrates how the File ID is used with PIP to delete a file:

Example:

```
$ MCR PIP /FI:12:20/DE
```

In this example, the file with File ID 12,20 is deleted from the system device. PIP issues the following error message

```
PIP -- FAILED TO MARK FILE FOR DELETE-NO SUCH FILE
```

since the file system denies the existence of files already marked-for-delete; however, the file is completely deleted.

Once files have been restored or deleted, run VFY with the /RE qualifier specified to assure the consistency of the volume's storage allocation map.

A.1.3 Deletion of Multiply-Allocated Blocks

If the file structure contains multiply-allocated blocks, delete files until there are no more such blocks. An automatic rescan of the volume identifies which files share which blocks. Using this information, the operator can determine which files can be saved and, using the PIP delete function, delete the rest.

Be careful when deleting multiply-allocated files. After the files have been deleted, VFY should be run once again to ensure that all of the multiply-allocated files have been deleted.

QUALIFIERS APPENDED TO THE MCR VFY COMMAND

A.1.4 Elimination of Free Blocks

Once there are no multiply-allocated blocks, eliminate blocks that are marked FREE in the storage allocation map, but which are actually allocated to a file. To reallocate these blocks in the storage allocation map, rerun the validity check specifying the /UP qualifier. This qualifier allocates all blocks that should have been marked as allocated. See the description of the /UP qualifier.

Once there are no multiply-allocated blocks and no blocks marked free that are actually in use, the file structure is safe for writing new files and extending existing files. However, if there were such errors, there may be files which have had data blocks overwritten as the result of multiple allocation.

A.1.5 Recovering Lost Blocks

To determine whether any blocks on a file-structured volume have been lost, examine the last two lines of output from the validity check. The last two lines of output give the free space on the volume. The first line of the two tells how much room is available according to the index file (that is, the number of blocks that are not in use by any file in the index file). The last line specifies how much room is available according to the storage allocation map. Assuming there are no other errors, these two figures should agree. If the index file indicates that more blocks are free than the storage allocation map, then those blocks are "lost" in the sense that they appear to be allocated, but no file contains them. Lost blocks may be recovered by rerunning the validity check specifying the /RE qualifier. See the description of the /RE qualifier.

A.2 FILE QUALIFIERS

This section describes the file qualifiers that can be appended to the VERIFY command.

A.2.1 Delete Qualifier (/DE)

The Delete qualifier (/DE) resets the marked-for-delete indicator in the file header of a file that was marked for deletion, but never actually deleted.

The volume must be mounted with the /UNLOCK qualifier.

The volume being deleted must be write-enabled; VFY requires write access to the index file ([0.0]INDEXF.SYS).

VFY must be running under a system UIC.

A.2.2 Free Qualifier (/FR)

The Free qualifier (/FR) displays the available space on a specified volume.

FREE qualifier output is the following message:

```
dev: HAS nnnn. BLOCKS FREE, nnnn. BLOCKS USED OUT OF nnnn.
```

QUALIFIERS APPENDED TO THE MCR VFY COMMAND

A.2.3 List Qualifier (/LI)

The List qualifier (/LI) lists the entire index file by file identification. The output for each file specifies the file number, file sequence number, file-name, and owner. A typical index file listing is illustrated in Figure A-1.

```
VFY>DML:/LI
LISTING OF INDEX ON DML:

FILE ID 000001,000001 INDEXF.SYS;1    OWNER [1,1]
FILE ID 000002,000002 BITMAP.SYS;1    OWNER [1,1]
FILE ID 000003,000003 BADBLK.SYS;1    OWNER [1,1]
FILE ID 000004,000004 000000.DIR;1    OWNER [1,1]
FILE ID 000005,000005 CORIMG.SYS;1    OWNER [1,1]
FILE ID 000006,000006 001001.DIR;1    OWNER [1,1]
FILE ID 000007,000007 001002.DIR;1    OWNER [1,2]
FILE ID 000010,000010 EXEMC.MLB;1     OWNER [1,1]
FILE ID 000011,000011 RSXMAC.SML;1    OWNER [1,1]
FILE ID 000012,000012 NODES.TBL;1     OWNER [1,1]
FILE ID 000013,000036 QIOSYM.MSG;311  OWNER [1,2]
FILE ID 000014,000037 F4PCOM.MSG;1    OWNER [1,2]
```

Figure A-1 VFY Listing Sample Using the /LI Qualifier

A.2.4 Lost Qualifier (/LO)

The Lost qualifier (/LO) scans the entire file structure looking for files that are not in any directory and, thus, are lost in the sense that they cannot be referenced by file name. A list of the files is produced, and if the "lost file directory" [1,3] exists on the volume, all the files will be entered in that directory.

A.2.5 Read Check Qualifier (/RC)

The Read Check qualifier (/RC) checks to ensure that every block of every file on a specified volume can be read.

The optional parameter [:n] is the blocking factor which indicates the number of file blocks to be read at a time. The default value is the maximum number of blocks in dynamic memory available to VFY.

Since the READ CHECK is a read-only operation, the volume can be write-protected.

When an error is detected, a file identification line is listed in the following format:

```
FILE ID nn,nn file-name,type;version. blocks used/blocks allocated
```

Following this line, an error message is listed. If a blocking factor other than 1 is in use, an error message in the following format will be issued:

```
ERROR STARTING AT VBN n1,n2 LBN n1,n2 - ERROR CODE -err
```

QUALIFIERS APPENDED TO THE MCR VFY COMMAND

Following the first error message, there should be one or more error messages indicating the exact block(s) in error. The second error message line(s) will be in the following format:

```
ERROR AT VBN n1,n2 - ERROR CODE -err
```

If an ERROR STARTING AT line is displayed without one or more ERROR AT lines, a multiblock read operation on the selected device has failed, but the data blocks appear to be individually readable.

If the VBN of the unreadable block listed in the ERROR AT line is beyond the block-used-count, the data portion of the file is all right.

The negative number printed after the ERROR CODE message is -4 to indicate a device parity error. Other error codes are contained in Appendix A of the VAX/VMS System Messages and Recovery Procedures Manual.

A.2.6 Rebuild Qualifier (/RE)

The Rebuild qualifier (/RE) recovers lost blocks, that is, blocks that appear to be allocated but which are not contained in any file.

Multiply-allocated blocks must be removed from the file structure before the rebuild can be run.

The volume being updated must be write-enabled; VFY requires write-access to the storage allocation map ([0.0] BITMAP.SYS).

VFY must be running under a system UIC.

The scratch file should be on another volume. If this is impossible, the volume must be dismounted immediately after VFY terminates. (Failure to do this may result in partial updating of the storage allocation map.) Then the volume should be mounted again, and the scratch file must be deleted manually. VFY issues a detailed message in this case, specifying the name of the scratch file to be deleted.

A.2.7 Update Qualifier (/UP)

The Update qualifier (/UP) allocates all blocks that appear to be available but are actually allocated to a file.

Files with multiply-allocated blocks must be deleted from the file structure before the update can be run.

The volume being updated must be write-enabled; VFY requires write-access to the storage map ([0.0] BITMAP.SYS).

VFY must be running under a system UIC.

QUALIFIERS APPENDED TO THE MCR VFY COMMAND

The scratch file should be on another volume. If this is impossible, the volume must be dismounted immediately after VFY terminates. (Failure to do this may result in partial updating of the storage allocation map.) Then the volume should be mounted again, and the scratch file must be deleted manually. VFY issues a detailed message specifying the name of the scratch file to be deleted.

The message is:

```
VFY -- TO COMPLETE THE STORAGE MAP UPDATE DISMOUNT THE VOLUME
      IMMEDIATELY. THEN MOUNT IT AND DELETE THE FOLLOWING
      FILE: [ufd]file-spec
```

```
[ufd] is the UFD.
```

```
file-spec is the name of the file to be deleted.
```

A.2.8 Validity Check (Default)

A validity check is performed whenever no qualifiers are specified with the VFY command. This default qualifier is fully described in Section A.1.

APPENDIX B

ERROR MESSAGES PRODUCED BY THE BAD BLOCK UTILITY (BAD)

This appendix lists the BAD error messages, gives a brief description of the condition that causes each message, and suggests a response to the condition.

BAD -- BAD BLOCK FILE OVERFLOW

Description: BAD detected more than 102 sets of contiguous bad blocks. For RK06 disks, the maximum number of bad blocks that can be recorded is 126. This message usually indicates a disk unit failure.

Suggested User Response: Call the DIGITAL Field Service Representative.

BAD -- BLOCK 0 BAD - DO NOT USE AS SYSTEM DISK

Description: This is a warning message and can be ignored. VAX/VMS systems do not use block zero for bootstrapping purposes as do other RSX/IAS systems which support Files-11 on-disk structure.

Suggested User Response: Ignore the message

BAD -- DEVICE NOT IN SYSTEM

Description: The requested device was not made part of the system generation or the device does not exist on the host configuration.

Suggested User Response: Ensure that the command line was entered correctly.

BAD -- DISK ERROR

Description: An I/O error caused BAD to terminate.

Suggested User Response: Execute BAD again and, if the same error occurs, check the integrity of the unit.

BAD -- DISK IS ALIGNMENT CARTRIDGE

Description: The factory-written label on the last track of an RK06 cartridge indicates an alignment cartridge.

Suggested User Response: Self-explanatory.

ERROR MESSAGES PRODUCED BY THE BAD BLOCK UTILITY (BAD)

BAD -- DISK MOUNTED

Description: The disk to be analyzed is mounted as a Files-11 disk; however, BAD cannot process disks unless they are mounted using the /FOREIGN qualifier.

Suggested User Response: Dismount the unit using the DISMOUNT command, remount the unit with the /FOREIGN qualifier, and reenter the command line.

See the description of the MOUNT command in the VAX/VMS Command Language User's Guide for more information on this subject.

BAD -- DISK NOT READY

Description: The disk unit had not reached operating speed when BAD attempted to access it.

Suggested User Response: Allow the disk unit to reach operating speed and reenter the command line.

BAD -- DISK WRITE LOCKED

Description: The disk unit is write-locked.

Suggested User Response: Unlock the unit and reenter the command line.

BAD -- FAILED TO READ MANUFACTURER'S BAD SECTOR FILE

Description: A disk-read hardware error occurred while attempting to read the factory-written bad block data on the last track of an RK06 cartridge.

Suggested User Response: Reenter the command. If the problem persists, call the DIGITAL Hardware Service Representative.

BAD -- FAILED TO READ MCR COMMAND BUFFER

Description: BAD issued a \$GMCR directive and no buffer exists. This condition occurs when BAD is invoked in the form:

\$ MCR BAD

which is not supported for BAD.

Suggested User Response: Reenter the command line in the form:

\$ MCR BAD device-name:

BAD -- FAILED TO WRITE BAD BLOCK FILE

Description: BAD could not write the bad block file. This condition usually results from a disk write error.

Suggested User Response: Reenter the command. If the problem persists, check the integrity of the disk unit.

ERROR MESSAGES PRODUCED BY THE BAD BLOCK UTILITY (BAD)

BAD -- FATAL HARDWARE ERROR

Description: Self-explanatory.

Suggested User Response: Call the DIGITAL Field Service Representative.

BAD -- ILLEGAL DEVICE

Description: The device at which bad block processing is directed is not a Files-11 device (disk).

Suggested User Response: Files-11 devices are the only devices supported for the BAD utility; other devices are illegal.

BAD -- MANUFACTURER'S BAD SECTOR FILE CORRUPT

Description: The factory-written bad block data in the last track of an RK06 cartridge is in an inconsistent format.

Suggested User Response: Call the DIGITAL Field Service Representative.

BAD -- SYNTAX ERROR

Description: BAD detected a syntax error on the command line.

Suggested User Response: Determine the correct syntax and reenter the command line.

BAD -- TOTAL NO. OF BAD BLOCKS = nnnnnnnn

Description: This is an information message indicating the total number of bad blocks on the disk.

Suggested User Response: Write the bad block count on the disk label.

VAX/VMS OPERATOR'S NOTES

VAX/VMS
OPERATOR'S NOTES

VAX/VMS OPERATOR'S NOTES

Subject:

Date:

VAX/VMS OPERATOR'S NOTES

Subject:

Date:

VAX/VMS OPERATOR'S NOTES

Subject:

Date:

VAX/VMS OPERATOR'S NOTES

Subject:

Date:

VAX/VMS OPERATOR'S NOTES

Subject:

Date:

VAX/VMS OPERATOR'S NOTES

Subject:

Date:

VAX/VMS OPERATOR'S NOTES

Subject:

Date:

VAX/VMS OPERATOR'S NOTES

Subject:

Date:

VAX/VMS OPERATOR'S NOTES

Subject:

Date:

INDEX

A

ASSIGN/MERGE command, 3-4
ASSIGN/QUEUE command, 3-5

B

Backing up a floppy diskette to disk, 2-23
Backing up a nonsystem disk to disk, 2-18
Backing up the system disk to disk, 2-10
Backing up the system disk to tape, 2-14
Backing up volumes, 2-3
BAD error messages, B-1
Batch job, processing, 2-33
Batch queues, 2-28
 terminating, 2-31
Brief error report, 3-32
Broadcasting messages to terminals, 3-22

C

Card reader,
 correcting defective cards, 2-33
 procedure for processing a card deck, 2-35
 processing a batch job, 2-33
 processing a data deck, 2-33
 translation modes, 2-33
Correcting defective cards, 2-33
Closing the log file, 3-23
CMKRNL privilege, 1-2
Commands,
 ASSIGN/MERGE, 3-4
 ASSIGN/QUEUE, 3-5
 DEASSIGN/QUEUE, 3-7
 DELETE/QUEUE, 3-8
 INITIALIZE/QUEUE, 3-9
 MCR BAD, 3-12
 MCR DSC, 3-14
 MCR VFY, 3-18
 REPLY, 3-21
 RUN SYS\$SYSTEM:INSTALL, 3-27
 RUN SYS\$SYSTEM:SYE, 3-31
 SET, 3-33
 SET ACCOUNTING, 3-34
 SET DEVICE, 3-36
 SET LOGINS, 3-38
 SET PRINTER, 3-39
 SET PROTECTION/DEVICE, 3-41

Commands (Cont.),
 SET TERMINAL/PERMANENT, 3-44
 SET UIC, 3-51
 SHOW TERMINAL/PERMANENT, 3-52
 START/QUEUE, 3-53
 STOP/ABORT, 3-56
 STOP/ENTRY, 3-57
 STOP/QUEUE, 3-58
 STOP/REQUEUE, 3-60
Cryptic error report, 3-32

D

Data deck, processing, 2-33
DEASSIGN/QUEUE command, 3-7
Default terminal characteristics, 3-45
DELETE/QUEUE command, 3-8
Deleting multiply-allocated blocks, A-3
Device status messages, 4-2
Disabling operator status, 3-22
Disabling terminal messages, 4-3
Duties, operator, 1-1

E

Eliminating free blocks, A-4
Emergency shutdown of the system, 2-46
Enabling operator status, 3-22
Enabling terminal messages, 4-3
Error log (ERRLOG.SYS) file, 2-36, 2-38
Error Reports,
 brief, 3-32
 cryptic, 3-32
 roll up, 3-32
 standard, 3-32
 unknown, 3-32

F

File error messages, A-1
Files and volumes,
 handling, 2-1
 procedures for management of, 2-4

G

GROUP privilege, 1-2
GRPNAM privilege, 1-2

INDEX (Cont.)

H

Handling the error log file, 2-36
Handling files and volumes, 2-2
Handling the operator's log file,
2-36

I

INITIALIZE/QUEUE command, 3-9
Initializing the log file, 4-1

L

Line printer, changing paper in,
2-30
Log file, operator's, 1-2, 2-36,
2-40, 3-1, 4-1
LOG_IO privilege, 1-2

M

MCR BAD command, 3-12
MCR DSC command, 3-14
MCR VFY command, 3-18
Merging line printer queues, 2-29
Messages,
device status, 4-2
disable, 4-3
enable, 4-3
time stamp, 4-4
operator reply, 4-4
user request, 4-4
Mounting public disks, 2-9
Mounting volumes, 2-2

O

OPER privilege, 1-2
Operator assistance in creating a
multivolume tape set, 2-7
Operator assistance in handling
disks and single-volume tapes,
2-4
Operator assistance in handling
multivolume tapes, 2-6
Operator assistance in processing
a card deck, 2-35
Operator duties, 1-1
Operator privileges, 1-2
Operator reply messages, 4-4
Operator's commands, 3-1
Operator's communication process
(OPCOM), 4-1
Operator's log (OPERATOR.LOG)
file, 1-2, 2-36, 2-40, 3-1,
4-1

Operator's terminal, 1-3, 4-3
Orderly shutdown of the system,
2-43

P

PHY_IO privilege, 1-2
Preventing the loss of data when
line printer runs out of
paper, 2-30
Print queues, 2-28
merging, 2-29
replacing paper in line printer,
2-30
terminating, 2-32
Printing the error log file, 2-38
Printing the operator's log file,
2-40
Privileges, operator, 1-2
PRMCEB privilege, 1-2
PRMMBX privilege, 1-2
Procedure for,
emergency shutdown of system, 2-46
merging line printer queues, 2-29
orderly shutdown of system, 2-43
processing a card deck, 2-35
replacing line printer paper, 2-30
restarting the system, 2-47
terminating a batch job, 2-31
terminating a print job, 2-32
Procedure for backing up,
floppy diskette to disk, 2-23
nonsystem disk to disk, 2-18
selected files to disk, 2-21
system disk to disk, 2-10
system disk to tape, 2-14
Procedure for creating,
a multivolume tape set, 2-7
Procedure for handling,
disks, 2-4
multivolume tapes, 2-6
single-volume tapes, 2-4
Procedure for mounting,
public disks, 2-9
Procedure for printing,
error log (ERROR.SYS) file, 2-38
operator's log (OPERATOR.LOG)
file, 2-40
Procedure for restoring,
floppy diskette from disk, 2-25
system disk from tape, 2-16
Processing a batch job, 2-33
Processing a data deck, 2-33

Q

Queues,
batch, 2-28
print, 2-28
procedures that control queues, 2-29

INDEX (Cont.)

R

Recovering lost blocks, A-4
 REPLY command, 3-21
 Responding to user requests, 3-22
 Restarting the system, 2-41
 procedure for, 2-47
 Restoring files marked-for-delete,
 A-3
 Restoring a floppy diskette from
 disk, 2-25
 Restoring the system disk from tape,
 2-16
 Roll up error report, 3-32
 RUN SYS\$SYSTEM:INSTALL command,
 3-27
 RUN SYS\$SYSTEM:SYE command, 3-31
 Running a validity check, A-1

S

Selective back-up of files, 2-21
 SET ACCOUNTING command, 3-34
 SET command, 3-33
 SET DEVICE command, 3-36
 SET LOGINS command, 3-38
 SET PRINTER command, 3-39
 SET PROTECTION/DEVICE command, 3-41
 SET TERMINAL/PERMANENT command,
 3-44
 SET UIC command, 3-51
 SHOW TERMINAL/PERMANENT command,
 3-52
 Shutting down the operating system,
 2-41
 procedures for, 2-43
 Standard error report, 3-32
 Starting the system, 2-41
 procedure for, 2-47
 START/QUEUE command, 3-53
 STOP/ABORT command, 3-56
 STOP/ENTRY command, 3-57
 STOP/QUEUE command, 3-58
 STOP/REQUEUE command, 3-60
 Summary of operator's commands, 3-1
 SYSNAM privilege, 1-2

System status and activity,
 error log file, 2-36
 maintaining the error log file
 and operator's log file, 2-36
 operator's log file, 2-36
 procedures for, 2-38

T

Terminal, the operator's, 1-3, 4-3
 Terminating a batch job, 2-31
 Terminating a print job, 2-32
 Time stamp messages, 4-4
 TMPMBX privilege, 1-2
 Translation modes (for card reader),
 2-33

U

Unknown error report, 3-32
 User request messages, 4-3
 Using REPLY/DISABLE command, 4-3
 Using REPLY/ENABLE command, 1-3, 4-3
 Using REPLY/LOG command, 4-1
 Using REQUEST/REPLY command, 4-4
 Using REQUEST/TO command, 4-4

V

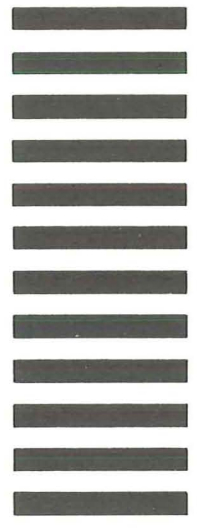
VAX/VMS operating system,
 procedures for halting and
 starting the, 2-43
 restarting the, 2-41
 shutting down the, 2-41
 VERIFY (VFY) qualifiers, A-1
 VFY file qualifiers,
 /DE, A-4
 /FR, A-4
 /LI, A-5
 /LO, A-5
 /RC, A-5
 /RE, A-6
 /UP, A-6
 VFY validity check, A-1
 VOLPRO privilege, 1-2

Do Not Tear - Fold Here and Tape

digital



No Postage
Necessary
if Mailed in the
United States



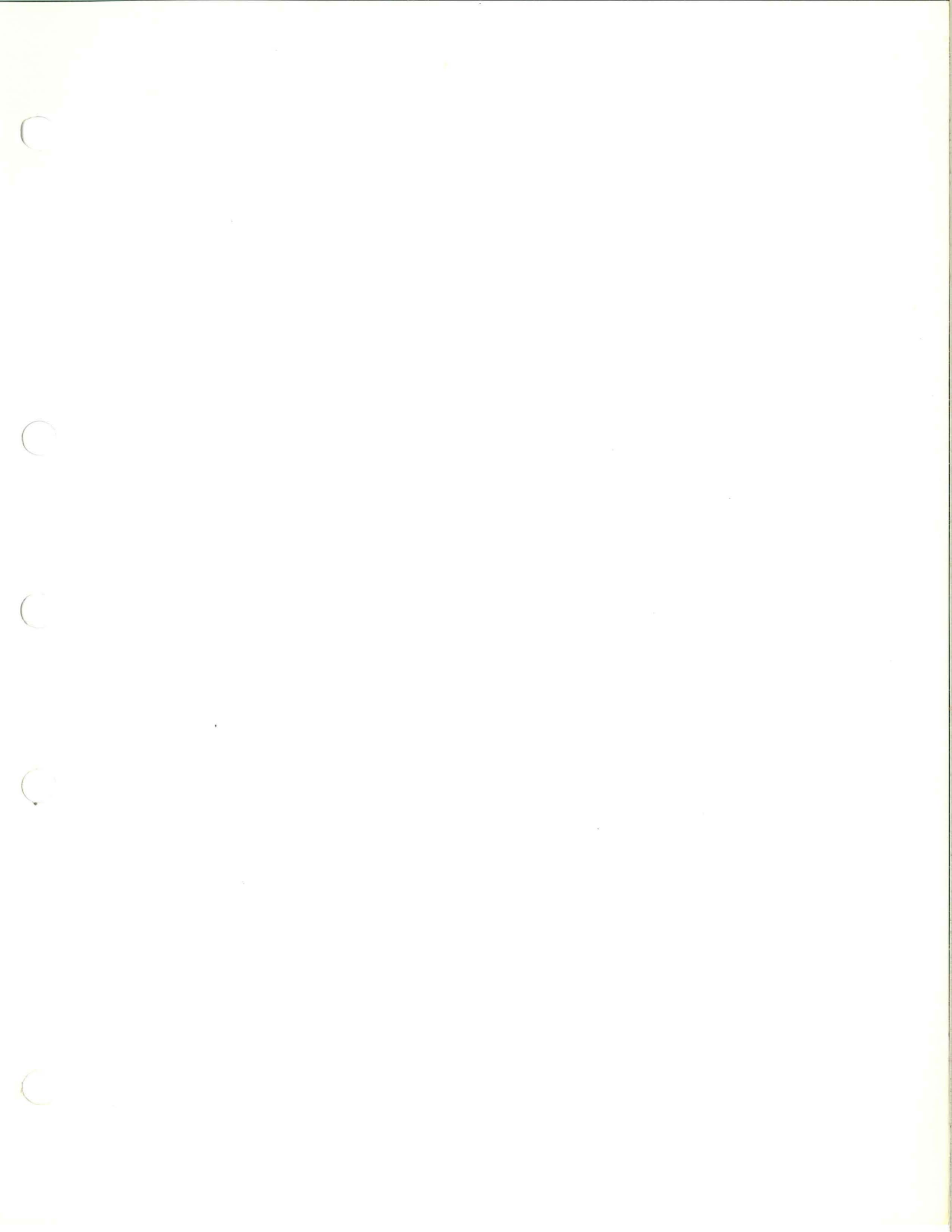
BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

RT/C SOFTWARE PUBLICATIONS TW/A14
DIGITAL EQUIPMENT CORPORATION
1925 ANDOVER STREET
TEWKSBURY, MASSACHUSETTS 01876

Do Not Tear - Fold Here

Cut Along Dotted Line



digital