
Guide to Maintaining a VMS System

Order Number: AA-LA34B-TE

November 1991

This manual provides the concepts and procedures system managers and operators need to maintain daily operations on a VMS operating system.

Revision/Update Information: This manual supersedes the *Guide to Maintaining a VMS System*, Version 5.0.

Software Version: VMS Version 5.5

**Digital Equipment Corporation
Maynard, Massachusetts**

November 1991

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 be used or copied only 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 Equipment Corporation or its affiliated companies.

© Digital Equipment Corporation 1991.

All Rights Reserved.

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation: DECwindows, Digital, VAX, VMS, and the DIGITAL logo.

PostScript is a registered trademark of Adobe Systems Incorporated.

ZK3386

This document was prepared using VAX DOCUMENT, Version 1.2

Contents

Preface	ix
1 Introduction	
2 Setting Up and Maintaining Public Volumes	
2.1 Planning Public Volumes	2-1
2.1.1 Establishing Volume-Level Protection	2-2
2.1.2 Using Volume Sets	2-2
2.2 Formatting Disks	2-3
2.3 Initializing Public Volumes	2-4
2.4 Mounting Public Volumes	2-5
2.5 Using the Exchange Utility to Transfer Information	2-8
2.5.1 Running the EXCHANGE Utility Interactively	2-9
2.5.2 Using a Command Procedure to Exchange Information	2-9
3 Performing Disk and Magnetic Tape Operations	
3.1 Using the Operator Communication Manager (OPCOM)	3-1
3.1.1 Enabling the Operator Terminal	3-1
3.1.2 Restarting OPCOM	3-2
3.1.3 Requests for Operator Assistance	3-2
3.1.4 Responding to User Requests	3-3
3.2 Handling Requests for Mounting Volumes	3-3
3.2.1 Mounting Disks and Single Magnetic Tape Volumes	3-4
3.2.2 Mounting Magnetic Tape Volume Sets	3-5
3.2.2.1 Mounting Volume Sets with Automatic Switching	3-5
3.2.2.2 Mounting Volume Sets Without Automatic Switching	3-5
3.3 Handling Requests from the Backup Utility	3-6
3.3.1 Writing to a Save Set	3-7
3.3.2 Reading from a Save Set	3-7
3.3.3 Recovering from an Error	3-7
3.4 Maintaining Volume Integrity	3-9
3.5 Mount Verification	3-9
3.5.1 Mount Verification for Offline Devices	3-10
3.5.2 Mount Verification for Write-Locked Devices	3-11
3.5.3 Canceling Mount Verification	3-12
3.5.3.1 Dismounting the Volume	3-12
3.5.3.2 MVTIMEOUT and TAPE_MVTIMEOUT System Parameters	3-12
3.5.3.3 Cancellation Commands	3-13

4 Managing Disk Space

4.1	Purging Old Versions of Files	4-1
4.2	Establishing Disk Quotas	4-2
4.2.1	Disk Quota Operations	4-2
4.2.1.1	Exceeding the Quota	4-2
4.2.1.2	Suspending Quotas	4-3
4.2.1.3	Ensuring Quota File Accuracy with REBUILD on Mount	4-3
4.3	Setting File Expiration Dates	4-3
4.4	Handling Error Conditions	4-4
4.4.1	Analyzing Disk Structure Errors	4-4
4.4.2	Analyzing Media Errors	4-5

5 Batch and Print Operations

5.1	Overview: Batch and Print Queuing System Concepts	5-1
5.1.1	The Queuing Process	5-1
5.1.2	The Queue Manager	5-1
5.1.3	The Queue Database	5-3
5.1.3.1	The Master File	5-3
5.1.3.2	The Queue File	5-3
5.1.3.3	The Journal File	5-3
5.1.4	The Job Controller	5-3
5.1.5	Types of Queues	5-3
5.1.5.1	Execution Queues	5-4
5.1.5.1.1	Batch Execution Queue	5-4
5.1.5.1.2	Output Execution Queue	5-4
5.1.5.2	Generic Queues	5-4
5.1.5.3	Logical Queues	5-5
5.1.6	The Autostart Feature	5-5
5.1.7	Job Scheduling	5-5
5.1.8	Spooled Output Devices	5-6
5.1.9	PostScript Printing	5-6
5.1.10	Distributed Printing	5-7
5.2	Setting Up the Batch and Print Queuing System	5-7
5.2.1	Planning Your Batch Queuing Environment	5-7
5.2.1.1	Simple Batch Queue Configuration: For Limited Batch Needs ...	5-8
5.2.1.2	Specialized Batch Queues: Meeting Special Needs	5-8
5.2.1.3	Generic Batch Queues in a VAXcluster: Distributing the Workload	5-9
5.2.1.4	Autostart Queues: For Ease of Maintenance and High Availability	5-11
5.2.2	Planning Your Output Queuing Environment	5-11
5.2.2.1	Simple Queue Configuration: For Limited Printing Needs	5-11
5.2.2.2	Mixed Printers: Specialized Output Execution Queues	5-12
5.2.2.3	LAT Printers: Sharing Printers Among Multiple Systems	5-13
5.2.2.4	Generic Output Queues: Distributing the Workload	5-14
5.2.2.5	VAXcluster Queues: Sharing Devices in a Cluster	5-15
5.2.2.6	Spooled Output Devices: Printing from Applications	5-16
5.2.2.7	Autostart Queues: For Ease of Maintenance and High Availability	5-17
5.2.3	Choosing Batch and Output Queue Options	5-17

5.2.4	Restricting Access to a Queue	5-21
5.2.4.1	UIC-Based Queue Protection	5-21
5.2.4.2	ACL-Based Queue Protection	5-23
5.2.4.3	Privileges Allowing Queue Access	5-23
5.2.5	Setting Up Queues to Retain Jobs	5-24
5.2.6	Using Queue Characteristics	5-25
5.2.6.1	Defining Characteristics	5-26
5.2.6.2	Displaying Characteristics Defined on a System	5-26
5.2.6.3	Assigning Characteristics to a Queue	5-26
5.2.6.4	Displaying Characteristics Assigned to a Queue	5-26
5.2.6.5	Cancelling Characteristics Assigned to a Queue	5-26
5.2.6.6	Deleting Characteristics	5-27
5.2.7	Options Specific to Batch Queues	5-27
5.2.7.1	Base Process Priority	5-28
5.2.7.2	Job Limit	5-28
5.2.7.3	Working Set Default, Quota, and Extent	5-28
5.2.7.4	CPU Default and Maximum	5-29
5.2.7.5	Swapping	5-29
5.2.7.6	Setting Up Batch Queues on Memory-Constrained Systems	5-30
5.2.7.7	Optimizing Batch Queues for the VMS Sort/Merge Utility	5-30
5.2.8	Options Specific to Output Queues	5-30
5.2.8.1	Scheduling Jobs Based on Job Size	5-30
5.2.8.2	Using Banner Pages	5-31
5.2.8.3	Using Forms	5-40
5.2.8.3.1	Procedure for Using Forms	5-41
5.2.8.3.2	Defining a Form	5-41
5.2.8.3.3	Displaying Forms Defined on a System	5-42
5.2.8.3.4	Changing the Systemwide Default Form	5-43
5.2.8.3.5	Assigning a Default Form for a Queue	5-43
5.2.8.3.6	Mounting a Form on a Queue	5-43
5.2.8.3.7	Displaying the Form Assigned to a Queue	5-44
5.2.8.3.8	Deleting a Form	5-44
5.2.8.4	Controlling Page and Line Overflow	5-44
5.2.8.5	Using Device Control Libraries	5-45
5.2.8.5.1	Procedure for Using Device Control Libraries	5-46
5.2.8.5.2	Creating a Device Control Library and Inserting Modules	5-47
5.2.8.5.3	Assigning a Library to a Queue	5-47
5.2.8.5.4	Creating Forms for Setup and Page Setup Modules	5-48
5.2.8.5.5	Assigning a Reset Module to a Queue	5-48
5.2.8.5.6	Order of Device Control Module Output	5-49
5.2.9	Implementing Your Planned Queue Configurations	5-50
5.2.9.1	Setting Up Output Devices	5-50
5.2.9.1.1	Setting Up Logical Ports for LAT Printers	5-51
5.2.9.1.2	Setting Printer Device Characteristics	5-51
5.2.9.1.3	Spooling Output Devices	5-52
5.2.9.1.4	Creating a Command Procedure to Setup Devices on Reboot	5-54
5.2.9.2	Starting the Queue Manager and Creating the Queue Database	5-54
5.2.9.3	Defining Forms and Characteristics You Have Planned	5-55
5.2.9.4	Creating and Starting Queues	5-56
5.2.9.4.1	Creating Execution Queues	5-56
5.2.9.4.2	Creating Generic Queues	5-57
5.2.9.4.3	Procedure for Creating and Starting Queues	5-57

5.2.9.4.4	Sample Queue Configuration Setup	5-58
5.2.9.4.5	Creating a Command Procedure to Restart Execution Queues on Reboot	5-61
5.2.9.5	Maintaining Your Queue Configuration	5-62
5.3	Maintaining the Queuing System	5-62
5.3.1	Maintaining the Queue Manager and Queue Database	5-62
5.3.1.1	Starting the Queue Manager	5-62
5.3.1.2	Stopping the Queue Manager	5-63
5.3.1.3	Saving the Queue Database	5-64
5.3.1.4	Restoring the Queue Database	5-64
5.3.1.5	Moving Queue Database Files	5-65
5.3.1.5.1	Moving the Master File	5-65
5.3.1.5.2	Moving the Queue and Journal Files	5-66
5.3.1.6	Submitting a Software Performance Report (SPR)	5-66
5.3.2	Managing Queues	5-67
5.3.2.1	Initializing Queues	5-68
5.3.2.2	Starting Nonautostart Queues	5-68
5.3.2.3	Starting Autostart Queues	5-68
5.3.2.4	Activating an Autostart Queue	5-68
5.3.2.5	Enabling Autostart on a Node	5-69
5.3.2.6	Monitoring Queue Information	5-69
5.3.2.7	Modifying a Queue	5-72
5.3.2.8	Pausing a Queue	5-73
5.3.2.9	Closing a Queue	5-73
5.3.2.10	Stopping a Queue	5-73
5.3.2.11	Preventing Autostart Queues from Starting	5-74
5.3.2.12	Disabling Autostart on a Node	5-74
5.3.2.13	Stopping All Queues on a Node	5-74
5.3.2.14	Stopping Queues Before Shutting Down a System	5-75
5.3.2.15	Assigning a Logical Queue	5-75
5.3.2.16	Moving All Jobs From One Queue to Another	5-76
5.3.2.17	Deleting a Queue	5-76
5.3.3	Managing Jobs	5-76
5.3.3.1	Monitoring Jobs	5-77
5.3.3.2	Modifying Job Processing Options	5-78
5.3.3.3	Holding and Releasing a Job	5-79
5.3.3.4	Changing the Scheduling Priority of a Job	5-80
5.3.3.5	Requeuing an Executing Job	5-80
5.3.3.6	Requeuing a Pending Job	5-81
5.3.3.7	Retaining Jobs in a Queue	5-81
5.3.3.8	Deleting a Job	5-82
5.3.3.9	Pausing an Output Queue to Control Print Job Position and Alignment	5-83
5.3.3.9.1	Specifying the Position of Print	5-83
5.3.3.9.2	Aligning Print Forms	5-84
5.3.4	Troubleshooting	5-85
5.3.4.1	Solving Queue Manager Problems	5-85
5.3.4.1.1	If the Queue Manager Becomes Unavailable	5-86
5.3.4.1.2	If the Queue Manager Does Not Start	5-86
5.3.4.2	Determining the Cause of General Printer Problems	5-87
5.3.4.3	Making Pending Jobs Eligible for Scheduling	5-88
5.3.4.3.1	Fixing Print Jobs That Are Pending Due to Stock Mismatch	5-89

5.3.4.3.2	Fixing Jobs That Are Pending Because of Characteristics Mismatch	5-89
5.3.4.4	Fixing a Stalled Output Queue	5-90
5.3.4.5	Determining Why an Autostart Queue Will Not Start	5-90
5.3.4.6	Solving Problems Deleting a Queue, Form, or Characteristic	5-91
5.3.4.7	Adding or Deleting a Device Control Library Module	5-92
5.4	Managing the Card Reader	5-93
5.4.1	Distinguishing the Type of Card Deck	5-94
5.4.1.1	Batch Job Card Deck	5-94
5.4.1.2	Data Card Deck	5-94
5.4.1.3	Setting Card Reader Translation Modes	5-95
5.4.2	Tending the Card Reader	5-95
5.4.2.1	Replacing Physically Defective Cards	5-95
5.4.2.2	Operating the Card Reader	5-95
5.4.3	Running the Input Symbiont Interactively	5-96

6 Gathering System Information

6.1	The Accounting Log File	6-2
6.1.1	Accounting Records	6-3
6.1.2	Accounting Report Formats	6-4
6.2	The Error Log File	6-4
6.2.1	Using Error Reports	6-5
6.2.2	Maintaining the Error Log File	6-6
6.2.3	Printing the Error Log File	6-6
6.3	The Operator Log File	6-7
6.3.1	Types of OPCOM Messages	6-7
6.3.1.1	Initialization Messages	6-7
6.3.1.2	Device Status Messages	6-7
6.3.1.3	Terminal Enable and Disable Messages	6-8
6.3.1.4	Volume Mount and Dismount Messages	6-9
6.3.1.5	User Request and Operator Reply Messages	6-9
6.3.1.6	SYSGEN Utility Messages	6-10
6.3.1.7	Security Alarm Messages	6-10
6.3.2	Setting Up the Operator Log File	6-11
6.3.3	Maintaining the Operator Log File	6-12
6.3.4	Printing the Operator Log File	6-13
6.4	The Security Audit Log File	6-13
6.4.1	Creating a New Version	6-14
6.5	The System Dump File	6-14
6.6	Obtaining Information About a Multiprocessing System	6-15
6.7	Obtaining Information About a Vector Processing System	6-15
6.7.1	DCL Lexical Functions F\$GETJPI and F\$GETSYI	6-16
6.7.2	SHOW CPU Command	6-16
6.7.3	SHOW PROCESS and LOGOUT/FULL Commands	6-16
6.7.4	Vector Processing Support Within the VMS Accounting Utility	6-17
6.7.5	Vector Support Within the VMS Monitor Utility	6-17

Index

Examples

5-1	Using Device Control Library Modules	5-49
5-2	Sample Commands for Setting Up Queue Configuration	5-59
5-3	Sample Queue Startup Commands	5-61
6-1	Sample Operator Log File (SYS\$MANAGER:OPERATOR.LOG)	6-8

Figures

5-1	The VMS Batch and Print Queuing System	5-2
5-2	A Default Batch Queue	5-8
5-3	Multiple Batch Queues with Special Resource and Performance Options	5-9
5-4	Batch Queue Configuration with Clusterwide Generic Queue	5-10
5-5	Single Laser Printer Queue	5-12
5-6	Queue Configuration with Mixed Printers	5-13
5-7	Configuration for Remote Printers on a Terminal Server	5-14
5-8	Queue Configuration with Three Like Printers and a Generic Queue	5-15
5-9	Output Queue Configuration in a VAXcluster	5-16
5-10	Queue Configuration with a Spooled Device	5-17
5-11	Job Flag and Burst Pages	5-33
5-12	File Flag and Burst Pages	5-35
5-13	Job Trailer Page	5-37
5-14	File Trailer Page	5-38

Tables

5-1	Qualifiers for Specifying Queue Options	5-18
5-2	Commands to Set Up Output Devices	5-51
5-3	Effects of Queue Commands on Autostart and Nonautostart Queues	5-67
5-4	Effects of Queue Commands on All Queues	5-67
5-5	Queue Status	5-71
5-6	Job States	5-78
5-7	Card Reader Errors: Causes and Corrective Actions	5-96

Preface

The *Guide to Maintaining a VMS System* provides guidelines and procedures system managers need to maintain daily operations on a VMS operating system. After reading this manual, system managers should be able to perform the following tasks:

- Set up public volumes
- Handle user requests for disk and tape operations
- Manage disk space
- Manage batch and print operations
- Get information about the system

This manual is not intended to be a complete one-volume source of information. For information about customizing the system after installation and setting up user accounts, see the *Guide to Setting Up a VMS System*. The utilities and commands used to perform specific system management tasks are described in detail in the individual VMS utility manuals and in the *VMS DCL Dictionary*.

Intended Audience

This manual addresses experienced users of the VMS operating system who perform the functions of a system manager or operator.

Document Structure

The *Guide to Maintaining a VMS System* contains the following chapters:

- Chapter 1 provides an overview of tasks described in this manual.
- Chapter 2 contains guidelines for setting up and maintaining public files and volumes.
- Chapter 3 describes several disk and tape operations you should be familiar with.
- Chapter 4 describes how to make the best use of available disk space on your system.
- Chapter 5 contains information about batch and print jobs and ways to submit, schedule, and execute them efficiently.
- Chapter 6 describes ways to gather information about what the system is doing.

Associated Documents

- For general background information about the system, see the *Introduction to VMS*.
- For computer-specific operations and installation information, see the upgrade and installation supplement for your VAX computer.
- For information about the current VMS installation and upgrade procedures, see the *VMS Upgrade and Installation Manual*.
- For information about setting up your system for daily operations, see the *Guide to Setting Up a VMS System*.
- For information about creating and maintaining volumes using the volume shadowing option, see the *VMS Volume Shadowing Manual*.
- For hardware operating instructions, see the appropriate hardware owner's manual.
- For managing network operations, see the *Guide to DECnet-VAX Networking*.
- For configuring and managing VAXclusters, see the *VMS VAXcluster Manual*.
- For information about performance tuning, see the *Guide to VMS Performance Management*.
- For information about the new System Management (SYSMAN) Utility, see the *VMS SYSMAN Utility Manual*. For detailed information about other VMS utilities, see the specific VMS utility manual.
- For supplemental reference information, see the *VMS DCL Dictionary* and the *VMS System Messages and Recovery Procedures Reference Manual*.

Conventions

The following conventions are used in this manual:

Ctrl/x	A sequence such as Ctrl/x indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
Return	In examples, a key name is shown enclosed in a box to indicate that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)
...	In examples, a horizontal ellipsis indicates one of the following possibilities: <ul style="list-style-type: none">• Additional optional arguments in a statement have been omitted.• The preceding item or items can be repeated one or more times.• Additional parameters, values, or other information can be entered.
.	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.

()	In format descriptions, parentheses indicate that, if you choose more than one option, you must enclose the choices in parentheses.
[]	In format descriptions, brackets indicate that whatever is enclosed within the brackets is optional; you can select none, one, or all of the choices. (Brackets are not, however, optional in the syntax of a directory name in a file specification or in the syntax of a substring specification in an assignment statement.)
red ink	Red ink indicates information that you must enter from the keyboard or a screen object that you must choose or click on. For online versions of the book, user input is shown in bold .
boldface text	Boldface text represents the introduction of a new term or the name of an argument, an attribute, or a reason. Boldface text is also used to show user input in online versions of the book.
<i>italic text</i>	Italic text represents information that can vary in system messages (for example, Internal error <i>number</i>).
UPPERCASE TEXT	Uppercase letters indicate that you must enter a command (for example, enter OPEN/READ), or they indicate the name of a routine, the name of a file, the name of a file protection code, or the abbreviation for a system privilege.
-	Hyphens in coding examples indicate that additional arguments to the request are provided on the line that follows.
numbers	Unless otherwise noted, all numbers in the text are assumed to be decimal. Nondecimal radices—binary, octal, or hexadecimal—are explicitly indicated.



Introduction

This manual provides guidelines and procedures for maintaining daily operations on a VMS operating system. Maintenance tasks can be performed by an operator, a system manager, or by several people. The following is an overview of the routine maintenance tasks described in this manual:

- **Setting up and maintaining public volumes**

A primary system management duty is setting up and maintaining public volumes. Public volumes, also called system volumes, are file-structured disk volumes that contain public files. Such files are made available to most, if not all, users. Public volumes can also contain files that individuals create for their own use.

As system manager, you should find the balance between your users' needs and the system's available mass storage resources. You must determine how to configure mass storage devices on your system, which devices will hold public system volumes, which devices will be available for users' private volumes, and how the public volumes will be configured. See Chapter 2 for detailed information.

- **Handling user requests for disk and magnetic tape operations**

Users may prepare and manipulate their own volumes, or they may rely on assistance from an operator. If users require assistance, they can use the operator communication facility (OPCOM), an online facility for communication between operator and user. See Chapter 3 for more information on handling user requests with OPCOM.

- **Managing disk space**

You control the amount of disk space available to users by creating and maintaining quota files on public volumes with the disk quota subcommand of the System Management Utility (SYSMAN). The absolute maximum number of blocks allotted to a user on a volume is the sum of the quota and overdraft values. In addition to maintaining disk quotas, you can also conserve disk space by periodically purging log files and setting file expiration dates. See Chapter 4 for more information.

- **Performing batch and print operations**

Setting up queues and managing batch and print jobs is of primary importance to maintaining daily operations on your system. As system manager, you must monitor the types of jobs that run on your system and develop ways to submit, schedule, and execute jobs efficiently. After you have set up the queues, you must monitor performance and troubleshoot for potential problems in the processing of jobs in the queues. For example, you may need to delete a job entry, or pause a queue temporarily to change the printer form or correct a defective printer device.

Introduction

You interact with the batch and print queueing system by entering DCL commands. Most of the commands for managing queues require privileges (see DCL Help or the *VMS DCL Dictionary* for information on privileges required for specific commands). See Chapter 5 for guidelines and procedures for managing batch and print operations.

- **Gathering information about the system**

The VMS operating system provides several system log files to assist you in monitoring system performance and analyzing errors and other system events. These log files include the following:

- Accounting log file
- Error log file
- Operator log file
- Security audit log file

In addition, if the system fails, the VMS operating system automatically shuts down and produces a crash dump of the system at the time the error was detected. The system dump file assists you in analyzing the cause of the system failure. (See the *VMS System Dump Analyzer Utility Manual* for more information on the System Dump Analyzer; see Chapter 6 for more information on the system log files.)

If you are running a special processing environment, such as multiprocessing or vector processing, DCL commands and utilities help you obtain information about that environment.

If you are manager of a cluster, or of a system with multiple nodes, you should also become familiar with SYSMAN, which centralizes the management of nodes and clusters. In a multinode environment, SYSMAN allows you to define your management environment to be a particular node, a group of nodes, or a cluster. After you define the management environment, you can perform system management tasks on all nodes in the target environment from your local node, without having to log in to each individual node.

Using SYSMAN to maintain your system is very similar to using the traditional system management tools. SYSMAN uses the software tools with which you may already be familiar, including DCL commands such as MOUNT, DEFINE, INITIALIZE, SET, and SHOW. SYSMAN also executes VMS system management utilities, such as AUTHORIZE, AUTOGEN, and INSTALL. In addition to giving you access to these system management tools, SYSMAN allows you to set disk quotas, to modify system parameters (with the same functions as SYSGEN), and to build site-specific startup procedures.

For more information about SYSMAN, see the *VMS SYSMAN Utility Manual*.

Setting Up and Maintaining Public Volumes

This chapter contains guidelines for setting up and maintaining public files and volumes. It also describes how to initialize and mount public volumes, and how to transfer information between VMS Files-11 formatted and **foreign** (non-Files-11 formatted) volumes. For information on mounting tapes and disks or backing up your files, see Chapter 3 or *VMS Backup Utility Manual*, respectively.

2.1 Planning Public Volumes

Public volumes are file-structured disk volumes that contain user and system files. Such files must be available to most, if not all, users. Public volumes contain files provided by Digital, such as the operating system, help files, and diagnostic programs. Public volumes also contain files that system managers and users make available for general use. As long as file protection permits it, all users have access to public volumes and to the files on them.

Successful maintenance results from balancing users' needs for disk storage and the system's available storage. You must determine how mass storage devices on your system will be configured, which devices will hold public volumes, which devices will be available for users' private volumes, and how the public volumes will be configured.

Generally, there are two types of files on public volumes: system files and user files. System files refer to the operating system and its related files, plus common tools such as editors, compilers, and linkers. Public volumes contain the following kinds of files supplied by Digital:

- The operating system in executable form and files related to the operating system
- Utility programs in executable form
- Diagnostic and test programs in executable form and files related to these programs
- Various system libraries such as macro libraries, object module libraries, shared run-time libraries, and error message libraries
- Text files such as help files
- Optional software in executable form plus related libraries and other files

In addition to these system files, you include on public volumes any files unique to an installation that must be accessible to many or all users. You also include all top-level user directories.

If you have a relatively small mass storage configuration, you will have both system files and user files on the same public volume. You should allocate disk space carefully, in such a case, and set disk quotas to ensure that user files do not exhaust the free space on the disk volume. Entering the command `SET DIRECTORY/VERSION_LIMIT=n` is one way to control the amount of disk space

Setting Up and Maintaining Public Volumes

2.1 Planning Public Volumes

consumed. This command limits the number of versions of a file that can exist in a directory at one time.

In large mass storage configurations, you should keep all system files on one disk volume (known henceforth as a system disk or a system volume), and keep all user files on separate volumes. The system disk will be kept active reading system images, paging and swapping, spooling files, maintaining system logs, and so forth. See the *Guide to Setting Up a VMS System* for a complete list of the files that belong on the system disk.

The most common arrangement is to have one public volume with system files and the directories of privileged users, and other public volumes dedicated to user directories, databases, and applications required by your site.

Whichever arrangement you select, plan each public volume and test disk performance once the system is running. Be sure that the system disk has sufficient space for the VMS operating system to boot and accept updates, and that any user directories on the system disk are carefully controlled in size. Once the system is running, use the Monitor Utility to analyze disk use and determine whether or not disk I/O is balanced across the configuration. Often it is possible to move system files off the system disk and use search lists or logical names to access them. In large configurations, secondary paging and swapping files can easily be placed on other devices to balance disk load. The *Guide to VMS Performance Management* provides detailed information on redistributing system files and achieving a balanced disk load.

2.1.1 Establishing Volume-Level Protection

You can apply file protection to disk volumes as well as to the files and directories on them. Even though users may have access to the directories and files on a volume, they cannot access any part of the volume without the proper volume protection code.

When you initialize or mount a public disk volume, user identification code (UIC) protection is set by default to full access, giving all users READ, WRITE, EXECUTE, and DELETE (RWED) access to the volume. You can use the /PROTECTION qualifier with the MOUNT or INITIALIZE command to specify a more restrictive protection code. Once the volume is mounted, you can change the protection code with the SET VOLUME/PROTECTION command; for example:

```
$ SET VOLUME/PROTECTION=(S:RWED,C:RWED,G:RE,W:RE) DBA0:
```

Use the SHOW DEVICES command to verify the new protection code:

```
$ SHOW DEVICES/FULL DBA0:
Disk $11$DBA0: ...      Dev Prot   S:RWED,O:RWED,G:RE,W:RE
```

Although access control lists (ACLs) are not typically set for disk volumes because of the performance expense, you can set ACLs on volumes for additional access control. For a full explanation of how to set UIC and ACL protection, see the *VMS DCL Concepts Manual* or the *Guide to VMS System Security*.

2.1.2 Using Volume Sets

A volume set is a collection of disk volumes bound into a single entity by the DCL command MOUNT/BIND. To users, a volume set looks like a single, large volume. Files are automatically allocated anywhere on the volume set that space is available, disk quotas are enforced over the entire set, and a single directory structure covers the whole volume set. To provide a large homogeneous public file space, use a volume set.

Setting Up and Maintaining Public Volumes

2.1 Planning Public Volumes

To create files that are larger than a single physical disk volume, you must use a volume set. Note that the file system attempts to balance the load on the volume sets with tactics such as creating new files on the volume that is the least full at the time.

If you want several distinct areas of file storage, with different types of users or different management policies, you must use a separate volume (or volume set) for each area. For example, you might want one volume for permanent user storage, with limited disk quotas and regular backups. You might want another volume for “scratch” use, which has liberal or no quotas, is not backed up, and whose files are purged on a periodic basis. Each separate volume or volume set must contain a top-level user file directory for each user who will keep files on that volume.

An advantage of using separate volumes is their modularity. If one of the drives holding a volume set is out of service, the whole volume set will be unavailable because of its interconnected directory structure. When a drive holding a single volume is not functioning, only the files on that volume are not available.

When planning, keep in mind the following:

- Any single volume can be turned into a volume set by binding it with a newly initialized volume. Likewise, you can always add another newly initialized volume to an existing volume set.
- You can bind disk volumes into the same volume set as long as the volume is not a system disk.
- You cannot bind two existing separate volumes containing files into a volume set. (The MOUNT command appears to let you do this, but the result is not a coherent volume set.)
- Enter the MOUNT/BIND command only once to bind a volume set; thereafter, the volume-set association is recorded on the volumes. (See the *VMS Mount Utility Manual* for more information.)
- Once you have bound two or more volumes into a volume set, they cannot be separated. The only way to separate a volume set is to copy selectively sets of directories using the Backup Utility (BACKUP). (See the *VMS Backup Utility Manual* for more information.)

Caution

Do not make the system disk part of a volume set because updates, upgrades, and optional product installations will not install correctly, and the VMS operating system will no longer boot successfully.

2.2 Formatting Disks

Disks purchased from Digital are preformatted by a field service diagnostic utility program. However, you may need or want to format a disk. Disks must be reformatted if they have been exposed to X-rays, degaussing (demagnetizing), or certain kinds of power disruptions. Also, you might want to format a disk if there are excessive parity errors. In such cases, contact your Digital Customer Services Group representative for assistance.

Setting Up and Maintaining Public Volumes

2.3 Initializing Public Volumes

2.3 Initializing Public Volumes

The purpose of initializing a disk volume is to delete all old information from the volume and to impart to the volume a Files-11 structure recognized by the VMS operating system. This structure prepares a volume to receive data and stores it so that the operating system can locate it easily.

When initializing a public volume, you must specify the /SYSTEM qualifier with the DCL command INITIALIZE. For example:

```
INITIALIZE/SYSTEM device name[:] volume-label
```

You might want to use one or more of the following qualifiers when initializing a public volume:

- /CLUSTER_SIZE=*n*
- /HEADERS=*n*
- /INDEX=*position*
- /MAXIMUM_FILES=*n*
- /WINDOWS=*n*

Selecting appropriate values for these qualifiers and selecting the appropriate position for the index file involve making tradeoffs. Use the following guidelines for initializing public volumes. Refer to the *VMS DCL Dictionary* for complete information.

/CLUSTER_SIZE Qualifier

The /CLUSTER_SIZE qualifier specifies the number of disk blocks to be allocated each time disk space is needed. This is the minimum amount allocated whenever a file is created. When you select a small value (for example, 1) for /CLUSTER_SIZE, you might save disk space compared to selecting a relatively large value (for example, 4), because you will reduce the wasted space at the end of files. However, a small value will also result in increased disk and file fragmentation.

/HEADERS Qualifier

The /HEADERS qualifier specifies the number of file entries (called file headers) that you expect to have in the index file. The index file is a file holding the addresses of all disk files. The system accesses this index file each time it needs to locate a file on disk. An accurate estimate of the index file size allows the system to build one contiguous file, which translates into better system performance. Specifying too large a value wastes space.

If your estimate is too low and the index file needs to expand, the system uses a different part of the disk. However, a fragmented index file takes more time to scan. Do not be too generous with the /HEADERS value, because space allocated to headers cannot later be made available for file storage. A small disk might allocate 2000 entries in the index file, while a large disk might allocate the following:

```
$ INITIALIZE/HEADERS=100000 DUA3:
```

Setting Up and Maintaining Public Volumes

2.3 Initializing Public Volumes

/INDEX Qualifier

The /INDEX qualifier determines the location of the index file on a volume, using the keyword BEGINNING, MIDDLE, or END. The index file lists the names and addresses of all disk files, so it is constantly referenced. The default position, MIDDLE, uses the smallest amount of head motion during most file processing if the disk is full. However, the BEGINNING keyword should be used if the disk is to contain either one or very few large, contiguous files. In rare instances, an application might require the use of the beginning of a disk; in this instance, use the END keyword.

When the Backup Utility copies a volume as the result of a BACKUP/IMAGE operation, it preserves the placement of the index file, if the output device is the same type. Otherwise, it defaults to MIDDLE.

/MAXIMUM_FILES Qualifier

The /MAXIMUM_FILES qualifier specifies the maximum number of entries in the index file, and therefore limits the number of files that a volume can contain. Once set, the maximum number of files for a volume cannot be increased without reinitializing the disk. Note that each directory and each extension header of a multiheader file counts as a file against this maximum value. For example, a small disk might have the following characteristic:

```
$ INITIALIZE/MAXIMUM_FILES=20000 DUA3:
```

/WINDOWS Qualifier

The /WINDOWS qualifier sets the default number of mapping pointers to be allocated for file windows. When a file is opened, the file system uses mapping pointers to access data in the file. The file system can read one file segment into memory for each available pointer. For a large disk of 500 MB, you can cite a large number of pointers. For example:

```
$ INITIALIZE/WINDOWS=10 DUA3:
```

2.4 Mounting Public Volumes

Mounting a volume or volume set establishes a relationship between the volume or volume set and the device on which the volume is physically loaded. When mounting a volume set, first physically mount all disks and put them on line. Then enter the MOUNT command. Once invoked, the Mount Utility allocates the device, checks to see that the device is correctly loaded, and reads and verifies the disk label.

To mount a public volume, include the /SYSTEM qualifier with the DCL command MOUNT:

```
MOUNT/SYSTEM device-name volume-label logical-name
```

In a VAXcluster, specify the /CLUSTER qualifier as well, to mount the disk clusterwide.

Upon successful completion of the operation, MOUNT notifies you with a message sent to SYS\$OUTPUT. If the operation fails for any reason, MOUNT notifies you with an error message.

Setting Up and Maintaining Public Volumes

2.4 Mounting Public Volumes

You can modify the command with any of the following qualifiers:

- /ACCESSED=*n*
- /BIND=*volumeset_name*
- /CLUSTER
- /COMMENT=*text*
- /EXTENSION=*n*
- /[NO]ASSIST
- /[NO]MOUNT_VERIFICATION
- /SHARE
- /SYSTEM
- /WINDOWS=*n*

These qualifiers are described in the following section. For a complete description of the command and its qualifiers, see the *VMS Mount Utility Manual*.

/ACCESSED Qualifier

The /ACCESSED qualifier specifies the number of directories that the system will keep in memory for ready access. The file system selects the most recently used directories and, by storing them in memory, substantially reduces the overhead associated in directory operations. (For volumes mounted with the /SYSTEM qualifier, the SYSGEN parameter ACP_DINDXCACHE overrides this value.)

When you create a volume set, specify reasonable values for the /ACCESSED qualifier on each volume because the total number of directory file control blocks retained will be the sum of the values of all the /ACCESSED qualifiers specified for the volume set.

For example, on a large 500Mb disk you might select a value of 40, but on a small disk you might specify the following:

```
$ MOUNT/ACCESSED=2 DUA3:
```

/BIND Qualifier

The /BIND qualifier creates a volume set of one or more disk volumes or adds an existing volume to a volume set. Include a volume set name as part of the qualifier. For example:

```
$ MOUNT/SYSTEM/BIND=CLIENTS DMA0:,DMA1: EUROPE,ASIA
```

/CLUSTER Qualifier

The /CLUSTER qualifier specifies that after a public volume is successfully mounted on the local node with the /SYSTEM qualifier, it is to be mounted on every other node in the existing VAXcluster. For example:

```
$ MOUNT/SYSTEM/CLUSTER DBA1: SALES_86
```

For more information on mounting clusterwide disk volumes, see the *VMS VAXcluster Manual*.

Setting Up and Maintaining Public Volumes

2.4 Mounting Public Volumes

/COMMENT Qualifier

The /COMMENT qualifier includes a quoted text string that you specify as part of the mount request. This qualifier is useful in situations where operator assistance is expected, because it passes on information such as the physical location of a particular volume that is required. For example:

```
$ MOUNT/SYSTEM DYA1: SALES_86/COMMENT="Volume in Rack 2."
```

You should encourage users to take advantage of this feature.

/EXTENSION Qualifier

The /EXTENSION qualifier specifies the number of disk blocks allocated to a file each time it is expanded. The cluster size sets the initial disk block allocation and the extension qualifier determines how the file grows. The quantity by which a file is extended is always a multiple of the cluster size for the volume, because the cluster size is the minimum allocation quantity in that volume. In the following example, a small disk with a cluster size of 1 disk block selects an extension size of 2 disk blocks:

```
$ MOUNT/EXTENSION=2 DUA3:
```

/[NO]ASSIST Qualifier

The /NOASSIST qualifier disables automatic notification of any mount failures. Normally, an operator is told whenever the system is unable to process a mount request. This gives the operator an opportunity to terminate the operation or correct the problem so the mount can continue. Encourage users to take advantage of this feature, which repeatedly alerts the operator of a mount request until the request is satisfied. If you want to disable operator-assisted mounts, you could enter the following command:

```
$ MOUNT/SYSTEM/NOASSIST DBA1: SALES_86
```

The /ASSIST qualifier is the default except during system startup. During the execution of STARTUP.COM, operator-assisted mounts are disabled because there is no operator defined at this time.

/[NO]MOUNT_VERIFICATION Qualifier

The /[NO]MOUNT_VERIFICATION qualifier enables or disables the mount verification feature on disks and magnetic tapes. By default, the mount verification feature is enabled. If a device goes off line or becomes write-locked, mount verification notifies the operator of the error condition, and then checks to see that the volume identification codes recorded before and after the error condition are identical. Once verified, the operations can continue from the error point, rather than restarting the mount from the beginning. Mount verification retains pending requests in a queue while attempting to recover, instead of aborting them. To disable mount verification, you could enter the following command:

```
$ MOUNT/SYSTEM/NOMOUNT_VERIFICATION MTB1: ACCOUNTS_DUE
```

Setting Up and Maintaining Public Volumes

2.4 Mounting Public Volumes

/SHARE Qualifier

The /SHARE qualifier specifies that other users can access the current volume. Use this qualifier instead of the /SYSTEM qualifier when you have READ access to a volume, but not the SYSNAM privilege that /SYSTEM requires. While the qualifier creates a volume that can be shared by several users, public volumes are created with the /SYSTEM qualifier. Use /SHARE in the following way:

```
$ MOUNT/SHARE DLA0: COST_ACCOUNT
```

/SYSTEM qualifier

The /SYSTEM qualifier creates a public volume (a volume available to all system users with the proper UIC). Use of this qualifier requires SYSNAM privilege. If you mount a volume with this qualifier in a VAXcluster environment, you must use a unique volume label, even if you are not mounting the volume clusterwide.

The following command mounts the volume labeled WORK and makes it available systemwide:

```
$ MOUNT/SYSTEM DUA1: WORK
```

/WINDOWS Qualifier

The /WINDOWS qualifier on the MOUNT command resets the number of mapping pointers to be allocated for file windows. The default number of windows is set with the INITIALIZE command. When a file is opened, the file system uses mapping pointers to access data in the file. The file system can read one file segment into memory for each retrieval pointer. For a small disk you may cite a modest number of pointers; for example:

```
$ MOUNT/WINDOWS=4 DUA3:
```

2.5 Using the Exchange Utility to Transfer Information

This section describes how to use the Exchange Utility to transfer information between foreign volumes and VMS Files-11 volumes. The Exchange Utility (EXCHANGE) converts the format of files, as appropriate, when transferring files between volumes of different structures. EXCHANGE recognizes all Files-11 and RT-11 disk volumes on VMS devices, as well as all DOS-11 and RT-11 formatted volumes on 9-track magnetic tape devices.

Use EXCHANGE commands to perform any of the following tasks:

- Locate bad blocks on volumes
- List directories of volumes
- Transfer files to and from volumes
- Delete and rename files for block-addressable devices
- Create foreign volumes
- Manipulate Files-11 files that are images of foreign volumes

For information on how to use EXCHANGE and for a description of all EXCHANGE commands, qualifiers, and parameters, see the *VMS Exchange Utility Manual*.

Setting Up and Maintaining Public Volumes

2.5 Using the Exchange Utility to Transfer Information

2.5.1 Running the EXCHANGE Utility Interactively

To invoke the Exchange Utility, enter the following command in response to the DCL prompt:

```
$ EXCHANGE
```

The utility displays the following prompt:

```
EXCHANGE>
```

You can now enter any EXCHANGE command. For example:

```
EXCHANGE> DIRECTORY MFA0:/VOLUME=DOS11/FULL
```

This command lists all the files on the DOS-11 magnetic tape mounted on MFA0:. In this case, the magnetic tape will be rewound before the files are listed.

The following example illustrates the use of the MOUNT command within the Exchange Utility. This command mounts the foreign volume that is loaded in the RK07 device DMA1:, making the volume available for subsequent commands. In this case, EXCHANGE recognizes that the volume itself is write-locked, and displays the following message:

```
EXCHANGE> MOUNT DMA1:
%EXCHANGE-I-WRITELOCK, volume is write-locked
%EXCHANGE-I-MOUNTED, volume DMA1: mounted
```

To exit from EXCHANGE and return to DCL level, use the EXCHANGE command EXIT or Ctrl/Z.

You can also type EXCHANGE at the DCL prompt and enter a single EXCHANGE command on the same line. EXCHANGE executes the single command and returns to the DCL prompt. For example, you can list the directory of a foreign volume by appending the EXCHANGE command DIRECTORY to the DCL command EXCHANGE, as follows:

```
$ EXCHANGE DIRECTORY DMA1:/VOLUME_FORMAT=RT11
```

This DCL command lists the directory of the RT-11 volume mounted /FOREIGN on DMA1.

Tasks requiring more than one EXCHANGE command cannot be performed at DCL level unless you design a command procedure. Then you execute the command procedure at the DCL command level, as the following section describes.

2.5.2 Using a Command Procedure to Exchange Information

The command procedure in the following example exchanges files between the console device and the current directory on disk. This procedure assumes that the files to be copied are in standard format, as determined by file type.

```
$ WRITE SYS$OUTPUT " Command file to copy files to/from the system"
$ WRITE SYS$OUTPUT " console storage medium and the current directory."
$ WRITE SYS$OUTPUT " "
$ INQUIRE MOUNT "Is system console storage medium mounted (Y/N)?"
$ IF MOUNT THEN GOTO MOUNTED
$ WRITE SYS$OUTPUT "Please place the system console medium in the console drive"
$ INQUIRE MOUNT "and press Return when ready"
```

Setting Up and Maintaining Public Volumes

2.5 Using the Exchange Utility to Transfer Information

```
$ RUN SYS$SYSTEM:SYSGEN
CONNECT CONSOLE
$ MOUNT/SYSTEM/FOREIGN CSA1: "VAX console"
$
$ MOUNTED:
$ INQUIRE DIR "Copy from console medium (Y/N)?"
$ IF DIR THEN GOTO FROMCON
$ INQUIRE SOURCE "Enter file name(s)"
$ IF SOURCE .EQS. "" THEN GOTO EXIT
$ EXCHANGE COPY /LOG 'SOURCE' CSA1:
$ GOTO EXIT
$
$ FROMCON:
$ INQUIRE SOURCE "Enter console file name"
$ IF SOURCE .EQS. "" THEN GOTO EXIT
$ EXCHANGE COPY /LOG CSA1:'SOURCE' *
$ EXIT:
$ DISMOUNT CSA1:
$ MOUNT/SYSTEM/FOREIGN/NOWRITE CSA1: "VAX console"
```


Performing Disk and Magnetic Tape Operations

This chapter describes the following disk and magnetic tape operations:

- Using the Operator Communication Manager (OPCOM)
- Servicing mount requests from users
- Servicing requests from the Backup Utility
- Maintaining volume integrity
- Performing mount verification

3.1 Using the Operator Communication Manager (OPCOM)

The Operator Communication Manager (OPCOM) is an online system management tool that does the following:

- Handles requests from users
- Delivers requests to the operator
- Records messages in the operator log file

The following sections describe how to set up an operator terminal to enable OPCOM and how to handle user requests. For more information on OPCOM, see the `REPLY` and `REQUEST` commands in the *VMS DCL Dictionary*. For more information on the operator log file, see Section 6.3.

3.1.1 Enabling the Operator Terminal

When you respond to system users' requests to mount disks and magnetic tapes, you work from a terminal that has been defined as an operator terminal. Normally, the console terminal is designated as the operator terminal (OPA0:).

To designate a particular terminal as an operator terminal, enter the DCL command `REPLY/ENABLE` from the desired terminal. The `REPLY/ENABLE` command requires `OPER` privilege. When OPCOM receives a `REPLY/ENABLE` command, it displays the following message on the enabled operator terminal:

```
%%%%%%%%%% OPCOM, dd-mmm-yyyy hh:mm:ss,cc, %%%%%%%%%%%
Operator _nodename$terminal-name: has been enabled, username USERNAME
```

OPCOM writes the same message to the operator log file. This message tells which terminal has been established as an operator terminal and when it was established.

By default, OPA0 is enabled for all classes on all systems except workstations in a VAXcluster configuration. You can change these defaults by defining certain logical names in the command procedure `SYS$MANAGER:SYLOGICALS.COM`.

Performing Disk and Magnetic Tape Operations

3.1 Using the Operator Communication Manager (OPCOM)

To specify whether OPA0 will be enabled, define the logical name OPC\$OPA0_ENABLE in the command procedure SYS\$MANAGER:SYLOGICALS.COM. For example, if you do not want OPA0 to be enabled, add the following command to SYLOGICALS.COM:

```
$ DEFINE/SYSTEM OPC$OPA0_ENABLE "FALSE"
```

To disable certain operator classes on OPA0, define the logical name OPC\$OPA0_CLASSES in SYLOGICALS.COM. Define OPC\$OPA0_CLASSES to be a search list of the allowed classes, a list of classes, or a combination of the two. For example, to disable all operator classes *except* CENTRAL, DISKS, and TAPES, add the following line to SYLOGICALS.COM:

```
$ DEFINE/SYSTEM OPC$OPA0_CLASSES CENTRAL,DISKS,TAPES
```

For information on the operator classes, see the REPLY/ENABLE command in the *VMS DCL Dictionary*. For information on SYLOGICALS.COM, see the *Guide to Setting Up a VMS System*.

To disable operator terminal functions, log out or enter the DCL command REPLY /DISABLE.

3.1.2 Restarting OPCOM

If, under abnormal conditions, OPCOM is deleted or suspended, you can restart it manually. To restart OPCOM, log in to the SYSTEM account and execute STARTUP.COM by entering the following command:

```
$ @SYSSYSTEM:STARTUP OPCOM
```

After a system reboot, OPCOM normally restarts automatically as soon as the system startup procedure executes.

3.1.3 Requests for Operator Assistance

Users communicate with the operator using the REQUEST command, specifying either the /TO or the /REPLY qualifier. The REQUEST command sends a text message to the operator, but it does not initiate any further system action.

The REQUEST command with the /REPLY qualifier means that the user is waiting for a response from the operator. The user's current process is put in a wait state until you respond. When the user enters a REQUEST/REPLY command, the request appears at the operator terminal in the following format:

```
%%%%%%%%% OPCOM,dd-mmm-yyyy hh:mm:ss.cc %%%%%%%%%%%  
request request-id from user USERNAME  
__terminal-name:, "message-text"
```

This message explains which user sent the message, the time it was sent, the request identification number assigned to the message, the originating terminal, and the request message.

If the user enters a REQUEST/TO command, the request appears at the operator terminal in a format similar to the previous one, but without a request identification number. The message will have the following format:

```
%%%%%%%%% OPCOM,dd-mmm-yyyy hh:mm:ss.cc %%%%%%%%%%%  
request from user USERNAME  
__terminal-name:, "message-text"
```

Performing Disk and Magnetic Tape Operations

3.1 Using the Operator Communication Manager (OPCOM)

If all operator terminals are disabled when a user enters a REQUEST/REPLY command, OPCOM returns a message to the user indicating that no operator is available. However, it records this request and all subsequent requests in the operator log file (SYS\$MANAGER:OPERATOR.LOG). For more information on the operator log file, see Chapter 6.

Users can also enter the MOUNT command for a disk or magnetic tape operation and request operator assistance. (See Section 3.2.) For more detailed information on the REQUEST and the MOUNT commands, including a list of all the command qualifiers, see the *VMS DCL Dictionary* and the *VMS Mount Utility Manual*.

3.1.4 Responding to User Requests

You communicate with system users by entering the REPLY command in one of the following forms:

- Use the REPLY/ABORT=*identification-number* "*message-text*" command to indicate that the user request is canceled.
- Use the REPLY/PENDING=*identification-number* "*message-text*" command when the request has been put in a wait state until it can be completed. This command implies that the originating request was either a REQUEST/REPLY or a MOUNT command. The user cannot enter other commands until the operator fulfills or aborts the request.
- Use the REPLY/TO=*identification-number* "*message-text*" command when the request is fulfilled.

In situations where users submit a MOUNT/ASSIST command and the desired device is unavailable, you can substitute another device. Load the requested volume on the alternate device and ready the device before entering the REPLY command in the following form.

```
REPLY/TO=identification-number "SUBSTITUTE device-name"
```

You can abbreviate the word SUBSTITUTE to "S" and can use uppercase or lowercase characters. The following example shows how an operator redirects the mount operation to DMA1:

```
$ REPLY/TO=24 "SUBSTITUTE DMA1"
```

3.2 Handling Requests for Mounting Volumes

At many installations, operators perform the physical mounting and dismounting of public and private disk and magnetic tape volumes. At these installations, users do not have access to the computer room so they communicate with operators using DCL commands. When users need to have a tape or disk mounted, they submit a request using the REQUEST command or the MOUNT command. The REQUEST command requests assistance and describes the volume to be mounted. The MOUNT command specifies the name of the device to be allocated, the volume to be mounted, and perhaps a comment indicating the location of the volume. For example:

```
$ MOUNT DYA0: ACCOUNTS_USA/COMMENT="Write enable, volume in rack 1"
```

Performing Disk and Magnetic Tape Operations

3.2 Handling Requests for Mounting Volumes

Use the following steps to respond to a user request to mount a disk or magnetic tape media:

1. Locate the designated volume. If the user intends to write to a disk or tape, toggle the disk drive's write-lock switch or insert a write ring in the tape volume. (Any write operation will fail if the volume is not write-enabled.)
2. Place the volume on the specified device. On disk drives, press the START or RUN button; on magnetic tape drives, press the LOAD button.
3. Enter the REPLY command with the appropriate qualifier.

If a user wants a particular disk to be initialized, but has already entered the MOUNT command, dismount the volume with the DCL command DISMOUNT, enter the DCL command INITIALIZE, and then mount the volume a second time, using the MOUNT command. Be careful to specify the user's ownership when you remount the device.

3.2.1 Mounting Disks and Single Magnetic Tape Volumes

When a user requests that a specific disk or magnetic tape be mounted on a device, the following type of message appears on the operator terminal:

```
%%%%%%%%%% OPCOM dd:mmm:yyyy:hh:mm:ss:cc %%%%%%%%%%%  
request request-id, from user USERNAME
```

For example, a user requesting to mount the volume TEST_FILES on the device DMA2 could enter the following command:

```
$ MOUNT DMA2: TEST_FILES/COMMENT="Shelf slot 6B"
```

OPCOM notifies you of the request by displaying a message similar to the following at the operator terminal:

```
%%%%%%%%%% OPCOM, 28-DEC-1990 15:47:50.26 %%%%%%%%%%%  
request 5, from user MALCOLM  
Please mount volume TEST_FILES in device _DMA2:  
Shelf slot 6B
```

Once you receive the request, OPCOM delivers a confirmation to the user, in a format similar to the following:

```
%MOUNT-I-OPRQST, Please mount volume TEST_FILES in device _DMA2:  
Shelf slot 6B
```

After you locate the volume and place it on the device, OPCOM notifies the user that the volume is on the device and that the task is complete.

```
%MOUNT-I-MOUNTED, TEST_FILES mounted on _DMA2:  
%MOUNT-I-RQSTDON, operator request canceled  
-- mount completed successfully.
```

Instead of requesting a specific hardware device, such as DMA2 or MTA0 for mounting a volume, the user can make a generic MOUNT request. A generic MOUNT request specifies a type of device and lets the operator find an available device in that class. For example, to mount the volume CITIES on any magnetic tape drive whose name begins with MT, the user would enter the following command:

```
$ MOUNT MT: CITIES/COMMENT="Slot 12c"
```

If the user has already allocated a drive whose name begins with MT, the Mount Utility requests that the operator mount CITIES on that particular drive. If no device has been allocated, the Mount Utility allocates the first available MT tape drive it finds and requests you to mount CITIES on that drive.

3.2.2 Mounting Magnetic Tape Volume Sets

The procedure for mounting a magnetic tape volume set is similar to the procedure for mounting a single magnetic tape volume described in Section 3.2.1. When mounting a volume set, however, make sure that all the volumes in the set contain write rings if the user intends to write to any of the volumes in the set. (If even one of the volumes in the set does not contain a write ring at mount time, all volumes will be write-locked; the system will be unable to write to any of them.) Load the volumes on the drives that have been allocated and place the drives on line.

The next two sections describe how to handle user requests to mount a magnetic tape volume set under the following conditions:

- Automatic volume switching is enabled by default
- Automatic volume switching is disabled, with the MOUNT/NOAUTOMATIC command

3.2.2.1 Mounting Volume Sets with Automatic Switching

The VMS operating system supports automatic volume recognition (AVR) and automatic volume labeling (AVL) for ANSI magnetic tapes. If the magnetic tape volume is mounted with AVR and AVL enabled, which it is by default, then your only task is to place the next volume in the drive. The system automatically reads the label and mounts the volume. The system then sends a message to the operator terminal stating that a volume switch has occurred.

With automatic volume switching, you can load each successive magnetic tape in the volume set on one of the allocated drives any time before the magnetic tape being processed reaches an end-of-tape position. The magnetic tape file system mounts or initializes (or both mounts and initializes) the next magnetic tape, and then notifies you that a switch has occurred. The magnetic tape file system switches volumes by issuing pending read or write requests to the next volume in the set.

Note that to take advantage of the automatic volume-switching capability, more than one magnetic tape drive must be allocated to the magnetic tape volume set; otherwise, automatic volume switching will be implicitly disabled.

3.2.2.2 Mounting Volume Sets Without Automatic Switching

This section describes how to handle requests for mounting continuation volumes in a volume set when automatic volume switching is disabled or when the magnetic tape file system cannot mount a given volume.

When a user is reading or writing to a magnetic tape and that tape reaches end-of-tape position, the system suspends processing and sends a request to mount the next magnetic tape in the volume set. For example:

```
%%%%%%%%%% OPCOM, 28-DEC-1990 15:23:31.78 %%%%%%%%%%%  
request 3, from user PLAW  
MOUNT new relative volume 2 (DWOQT2) on MTA1:
```

The user does not see this message and may not realize that another tape is needed to complete the read or write operation.

After loading the continuation volume on the drive specified in the mount request, you mount the volume by entering the REPLY command with one of three qualifiers:

- Use the REPLY/TO=*request_id* [*volume identifier*] command for both read and write operations. During a write operation, you use the /TO qualifier if you

Performing Disk and Magnetic Tape Operations

3.2 Handling Requests for Mounting Volumes

want the volume identifier that is specified in the mount request to be written on the continuation volume.

For example, to respond to the mount request 3, you would mount volume DW0QT2 on drive MTA1: and enter the following command:

```
$ REPLY/TO=3 "DW0QT2"
```

The volume identifier is required in certain instances, but not in others. See the explanation that follows this list.

- Use the `REPLY/INITIALIZE_TAPE` command for write operations if the volume identifier on the continuation volume does not match the one specified in the mount request. The file system reinitializes the tape and mounts the volume with the new volume identifier. The magnetic tape file system then performs access checks and initializes the volume as if the `INITIALIZE` command had been specified. For example:

```
$ REPLY/INITIALIZE_TAPE=3 "DW0QT2"
```

- Use the `REPLY/BLANK_TAPE` command to write to an unformatted volume. This qualifier initializes the volume and requires the `VOLPRO` and `OPER` privileges to avoid a runaway tape or timeout condition. Either of the following `REPLY` commands is valid:

```
$ REPLY/BLANK_TAPE=3  
$ REPLY/BLANK_TAPE=3 "DW0QT2"
```

Specifying the volume identifier in either the `MOUNT` command or the `REPLY/TO` command is essential during write operations because it ensures that the correct volume is mounted on the drive and links the continuation volume to the volume set.

You can omit the volume identifier with the `REPLY/TO` command under two circumstances. When reading from tape, the volume identifier is optional. During a write operation, you must omit the volume identifier to preserve the accessibility code on a volume. If you initialize and mount a volume set in which each volume has a unique accessibility character that you want to maintain, avoid using the volume identifier because it causes the accessibility character of the first volume in the set to overwrite the accessibility code on the continuation volume. For example, to preserve the accessibility character you would enter the following command, where 3 is the request identification number:

```
$ REPLY/TO=3
```

Once it receives the `REPLY` command, the magnetic tape file system performs checks on the continuation volume to ensure that it is the correct volume. As long as it is the correct volume with proper access codes, the system mounts the volume and reissues pending read or write requests to the continuation volume. If the volume fails any of these access checks, the volume is not mounted (or initialized and mounted, in the case of a blank tape).

3.3 Handling Requests from the Backup Utility

When a `BACKUP` operation is submitted as a batch job, operators receive requests to load the next volume of a save set or to reload a volume in the event of an error. (These requests go to the user when the Backup Utility is invoked interactively.)

Performing Disk and Magnetic Tape Operations

3.3 Handling Requests from the Backup Utility

3.3.1 Writing to a Save Set

During a BACKUP save operation, while the file system is copying data from disk to magnetic tape, you receive a message if an operation requires the loading of an additional volume. The message states the date and time, a request number, the user name, and the device name, as shown in the following example:

```
%%%%%%%%%% OPCOM,28-DEC-1990 17:02:32.31 %%%%%%%%%%%  
Request 24, from user TOM.  
%BACKUP-I-READYWRITE, mount volume 2 on _MTA0: for writing
```

To continue the BACKUP operation, load a scratch volume, put the drive on line, and enter a **REPLY/TO=*n*** command, specifying the request number as parameter *n* as shown in the following example:

```
$ REPLY/TO=24  
%%%%%%%%%% OPCOM,28-DEC-1990 17:02:34.14 %%%%%%%%%%%  
Request 24 completed by operator OPA0.
```

You can also abort the BACKUP operation by entering a **REPLY/ABORT** command, as shown in the following example:

```
$ REPLY/ABORT=24  
%%%%%%%%%% OPCOM,28-DEC-1990 17:02:34.14 %%%%%%%%%%%  
Request 24 aborted by operator OPA0.
```

3.3.2 Reading from a Save Set

During a restore operation, when the file system reads data from BACKUP media and writes it to disk, you receive a message if a restore operation requires the loading of an additional volume. The message states the date and time, a request number, the user name, and the device name, as shown in the following example:

```
%%%%%%%%%% OPCOM,28-DEC-1990 17:02:32.31 %%%%%%%%%%%  
Request 24, from user TOM.  
%BACKUP-I-READYREAD, mount volume 2 on _MTA0: for reading
```

To continue the restore operation, place the next volume of the save set on the drive, put the drive on line, and enter a **REPLY/TO** command, as shown in the following example:

```
$ REPLY/TO=24  
%%%%%%%%%% OPCOM,28-DEC-1990 17:02:34.14 %%%%%%%%%%%  
Request 24 completed by operator OPA0.
```

You can also abort the restore operation by entering a **REPLY/ABORT** command, as shown in the following example:

```
$ REPLY/ABORT=24  
%%%%%%%%%% OPCOM,28-DEC-1990 17:02:34.14 %%%%%%%%%%%  
Request 24 aborted by operator OPA0.
```

3.3.3 Recovering from an Error

On certain BACKUP save or restore errors, you receive messages stating the date and time, a request number, the user name, the device name, and reply options. On read errors from the save set, the options are usually **CONTINUE** and **QUIT**.

Performing Disk and Magnetic Tape Operations

3.3 Handling Requests from the Backup Utility

On write errors to the save set, the options are usually RESTART and QUIT. You should take one of the following actions:

- If you can fix the problem and are given the CONTINUE option, put the drive on line and enter a REPLY/TO=*n* command specifying the request number in parameter *n* and the word CONTINUE as the text. For example:

```
REPLY/TO=n "CONTINUE"
```

- If you can fix the problem and are given the RESTART option, load the volume (if necessary), put the drive on line, and enter a REPLY/TO=*n* command specifying the request number and the word RESTART as the text. For example:

```
REPLY/TO=n "RESTART"
```

- If you cannot fix the problem, enter a REPLY/TO=*n* command specifying the request number and the word QUIT as the text. For example:

```
REPLY/TO=n "QUIT"
```

The text in the REPLY/TO command can be uppercase or lowercase, and can be abbreviated to the first character.

Assume, for example, that you have mounted a tape without mount verification enabled and you receive the following messages during a restore operation (when the save set is being read):

```
%%%%%%%%%% OPCOM,28-DEC-1990 17:02:30.58 %%%%%%%%%%%
Message from user RESJOB.
%BACKUP-E-FATALERR, fatal error on MT:[]SAVE.;
%%%%%%%%%% OPCOM,28-DEC-1990 17:02:30.89 %%%%%%%%%%%
Message from user RESJOB.
-SYSTEM-F-MEDOFLL, medium is offline

%%%%%%%%%% %OPCOM,28-DEC-1990 17:02:31.05 %%%%%%%%%%%
Request 24, from user RESJOB.
%BACKUP-I-SPECIFY, specify option (QUIT or CONTINUE)
```

If you check the drive and find that it lost its vacuum, put the drive on line and enter the following command:

```
$ REPLY/TO=24 "CONTINUE"
CONTINUE
%%%%%%%%%% OPCOM,28-DEC-1990 17:02:33.41 %%%%%%%%%%%
Request 24 completed by operator OPA0.
```

If you find the drive inoperable, enter the QUIT reply:

```
$ REPLY/TO=24 "QUIT"
QUIT
%%%%%%%%%% OPCOM,28-DEC-1990 17:02:33.41 %%%%%%%%%%%
Request 24 completed by operator OPA0.
```

If the error occurs during a save operation (when the tape save set is being written), you have a choice of QUIT or RESTART:

```
%%%%%%%%%% OPCOM,28-DEC-1990 17:02:30.58 %%%%%%%%%%%
Message from user RESJOB.
%BACKUP-E-FATALERR, fatal error on MT:[]SAVE.;
%%%%%%%%%% OPCOM,28-DEC-1990 17:02:30.89 %%%%%%%%%%%
Message from user RESJOB.
-SYSTEM-F-MEDOFLL, medium is offline
%%%%%%%%%% OPCOM,28-DEC-1990 17:02:31.05 %%%%%%%%%%%
Request 24, from user RESJOB.
%BACKUP-I-SPECIFY, specify option (QUIT or RESTART)
```


Performing Disk and Magnetic Tape Operations

3.3 Handling Requests from the Backup Utility

Put the tape drive on line, and choose the RESTART option:

```
.  
. .  
$ REPLY/TO=24 "RESTART"  
RESTART  
%%%%%%%%%% OPCOM,28-DEC-1990 17:02:33.35 %%%%%%%%%%%  
Request 24 completed by operator OPA0.
```

The RESTART option permits you to restart a multivolume BACKUP operation at the beginning of the current volume. If you specify the QUIT option, you must restart the BACKUP operation from the beginning of the first volume.

3.4 Maintaining Volume Integrity

To enhance performance, the system caches in memory information about a disk volume's free space, file identifications, quota file entries, and file headers. You determine the degree of caching with the ACP cache system parameters, discussed in the *VMS System Generation Utility Manual*. Individual users can alter cache sizes on their volumes with qualifiers of the DCL command MOUNT, described in the *VMS Mount Utility Manual*.

The system writes the information in the caches to the disk when the disk is dismounted or the system is shut down. Removing a disk from a drive before the caches are written to disk causes the information in the caches to be lost. Therefore, avoid doing the following:

- Write-locking a volume while it is mounted
- Removing a volume from a drive before it has been dismounted
- Halting the system without performing an orderly shutdown procedure (see the *Guide to Setting Up a VMS System*)

If anyone write-locks a volume at mount time, the system additionally applies a software write-lock. To write-enable a volume that was mounted while the WRITE LOCK switch was on, you must first dismount the volume, write-enable the drive, and then remount the volume. If a volume was mounted on a drive with write-lock off and someone toggles the WRITE LOCK switch while mount verification is enabled for the volume, the volume enters mount verification. All I/O operations to the volume are suspended. Section 3.5.2 describes how recovery is affected with write-lock mount verification. Without mount verification, you would have to dismount the volume, write-enable the drive, and then remount the volume.

At mount time, if the system detects that the caches were not written back the last time the volume was used, the system automatically rebuilds the file information by scanning the contents of the volume. However, file headers for files open at the time of the improper dismount may be partially or entirely lost.

3.5 Mount Verification

Mount verification provides a recovery mechanism for disk and tape operations. If a device is write-locked or goes off line while mount verification is enabled, you can correct the problem and continue the operation, rather than abort the entire disk or tape operation. As soon as mount verification detects an error condition, it stalls all I/O to the disk or tape, waits for the error resolution, and then validates the volume on the enabled device.

Performing Disk and Magnetic Tape Operations

3.5 Mount Verification

The mount verification feature of disk and tape handling generally leaves users unaware that a mounted disk or tape has gone off line and returned on line, or in some other way has become unreachable and then restored. Mount verification is enabled by default with the /MOUNT_VERIFICATION qualifier when the disk or tape is mounted. To disable mount verification, you must specify /NOMOUNT_VERIFICATION when mounting the disk or tape. Note that this feature applies to ANSI mounted tapes, foreign mounted tapes, and Files-11 disks.

Mount verification sends two messages: one goes to OPCOM and the other, distinguished by the prefix %SYSTEM-I-MOUNTVER, goes directly to the system console (OPA0, bypassing OPCOM). The second message is a form of insurance in cases in which OPCOM is unavailable. For example, if the system disk undergoes mount verification or if OPCOM is not present on a system, you would at least receive the messages with the %SYSTEM-I-MOUNTVER prefix. Under normal circumstances, both messages are received at the operator terminal, with the %SYSTEM-I-MOUNTVER message arriving first.

3.5.1 Mount Verification for Offline Devices

Any offline condition initiates mount verification. Usually an offline condition results from a hardware or user error—for example, someone inadvertently presses the offline button. Once a device is off line, the hardware (and for some disks, the software) marks the disk or tape as “invalid,” and I/O requests fail. As long as mount verification is enabled, the software marks the volume to indicate that it is undergoing mount verification and stalls all I/O operations to the disk or tape until the problem is corrected. OPCOM then issues a message in the following format to the operators enabled for DISKS and DEVICES or TAPES and DEVICES to announce the disk’s or tape’s unavailability:

```
%%%%%%%%%% OPCOM, dd-mmm-yyyy hh:mm:ss.cc %%%%%%%%%%%  
Device 'device-name' is offline.  
Mount verification in progress.
```

When a mounted disk or tape volume goes off line while mount verification is enabled, you can try to recover or you can terminate the mount request. The following options are available:

- Try to put the device on line by toggling the START or RUN button on disks or the LOAD button on tapes.
- If the disk or tape drive is faulty, but another functioning drive is available on the same controller, move the disk or tape to the functioning drive and swap the unit select plugs.
- Take the disk or tape out of the offline and verification pending state by shutting down mount verification with one of the three techniques described in Section 3.5.3. These techniques include canceling the mount request, dismounting the volume, and allowing mount verification to time out.

If the device is successfully put back on line, the mount verification software that polls the disk or tape drive will begin verification. The system checks to see that the currently mounted disk or tape has the same identification as the previously mounted volume. Thus mount verification confirms that this is the same disk or tape as was previously mounted and no switching has occurred.

If the drive contains the wrong volume, OPCOM issues a message in the following format:

Performing Disk and Magnetic Tape Operations

3.5 Mount Verification

```
%%%%%%%%%% OPCOM, dd-mmm-yyyy hh:mm:ss.cc %%%%%%%%%%%  
Device 'device-name' contains the wrong volume.  
Mount verification in progress.
```

Once mount verification succeeds, the disk is marked as valid, and OPCOM issues the following message:

```
%%%%%%%%%% OPCOM, dd-mmm-yyyy hh:mm:ss.cc %%%%%%%%%%%  
Mount verification completed for device 'device-name.'
```

At this point I/O operations to the disk or tape proceed, as shown in the following example:

```
%%%%%%%%%% OPCOM, 28-DEC-1990 11:54:54.12 %%%%%%%%%%%  
Device DMA0: is offline.  
Mount verification in progress.  
  
%%%%%%%%%% OPCOM, 28-DEC-1990 11:57:34.22 %%%%%%%%%%%  
Mount verification completed for device DMA0:.
```

In this example, the message from OPCOM informs the operator that device DMA0 went off line and mount verification was initiated. The operator finds that the drive was accidentally powered down and successfully powers it up again. The next message indicates that mount verification is satisfied that the same volume is on the drive as was there before the error. All I/O operations to the volume resume.

3.5.2 Mount Verification for Write-Locked Devices

Devices become write-locked when a hardware or user error occurs while a disk or a magnetic tape volume is mounted for a write operation. For example, if a disk is write-locked or a tape is missing a write ring, the hardware generates an error. As soon as the software discovers that the disk or tape is write-locked (for example, when an I/O operation fails with a write-lock error), mount verification begins.

OPCOM issues a message to the operators enabled for DISKS and DEVICES or TAPES and DEVICES to announce the disk's or tape's unavailability:

```
%%%%%%%%%% OPCOM, dd-mmm-yyyy hh:mm:ss.cc %%%%%%%%%%%  
Device 'device-name' has been write-locked.  
Mount verification in progress.
```

You can either recover the operation or terminate mount verification. Your options include the following:

- Enable the drive for writing by toggling the disk's or tape drive's hardware WRITE LOCK switch, or check to see that a tape volume has a write ring.
- If the disk or tape drive is faulty, but another functioning drive is available on the same controller, move the disk or tape to the functioning drive and swap the unit select plugs. (Note that switching to another drive causes the volume to undergo offline mount verification. Once that completes, the write-lock mount verification continues.)
- Terminate the mount operation by shutting down mount verification with one of the techniques described in Section 3.5.3. These techniques include canceling the mount request and dismounting the volume.

Once the mount verification software determines that the volume is in a write-enabled state, I/O operations to the disk resume with no further messages.

Performing Disk and Magnetic Tape Operations

3.5 Mount Verification

3.5.3 Canceling Mount Verification

You can cancel a mount verification request in one of the following ways:

- Dismount the volume with the DCL command DISMOUNT from a process that is not hung.
- If the device is off line, allow mount verification to time out. The expiration value is 10 minutes by default; however, you can use the system parameter MVTIMEOUT (for disk) or TAPE_MVTIMEOUT (for magnetic tape) to set the value to whatever you want. When the time expires, the system automatically cancels the pending mount verification. Note that a mount verification initiated by a write-lock condition will not time out.
- Invoke a special canceling routine from the console terminal.

The following sections describe these methods in more detail.

3.5.3.1 Dismounting the Volume

If it is possible for you to enter the DCL command DISMOUNT for the volume, then you can abort mount verification by dismounting the volume in question. (DISMOUNT requires GRPNAM and SYSNAM privileges to dismount group and system volumes, respectively.) To dismount a volume, follow these steps:

1. Log in at another terminal or use any logged-in terminal that has access to the volume. It does not have to be an operator terminal.
2. Enter the DISMOUNT/ABORT command for the volume. If your system is a VAXcluster environment, also specify the /CLUSTER qualifier.

When you cancel a pending mount verification by dismounting the volume, OPCOM issues a message in the following format:

```
%%%%%%%%%% OPCOM, dd-mmm-yyyy hh:mm:ss.cc. %%%%%%%%%%%  
Mount verification aborted for device 'device-name.'
```

If you do not have access to the volume, you will receive an error message. You can try again if you can find an appropriate process to use. If your process hangs, it is the system file ACP that is hung, and you cannot use this technique to cancel mount verification.

3. Once the cancellation succeeds, remove the volume from the drive.

3.5.3.2 MVTIMEOUT and TAPE_MVTIMEOUT System Parameters

The MVTIMEOUT system parameter for disks and the TAPE_MVTIMEOUT system parameter for tapes define the time (in seconds) that is allowed for a pending mount verification to complete before it is automatically canceled. (See the *VMS System Generation Utility Manual* for more information on system parameters.) This parameter should always be set to a reasonable value for the typical operations at your site. Note that resetting the value of the MVTIMEOUT parameter or the TAPE_MVTIMEOUT parameter will not affect a mount verification that is currently in progress.

Ten minutes (600 seconds) is usually a good value for MVTIMEOUT and TAPE_MVTIMEOUT, whether you usually operate with or without an operator.

When a pending mount verification is canceled by timing out, OPCOM prints a message in the following format:

```
%%%%%%%%%% OPCOM, dd-mmm-yyyy hh:mm:ss.cc %%%%%%%%%%%  
Mount verification aborted for device 'device-name'.
```

Performing Disk and Magnetic Tape Operations

3.5 Mount Verification

After a mount verification times out, all pending and future I/O requests to the volume will fail. Thus, the disk must be dismounted and remounted before it can be accessed again.

Note

Mount verification caused by a write-lock error will not time out.

3.5.3.3 Cancellation Commands

To cancel mount verification, you run a special interrupt program called Interrupt Priority C (IPC). The IPC issues a software interrupt to gain the attention of the VMS operating system. You use the IPC to cancel mount verification, enter the debugger, or recalculate the quorum on a cluster. (The debugger in this case refers to the system level debugger, XDELTA.)

To invoke the IPC, enter the following sequence of commands from the console terminal of your VAX computer:

```
Ctrl/P
>>> HALT
>>> D/I 14 C
>>> CONT
IPC>
```

To exit from the IPC and resume normal operations, press Ctrl/Z.

The IPC program converts lowercase characters to uppercase, issues the terminal bell character whenever it receives illegal characters (such as most control characters), compresses multiple spaces, and ignores leading spaces.

You can enter one of the following commands in response to the IPC> prompt:

- Use the **C** *device-name* command to cancel any pending mount verification on the device specified. (A warning is given if no mount verification was in progress for that device.) For example:

```
IPC> C MTA1:
```

- Use the **X** command to transfer control to the debugging tool XDELTA (provided it was loaded with the system by setting the appropriate value in the boot file). If XDELTA has not been loaded, the prompt IPC> is reissued. For example:

```
IPC> X
```

- Use the **Q** command to recalculate the quorum on a cluster.

When a pending mount verification is canceled, OPCOM prints a message in the following format:

```
%%%%%%%%%% OPCOM, dd-mmm-yyyy hh:mm:ss.cc %%%%%%%%%%%
Mount verification aborted for device 'device-name.'
```

After you successfully cancel a pending mount verification with this technique, you must dismount and then remount the volume before you can access it again, as the following example demonstrates:

```
%%%%%%%%%% OPCOM, 28-DEC-1990 10:54:54.12 %%%%%%%%%%%
Device DBA0: is offline.
Mount verification in progress.
```

Performing Disk and Magnetic Tape Operations

3.5 Mount Verification

```
Ctrl/P
>>> HALT
>>> D/I 14 C
>>> CONT
IPC> C DBA0
IPC> ^Z
%SYSTEM-I-MOUNTVER, DBA0: has aborted mount verification.
%%%%%%%%%% OPCOM, 28-DEC-1990 10:56:26.13 %%%%%%%%%%%
Mount verification aborted for device DBA0:
```

In this example, device DBA0 is off line, but you are unable to spin the disk back up. There is no other available drive on the controller, so it is not possible to switch the unit select plugs of the two drives. You do not enter a DISMOUNT command for the disk, because it was mounted as a private volume and you do not have access to it. The %SYSTEM-I-MOUNTVER message also appears here because this is the console terminal.

Managing Disk Space

The disk space that is available for files is a finite amount. As system manager, you share a responsibility with your users for making the best use of the available disk space. The following techniques are some of the methods that can be used to conserve and monitor disk space usage:

- Purging old versions of files
- Establishing disk quotas
- Setting file expiration dates
- Analyzing and repairing error conditions

The following sections discuss these tasks in detail.

4.1 Purging Old Versions of Files

One of the best ways of conserving disk space is to purge old versions of files periodically. Before purging files, make sure that the most recent versions are the ones you want to preserve.

You should periodically purge certain system log files that are automatically generated, such as the operator log and the accounting log files. Also, purge log files created by the PRINT and SUBMIT commands.

You can encourage individual users to purge files that are in their own areas and directories. If desired or necessary, the system manager can purge files from some or all directories. The following example purges all files in the directory [JONES] and all the subdirectories below [JONES] on disk \$DISK1: and logs the files that are deleted (displaying their names on the terminal as they are deleted):

```
$ PURGE/LOG $DISK1:[JONES...]
```

You can use wildcard characters to perform global purges, and you can use the /KEEP qualifier to retain only a specified number of versions of each file, as shown in the following example:

```
$ PURGE/KEEP=3 $DISK1:[*...]
```

You can restrict the number of file versions that can be generated in a user's directory by using the /VERSION_LIMIT qualifier with the SET DIRECTORY or CREATE DIRECTORY command. When you create a default directory for a user, use the /VERSION_LIMIT qualifier with the CREATE/DIRECTORY command. In the following example, files in account [JONES] cannot exceed three versions:

```
$ CREATE/DIRECTORY $DISK1:[JONES]-  
_ $ /OWNER_UIC=[200,1]/VERSION_LIMIT=3
```

If a user in this directory attempts to exceed the three-version limit, the system purges the file, leaving only the three most recent versions.

4.2 Establishing Disk Quotas

You limit the amount of space available to individual users on public volumes (or volume sets) by creating and maintaining quota files on those volumes. Individual users can similarly restrict usage on private volumes. Quotas are maintained and enforced on a per-volume basis. Each volume or volume set has its own quota file; a volume on which quotas are not maintained has no quota file; on a volume set, volume 1 contains the quota file. Each entry in a quota file includes the following information:

- **General Identifier or UIC**—Identification code of a user entitled to maintain files on the volume
- **Usage**—Number of blocks on the volume taken up by the user's files
- **Quota**—Maximum number of blocks on the volume that the user's files can take up before an error message is issued
- **Overdraft**—Number of blocks over the quota that the user's files can take up

The absolute maximum number of blocks permitted a user on a volume is the sum of the quota and the overdraft.

You (or the user maintaining the volume) supply identifiers and assign quotas and overdrafts with the System Management Utility (SYSMAN). (For more information, see the *VMS SYSMAN Utility Manual*). Usage counts are maintained automatically by the system during normal file activities.

The name of the quota file is [000000]QUOTA.SYS on the applicable volume. A quota file requires 1 block of disk storage for each 16 entries.

A quota file is initialized with an entry for UIC [0,0]. The usage count for this UIC should not change from 0—the UIC should own no files. Its quota and overdraft, however, serve as defaults in certain situations, and so should be set to values most likely to be assigned as quotas and overdrafts to other UICs.

4.2.1 Disk Quota Operations

During normal use of a volume with a quota file, the system automatically updates the usage counts as users create, delete, extend, and truncate files. Users without entries in the quota file are not allowed to create files or allocate space on the volume unless they have the EXQUOTA privilege.

4.2.1.1 Exceeding the Quota

To create new files, a user's disk space usage must be below quota (not overdraft). If an operation to add a new file or expand a current file will exceed a user's quota, the system prohibits the operation and issues an error message.

A user with an overdraft might be able to extend an open file after exceeding the disk quota (for example, during an editing session). Operations to extend an open file will succeed until usage exceeds the sum of the quota and the overdraft. At this point, the system prohibits further extensions to the file.

Quota restrictions are not enforced for users with the EXQUOTA privilege; however, their usage counts are maintained.

4.2.1.2 Suspending Quotas

The `SYSMAN` subcommand `DISKQUOTA DISABLE` suspends quota operations on a volume; the `DISKQUOTA ENABLE` subcommand lifts the suspension. In addition, quota operations on a volume can be suspended at mount time by specifying the `/NOQUOTA` qualifier to the `DCL` command `MOUNT`. Note that when you suspend and then resume quota operations on a volume, you will probably find incorrect usage values in the quota file. To correct the usage values, use the `SYSMAN` subcommand `DISKQUOTA REBUILD`. (See the *VMS SYSMAN Utility Manual* for more information.)

To discontinue quota operations on a volume, execute the `DISKQUOTA DISABLE` subcommand, exit from `SYSMAN`, and delete the `QUOTA.SYS` file. Note that quotas are not suspended across disk mounts if the `QUOTA.SYS` file is still present.

4.2.1.3 Ensuring Quota File Accuracy with `REBUILD` on Mount

When a volume is mounted that was not properly dismounted the last time it was used, the system performs an automatic `REBUILD` operation. If quotas are enforced on the volume, this action ensures that the quota file accurately reflects usage of the disk, under any of the following conditions:

- The system failed
- The volume was physically removed before being dismounted
- The `WRITE PROTECT` button was pushed

4.3 Setting File Expiration Dates

File expiration is a file system feature (available on Files-11 Structure Level 2 disks only) that uses the expiration date of each file to track the file's use. The expiration dates aid the disposal of seldom used files.

To enable the setting of expiration dates, use the `DCL` command `SET VOLUME`:

```
$ SET VOLUME device-name: /RETENTION=(min,max)
```

For *min* and *max*, specify the minimum and maximum retention periods for files on the volume, expressed as delta time values. For example, the following command sets the minimum retention period to 15 days and the maximum to 20 days:

```
$ SET VOLUME DRA0: /RETENTION=(15-0:0,20-0:0)
```

The retention periods operate as follows:

- When a user creates a file, the file's expiration date is the current date plus the maximum retention period.
- Every time a user accesses a file (for either a read or write operation), the file's expiration date is either the current expiration date or the current time added to the minimum retention period, whichever is greater.

Thus, the expiration date of a frequently accessed file fluctuates between the minimum and maximum period plus the current date. When you set a suitable interval between minimum and maximum retention periods, you can trade between accuracy and efficiency in maintaining expiration dates. The difference between the two periods is the interval at which the expiration date of a frequently accessed file will be updated.

Managing Disk Space

4.3 Setting File Expiration Dates

If you specify only a single value in the SET VOLUME/RETENTION command, it becomes the minimum retention period; the maximum retention period is set to twice the minimum or the minimum plus 7 days, whichever is less.

You can simulate the maintenance of “access dates,” available in some other operating systems, by setting the retention periods to very small values (for example, 1 hour). Note, however, that doing so substantially increases overhead in the file system.

This feature does not automatically remove unused files; it maintains expiration dates to permit you to develop your own policy for handling files with little or no activity. For example, the following BACKUP command will copy to tape and then delete all expired files:

```
$ BACKUP/DELETE PUBLIC:[*...]/BEFORE=TODAY/EXPIRED -  
_ $ MTA0:ARCH20JUN
```

Users may not always be aware of file expiration dates, so retain the tape for a substantial period of time.

Note

If you start maintaining expiration dates on a previously existing volume, you should be aware that the expiration dates on existing files are 0 (until the files are accessed). Files with expiration dates of 0 are considered expired.

4.4 Handling Error Conditions

This section describes two utilities that analyze errors conditions relating to disk media: ANALYZE/DISK_STRUCTURE and ANALYZE/MEDIA. You use ANALYZE/DISK_STRUCTURE to analyze disk structure errors, and you use ANALYZE/MEDIA to analyze media for bad blocks. See the *VMS Analyze/Disk Structure Utility Manual* and the *VMS Bad Block Locator Utility Manual* for detailed information.

4.4.1 Analyzing Disk Structure Errors

The DCL command ANALYZE/DISK_STRUCTURE invokes the Analyze/Disk Structure Utility. This utility verifies the Files-11 structure on disk volumes, reports errors, and (when you specify the /REPAIR qualifier) repairs errors. By default, ANALYZE/DISK_STRUCTURE reports errors but does not make repairs. The following command reports all disk structure errors on device DBA1:

```
$ ANALYZE/DISK_STRUCTURE DBA1:
```

You should periodically use the Analyze/Disk Structure Utility to check for disk structure errors such as lost files on the disk. Lost files are files with a backlink to a directory file that has been deleted. (See the *VMS Analyze/Disk Structure Utility Manual* for more information.) The command in this example analyzes and repairs all errors and lost files on device DDA0:

```
$ ANALYZE/DISK_STRUCTURE/REPAIR/CONFIRM DDA0:
```

It is a good idea to execute ANALYZE/DISK_STRUCTURE in two passes: one pass to report all errors, and a second pass with the /REPAIR and /CONFIRM qualifiers to repair selected errors.

Another opportunity to check for lost files on your system is during a BACKUP operation. The following BACKUP command returns a formatted listing of the files in the BACKUP save set as the operation proceeds:

```
$ BACKUP/IMAGE/LIST device_A: device_B:saveset.sav/SAVESET
```

Lost files appear at the end of the save-set listing as files with empty brackets, indicating that they are not found within any directory. For example:

```
[ ]ALPHA.DAT;4  
[ ]OMEGA.DAT;2  
[ ]SUB_A.DIR;1  
End of saveset
```

4.4.2 Analyzing Media Errors

The DCL command ANALYZE/MEDIA invokes the Bad Block Locator Utility, which analyzes block-addressable devices and records the location of blocks that cannot reliably store data.

To ensure that the device is not accessed by any other user, you must first allocate the device with the DCL command ALLOCATE. After allocating the device, enter the DCL command MOUNT/FOREIGN. This ensures that the system does not recognize the device as a Files-11 volume, therefore allowing the Bad Block Locator Utility to execute.

To test the blocks on a volume, ANALYZE/MEDIA does the following:

- Writes a test pattern to each block on the medium
- Reads the contents of the block into a buffer
- Compares the data read back with the data written

If the data does not compare exactly, a block cannot reliably store data.

When the Bad Block Locator Utility locates a bad block, it records the address of the block. Consecutive bad blocks are recorded as single entries for non-last-track devices. After it finishes testing the disk, BAD writes the addresses of the bad blocks into an area called the Detected Bad Block File (DBBF).

After the Bad Block Locator Utility locates and records the bad blocks, enter the DCL command INITIALIZE to change the volume from unstructured format to Files-11 format and to allocate the faulty blocks to a special file on the volume called [000000]BADBLK.SYS. In this way, users are protected from accessing bad blocks for their files.

Caution

There is no way to test the volume for bad blocks without destroying its contents. However, you can update the Detected Bad Block File without erasing the volume's contents by using the ANALYZE/MEDIA qualifiers /NOEXERCISE and /BAD_BLOCKS.

Managing Disk Space

4.4 Handling Error Conditions

The following example shows the command sequence for executing ANALYZE /MEDIA interactively from your terminal:

```
$ ALLOCATE DBA2:
DBA2: ALLOCATED
$ MOUNT/FOREIGN DBA2:
%MOUNT-I-MOUNTED  mounted on DBA2:
$ ANALYZE/MEDIA/EXERCISE DBA2:
$ INITIALIZE DBA2:
```

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 ANALYZE/MEDIA command invokes the Bad Block Locator Utility. Specifying DBA2 and the /EXERCISE qualifier causes the Bad Block Locator Utility to analyze each block on this disk volume and to record the bad blocks. The INITIALIZE command reformats the volume as a Files-11 volume and allocates the bad blocks to a [000000]BADBLK.SYS. Once allocated, the bad blocks cannot be used by other files.

You do not need to execute ANALYZE/MEDIA on mass storage control protocol (MSCP) devices. All MSCP devices appear to the operating system as a contiguous stream of logical blocks. If part of the disk goes bad, the drive revector the bad block automatically. See the *VMS Bad Block Locator Utility Manual* for complete details on using the Bad Block Locator Utility.

Batch and Print Operations

This chapter is made up of the following sections:

1. Overview: Batch and Print Queuing System Concepts. See Section 5.1.
2. Setting Up Batch and Print Queues. See Section 5.2.
3. Maintaining Batch and Print Queues. See Section 5.3.
4. Troubleshooting Batch and Print Queues. See Section 5.3.4.
5. Managing a Card Reader. See Section 5.4.

5.1 Overview: Batch and Print Queuing System Concepts

Figure 5–1 illustrates the basic components of the VMS batch and print queuing system.

5.1.1 The Queuing Process

When a user submits a batch or print job, either by entering the DCL command `SUBMIT` or `PRINT` or through an application, the job is sent to a queue for processing. Information about the user's queue request, including the type of job, the file name or names, the name of the queue, and any special options, is sent to the queue manager. The queue manager stores and retrieves appropriate information from the queue database to print or execute the job. The job is then placed in the appropriate queue to await its turn for processing. Only one print job can be printed on a printer at a single time; however, more than one batch job can execute simultaneously in a batch queue.

Print jobs are sent to an independent process, called a **symbiont**, for formatting and finally sent to the printer for printing. For batch jobs, the job controller creates a process to perform the tasks in the batch job.

5.1.2 The Queue Manager

The queue manager process handles all queuing requests on a standalone node or in a cluster.

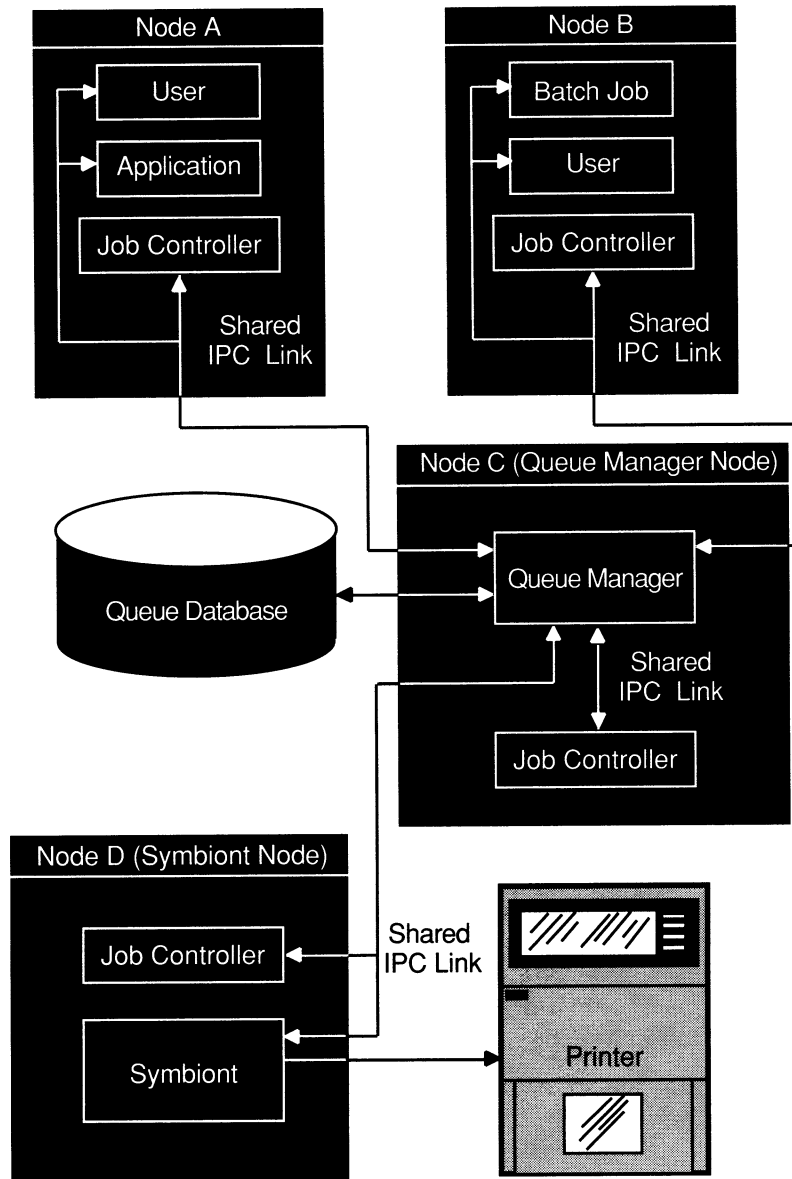
A single queue manager process accesses the queue database for all processes on a node or in a cluster. User processes, symbionts, and job controllers on each node communicate directly with the centralized queue manager through a shared interprocess communication (IPC) link. IPC is an internal VMS communication mechanism. Jobs can be submitted from one node and executed on a queue running on another node in the cluster.

Before you can enter any queue command, you must start the queue manager. For information, see Section 5.2.9.2.

Batch and Print Operations

5.1 Overview: Batch and Print Queuing System Concepts

Figure 5-1 The VMS Batch and Print Queuing System



ZK-3521A -GE

By default, in a VAXcluster environment, the queue manager will attempt to failover to another node if the node on which the queue manager is running leaves the cluster. You can specify the order in which nodes in the cluster should claim the queue manager process or limit the nodes that can run the queue manager. For more information, see Section 5.2.9.2.

5.1.3 The Queue Database

The queue database stores information used to keep the queuing system operating, including information about jobs, queues, and the queue manager. The queue database is made up of the following files:

QMAN\$MASTER.DAT, the master file
SYS\$QUEUE_MANAGER.QMAN\$QUEUES, the queue file
SYS\$QUEUE_MANAGER.QMAN\$JOURNAL, the journal file

SYS\$COMMON:[SYSEXE] is the default location for all three queue database files. However, you can move the files to another location. For more information, see Section 5.3.1.5.

5.1.3.1 The Master File

The master file contains the following information:

- The location of the queue and journal files
- Definitions of forms and characteristics
- A list of queue names
- A list of nodes allowed to run the queue manager

5.1.3.2 The Queue File

The queue file contains the queue definitions you form when you create, start, or modify queues.

5.1.3.3 The Journal File

The journal file contains information allowing the queue manager to return to the last known state should a standalone machine be stopped unexpectedly or should a VAXcluster member running the queue manager leave its cluster. The journal file also contains job definitions.

5.1.4 The Job Controller

The job controller does the following queue management tasks:

- Creates and monitors batch, symbiont, and queue manager processes
- Restarts the queue manager process on reboot
- Handles failover of the queue manager in a cluster

5.1.5 Types of Queues

The VMS batch and print queuing system supports two general classes of queues: execution and generic. Execution queues accept batch or print jobs for processing. Generic queues hold jobs until they are transferred to an execution queue for processing.

Execution and generic queues can be classified as batch or output queues.

For information about creating and starting queues, see Section 5.2.9.4. For information about maintaining queues and jobs in queues, see Section 5.3.

The following sections describe the specific types of execution and generic queues.

Batch and Print Operations

5.1 Overview: Batch and Print Queuing System Concepts

5.1.5.1 Execution Queues

An execution queue accepts jobs for processing. The two types of execution queue, batch queue and output queue, are described in the following sections.

5.1.5.1.1 Batch Execution Queue A batch execution queue can accept only batch jobs. A batch job executes as a detached process that sequentially runs one or more command procedures. The user defines the list of command procedures when submitting the job.

Section 5.2.1 contains information you need to plan and setup batch queues.

5.1.5.1.2 Output Execution Queue An output execution queue accepts jobs for processing by a symbiont. The queue manager sends the symbiont a list of files, which the user defines when submitting the job. An output symbiont transfers data from a disk to an output device. As the symbiont processes each file, it produces output for the device it controls, such as a printer or a terminal.

The standard print symbiont provided by the VMS operating system is named PRTSMB and is designed to print files on hardcopy devices. The LAT print symbiont LATSMB is used to print files on output devices attached to LAT ports. User-modified or user-written symbionts can also be designed for this or any other file processing activity managed by the VMS batch and print queuing system. (See the *VMS Utility Routines Manual* for more information.) Output execution queues include the following types:

Type	Description
Printer queue	Uses a symbiont to direct output to a printer.
Terminal queue	Uses a symbiont to direct output to a terminal printer.
Server queue	Uses a user-modified or user-written symbiont to process the files that belong to jobs in the queue.

When you create an output execution queue you can initially designate it as either a printer, a terminal, or a server execution queue; you can also specify the symbiont to be associated with the queue. However, when the queue is started, the symbiont process associated with the queue can override the queue designation if the queue type specified does not match the type of device. The standard VMS symbiont determines whether it is controlling a printer or a terminal. It communicates this information to the queue manager and, if necessary, the queue manager changes the type designation of the output execution queue. By convention, a user-written or user-modified symbiont that does not deliver output to a printer defines its queue as a server queue.

Section 5.2.2 contains information you need to plan and set up output execution queues.

5.1.5.2 Generic Queues

A generic queue holds a job until an appropriate execution queue becomes available to initiate the job. The queue manager then requeues the job to the available execution queue. In a VAXcluster environment, a generic queue can direct jobs to execution queues located on different nodes in the VAXcluster. The two types of generic queues, generic batch queues and generic output queues, are described as follows:

Batch and Print Operations

5.1 Overview: Batch and Print Queuing System Concepts

Type	Description
Generic batch queue	A generic batch queue can direct jobs only to batch execution queues. Generic batch queues are typically used in clustered systems to distribute the batch workload across several nodes (see Section 5.2.1.3).
Generic output queue	A generic output queue can direct jobs to any of the three types of output execution queues: print, terminal, or server queues. Generic output queues are typically used to distribute the output workload among several like printers (see Section 5.2.2.4).

Generic queues are not automatically stopped when a node is shutdown. Therefore, they do not need to be started when a node reboots.

5.1.5.3 Logical Queues

A **logical queue** is a special type of generic output queue that transfers jobs to another output execution queue. You might use this kind of queue to temporarily redirect a queue when the device on which it runs is broken.

A logical queue transfers its jobs into the execution queue specified with the ASSIGN/QUEUE command. For information on setting up a logical queue, see Section 5.3.2.15.

5.1.6 The Autostart Feature

The autostart feature simplifies startup and ensures high availability of execution queues in a cluster. It lets you do the following:

- Start all autostart queues on a node with a single command
- Specify a list of nodes (within a VAXcluster environment) to which a queue can automatically attempt to fail over if necessary.

For these reasons, Digital recommends you use autostart queues whenever possible.

When you set up an autostart execution queue, you specify one or more nodes (for batch queues) or nodes and devices (for output queues) on which the queue can be started, and you activate the queue for autostart. When you enable autostart on a node, all stopped, active autostart queues capable of running on the node are automatically started by the queue manager. Any autostart queue that fails over to the node is also automatically started.

To use autostart queues, you must do the following:

1. Enable autostart on a node. For more information, see Section 5.3.2.5.
2. Create the queue as an autostart queue and initially activate the queue for autostart. For more information, see Section 5.2.9.4.1.

5.1.7 Job Scheduling

This section describes how the queue manager determines the order in which batch and print jobs are processed. (If a batch or print job cannot be processed, it is placed in a pending state and is not processed until the cause of the pending state is resolved. For more information, see Section 5.3.4.3.)

Batch and Print Operations

5.1 Overview: Batch and Print Queuing System Concepts

The queue manager uses the following criteria to determine the scheduling order for batch and print jobs that are eligible for processing:

1. Job scheduling priority

The queue manager checks the job's scheduling priority. The job with the highest scheduling priority value is processed first. The job scheduling priority is different than the base process priority or current process priority. The user can specify job scheduling priority with the /PRIORITY qualifier of the PRINT or SUBMIT command.

2. Size (optional and applies only to output jobs)

By default, the job size of an output job is checked. Among jobs of identical scheduling priority, the smallest sized job is processed first.

The job size is not checked if the queue has been created, started, or modified to use the /SCHEDULE=NOSIZE option.

3. Submission time

If the jobs' scheduling priorities are identical, the job that was submitted first is processed first.

5.1.8 Spooled Output Devices

Certain application programs print output by writing data directly to an output device rather than submitting it to a queue. A **spooled output device** causes such an application program to write output to an intermediate storage device (such as a disk) so that the printer targeted to print the output remains available to other system users while the program is running.

When you set up a printer as a spooled device, you specify a storage device and an output queue to be associated with that printer. When a process running an application directs its output to the spooled printer, the output is instead placed in a temporary file on the storage device. When the file is closed, the system submits the file for printing on the associated output queue. Both the spooling of the output file to an intermediate storage device and the subsequent queuing of a job consisting of this file occur without the direct intervention of the user.

If your system runs application programs that might write output directly to a printer, Digital recommends you set up your printers as spooled devices. Digital recommends you also spool your LAT printers to prevent privileged users from writing directly to a LAT printer. Writing directly to a LAT printer can cause problems for output queues that use the printer.

Section 5.2.2.6 contains a sample configuration using spooled output devices. Section 5.2.9.1.3 describes how to set up a spooled output device.

5.1.9 PostScript Printing

The VMS operating system does not include software to support PostScript printing. In order to print PostScript files, you must have either of the following:

- A printer capable of printing PostScript files and supporting software
- Software that provides PostScript to Sixel printing and a supported printer.

For more information, see your Digital sales representative.

5.1 Overview: Batch and Print Queuing System Concepts

5.1.10 Distributed Printing

The VMS batch and print queuing system enables users to print files on output devices attached to the local system or VAXcluster.

The VAX Distributed Queuing Service (DQS) is a VMS layered product that extends the printing capabilities of the VMS queuing system to a distributed environment. DQS enables users to print files on output devices attached to remote VMS nodes in your network.

For more information, see the DQS documentation or your Digital sales representative.

5.2 Setting Up the Batch and Print Queuing System

The following sections provide information about setting up your batch and print queuing system. For information on maintaining your queuing system, see Section 5.3.

To set up and start the batch and print queuing system, perform the following steps:

1. Plan your batch queuing environment—see Section 5.2.1.
2. Plan your output queuing environment—see Section 5.2.2.
3. Choose batch and output queue options—see Section 5.2.3.
4. Implement the batch and output queue environment you have planned—see Section 5.2.9.

5.2.1 Planning Your Batch Queuing Environment

To set up your batch queuing environment, you must understand your users' batch needs and determine how to set up your batch queues to meet those needs. To do so, you must design your batch configuration and choose certain options for your batch queues.

This section provides examples of some typical batch configurations. Use the information provided in these examples to design your own configuration. Your configuration will probably combine elements from several of these examples.

Figure 5-2 illustrates a simple batch configuration, suitable for a standalone system supporting mainly interactive processing.

If your system has special requirements, you might need a more complex batch queue configuration. The following table lists some special requirements, and the section that describes an appropriate configuration:

Requirement	Section
Users rely heavily on batch processing or have special batch needs.	Section 5.2.1.2.
Your system is a clustered environment.	Section 5.2.1.3.
Your batch queues must be highly available.	Section 5.2.1.4.

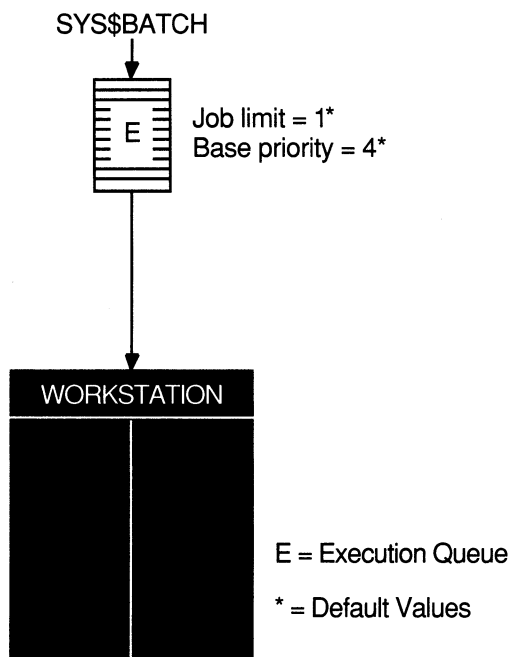
Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

5.2.1.1 Simple Batch Queue Configuration: For Limited Batch Needs

Figure 5–2 shows a single, default batch queue.

Figure 5–2 A Default Batch Queue



ZK-3820A-GE

By default, when a user submits a batch job with the `SUBMIT` command, the job is placed in the queue named `SYS$BATCH`. To set up a single default queue on a standalone system, name the queue `SYS$BATCH`.

5.2.1.2 Specialized Batch Queues: Meeting Special Needs

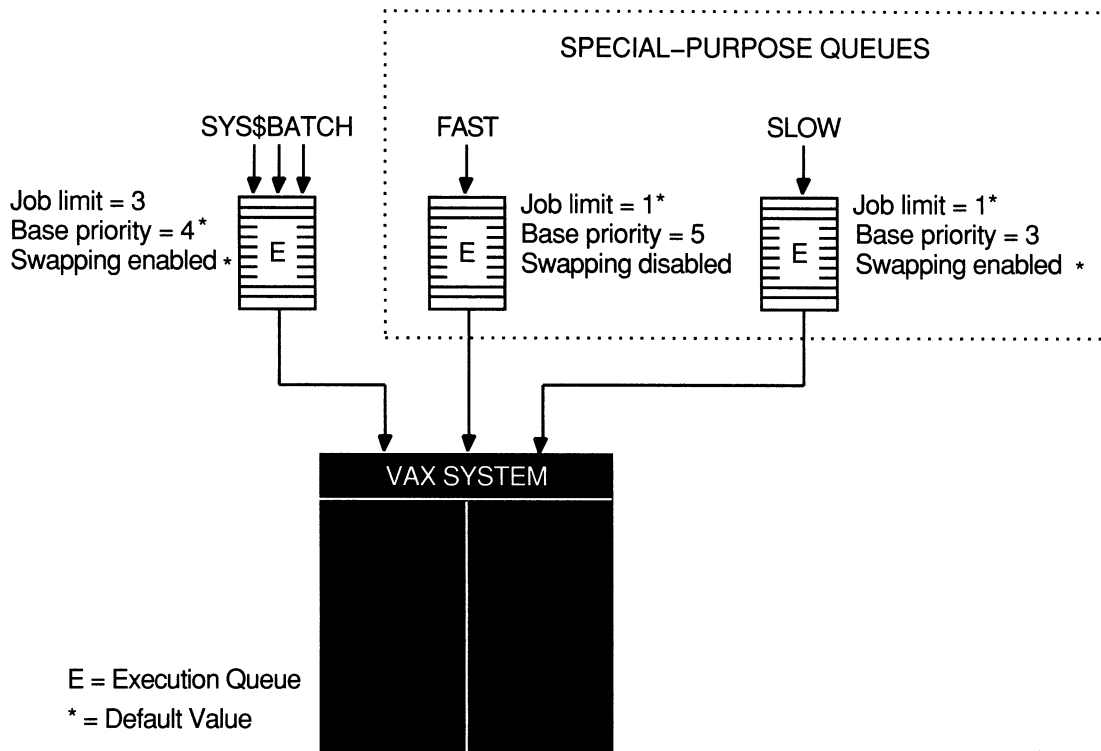
If your users rely heavily on batch processing or have special processing needs, you might want to set up more than one queue. You can customize batch queues to handle specialized jobs by specifying performance and resource options for jobs in the queue. For more information, see Section 5.2.7.

The configuration illustrated in Figure 5–3 has several queues, each customized to process certain types of batch jobs.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Figure 5-3 Multiple Batch Queues with Special Resource and Performance Options



ZK-3819A-GE

In this example, **SYS\$BATCH** is the default queue. Normal batch jobs of no special importance would typically be submitted to this queue. **FAST** is a queue for executing high priority jobs that should not be swapped out of memory. **SLOW** is a background queue for processing low priority jobs. Typically, these are large jobs with large requirements for physical memory. Be conservative when changing base priority and swapping on a queue. Even a slight change can have a significant negative effect on batch and interactive performance. For example, even an increase of 1 in a queue's base priority can affect performance significantly.

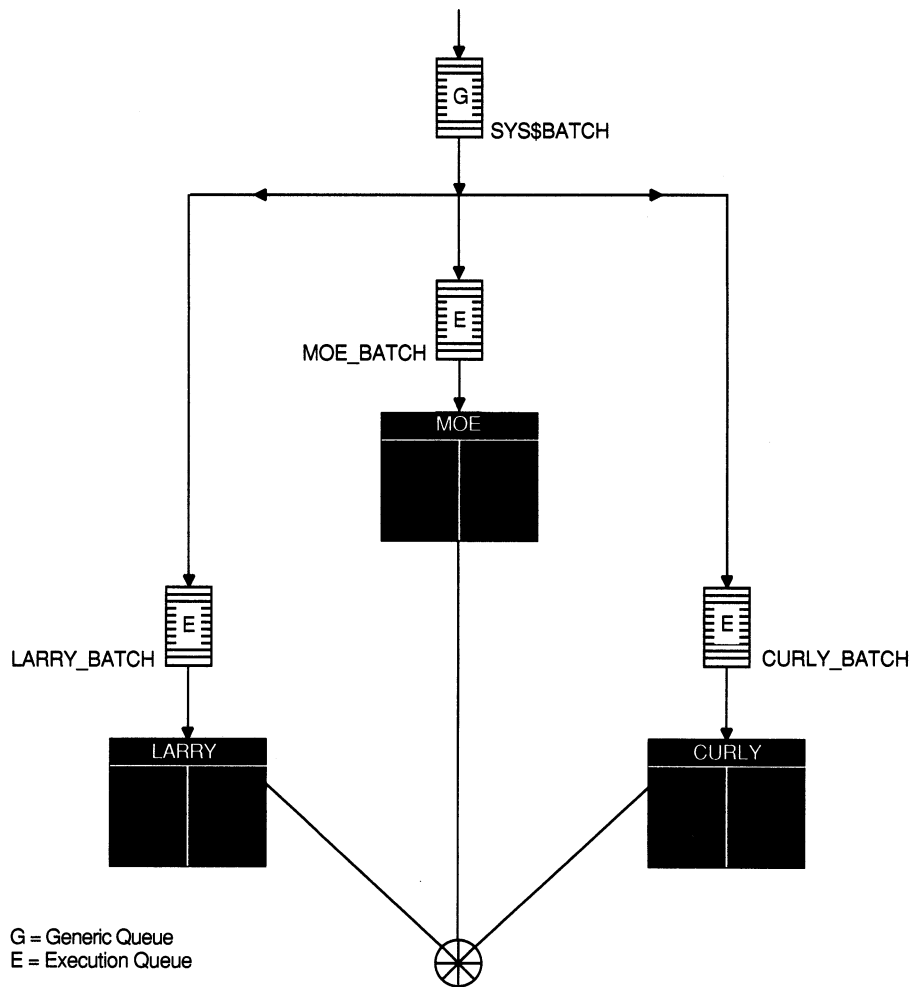
5.2.1.3 Generic Batch Queues in a VAXcluster: Distributing the Workload

Generic queues can be used in a VAXcluster to balance the use of processing resources by distributing batch processing across nodes in a cluster. (For an explanation of generic queues, see Section 5.1.5.2.) Figure 5-4 illustrates a typical configuration.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Figure 5-4 Batch Queue Configuration with Clusterwide Generic Queue



ZK-3815A-GE

In this example, a generic batch queue named SYS\$BATCH is set up to feed jobs to execution queues on each node in the cluster. When a user submits a job to the clusterwide generic queue SYS\$BATCH, the job is placed in the appropriate execution queue to minimize the ratio of executing jobs to job limits for all execution queues fed by SYS\$BATCH. For example, suppose execution queues MOE_BATCH, LARRY_BATCH and CURLY_BATCH all have a job limit of 5. If MOE_BATCH and LARRY_BATCH are executing four jobs and CURLY_BATCH is executing one job, the generic queue SYS\$BATCH will feed the next job to CURLY_BATCH.

See the *VMS VAXcluster Manual* for more information about VAXcluster queue configurations. For information about how to create a generic queue, see Section 5.2.9.4.2.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

5.2.1.4 Autostart Queues: For Ease of Maintenance and High Availability

You should use autostart queues whenever possible to simplify the startup of queues on your node. In a VAXcluster environment, you should also use autostart queues to keep your queues available when a node or nodes leave the cluster.

For more information on the autostart feature, see Section 5.1.6. For information about how to create autostart queues, see Section 5.2.9.4.1.

5.2.2 Planning Your Output Queuing Environment

This section describes the bolded step in the following list of steps for setting up a batch and print queuing system:

1. Plan your batch queuing environment. See Section 5.2.1.
2. **Plan your output queuing environment.**
3. Choose batch and output queue options. See Section 5.2.3.
4. Implement the batch and output queue environment you have planned. See Section 5.2.9.

Before you set up your output queuing environment, you must understand your users' printing needs and determine how to set up your output queues to meet those needs. To do so, you must design your output queue configuration and choose certain queue options.

This section provides examples of some typical output queue configurations. Use the information provided in these examples to design your own configuration. Your configuration will probably combine elements from several of these examples.

Figure 5-5 illustrates a simple queue configuration, suitable for a standalone system with a single printer.

If your system has special requirements you might need a more complex output queue configuration. The following table lists some special requirements, and the section that describes an appropriate configuration:

Requirement	Section
Printers of different kinds. For example, LN03 and line printers.	Section 5.2.2.2.
Remote printers attached to a LAT port.	Section 5.2.2.3.
More than one printer of the same kind.	Section 5.2.2.4.
A VAXcluster environment.	Section 5.2.2.5.
Applications that print output by writing directly to a printer, rather than submitting to an output queue.	Section 5.2.2.6.
Queues that must be highly available.	Section 5.2.2.7.

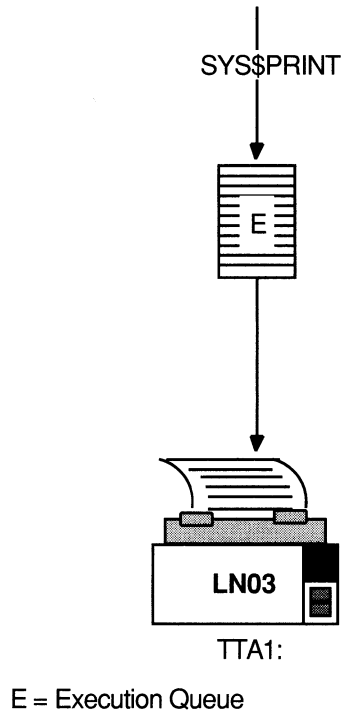
5.2.2.1 Simple Queue Configuration: For Limited Printing Needs

For a standalone system, such as a standalone workstation, you might need only a single printer. Figure 5-5 illustrates a typical queue configuration.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Figure 5-5 Single Laser Printer Queue



ZK-3816A-GE

By default, when a user submits a print job with the PRINT command, the job is placed in the queue named SYSS\$PRINT. To set up a single default printer queue on a standalone system, name the queue SYSS\$PRINT.

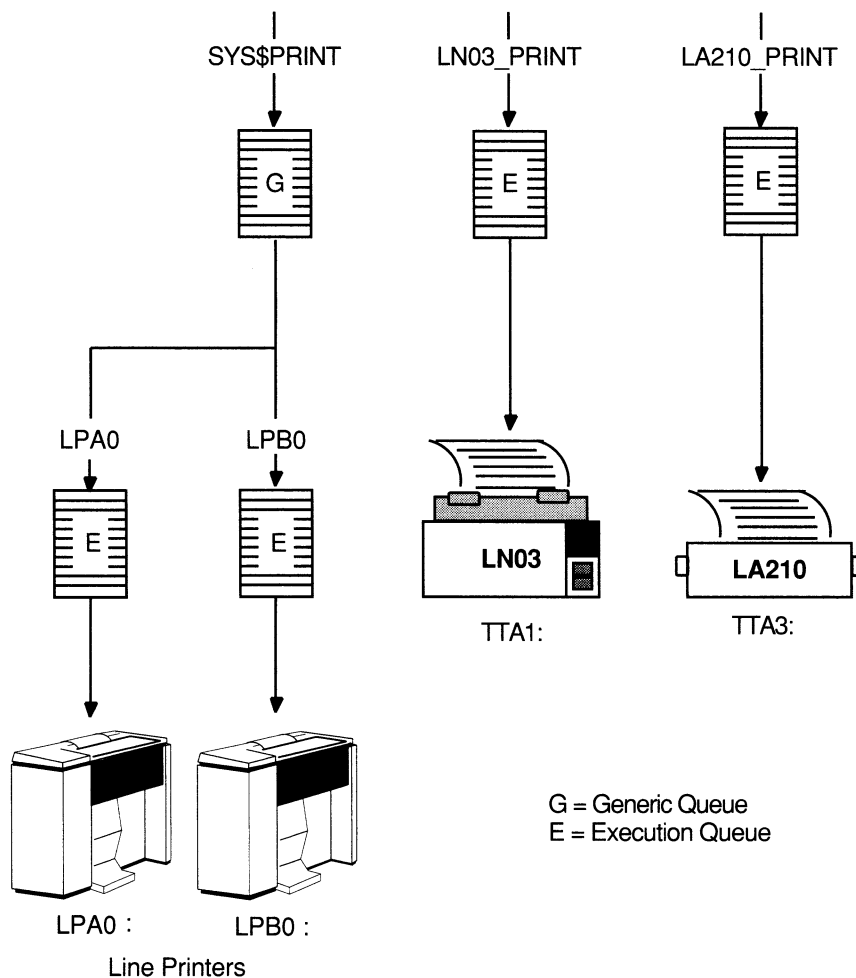
5.2.2.2 Mixed Printers: Specialized Output Execution Queues

If you have several printers of different kinds (for example, an LN03 printer, an LA210 printer, and an LP27 line printer) you will need to set up a separate queue for each printer. The options, such as the default form or device control library that you use with these queues will probably differ according to the printer to which the queue's output is sent. For example, the default form for a line printer might have a width of 132 columns, while the default form for an LN03 printer might have a width of 80 columns. Figure 5-6 illustrates such a configuration.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Figure 5-6 Queue Configuration with Mixed Printers



In addition, if you have Common Print Symbiont (CPS) printers, such as the PrintServer 20 or PrintServer 40 printers, that can print files of different data formats, you might need to set up a separate queue for each data format. For example, you might set up an `ANSI_PS40` queue to print ANSI format files and a `POSTSCRIPT_PS40` queue to print PostScript files. For more information about CPS printers, see the documentation provided with the printer.

5.2.2.3 LAT Printers: Sharing Printers Among Multiple Systems

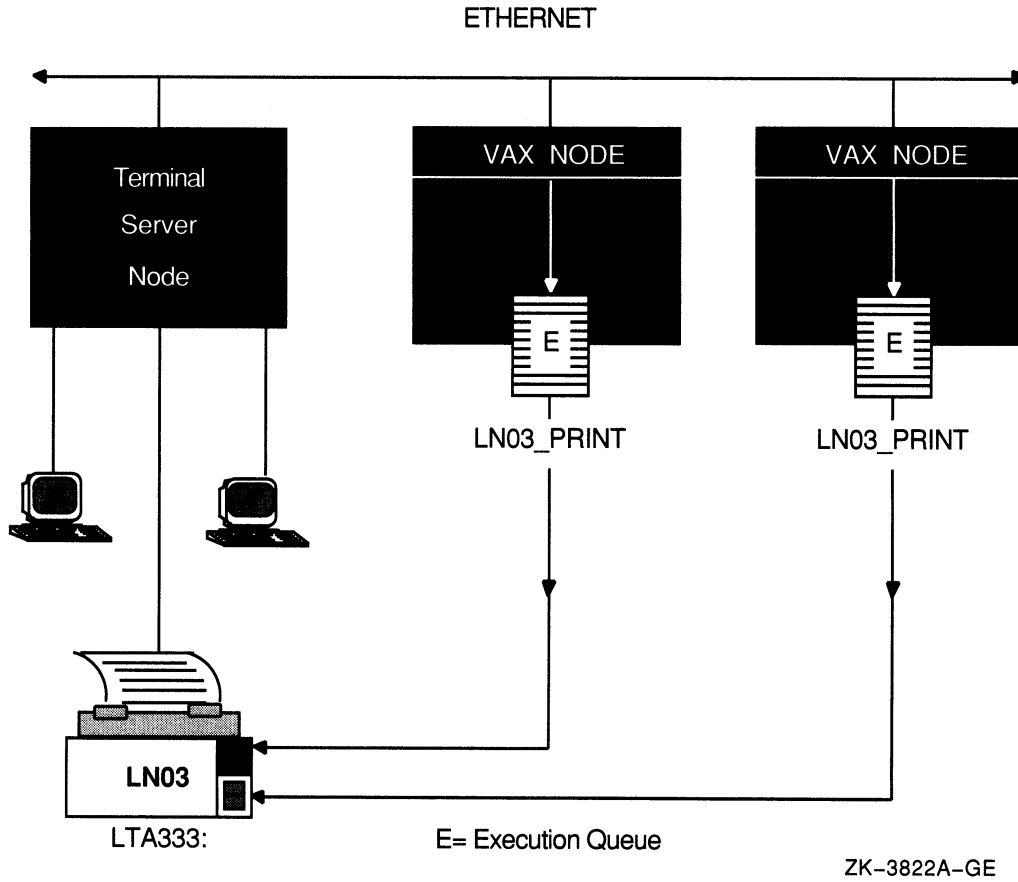
To share printers among multiple systems or VAXclusters, you can connect printers to a LAT port on a terminal server. Figure 5-7 illustrates an output queue configuration with a remote printer on a terminal server.

Because LAT printers are usually shared among users of multiple systems or clusters, more users will be affected if a LAT queue is unavailable. For this reason, you might want to set up your LAT queues as autostart queues (as explained in Section 5.2.2.7) to ensure that these queues are highly available.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Figure 5-7 Configuration for Remote Printers on a Terminal Server



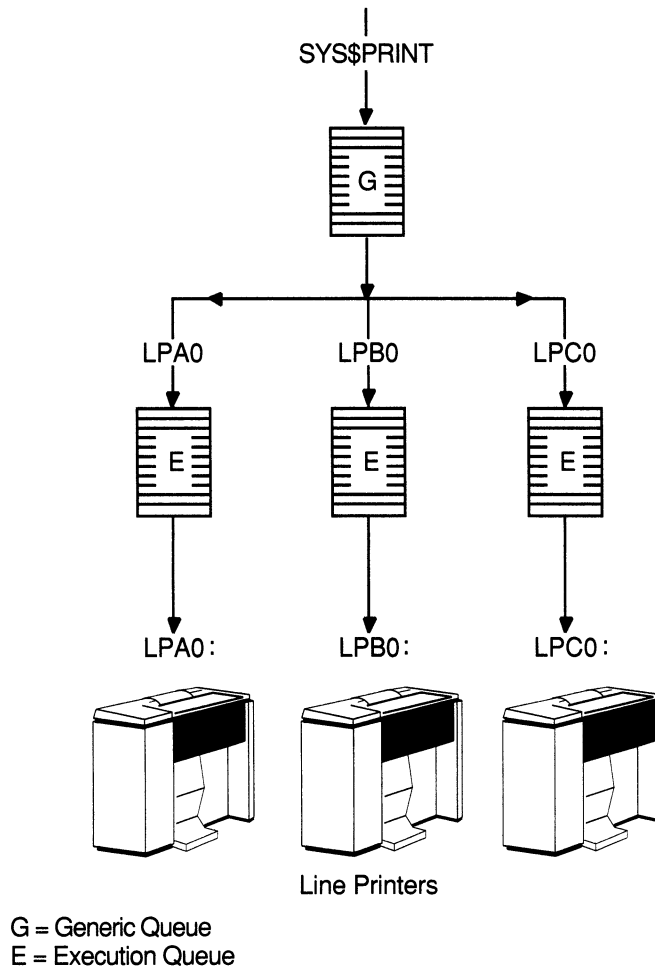
5.2.2.4 Generic Output Queues: Distributing the Workload

If you have more than one printer of the same kind (for example, if you have three line printers), you should use generic queues to balance the print load among the printers. Figure 5-8 illustrates such a configuration.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Figure 5-8 Queue Configuration with Three Like Printers and a Generic Queue



ZK-3818A-GE

For information about how to create a generic queue, see Section 5.2.9.4.2.

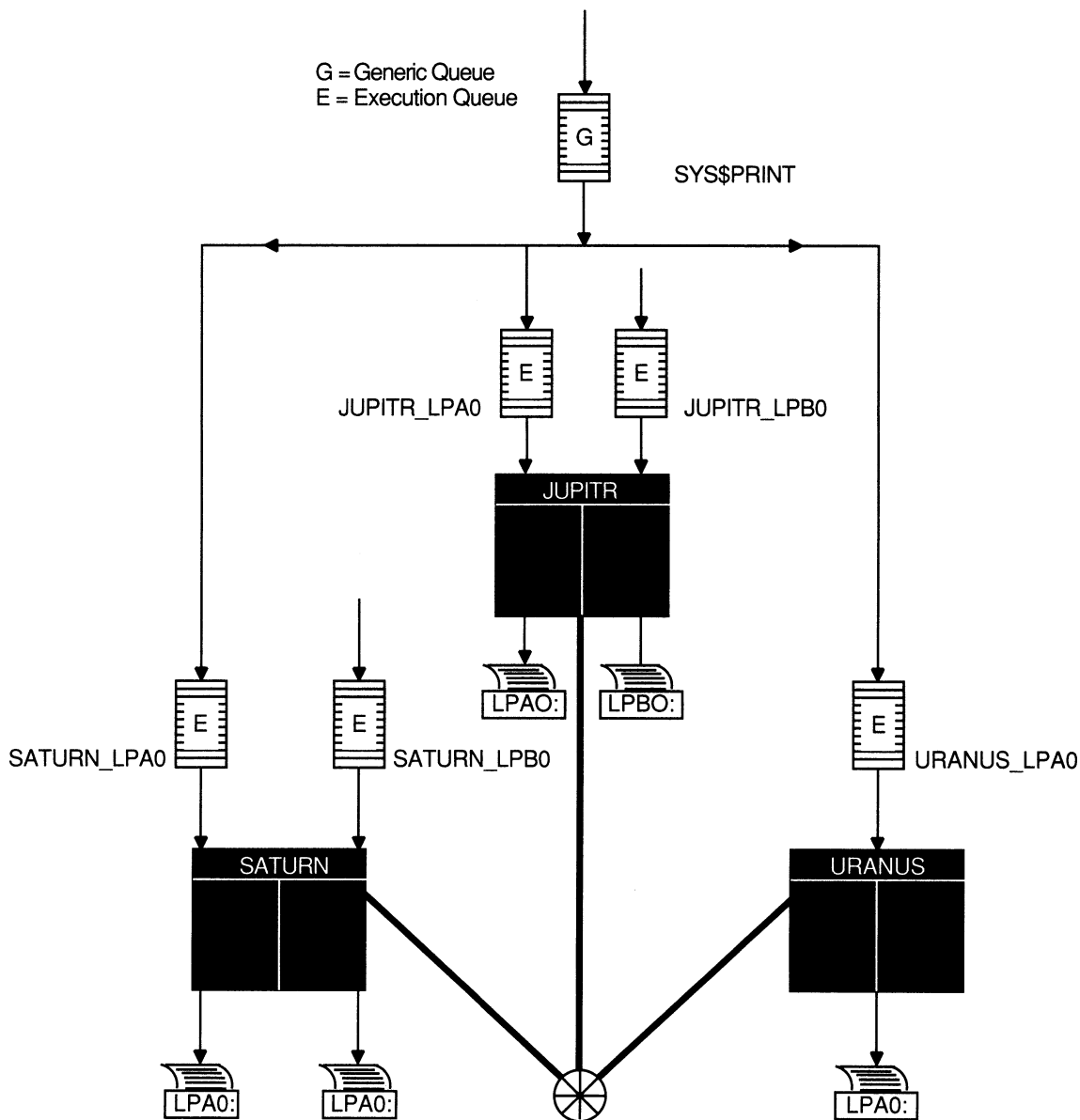
5.2.2.5 VAXcluster Queues: Sharing Devices in a Cluster

Figure 5-9 illustrates a typical VAXcluster output queue configuration. For information about cluster queue configurations, see the *VMS VAXcluster Manual*.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Figure 5-9 Output Queue Configuration in a VAXcluster



ZK-1634-GE

5.2.2.6 Spooled Output Devices: Printing from Applications

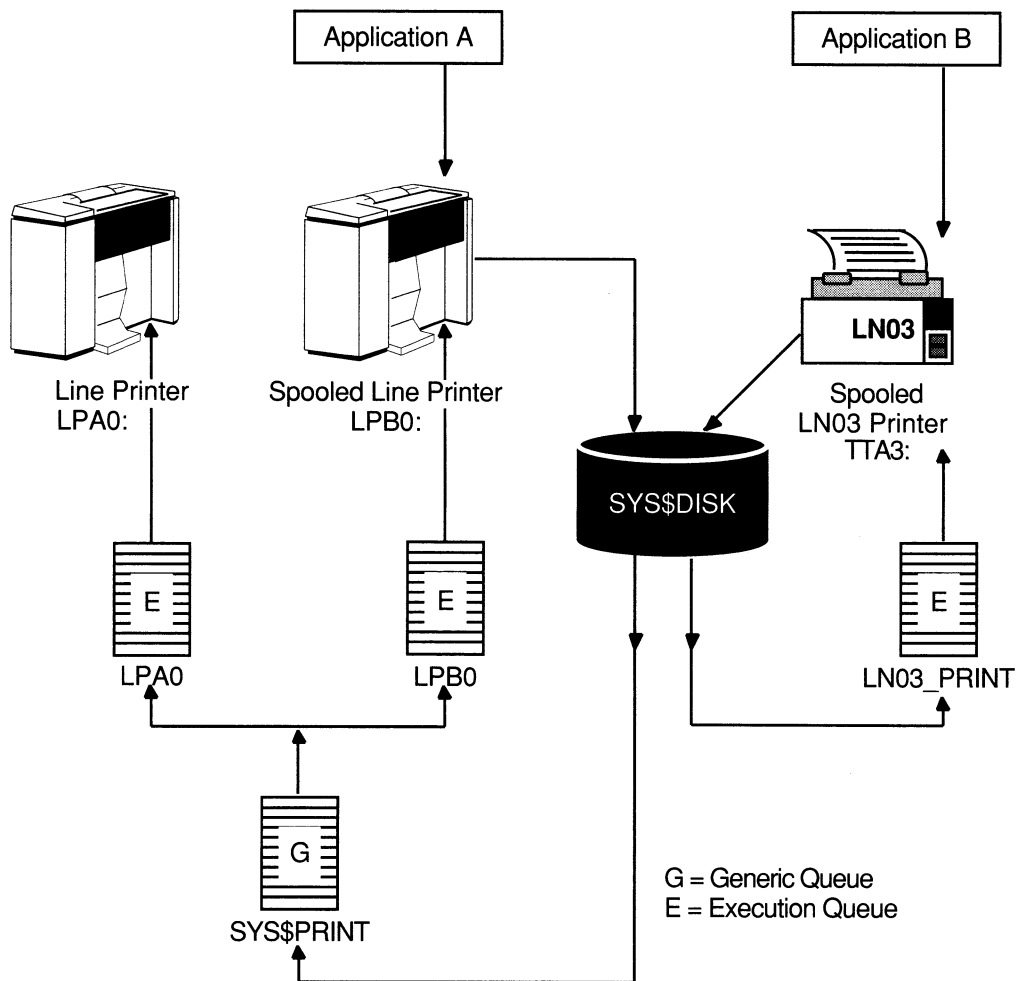
If your system runs application programs that write output directly to a printer rather than submitting it to an output queue or if you will be using LAT queues, you should set up your printers as spooled devices. Spooling your output devices causes application programs to write output to an intermediate storage device so that the printer remains available to other users while the program is running.

Figure 5-10 illustrates a spooled output device configuration.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Figure 5-10 Queue Configuration with a Spooled Device



ZK-3817A-GE

For an explanation of spooled output devices, see Section 5.1.8. For information about how to set up a spooled output device, see Section 5.2.9.1.3.

5.2.2.7 Autostart Queues: For Ease of Maintenance and High Availability

You should use autostart queues whenever possible to simplify the startup of queues on your node. In a VAXcluster environment, you should also use autostart queues to keep your queues available when a node or nodes leave the cluster.

For more information about the autostart feature, see Section 5.1.6. For information about how to create autostart queues, see Section 5.2.9.4.1.

5.2.3 Choosing Batch and Output Queue Options

This section describes the bolded step in the following list of steps for setting up a batch and print queuing system:

1. Plan your batch queuing environment. See Section 5.2.1.
2. Plan your output queuing environment. See Section 5.2.2.
3. **Choose batch and output queue options.**

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

4. Implement the batch and output queue environment you have planned. See Section 5.2.9.

To specify options for a queue, you specify qualifiers to the INITIALIZE/QUEUE command when you create the queue. (You can also specify options after you create a queue by including qualifiers with the START/QUEUE or SET QUEUE command.) Table 5–1 summarizes the options you can use with batch and output queues, and lists the type of queue for which you can specify the option. See DCL Help or the *VMS DCL Dictionary* for more details about these qualifiers.

Table 5–1 Qualifiers for Specifying Queue Options

Qualifier	Type of Queue	Description
/CPUDEFAULT	Batch	Defines the base value for the default CPU time limit for batch jobs executed in the queue. For more information, see Section 5.2.7 and the <i>VMS DCL Dictionary</i> .
/CPUMAXIMUM	Batch	Defines a maximum CPU time limit for batch jobs executed in the queue. For more information, see Section 5.2.7 and the <i>VMS DCL Dictionary</i> .
/DISABLE_SWAPPING	Batch	Specifies whether batch jobs executed from a queue can be swapped in and out of memory. For more information, see Section 5.2.7.
/JOB_LIMIT	Batch	Indicates the number of batch jobs that can be executed concurrently from a batch queue. For more information, see Section 5.2.7.
/BLOCK_LIMIT	Output	Limits the size of print jobs that can be processed on an output execution queue. For more information, see Section 5.2.8.1.
/DEFAULT	Output	Establishes defaults for certain options of the PRINT command. After you set an option for the queue with the /DEFAULT qualifier, users do not have to specify that option in their PRINT commands. However, they can specify options to override the defaults set on the queues.

Possible default options are as follows:

BURST—Controls whether file flag pages with a burst bar between them are printed preceding output. For more information, see Section 5.2.8.2.

FEED—Specifies whether a form feed is inserted automatically at the end of a page. For more information, see Section 5.2.8.4.

FLAG—Controls whether a file flag page is printed preceding output. For more information, see Section 5.2.8.2.

(continued on next page)

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Table 5–1 (Cont.) Qualifiers for Specifying Queue Options

Qualifier	Type of Queue	Description
		FORM—Specifies the default form for an output execution queue. If a job is submitted without an explicit form definition, this form is used to process the job. For more information, see Section 5.2.8.3.
		TRAILER—Controls whether a file trailer page is printed following output. For more information, see Section 5.2.8.2.
/DEVICE	Output	Specifies the type of output execution queue. The keywords are as follows: PRINTER (default)—Specifies a printer queue. This is the default. TERMINAL—Specifies a terminal queue. SERVER—Specifies a server queue. For more information, see the Section 5.1.5.1.2 and the <i>VMS DCL Dictionary</i> .
/FORM_MOUNTED	Output	Specifies the mounted form for an output execution queue. For more information, see Section 5.2.8.3.
/LIBRARY	Output	Specifies the file name for a device control library. For more information, see Section 5.2.8.5.
/PROCESSOR	Output	Specifies the symbiont to be used with an output execution queue. The default is the standard VMS print symbiont PRTSMB. For more information, see Section 5.1.5.1.2 and the <i>VMS DCL Dictionary</i> .
/RECORD_BLOCKING	Output	Determines whether the symbiont can concatenate (or block together) output records for transmission to the output device. For more information, see the <i>VMS DCL Dictionary</i> .

(continued on next page)

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Table 5-1 (Cont.) Qualifiers for Specifying Queue Options

Qualifier	Type of Queue	Description
/SEPARATE	Output	<p>Specifies required job separation or job reset options for an output execution queue. Required options cannot be overridden by the PRINT command. Possible options are as follows:</p> <p>BURST—Controls whether two job flag pages with a burst bar between them are printed at the beginning of each job. For more information, see Section 5.2.8.2.</p> <p>FLAG—Controls whether a job flag page is printed at the beginning of each job. For more information, see Section 5.2.8.2.</p> <p>RESET—Specifies one or more device control library modules you want to use to reset the device after any file trailer and before any job trailer. For more information, see Section 5.2.8.5.</p> <p>TRAILER—Controls whether a job trailer page is printed at the end of each job. For more information, see Section 5.2.8.2.</p>
/SCHEDULE	Output	<p>Specifies whether pending jobs in a queue are scheduled based on the size of the job. For more information, see Section 5.2.8.1.</p>
/BASE_PRIORITY	Batch and output	<p>Specifies a base process priority (not the same as the job scheduling priority). For a batch queue, specifies the base priority for processes executing jobs in the queue. For output queues, specifies the base priority of the symbiont process. For more information, see Section 5.2.7.1 and the <i>VMS DCL Dictionary</i>.</p>
/WSDEFAULT	Batch and output	<p>For batch queues, specifies a default working set size for batch jobs executed in the queue. For output queues, specifies a default working set size for the symbiont process.</p> <p>The value set by this qualifier overrides the value defined in the UAF of any user submitting a job to the queue. For more information, see Section 5.2.7.</p>
/WSEXTENT	Batch and output	<p>For batch queues, specifies the working set extent for batch jobs executed in the queue. For output queues, specifies a working set extent for the symbiont process.</p> <p>The value set by this qualifier overrides the value defined in the UAF of any user submitting a job to the queue. For more information, see Section 5.2.7.</p>

(continued on next page)

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Table 5–1 (Cont.) Qualifiers for Specifying Queue Options

Qualifier	Type of Queue	Description
/WSQUOTA	Batch and output	For batch queues, specifies the working set quota for batch jobs executed in the queue. For output queues, specifies a working set quota for the symbiont process. The value set by this qualifier overrides the value defined in the UAF of any user submitting a job to the queue. For more information, see Section 5.2.7.
/AUTOSTART_ON	Batch and output	Creates an autostart execution queue and specifies the node or nodes (and for output queues, the device or devices) on which the queues can run. For more information, see Section 5.2.9.4.1.
/ON	Batch and output	Creates a nonautostart execution queue and specifies the node (and for output queues, the device) on which the queue is to run. For more information, see Section 5.2.9.4.1.
/GENERIC	Batch and output	Creates a generic queue and names the execution queues it feeds. For more information, see Section 5.2.9.4.2.
/CHARACTERISTICS	Batch and output	Specifies one or more characteristics associated with the queue. For more information, see Section 5.2.6.
/DESCRIPTION	Batch and output	Specifies a text string to provide users with information about the queue. For more information, see DCL Help.
/RETAIN	Batch and output	Holds jobs in the queue after they have executed. For more information, see Section 5.2.5.
/OWNER_UIC	Batch and output	Specifies the user identification code (UIC) for the queue. For more information, see Section 5.2.4.
/PROTECTION	Batch and output	Specifies a protection for the queue. For more information, see Section 5.2.4.

5.2.4 Restricting Access to a Queue

As with a files or directory, you can use UIC-based or ACL-based protection to control access to a queue.

See the *Guide to VMS System Security* for detailed information about establishing system security.

5.2.4.1 UIC-Based Queue Protection

UIC-based protection restricts the jobs and the users who have access to a queue. Operations that apply to queues are controlled by UIC-based protection in the same way that access to other system objects (such as files) is controlled.

When you create a queue, the queue is assigned an owner UIC and a protection mask based on your UIC (unless you specify another UIC with the OWNER_UIC qualifier). Jobs are assigned an owner UIC equal to the UIC of the process that submitted the job, unless the job was submitted with the /USER qualifier. Each

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

job in a queue, and each operation that is performed on a queue, is checked against the UIC of the owner, the protection of the queue, and the privileges of the requester.

All operations are checked as follows:

Operation Type	Check
Operations that apply to jobs	Checked against the READ and DELETE protection specified for the queue and the owner UIC of the job. In general, READ access to a job allows a user to see the information about a job, and DELETE access allows the user to modify and delete the job.
Operations that apply to queues	Checked against the WRITE and EXECUTE protection specified for the queue and the owner UIC of the queue. A user with WRITE access to a queue can submit jobs to that queue. Users with EXECUTE access to a queue can act as the operator for that queue, with the ability to affect any jobs in the queue. Users with operator (OPER) privilege have EXECUTE access to all queues. OPER privilege also enables users to create queues.

The default UIC-based protection for a queue is as follows:

(System:E,Owner:D,Group:R,World:W)

The following table lists each type of access and explains what functions the access controls when applied to a queue. It also explains the default value provided on a queue for the access type.

Access	Function	Default
WRITE	Submit jobs to the queue	World:Write—lets all users submit jobs to the queue.
DELETE	Modify and delete jobs in a queue	Owner:Delete —lets users modify and delete their own jobs.
READ	Display attributes of jobs in a queue	Group:Read—lets users see the attributes of the jobs of all users in their UIC group.
EXECUTE	Modify any non-security attribute of a queue and delete the queue, plus WRITE access to the queue and READ and DELETE access to all jobs in the queue	System:Execute—lets users with a system UIC act as operators for the queue.

To find out the protection specified for a particular queue, enter the DCL command SHOW QUEUE/FULL, as shown in the following example:

```
$ SHOW QUEUE/BATCH/FULL
Batch queue RUBY_BATCH, available, on RUBY::
/AUTO_START_ON=(RUBY::) /BASE_PRIORITY=3 /JOB_LIMIT=4 /OWNER=[SYSTEM]
/PROTECTION=(S:E,O:D,G:R,W:W)
```

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

5.2.4.2 ACL-Based Queue Protection

In addition to UIC-based protection, you can associate access control lists (ACLs) with a queue. ACL-based protection provides a more refined level of protection in cases where certain members of a project group require access to a queue, excluding others of the same UIC group or of other groups.

You use the DCL command SET ACL to create, modify, or delete ACLs or access control entries (ACEs) established for a queue. Use the following command format:

```
SET ACL/OBJECT_TYPE=QUEUE/ACL=(IDENTIFIER=(identifier-string,ACCESS=access-code)[,...]) queue-name
```

The keyword QUEUE specifies that the object type is a batch or output queue. Use the /ACL qualifier to add or modify the ACEs of a particular ACL. The ACL defines the identifier that a user must hold in order to perform operations on a queue and defines the access code associated with that identifier. You use the Authorize Utility to grant specific ACL identifiers to users. (See the *VMS Authorize Utility Manual* for more information.)

You can use ACLs to restrict queue access to members of a particular project group, as shown in the following example:

```
$ SET QUEUE/PROTECTION=(S,O,G,W)
$ SET ACL/OBJECT_TYPE=QUEUE SYS_QUEUE1 -
_$ /ACL=((IDENTIFIER=ULTRA_LITE,ACCESS=READ+WRITE+EXECUTE+DELETE), -
_$ (IDENTIFIER=MINUTES,ACCESS=READ)) SYS_QUEUE1
```

The SET QUEUE/PROTECTION command in this example modifies the default UIC-based protection of queue SYS_QUEUE1 to prevent access by nonprivileged users. The SET ACL command then restricts access to only those members of a project group who hold the ULTRA_LITE or MINUTES identifiers. Members with the MINUTES identifier only have read access to the queue.

If a queue has ACL protection, the SHOW QUEUE/FULL command displays the ACL information, as shown in the following example:

```
$ SHOW QUEUE/FULL SYS_QUEUE1
Batch queue SYS_QUEUE1, stopped
  /BASE_PRIORITY=4 /JOB_LIMIT=1 /OWNER=[1,4] /PROTECTION=(S,O,G,W)
  (IDENTIFIER=ULTRA_LITE,ACCESS=READ+WRITE+EXECUTE+DELETE)
  (IDENTIFIER=MINUTES,ACCESS=READ)
```

See the *Guide to VMS System Security* for detailed information about establishing ACLs for system objects.

5.2.4.3 Privileges Allowing Queue Access

Certain account privileges allow users to access a queue in spite of UIC-based and ACL-based protection. The following table lists these account privileges and the type of access they allow on a queue:

Privilege	Access
OPER	Execute access to all queues
BYPASS	Execute access to all queues
READALL	Read access to all jobs
SYSPRV	The access specified for users with system UICs
GRPPRV	The access specified for users with system UICs

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

5.2.5 Setting Up Queues to Retain Jobs

Users can request that a job be retained in a queue after the job completes by specifying the `/RETAIN` qualifier when they submit a job with the `PRINT` or `SUBMIT` command.

However, you might want to set a job retention policy on a queue to keep information about all jobs in the queue after the jobs complete. This might be helpful when an individual must track jobs submitted by other users.

By default, no job retention policy is set on a queue. To specify a job retention policy for a queue, use the `/RETAIN=option` qualifier with the `INITIALIZE /QUEUE`, `START/QUEUE`, or `SET QUEUE` command. The `/RETAIN` qualifier has the following format:

```
/RETAIN=option
```

You can specify one of the following options:

Option	Description
ALL	Holds all jobs in the queue regardless of the jobs' completion statuses.
ERROR	Holds jobs in the queue if they complete unsuccessfully.

The following command specifies that the queue retain all jobs that complete with a status other than success:

```
$ SET QUEUE/RETAIN=ERROR BATCH_QUE
```

For example, if an operator needs to be aware of all batch jobs that do not complete successfully on a certain queue, you could set the queue to retain jobs that complete with an error status. This would let the operator enter the `SHOW QUEUE` command to display a list of jobs that completed unsuccessfully. The `SHOW QUEUE` display includes the completion status of jobs retained in the queue. If a job completes unsuccessfully, this message can help determine why. The displays also include the date and time at which a retained job completed.

If you set a job retention policy on a queue, the queue's job retention option overrides the job's retention option.

Whether and where a job is retained is determined by the following:

- The retention setting on the execution queue in which the job executes.
- The retention setting on the generic queue (if the job is submitted to a generic queue).
- The completion status of the job.
- The retention requested by the user upon submitting the job (if retention was requested).

If jobs are retained in queues, you should periodically delete jobs that no longer need to be retained.

Users can specify timed job retention to retain a job only as long as they need it. For example:

```
$ SUBMIT/RETAIN=UNTIL=19-OCT-1991:07:31:0.0 MYFILE.DAT
```

This eliminates the need for you to delete retained jobs from queues. If your users make use of the `/RETAIN` qualifier, encourage them to use timed retention.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

For more information about job retention, see the INITIALIZE/QUEUE/RETAIN and SUBMIT/RETAIN commands in the *VMS DCL Dictionary*.

To remove a job retention option from a queue, specify /NORETAIN qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command.

5.2.6 Using Queue Characteristics

You can define characteristics and assign them to queues to control the jobs that execute on certain queues. Similarly, you can define print forms for output queues (including, for example, a specific paper stock and page format) and assign the forms to queues to control the jobs that execute on certain queues. For more information, see Section 5.2.8.3.

When users include the /CHARACTERISTIC qualifier with a PRINT or SUBMIT command, all the characteristics they specify must also be specified for the queue that will execute the job. If a job is placed in a queue that does not have the characteristics required by that job, the job enters a pending state. The job remains pending in the queue until you correct the characteristic mismatch as explained in Section 5.3.4.3.2. Users do not need to specify all characteristics of a queue when they enter a PRINT or SUBMIT command, as long as the characteristics specified are a subset of those established for the queue.

For example, suppose a system manager manages twelve LN03 printers, three of which are located in each of the four corners of a building. A generic queue LN03\$PRINT feeds execution queues for each of the LN03 printers. The system manager defines four characteristics, EAST, WEST, NORTH, and SOUTH. The EAST characteristic is assigned to the execution queues that feed printers in the eastern corner of a building. WEST is assigned to the execution queues feeding the printers in the western corner, and so on. When a user submits a print job to LN03\$PRINT and specifies the EAST characteristic, the job prints on the first idle LN03 printer in the eastern corner of the building.

If the system had queues for printers on multiple floors, the system manager could further define a characteristic for each floor, for example FIRST, SECOND, and THIRD.

Use the following commands when working with characteristics:

Command	Description
DEFINE/CHARACTERISTIC	Creates a characteristic and assigns a name and number.
DELETE/CHARACTERISTIC	Deletes a characteristic.
SHOW QUEUE/CHARACTERISTIC	Displays information about characteristics defined for the system.
INITIALIZE/QUEUE/CHARACTERISTICS SET QUEUE/CHARACTERISTICS START/QUEUE/CHARACTERISTICS	Specifies one or more characteristics for processing jobs on a queue.
SHOW QUEUE/FULL	Displays information about a queue, including any characteristics assigned to the queue.
PRINT/CHARACTERISTIC SUBMIT/CHARACTERISTIC SET ENTRY/CHARACTERISTIC	Specifies the name or number of one or more characteristics to be associated with the job.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

5.2.6.1 Defining Characteristics

No characteristics are defined by default. To define a characteristic, use the DEFINE/CHARACTERISTIC command in the following format:

```
DEFINE/CHARACTERISTIC characteristic-name characteristic-number
```

For example:

```
$ DEFINE/CHARACTERISTIC EAST 1
```

For more information, see the DEFINE/CHARACTERISTIC command in the *VMS DCL Dictionary*.

5.2.6.2 Displaying Characteristics Defined on a System

To see the characteristics defined on a system, enter the SHOW QUEUE /CHARACTERISTIC command. For example:

```
$ SHOW QUEUE/CHARACTERISTIC
```

Characteristic name	Number
-----	-----
EAST	1
WEST	2
NORTH	3
SOUTH	4

5.2.6.3 Assigning Characteristics to a Queue

No characteristics are assigned to a queue by default. To assign one or more characteristics to a queue, include the /CHARACTERISTICS qualifier with the INITIALIZE/QUEUE, START/QUEUE or SET QUEUE command as follows:

```
SET QUEUE/CHARACTERISTICS=characteristic[...] queue-name
```

For example:

```
$ SET QUEUE/CHARACTERISTICS=(EAST) LN03_1
```

For more information, see the /CHARACTERISTICS qualifier of the INITIALIZE /QUEUE command in the *VMS DCL Dictionary*.

5.2.6.4 Displaying Characteristics Assigned to a Queue

To determine the characteristics defined for a queue, enter the DCL command SHOW QUEUE/FULL. For example:

```
$ SHOW QUEUE/FULL LN03_1
Printer queue LN03_1, idle, on HERA::TTA3, mounted form DEFAULT
<Printer queue on node HERA for an LN03 printer>
 /BASE_PRIORITY=4 /CHAR=(1) /DEFAULT=(FLAG=ONE,FORM=LN03$PORTRAIT
 (stock=DEFAULT)) /LIBRARY=LN03LIBRARY Lowercase
 /OWNER=[SYSTEM] /PROCESSOR=LATSYM /PROTECTION=(S:E,O:D,G:R,W:W)
 /SEPARATE=(RESET=(ANSI$RESET))
```

5.2.6.5 Cancelling Characteristics Assigned to a Queue

To cancel characteristics assigned to a queue, specify the /NOCHARACTERISTICS qualifier with the INITIALIZE/QUEUE, START /QUEUE or SET QUEUE command.

```
$ SET QUEUE/NOCHARACTERISTICS
```

5.2.6.6 Deleting Characteristics

To delete a characteristic definition, enter the DCL command DELETE/CHARACTERISTIC. For example:

```
$ DELETE/CHARACTERISTIC EAST
```

You must specify the characteristic *name* with the DELETE/CHARACTERISTIC command. If you know the number assigned to the characteristic but do not know the name, enter the SHOW/QUEUE/CHARACTERISTIC command to display the names and numbers assigned to characteristics on the system.

For more information about the DELETE/CHARACTERISTIC command, see the *VMS DCL Dictionary*.

If the system displays the following messages, a queue or job refers to the characteristic:

```
%DELETE-E-NOTDELETED, error deleting characteristic
-JBC-E-REFERENCED, existing references prevent deletion
```

You must remove all references to the characteristic before you can delete the characteristic. For information about removing references to a characteristic, see Section 5.3.4.6.

5.2.7 Options Specific to Batch Queues

You can use the following qualifiers with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command to set queue options that control batch job performance and the use of system resources by processes executing batch jobs:

Option	Description
/JOB_LIMIT= <i>n</i>	Specifies the number of jobs that can execute simultaneously in the queue.
/[NO]DISABLE_SWAPPING	Specifies whether or not the processes running jobs on the queue can be swapped out of memory.
/CPUDEFAULT= <i>time</i>	Specifies the base value for the default CPU time limit. The time cannot exceed the time limit set by the /CPUMAXIMUM qualifier.
/CPUMAXIMUM= <i>time</i>	Specifies the maximum CPU time limit.

You can also use the following qualifiers. Although these qualifiers are not specific to batch queues, they are commonly used like the batch-specific options, to control batch job performance and the use of system resources by batch processes.

Option	Description
/BASE_PRIORITY= <i>n</i>	Specifies the base process priority at which jobs are initiated from a batch queue.
/WSDEFAULT= <i>n</i>	Specifies the default working set size for jobs executed in a batch queue. (For output queues, specifies the default working set size for symbiont processes.)
/WSEXTENT= <i>n</i>	Specifies the working set extent for jobs executed in a batch queue. (For output queues, specifies the working set extent for symbiont processes.)

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Option	Description
<code>/WSQUOTA=<i>n</i></code>	Specifies the working set quota for jobs executed in a batch queue. (For output queues, specifies the working set quota for symbiont processes.)

For more information about these limits, quotas, and priorities, see the *Guide to VMS Performance Management* and the chapter on system resources in the *Guide to Setting Up a VMS System*.

By default, a process running a batch job uses values taken from the UAF record of the user submitting the job or from SYSGEN parameter settings. For more information about the default values for these options, see the description of the qualifiers for the INITIALIZE/QUEUE command in the *VMS DCL Dictionary*. If you specify values for any of these options, processes for jobs executed in the queue will use the values you set unless the user specifies values when the job is submitted. (A user can specify values for CPU time and for the working set options default, quota, and extent.) If a user specifies a value for one of these options when the job is submitted, this value cannot exceed the value you set for the queue. If you did not specify a value, the value specified by the user cannot exceed the value specified in the associated UAF limit or SYSGEN parameter.

The following sections provide guidelines for choosing values for these options.

5.2.7.1 Base Process Priority

Choose a value based on how quickly you will allow batch jobs to progress. If you choose a value equal to the system parameter value DEFPRI (typically 4), jobs in this queue will progress on an equal footing with typical interactive jobs. This choice might be appropriate for systems that have an abundance of available CPU time.

If, on the other hand, you choose a value less than DEFPRI, jobs in this queue will potentially progress more slowly than the typical interactive job. CPU time will be allocated to jobs in this queue only after servicing jobs of DEFPRI priority.

In the case of a queue defined for a very special purpose—for example, high priority jobs—a value greater than DEFPRI (for example, 5) might be appropriate. However, this choice can have a significant negative effect on the performance of other users and batch jobs.

5.2.7.2 Job Limit

Keep this value low when using a base process priority of DEFPRI or greater, since each batch job can affect the performance of interactive jobs.

5.2.7.3 Working Set Default, Quota, and Extent

If you do not specify values for these options, a job uses the value specified in its owner's User Authorization file (UAF) record.

Working set default is the value to which the working set returns at the exit of each image. Typically, the value should be relatively small and is usually best left at the value specified in the user's UAF record.

Working set quota is the value that approximates the amount of physical memory used by each batch job in the queue in a *memory-constrained* system.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Working set extent is the value that approximates the amount of physical memory in a *memory-rich* system. Typically, you should set this to a high value. The working set extent value is an upper limit for the size of the working set; the working set cannot be expanded beyond this value even if more memory is required by the job. If you set this value too low, a job might page fault heavily even if the system has plenty of memory available. To be safe, choose a working set extent value equal to the system parameter value of WSMAX, which specifies the maximum working set size possible for your system.

In general, the working set quota and extent values specified in the user's UAF are sufficient. However, you might want to specify more generous or stringent values for a queue, depending on the purpose of the queue. For example, you might encourage users to submit large jobs (such as compiling and linking large programs) as batch jobs to reserve interactive use of the system for jobs that do not require extensive resources, as follows:

- Set a large working set size for batch jobs by specifying larger WSQUOTA and WSEXTENT values when you create the batch queue.
- Restrict the working set size of interactive jobs by providing smaller WSQUOTA and WSEXTENT values in users' UAF records.

5.2.7.4 CPU Default and Maximum

You can restrict and expand the amount of time a job is allowed in the CPU by setting CPU limit values.

When a job's CPU time limit is exceeded, the job is terminated with an error status. So, be careful not to prematurely terminate jobs that are making legitimate and authorized use of the CPU. If working set values are set too low, you might make less efficient use of your memory, but the jobs can still complete normally. However, if CPU time limit values are set too low, jobs can terminate prematurely.

For example, you might use a CPU time limit on a batch queue devoted to execution of newly coded software that could unexpectedly enter a CPU loop. The CPU time limit would terminate infinitely looping jobs so they do not waste the CPU resource. However, you must be careful to choose a sufficiently high time limit so normally executing jobs do not terminate prematurely.

Another way to control allocation of the CPU resource is to create a special-purpose batch queue that shifts execution of its jobs to a less busy time of day when CPU time is more available.

5.2.7.5 Swapping

Typically, swapping is enabled on batch queues. However, for a very special purpose high priority queue, you might disable swapping. This provides a favorable status to jobs in this queue by removing them from consideration when memory must be reclaimed from processes (through the swapping and trimming mechanism of VMS). For more information, see the *Guide to VMS Performance Management*.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

5.2.7.6 Setting Up Batch Queues on Memory-Constrained Systems

On memory constrained systems, you should add up the pages required for the batch working sets on all batch queues and make sure that enough fluid memory remains for interactive jobs. Fluid memory can be reassigned from one process to another through swapping and paging. (You can calculate fluid memory as the space that remains when you subtract the number of pages permanently allocated to the VMS operating system from the total memory. To obtain these values, enter the DCL command SHOW MEMORY.)

5.2.7.7 Optimizing Batch Queues for the VMS Sort/Merge Utility

You can improve the efficiency of the VMS Sort/Merge Utility (SORT) by designating one batch queue for sorting jobs and giving this queue a very large working set quota.

In addition, you can set process quotas for greatest SORT efficiency. The values recommended below are based solely on SORT considerations; you should integrate other system considerations with these to determine appropriate values.

- Working set extent per process (WSEXTENT)—For maximum sorting efficiency, use the /WSEXTENT qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command to set this value on the queue to the largest value any sorting job would ever need. Generally, the smaller the working set, the slower the sort. For very large files, working sets of 4000 or more pages are not at all unreasonable, provided the system has enough physical memory to support them.
- Virtual page count per process (VIRTUALPAGECNT)—For maximum sorting efficiency, set this SYSGEN parameter to at least 1000 pages more than the maximum working set extent. Although SORT limits the size of the sort data structure to the process's working set extent, if work files are required, SORT can use the extra memory for I/O buffers. If this parameter is too low relative to the working set extent, SORT might be unable to create buffers for the work files when the files cannot be sorted in memory.

5.2.8 Options Specific to Output Queues

You can use the following options with output queues.

Option	Section
Job scheduling based on size.	See Section 5.2.8.1.
Banner pages for ease in separating jobs and files within jobs.	See Section 5.2.8.2.
Forms and stock.	See Section 5.2.8.3.
Controlling page and line overflow.	See Section 5.2.8.4.
Device control libraries to set up printers for certain formats.	See Section 5.2.8.5.

5.2.8.1 Scheduling Jobs Based on Job Size

The following qualifiers let you specify options that affect how the size of a print job affects its scheduling:

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Option	Description
<code>/SCHEDULE=[NO]SIZE</code>	Lets you control whether pending jobs are scheduled for printing based on the size of the job.
<code>/[NO]BLOCK_LIMIT=([lowlim,]uplim)</code>	Lets you reserve certain printers for certain size jobs.

For more information on job scheduling, see Section 5.1.7.

/SCHEDULE

By default, printer and terminal queues are created with the `/SCHEDULE=SIZE` option, causing shorter jobs to print before long ones. However, you can specify the `/SCHEDULE=NOSIZE` option, causing jobs to be printed in the order they were submitted, regardless of size. For more information, see the `/SCHEDULE` qualifier for the `INITIALIZE/QUEUE`, `START/QUEUE`, or `SET QUEUE` command in the *VMS DCL Dictionary*.

/BLOCK_LIMIT

By default, printer and terminal queues are created with no block limit, so jobs of any size will be printed. However, you can specify the `/BLOCK_LIMIT` qualifier to limit the size of jobs that can be printed. You can specify just an upper limit or both an upper and lower limit.

For example, you could submit a batch job that runs a command procedure to set a maximum block limit of 500 blocks for a queue at 8:00 a.m.. The command procedure might resubmit itself and remove the block limit after 5:00 p.m. each day. This would let users print jobs of any size after normal work hours, when the printing workload is lighter. Users can specify the `/AFTER` qualifier with the `PRINT` qualifier to specify that a job is to be printed after a certain time.

For more information about restricting print job sizes, see the `/BLOCK_LIMIT` qualifier for the `INITIALIZE/QUEUE`, `START/QUEUE`, or `SET QUEUE` command in the *VMS DCL Dictionary*.

5.2.8.2 Using Banner Pages

Banner pages are helpful in identifying and separating jobs and the files within those jobs. Banner pages can be any of the following types:

Type	Subtype	Description
Job pages	Flag page	A single page printed before each job.
	Burst page	Two flag pages, divided by a separation (burst) bar, printed before each job.
	Trailer page	A page printed after each job.
File pages	Flag page	A single page printed before each file in a job.
	Burst page	Two flag pages, divided by a separation (burst) bar, printed before each file in a job.
	Trailer page	A page printed after each file in a job.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Most sites use only a subset of the available banner pages at any given time. Use the following commands when working with banner pages:

Command	Description
INITIALIZE/QUEUE/SEPARATE= <i>option</i> START/QUEUE/SEPARATE= <i>option</i> SET QUEUE/SEPARATE= <i>option</i>	Specifies one or more of the following job banner page options: [NO]BURST [NO]FLAG [NO]TRAILER The job banner pages you specify for a queue, cannot be overridden by the user.
INITIALIZE/QUEUE/DEFAULT= <i>option=keyword</i> START/QUEUE/DEFAULT= <i>option=keyword</i> SET QUEUE/DEFAULT= <i>option=keyword</i>	Specifies one or more of the following file banner page options: [NO]BURST [NO]FLAG [NO]TRAILER <i>Keyword</i> can be either ALL or ONE. Users can override the file banner page settings you specify for a queue by specifying the BURST, FLAG, and TRAILER qualifiers with the PRINT command.
PRINT/BURST[= <i>keyword</i>]	Specifies one of the following keywords to override, for the job, the file burst pages specified for the queue: ALL or ONE.
PRINT/FLAG[= <i>keyword</i>]	Specifies one of the following keywords to override, for the job, the file flag pages specified for the queue: ALL or ONE.
PRINT/TRAILER[= <i>keyword</i>]	Specifies one of the following keywords to override, for the job, the file trailer pages specified for the queue: ALL or ONE.

Job and File Flag and Burst Pages

Flag and burst pages both indicate that a print file follows. However, a burst page is two flag pages divided by a **burst bar**, which is designed to print over the crease between two pages of continuous form paper. The burst bar is used for ease in bursting continuous form paper. If you do not need to burst pages of a printer's output, for example, if your printer uses cut sheet paper, avoid the burst page option. Flag pages, or flag and trailer pages, are usually sufficient for identifying the end of a job.

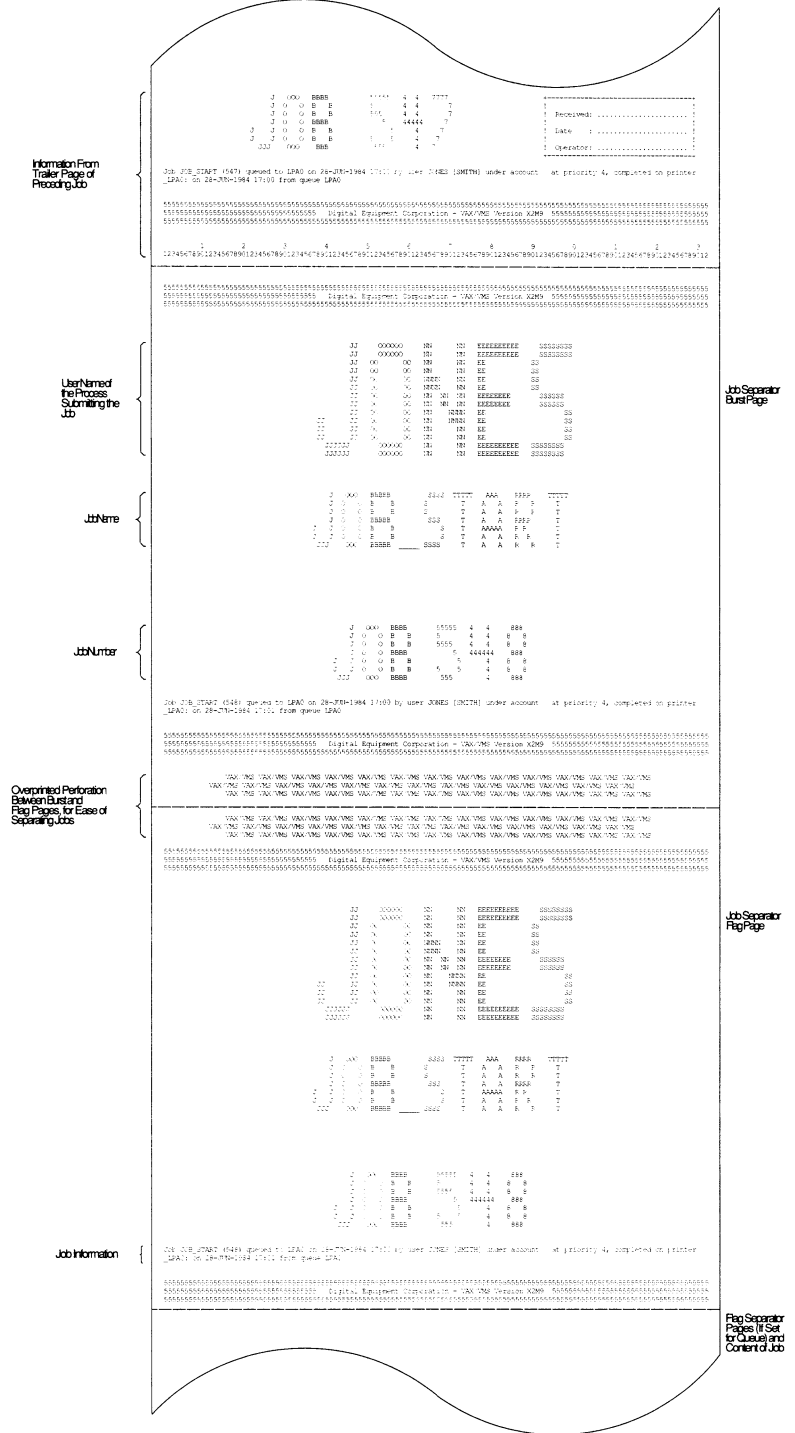
Figure 5-11 illustrates job flag and burst pages produced by the default VMS print symbiont, PRTSMB.

Figure 5-12 illustrates file flag and burst pages produced by PRTSMB.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Figure 5-11 Job Flag and Burst Pages



24-88-C

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

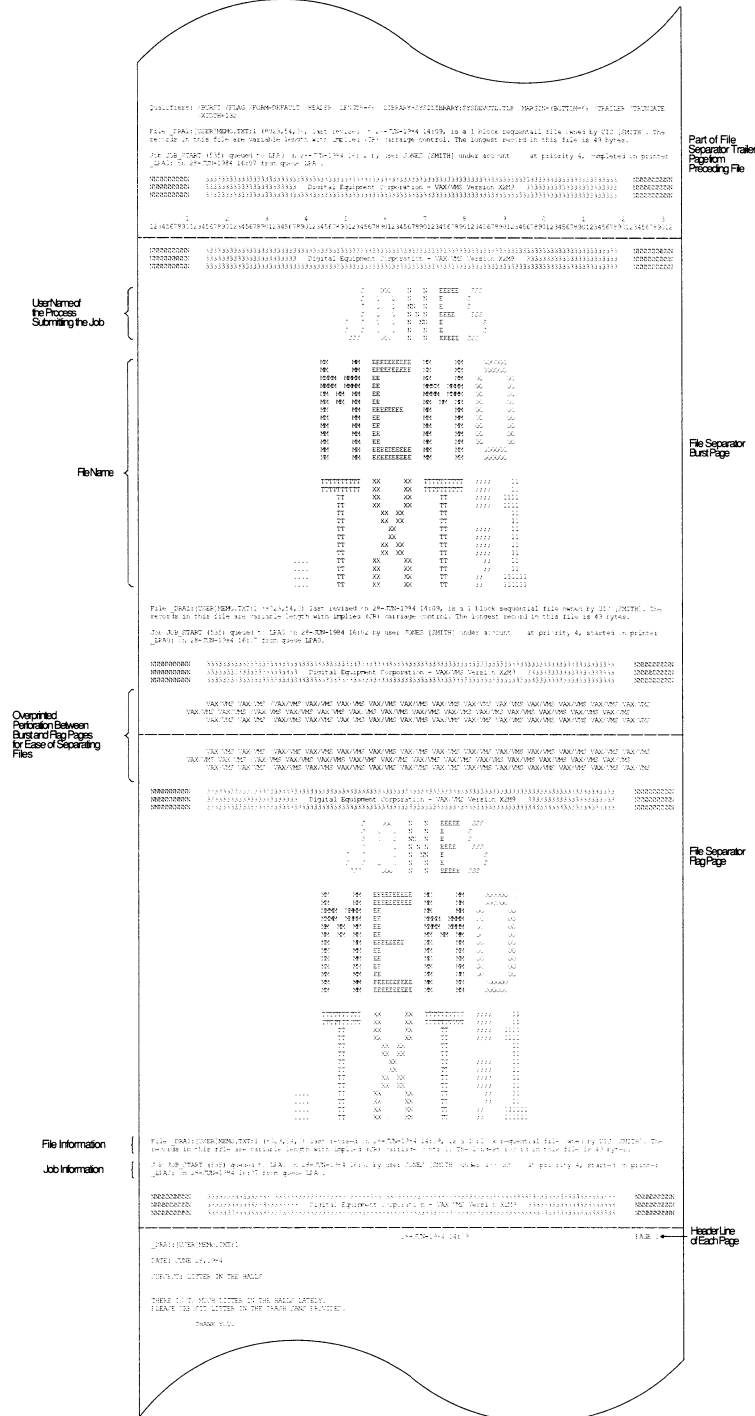
The job flag page indicates that a new job follows; the job can include one or more files. Information found on a job flag page includes the following:

Item	Description
Header bar	A single-segment bar composed of the following elements: <ul style="list-style-type: none">• Rows of a repeated numeral indicating the sequence of the job in the printer queue.• An embedded text string specified by defining the PSM\$ANNOUNCE system logical name. The length of the string is limited to one form width. If PSM\$ANNOUNCE is not defined, the default text string is “Digital Equipment Corporation” followed by the system version number. (“Digital Equipment Corporation” is abbreviated to “DEC” for shorter form widths.)
Note	Users can specify a string of up to 255 characters with the /NOTE qualifier of the DCL command PRINT.
Identification banner	This includes user name of the process submitting the job, the job name, and the job number.
Job sentence	Indicates the following information: <ul style="list-style-type: none">• Job name and number• Name of queue to which job was submitted• Submission date and time• Process user name, UIC, and account• Job scheduling priority• Print device name• Job start time• Execution queue name
Footer bar	Identical to the header bar with the exception that the definition of the system logical name (PSM\$ANNOUNCE) is not used as the embedded string. The default text is always used as the embedded string.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Figure 5-12 File Flag and Burst Pages



7K-4B-CE

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Information found in the file flag page includes the following items:

Item	Description
Header bar	A three-segment bar composed of the following elements: <ul style="list-style-type: none">• A central segment identical to the job header bar• Flanking segments with rows of a repeated character indicating the sequence of the file in the job.
Note	Users can specify a string up to 255 characters in length using the /NOTE qualifier of the DCL command PRINT.
Identification banner for the process submitting the job	Includes the user name and file name.
File sentence	Indicates the following information: <ul style="list-style-type: none">• Full file specification, including device and directory, file name, type, version, and revision time and date• File size in blocks• File organization• File owner's UIC• File record characteristics: record type, carriage control, and size of longest record.
Job sentence	Indicates the following information: <ul style="list-style-type: none">• Job name and number• Name of queue to which job was submitted• Submission date and time• Process username, UIC, and account• Job scheduling priority• Print device name• Job start time• Execution queue name
Footer bar	Identical to header bar except that the embedded text string is "Digital Equipment Corporation" followed by the operating system version number.

Job and File Trailer Pages

A trailer page indicates the end of a print job or print file. Figure 5-13 illustrates a job trailer page. Figure 5-14 illustrates a file trailer page.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

The job trailer page indicates the end of a print job. Information found on the job trailer page includes the following:

Item	Description
Header bar	A three-segment bar composed of the following elements: <ul style="list-style-type: none">• Central segment with “END OF JOB” banner• Flanking segments with job-sequence numeral
Identification banner	For the process submitting the job. Includes job name and job number.
Receipt box	Contains the following signoff fields: Received; Date; and Operator: .
Job sentence	
Ruler	A sequence of numbers counting to the end of the form.

The file trailer page can include the following items:

Item	Description
Header bar	A five-segment bar composed of the following elements: <ul style="list-style-type: none">– Central segment with “END OF FILE” banner– Inner flanking segments with job-sequence numeral– Outer flanking segments with file-sequence character
Identification banner	For the process submitting the job. Includes user name, file name.
Qualifier phrase	Indicates the print, queue, and form qualifiers active when the job was submitted; nonactive qualifiers (except /NORECORD_BLOCKING and /NOFEED) are not included.
File sentence	
Job sentence	
Footer bar	Identical to file flag/file burst footer bar.
Ruler	A sequence of numbers counting to the end of the form.

Widths greater than 40 characters and less than 200 characters and lengths of any size greater than 40 characters are supported for file and job banner pages. Pages requested for widths greater than 200 characters are formatted and printed at 200-character widths. Lengths less than 40 characters are extended by that form length until the 40-character threshold is exceeded. Margins are not taken into account when formatting banner pages.

Note

All banner pages format information to the width and length of a VMS default form size of 80 characters by 51 lines. Information, therefore, might be truncated, depending on the form sizes you specify. See Section 5.2.8.4 for information about controlling line and page overflow.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

5.2.8.3 Using Forms

Print forms serve the following two functions:

- Forms determine certain page formatting attributes (such as margins and page length).
- The paper stock specified in the form definition is used in determining whether a job is eligible to print.

If your printing needs are limited, you do not need to use special forms; Digital supplies a systemwide default form (named DEFAULT) for all queues on a VMS operating system.

However, if you want to help users format output or if certain print jobs require special paper, you can create one or more forms using the DCL command DEFINE /FORM. Users can then request a form for a print job by specifying the form when they submit a job using the /FORM qualifier with the PRINT command. If a user does not specify a form, the job uses the queue's **default form**.

Use the following commands when working with forms:

Command	Description
DEFINE/FORM	Creates a form and assigns a name and number.
SHOW QUEUE/FORM/FULL	Displays information about forms available on a system.
DELETE/FORM	Deletes a form.
INITIALIZE/QUEUE/DEFAULT=FORM START/QUEUE/DEFAULT=FORM SET QUEUE/DEFAULT=FORM	Specifies the name or number of the default form for an output execution queue.
PRINT/FORM SET ENTRY/FORM	Specifies the name or number of the form to be associated with a print job.
INITIALIZE/QUEUE/FORM_MOUNTED START/QUEUE/FORM_MOUNTED SET QUEUE/FORM_MOUNTED	Specifies the name or number of the mounted form for an output execution queue.
SHOW QUEUE/FULL	Displays information about a queue, including the default form for the queue, and the form mounted on the queue.

You can create forms to format output in the following ways:

- Control page width and length and margin size. For more information, see Section 5.2.8.3.2.
- Control line overflow by truncating or wrapping lines. For more information, see Section 5.2.8.4.
- Specify device control library modules to setup the device at the start of each page or each file or both. For more information, see Section 5.2.8.5.4.
- Pause output at the end of each page so a new sheet of paper can be inserted (use this for printers that require sheet feed paper).
- Specify that a certain paper stock must be physically loaded in the printer.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

The stock of a form is particularly important because it affects whether a job is eligible to print. The stock of a job's form must match the stock of the form currently being used by the queue (the queue's **mounted form**), or the job will not be scheduled to print; instead, it remains pending in the queue until you perform special steps. (These steps are described in Section 5.3.4.3.1.)

A queue's mounted form is the form of the current job or, if no job is processing, the form of the last job printed in a queue. As long as the stock of a job's form matches the stock of the queue's mounted form, the job is processed using the options in the job's form, and the mounted form changes automatically to the form of the job being processed in the queue.

If a user does not specify a form, the job uses the queue's **default form**. See Section 5.2.8.3.5 for information about specifying a default form for a queue.

5.2.8.3.1 Procedure for Using Forms To use forms with your queues, take the following steps:

1. Create a form as explained in Section 5.2.8.3.2.
2. If desired, assign a default form for each output execution queue. For more information, see Section 5.2.8.3.5.

If you do not assign a default form to a queue, the queue uses the systemwide default form named DEFAULT. If you want to change the DEFAULT form, you must do so *before* creating any queues that reference the DEFAULT form. For more information, see Section 5.2.8.3.4.

3. Inform users of the available forms and the queues with which they should be used. In addition, you can create symbols to automatically include the form with the PRINT command as follows:

```
$ PRINT_REPORT ::= PRINT/FORM=REPORT
```

4. If you see a print job pending because of a stock mismatch, change the stock of the printer to the requested stock and mount the form associated with the requested stock on the queue, as explained in Section 5.2.8.3.6, or perform one of the other steps explained in Section 5.3.4.3.1.

5.2.8.3.2 Defining a Form To create a form, enter the DEFINE/FORM command as follows:

```
DEFINE/FORM form-name form-number [/qualifiers]
```

You can specify any of the following qualifiers:

Qualifier	Purpose
/WIDTH= <i>n</i>	Specifies the physical width of the paper in characters.
/LENGTH= <i>n</i>	Specifies the physical length of a form page in lines.
/[NO]TRUNCATE	Specifies that characters exceeding the line length specified by /WIDTH and /MARGIN be discarded. See Section 5.2.8.4.
/[NO]WRAP	Specifies that characters exceeding the line length specified by /WIDTH and /MARGIN be wrapped to the next line. See Section 5.2.8.4.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Qualifier	Purpose
<code>/MARGIN=(option=n[,...])</code>	Specifies the number of blank spaces for one or more of the four margin options: BOTTOM, LEFT, RIGHT, and TOP.
<code>/[NO]PAGE_SETUP=(module[,...])</code>	Specifies one or more device control modules that set up a device at the start of each page. See Section 5.2.8.5.4.
<code>/SETUP=(module[,...])</code>	Specifies one or more device control modules that set up the device at the start of each file. See Section 5.2.8.5.4.
<code>/[NO]SHEET FEED</code>	Specifies that print jobs pause at the end of every physical page so that a new sheet of paper can be inserted.
<code>/STOCK=string</code>	Specifies the type of paper stock to be associated with the form.
<code>/DESCRIPTION=string</code>	Specifies a string used to provide information about the form.

For example:

```
$ DEFINE/FORM MEMO 3/STOCK=DEFAULT -
_$ /MARGIN=(TOP=2,BOTTOM=2,LEFT=6)/WIDTH=80/LENGTH=66/TRUNCATE -
_$ /DESCRIPTION="LN03 indented memo format"
```

The command in the example defines the form MEMO as the number 3 and defines formatting options for the form.

If you create forms only to provide different formatting options (and not to specify paper stock), specify the same stock type for each of the different forms. That way, jobs requesting any of these forms will print on the same queue without requiring you to enter any additional commands to modify the queue. Unless you specify the `/STOCK` qualifier, the form's stock is the same as the name of the form.

For more information about the `DEFINE/FORM` command, see the *VMS DCL Dictionary*.

5.2.8.3.3 Displaying Forms Defined on a System To display forms defined on a system, enter the `SHOW/QUEUE/FORM/FULL` command.

If you know the name of the form you want to display, you can specify the form name as a parameter to the command. For example:

```
$ SHOW QUEUE/FORM/FULL MEMO
Form name                               Number  Description
-----                               -
MEMO (stock=DEFAULT)                    110    LN03 indented memo format
      /LENGTH=66 /MARGIN=(TOP=2,BOTTOM=2,LEFT=5) /STOCK=DEFAULT /TRUNCATE
      /WIDTH=80
```

For more information, see the `SHOW QUEUE/FORM` command in the *VMS DCL Dictionary*.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

5.2.8.3.4 Changing the Systemwide Default Form The VMS operating system includes a systemwide default form, named DEFAULT, that corresponds to the form number 0 and has a stock named DEFAULT. A queue initialized without the /DEFAULT=FORM qualifier uses the systemwide default form to process print jobs not explicitly associated with a form definition. The systemwide DEFAULT form uses the following options:

Form name	Number	Description
-----	-----	-----
DEFAULT	0	System-defined default
/MARGIN=(BOTTOM=6) /STOCK=DEFAULT /TRUNCATE /WIDTH=132		
/LENGTH=66		

To change the systemwide default form, enter the DEFINE/FORM command in the following format:

```
DEFINE/FORM DEFAULT 0 /qualifier[s]
```

For example, to change the systemwide default form's bottom margin from 6 to 4, and the page length from 66 to 55, enter this command:

```
$ DEFINE/FORM DEFAULT 0 /MARGIN=(BOTTOM=4) /LENGTH=55
```

Note

Once a queue or job references a form, you cannot change the stock for that form. Therefore, if you want to change to stock of the DEFAULT form, you should do so before any queues are created.

5.2.8.3.5 Assigning a Default Form for a Queue If a user does not specify the /FORM qualifier when submitting a job with the PRINT command, the job uses the default form for the execution queue on which the job is printed.

To assign a default form for an output execution queue, specify the /DEFAULT qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command as follows:

```
INITIALIZE/QUEUE/DEFAULT=FORM=form-name-or-number
```

Note

The queue's default form is associated with a print job at the time the job is processed unless a specific form is requested by the user when the job is submitted. Therefore, if a user submits a job to a generic queue without specifying the /FORM qualifier, no form is associated with the job until it is transferred to an execution queue.

If you do not establish a default form for a queue, the queue uses the systemwide default form, DEFAULT.

5.2.8.3.6 Mounting a Form on a Queue Mounting a form on a queue associates the stock of the form with the queue. The stock of the form mounted on a queue should match the stock of the paper physically loaded in the queue's printer. A queue's mounted form affects the scheduling of all jobs in the queue. If the stock of a job does not match the stock of the form currently mounted, the job is held pending because of stock mismatch and is not scheduled. When this happens, you can mount a different form on the queue by using the /FORM_MOUNTED qualifier with the SET QUEUE command.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

You mount a form on a queue by specifying the `/FORM_MOUNTED` qualifier with the `INITIALIZE/QUEUE`, `START/QUEUE` or `SET QUEUE` command as follows:

```
INITIALIZE/QUEUE/FORM_MOUNTED=form-name-or-number
```

See Section 5.3.4.3.1 for other steps you can take to fix jobs that are pending because of stock mismatch. See the *VMS DCL Dictionary* for more information about the `/FORM_MOUNTED` qualifier.

5.2.8.3.7 Displaying the Form Assigned to a Queue To find the default form for a queue, enter the DCL command `SHOW QUEUE/FULL`. For example:

```
$ SHOW QUEUE/FULL JEAN_PRINT
Printer queue JEAN_PRINT, idle, on BAY::TTA3:, mounted form 8_5x11
  <Queue for printer in Jean's office>
  /BASE_PRIORITY=4 /DEFAULT=(FEED,FORM=REPORT (stock=8_5x11)) /OWNER=[SYSTEM]
  /PROTECTION=(S:E,O:D,G:R,W:W)
```

In this example, the default form is `REPORT` and its stock is `8_5x11`. All jobs processed on this queue that are not associated with an explicit form definition in the `PRINT` command have the default form `REPORT`. As long as the stock of the mounted form matches the stock of the default form, all jobs submitted to this queue without an explicit form definition will be scheduled to print.

5.2.8.3.8 Deleting a Form To delete a defined form, enter the DCL command `DELETE/FORM`. For example:

```
$ DELETE/FORM MEMO
```

You must specify the form *name* with the `DELETE/FORM` command (not the form *number*). If you know the number assigned to the form but do not know the name, enter the `SHOW/QUEUE/FORM` command to display the names and numbers assigned to forms on the system.

For information about the `DELETE/FORM` command, see the *VMS DCL Dictionary*.

If the system displays the following messages, a queue or job exists with a reference to the form:

```
%DELETE-E-NOTDELETED, error deleting form-name
-JBC-E-REFERENCED, existing references prevent deletion
```

You must remove all references to the form before you can delete the form. For information about removing references to a form, see Section 5.3.4.6.

5.2.8.4 Controlling Page and Line Overflow

Digital recommends that you control line overflow by using form definitions. To do this, you must set terminals and printers to avoid wrapping or truncating the print line before the physical limit of the device's width.

Note

The print symbiont uses the form to determine the width of a line. Once the print symbiont has finished formatting the data, if the width of the line exceeds the `/WIDTH` setting for the device, the device driver will use the `/TRUNCATE` or `/WRAP` settings (if set) to truncate or wrap the line.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Different forms can have different right, left, top, and bottom margin settings. By using forms to control page and line overflow, users can switch from one form to another without requiring operators to stop the queue, alter the device setup, and restart the queue. The queue manager automatically mounts any forms with the same stock as the currently mounted form. Operator assistance is only needed to mount a form which has a stock that differs from the stock of the currently mounted form. For more information, see Section 5.3.4.3.1.

You control line overflow by using the following qualifiers to the DCL command DEFINE/FORM:

Qualifier	Function
/[NO]TRUNCATE	Specifies that data in the right margin be discarded. The default is /TRUNCATE. If you specify /TRUNCATE, you cannot specify /WRAP; the /TRUNCATE qualifier forces /NOWRAP.
/[NO]WRAP	Specifies that a carriage return and line feed be inserted when the right margin is exceeded. The default is /NOWRAP. If you specify /WRAP, you cannot specify /TRUNCATE.

To control page overflow errors, /DEFAULT=[NO]FEED option with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command. This option controls whether a form-feed character is automatically inserted when the symbiont detects entry into the bottom margin area. Users can use the PRINT /FEED or PRINT/NOFEED command to override the default feed option specified for a queue.

Users can specify the /PASSALL qualifier with the PRINT command to bypass all formatting performed by the print symbiont. The default is /NOPASSALL. This qualifier should be used in cases where the print symbiont formatting might interfere with the desired formatting of the file. The /PASSALL qualifier causes the print symbiont to send I/O to the device driver with all formatting suppressed and to behave as follows:

- Not interpret FORTRAN or print carriage control characters
- Not perform line or page overflow error handling
- Not interpret escape sequences

5.2.8.5 Using Device Control Libraries

A device control library is a text library that contains user-written modules consisting of text or escape sequences. Device control library modules can be used for the following purposes:

- With programmable printers, to insert device-dependent escape sequences that set up a printer for selected print options such as point size, character set, and bold or italic print.
- With both programmable and nonprogrammable printers, to insert text at specific points in the processing of a print job.

There are three types of device control modules, distinguished by their placement in a print job:

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Module Type	Description
SETUP modules	Inserted at the beginning of a file
PAGE_SETUP modules	Inserted at the beginning of each page
RESET modules	Inserted at the end of each job

Use the following DCL commands to set up device control library modules for processing print jobs:

Command	Description
DEFINE/FORM/SETUP= <i>module</i> [,...]	Specifies one or more modules that set up the device before each file of a job that is printed when the form is mounted.
DEFINE/FORM/[NO]PAGE_SETUP= <i>module</i> [,...]	Specifies one or more modules that set up the device before each page of a job that is printed when the form is mounted.
INITIALIZE/QUEUE/LIBRARY= <i>filename</i> START/QUEUE/LIBRARY= <i>filename</i>	Specifies the file name of the device control library. The default library is SYS\$LIBRARY:SYSDEVCTL.TLB. The value specified as <i>filename</i> must be a file name only, without an extension or a directory. The file must be found in the SYS\$LIBRARY directory.
INITIALIZE/QUEUE/SEPARATE=[NO]RESET= <i>module</i> [,...] START/QUEUE/SEPARATE=[NO]RESET= <i>module</i> [,...] SET QUEUE/SEPARATE=[NO]RESET= <i>module</i> [,...]	Assigns one or more modules to the queue. This causes the modules to be extracted from the device control library and copied to the device at the end of each job. The module can be used to restore the device to a known state at the end of each job. The default is NORESET.
PRINT/FORM= <i>form-name</i>	Calls for a print job to be processed according to the specified form. You can create a form to contain setup or page setup modules to be to be extracted from the device control library and copied to the printer at the start of each file or page.

5.2.8.5.1 Procedure for Using Device Control Libraries To use a device control library module, perform the following steps:

1. Create a library and insert modules. For more information, see Section 5.2.8.5.2.
2. Assign the device control library to a queue. (This step is not necessary if you use the default library SYSDEVCTL.TLB.) For more information, see Section 5.2.8.5.3.
3. Create one or more forms with setup or page setup modules. This lets users request the form (and the accompanying setup or page setup modules) when they enter the PRINT/FORM command. For more information, see Section 5.2.8.5.4.
4. Assign a reset module to a queue. This specifies the reset module to follow each job printed on a queue. For more information, see Section 5.2.8.5.5.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

5.2.8.5.2 Creating a Device Control Library and Inserting Modules To create a device control library and insert modules, perform the following steps:

1. Create a device control library by entering a command in the following format:

```
LIBRARY/CREATE/TEXT SYS$LIBRARY:filename.TLB
```

2. Determine the contents of the module—either the text to be inserted or the escape sequences needed for the desired printer setup. To determine the proper escape sequences for a printer option, refer to the operation guide for the specific printer.
3. Create a module file and enter the appropriate escape sequences, carriage control characters, or text. You create and edit a module file as you would any other text file. Make sure the text editor you use does not insert a carriage return or line feed at the end of your file. This will affect your printer output.

4. Insert the module into the device control library by entering the following command:

```
LIBRARY/INSERT/TEXT library-file module-file
```

Note

To add or delete a module from a library, you must stop all output queues assigned to that library.

See the *VMS Librarian Utility Manual* for more information about creating libraries and inserting modules.

5.2.8.5.3 Assigning a Library to a Queue You assign a device control library to an output queue by specifying the /LIBRARY qualifier with the INITIALIZE /QUEUE or START/QUEUE command in the following format:

```
INITIALIZE/QUEUE/START/LIBRARY=filename queue-name
```

The file name is the name of the library file that contains the desired modules. Libraries must be in SYS\$LIBRARY and must have the file type TLB. If you do not specify the /LIBRARY qualifier, the default library is SYS\$LIBRARY:SYSDEVCTL.TLB. You can use the /LIBRARY qualifier to specify an alternate device control library. For example:

```
$ INITIALIZE/QUEUE/LIBRARY=LN03DEVCTL LN03_A_QUE
```

Note

If you specify a value for the /LIBRARY qualifier, do *not* include the directory, file type, or version number. The system assumes that the file is in SYS\$LIBRARY and has the type TLB. If you copy a library file from another node, be sure that the new library has a unique file name.

Operations that request a particular device control library module use the module from the library specified for the queue. If you have a small configuration of printers and normally use only a few modules, you usually store all modules in a single library and assign that same library to each printer queue.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

For sites with a large number of different printers, you usually create and assign a separate device control library for each type of printer. If you have a separate device control library for each type of printer, Digital recommends you give modules that perform the same function an identical name in all libraries, even though the modules contain escape sequences unique to the specific type of printer. For example, if three libraries contained modules that set up a printer for landscape orientation, these modules might be named LANDSCAPE in all three libraries. This allows you to use the same form on any queue for which a library contains a module of the specified name, even though the modules might contain different device-specific sequences.

If you use a single library to store modules for different types of printers, make sure that each module has a unique name.

Use the following command format to display a listing of all modules contained within a specified library:

```
LIBRARY/LIST/FULL SYS$LIBRARY:library-name.TLB
```

5.2.8.5.4 Creating Forms for Setup and Page Setup Modules The recommended method of using setup or page setup modules with print jobs is through forms. Use the `/SETUP=module` or `/PAGE_SETUP=module` qualifiers with the DCL command `DEFINE/FORM` to associate modules with forms. Modules specified with the `/SETUP` qualifier are sent to the printer when the form is mounted, before each file of a job is printed. Similarly, modules specified with `/PAGE_SETUP` are sent to the printer before each page of a job.

Once you have associated a setup or page setup module with a form, users can request the module by including the `/FORM` qualifier with the `PRINT` command as follows:

```
PRINT/FORM=form-name file-name /QUEUE=queue-name
```

Users can also request a setup module by specifying the `/SETUP=module` qualifier with the `PRINT` command. However, the `PRINT/FORM` command has certain advantages over the `PRINT/SETUP` command.

Caution

When a user enters the `PRINT/FORM` command, the form name is checked for validity when the command is entered. However, when a user enters the `PRINT/SETUP` command, the module names are not checked until the job attempts to print. If a user makes an error while entering a module name, the job will not print, and the user will not be notified unless the user also specified the `/NOTIFY` qualifier.

Digital recommends you assign setup modules in forms. Once the form is properly defined, users can enter the `PRINT/FORM` command to request a setup module.

5.2.8.5.5 Assigning a Reset Module to a Queue To assign a module to an output execution queue to reset the printer to a known state at the end of each job, you use the `/SEPARATE=RESET=module` qualifier with the `INITIALIZE /QUEUE`, `START/QUEUE` or `SET QUEUE` command. For example:

```
$ INITIALIZE/QUEUE/LIBRARY=MYDEVCTL /SEPARATE=RESET=MODULE2 PDQ_QUE
```

The reset sequence contained in the module resets the printer at the end of each job. It also resets the printer when the queue is started to ensure that the first job prints correctly.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Because the /SEPARATE qualifier specifies mandatory queue options, the RESET module you specify is sent to the queue at the end of every job. The user cannot override this option.

See Example 5-1 for a sample session in which device control library modules are used to process a print job. In this session, two device control modules are created and inserted into the library file MYDEVCTL.TLB. The escape sequence or text in the setup module named MODULE1 is sent to the printer to set the printer up before REPORT.TXT is printed and again before MEMO.TXT is printed. The escape sequence or text in the reset module named MODULE2 is sent to the printer only once after both of the files in job REPORT have printed.

Example 5-1 Using Device Control Library Modules

```
$ LIBRARY/CREATE/TEXT SYSS$LIBRARY:MYDEVCTL.TLB
$ EDIT MODULE1.TXT
    !enter printer escape sequences or text for module1
$ EDIT MODULE2.TXT
    !enter printer escape sequences or text for module2
$ LIBRARY/INSERT SYSS$LIBRARY:MYDEVCTL.TLB/TEXT MODULE1
$ LIBRARY/INSERT SYSS$LIBRARY:MYDEVCTL.TLB/TEXT MODULE2
$ INITIALIZE/QUEUE/START/ON=TTA9:/LIBRARY=MYDEVCTL PDQ_QUE
$ SET QUEUE/SEPARATE=RESET=MODULE2 PDQ_QUE
$ SHOW QUEUE/FULL PDQ_QUE

Terminal queue PDQ_QUE, idle on TOAD::TTA9, mounted form DEFAULT
  /BASE PRIORITY=4 /DEFAULT=(FEED,FORM=DEFAULT) /LIBRARY=MYDEVCTL
  /OWNER=[1,4] /PROTECTION=(S:E,O:D,G:R,W:W) /SEPARATE=(RESET=(MODULE2))

$ DEFINE/FORM/SETUP=MODULE1/STOCK=DEFAULT FORM1 1
$ PRINT/FORM=FORM1 REPORT.TXT,MEMO.TXT/QUEUE=PDQ_QUE
Job REPORT (Queue PDQ_QUE, entry 619) started on PDQ_QUE
$
```

5.2.8.5.6 Order of Device Control Module Output The following list shows the order in which device control modules are sent to the printer within a print job:

1. Reset modules assigned to the queue. (Reset modules are only used at this point for the first job printed after a queue is started.)
2. Setup modules specified in the form definition.
3. Page setup modules specified in the form definition.
4. Setup modules specified with the PRINT command
5. Page 1 of file 1.
6. Page setup modules specified in the form definition.
7. Page 2 of file 1.
8. Page setup modules specified in the form definition.
9. Last page of file 1.
10. Setup modules specified in the form definition.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

11. Page setup modules specified in the form definition.
12. Setup modules specified with the PRINT command
13. Page 1 of file 2.
14. Page setup modules specified in the form definition.
15. Page 2 of file 2.
.
16. Page setup modules specified in the form definition.
17. Last page of file 2.
18. Reset modules assigned to the queue.

5.2.9 Implementing Your Planned Queue Configurations

This section describes the bolded step in the following list of steps for setting up a batch and print queuing system:

1. Plan your batch queuing environment. See Section 5.2.1.
2. Plan your output queuing environment. See Section 5.2.2.
3. Choose batch and output queue options. See Section 5.2.3.
4. **Implement the batch and output queue environment you have planned.**

Once you have planned your batch and output environments, you are ready to implement your batch and output queue configurations. To do so, perform the following steps:

- a. If your configuration includes output queues, set up output devices and create a command procedure to setup the devices on reboot. See Section 5.2.9.1.
- b. Start the queue manager and create the queue database. See Section 5.2.9.2.
- c. Define any forms and characteristics you have planned. See Section 5.2.9.3.
- d. Create and start queues and create a command procedure to start execution queues on reboot. See Section 5.2.9.4.

5.2.9.1 Setting Up Output Devices

Before creating queues and starting the queuing system, you must install any printers, plotters, and other output devices to which your users will have access. For information, see the documentation provided with the hardware.

In addition, you should perform some or all of the following tasks to set up your system to use the output devices:

1. If you will use LAT printers, set up logical ports. See Section 5.2.9.1.1.
2. Set device characteristics for printers. See Section 5.2.9.1.2.
3. Spool printers. See Section 5.2.9.1.3.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

5.2.9.1.1 Setting Up Logical Ports for LAT Printers To set up a printer attached to a LAT port, you must create a logical LAT port on each VMS service node to which the printer is to be available and associate the logical port with a physical port or service on the terminal server node. To do so, use the LATCP commands CREATE PORT and SET PORT. For more information, see the *VMS LAT Control Program (LATCP) Manual*.

SYS\$MANAGER:LAT\$SYSTARTUP.COM contains sample Sample LATCP CREATE PORT and SET PORT commands for setting up ports are contained in the site-specific LAT command procedure SYS\$MANAGER:LAT\$SYSTARTUP.COM. Edit the procedure to remove the exclamation point (!) character that precedes the appropriate commands or add similar CREATE PORT and SET PORT commands to meet your needs. For more information, see the *VMS LAT Control Program (LATCP) Manual*.

On each node that is to have access to a LAT printer, you should also set device characteristics for the printer, as explained in Section 5.2.9.1.2. Digital recommends you also spool your LAT printers, as explained in Section 5.2.9.1.3, to prevent privileged users from writing directly to the device. Writing directly to a LAT device can cause problems for output queues that use the device.

5.2.9.1.2 Setting Printer Device Characteristics You should set up device characteristics for line printers and printers attached to terminal ports. To do so, use a series of SET commands. Table 5-2 lists the commands you use to set up device characteristics for printers. For more information about these commands and their qualifiers, see the *VMS DCL Dictionary*.

Table 5-2 Commands to Set Up Output Devices

Command	Description
SET PRINTER	Sets device characteristics for a line printer.
SET TERMINAL	Sets device characteristics for an output device attached to a terminal port.

The following example provides sample commands you might enter to set up printers:

```
$ SET PRINTER/PAGE=60/WIDTH=80/LOWERCASE/TRUNCATE LPA0: ❶
$ SET PRINTER/LA11/UPPERCASE/WRAP LPB0: ❷
$ SET TERMINAL LTA3331: /SPEED=9600/PAGE=100/WIDTH=200/DEVICE=LN03 -
  /NOBROADCAST /NOECHO /HARDCOPY /NOTYPE AHEAD /PERMANENT ❸
$ SET TERMINAL LTA332: /DEVICE=LA210 /WIDTH=255 /PAGE=66 -
  /NOBROAD/PERMANENT ❹
```

- ❶ Sets characteristics for the LPA0: line printer. The /PAGE qualifier establishes the length of the output page as 60 lines. The /WIDTH qualifier establishes the width of the output page as 80 characters. † /LOWERCASE enables lowercase printing, and /TRUNCATE causes the printer to truncate data if a line exceeds the width specified with the /WIDTH qualifier.

† The print symbiont first uses the form to determine the width of a line. Once the print symbiont has finished formatting the data, if the width of the line exceeds the /WIDTH setting for the device, the device driver will use the /TRUNCATE or /WRAP settings (if set) to truncate or wrap the line. For more information, see Section 5.2.8.4.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

- ② Sets characteristics for the LPB0: line printer. The /LA11 qualifier establishes it as an LA11 printer. /UPPERCASE disables lowercase printing. /WRAP causes the printer to perform a carriage return or line feed whenever the line width (as specified by the /WIDTH qualifier) is reached. The rest of the record will continue on the next line.†
- ③ Sets up the LN03 printer attached to the LTA3331 LAT port. The qualifiers set the following characteristics for the printer:

Qualifier	Description
/SPEED	Sets the input and output baud rates to 9600.
/PAGE	Sets page length as 100 lines.
/WIDTH	Sets the line width as 200 characters.†
/NOBROADCAST	Disables the reception of broadcast messages to prevent interruption of output (because the terminal port is being used for a non-interactive device).
/NOECHO	Causes the printer to print only system or user application output, or both.
/HARDCOPY	Establishes the terminal as a hardcopy device.
/NOTYPE_AHEAD	Disables the type ahead buffer so that the device is dedicated.
/PERMANENT	Sets the characteristics as permanent until the system is halted and restarted.

†The print symbiont first uses the form to determine the width of a line. Once the print symbiont has finished formatting the data, if the width of the line exceeds the /WIDTH setting for the device, the device driver will use the /TRUNCATE or /WRAP settings (if set) to truncate or wrap the line. For more information, see Section 5.2.8.4.

- ④ Sets characteristics for the LA210 printer attached to the LTA3332 LAT port. Later, you will create a command procedure to set up your device characteristics each time the system reboots. The commands you enter now must be included in the command procedure. (This step is described in Section 5.2.9.1.4.)

For more information about the SET PRINTER and SET TERMINAL commands and their qualifiers, see the *VMS DCL Dictionary*.

5.2.9.1.3 Spooling Output Devices To establish an output device as a spooled device, use the SET DEVICE/SPOOLED command. This command associates the output device with a storage device (such as a disk) and an output queue. (For an explanation of the use of spooled output devices, see Section 5.1.8.)

You must set up a spooled output device before the queue to be associated with the device is started.

Enter the DCL command SET DEVICE/SPOOLED in the following format:

```
SET DEVICE/SPOOLED[=(queue-name[:],intermediate-disk-name[:])] output-device-name
```

You should always specify the intermediate disk and queue explicitly. If the queue you associate with the spooled output device is a generic queue, a file written to that device is sent to the generic queue, which in turn places the job in one of its target queues. As a result, a job copied to LPA0, for example, might not necessarily print on the printer LPA0 but, instead, might print on one of the other printers targetted by the generic queue.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

When you select an intermediate storage device, make sure that it has sufficient free space for the volume of spooled output. If you plan to enforce disk quotas on the intermediate device, make sure that all expected users have a quota authorized on the intermediate device. The intermediate device must be mounted before files can be written to it.

The following example illustrates sample commands used to set up spooled output devices:

```
$ SET DEVICE/SPOOLED=(SYS$PRINT,SYS$SYSDEVICE) LPA0: ❶
$ SET DEVICE/SPOOLED=(MKTG$LN03_1,SYS$SYSDEVICE) LTA3331: ❷
$ SET DEVICE/SPOOLED=(LA210$PRINT,SYS$SYSDEVICE:) LTA3332: ❸
```

- ❶ Spools the output device LPA0: by associating it with the storage device SYS\$SYSDEVICE and the queue SYS\$PRINT. When output from an application is directed to LPA0:, the data is temporarily stored on SYS\$SYSDEVICE until the application completes. This keeps the output device LPA0: available for other jobs until the application's output is ready for printing. When the application does complete, its output is submitted to the queue SYS\$PRINT.
- ❷ Spools the LN03 device on LAT port LTA3331: by associating it with the storage device SYS\$SYSDEVICE and the queue MKTG\$LN03_1. Digital recommends you spool LAT printers.
- ❸ Spools the LA210 device on LAT port LTA3332: by associating it with the storage device SYS\$SYSDEVICE and the queue LA210\$PRINT.

Later, you will create a command procedure to set up your devices each time the system reboots. The commands you enter to set up spooled devices must be included in this command procedure. (This step is described in Section 5.2.9.1.4.)

If you need to disable spooling to an output device, use the SET DEVICE command with the /NOSPOOLED qualifier. For example, the SET TERMINAL command can be executed only on a despoiled output device. You must stop the corresponding queues before you can change the spooling status.

For more information about the SET DEVICE/[NO]SPOOLED command, see the *VMS DCL Dictionary*.

Testing a Spooled Device

After establishing an output device as spooled, you should test the device because errors in disk or queue names are not detected until spooling is attempted. To test a spooled device, use a command procedure similar to the following:

```
$!          *****TESTING SPOOLED DEVICE***
$!
$! set the device spooled
$ SET DEVICE/SPOOLED=(SYS$PRINT,SYS$SYSDEVICE:) LPA0:
$!
$! create a test file
$ CREATE TEST.LIS
    !Add the first test record here.
    !Ctrl/Z to exit the file
$!
$! write the file to the output device
$ COPY TEST.LIS LPA0:
$ EXIT
```

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

5.2.9.1.4 Creating a Command Procedure to Setup Devices on Reboot You must set up an output device each time the system on which the device is located reboots. To make sure this happens, create a command procedure on each node to execute the device setup commands explained in Sections 5.2.9.1.2 and 5.2.9.1.3. (The commands used to set up logical ports for LAT printers, explained in Section 5.2.9.1.1, should be added to your site-specific LAT startup command procedure `SY$MANAGER:LAT$SYSTARTUP.COM`.)

If your configuration is simple, you can add the commands to `SYSTARTUP_V5.COM`. If your configuration requires a large number of commands, create a separate command procedure (for example, `DEVICE_SETUP.COM`) and execute it from `SYSTARTUP_V5.COM`. When the device setup command procedure finishes executing, control returns to `SYSTARTUP_V5.COM`.

In the command procedure, a `SET TERMINAL` command must precede a `SET DEVICE/SPOOLED` command for the same output device.

The following example provides a sample `DEVICE_SETUP.COM` you might call from `SYSTARTUP_V5.COM` to set up printers:

```
$! Set up line printer devices
$!
$ SET PRINTER/PAGE=60/WIDTH=80/LOWERCASE/TRUNCATE LPA0:
$ SET PRINTER/LA11/UPPERCASE/WRAP LPB0:
$!
$! Set up terminal devices
$!
$ SET TERMINAL LTA3331: /SPEED=9600/PAGE=100/WIDTH=200/DEVICE=LN03
    /NOBROADCAST /NOECHO /HARDCOPY /NOTYPE_AHEAD /PERMANENT
$ SET TERMINAL LTA3332: /DEVICE=LA210 /WIDTH=255 /PAGE=66
    /NOBROAD/PERMANENT
$!
$! Spool output devices
$!
$ SET DEVICE/SPOOLED=(SY$PRINT,SY$SYSDEVICE) LPA0:
$ SET DEVICE/SPOOLED=(SY$PRINT,SY$SYSDEVICE) LPB0:
$ SET DEVICE/SPOOLED=(MKTG$LN03_1,SY$SYSDEVICE) LTA3331:
$ SET DEVICE/SPOOLED=(LA210$PRINT,SY$SYSDEVICE:) LTA3332:
```

5.2.9.2 Starting the Queue Manager and Creating the Queue Database

Before you can create any queues, you must create the queue database. The `START/QUEUE/MANAGER/NEW_VERSION` command starts the queue manager and creates the queue database. You normally need to perform this task only once.

To start the queue manager and create the queue database, perform the following steps:

1. Determine the location where you want to store the master file and queue and journal files of the queue database. By default, all three files are created in `SY$COMMON:[SY$EXE]`. However, you can create the files in other locations. The master file can be kept in a different location than the queue and journal files, but the queue and journal files must be kept together.
2. Make sure the disk or disks that are to hold the queue database files are mounted.
3. Make sure your account has `OPER` privilege.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

4. If you want to create your master file in a location other than the default, enter a command in the following format:

```
DEFINE/SYSTEM/EXECUTIVE_MODE QMAN$MASTER location
```

Specify the device and directory where the master file is to be created for *location*.

If you are performing this task in a VAXcluster environment, enter this command on every node in the cluster.

Be sure to add this command to the SYLOGICALS.COM startup command procedure. In a VAXcluster, add this command to the SYLOGICALS.COM command procedure on every node in the cluster.

5. In a VAXcluster environment, determine the preferred order in which nodes should claim the queue manager process.
6. Enter a command in the following format:

```
START/QUEUE/MANAGER/NEW_VERSION[/ON=(node-list)] [dirspec]
```

This command starts the queue manager process for a system or cluster and creates the queue database files.

Caution

Specify the /NEW_VERSION qualifier only if you want to create a new queue database.

The /ON qualifier specifies a list of VAXcluster nodes that can run the queue manager, in the preferred order in which they should claim the process. By default, all nodes are able to run the queue manager, in no specified order. Digital recommends you specify the asterisk character (*) at the end of the node list to make sure that at least one node is always available to run the queue manager. For more information, see the /ON qualifier for the START/QUEUE/MANAGER command in DCL help.

The *dirspec* parameter is the location where the queue and journal files are to be created. You need include this parameter only if you want to create the queue and journal files in a location other than the default. For more information, see the START/QUEUE/MANAGER command in DCL help.

Once you enter the START/QUEUE/MANAGER command, it is stored, along with any qualifier or parameter you entered, in the queue database. Thereafter, the job controller automatically starts the queue manager during reboot unless a STOP/QUEUE/MANAGER/CLUSTER command has been entered. For this reason, the START/QUEUE/MANAGER command does not need to be included in your startup command procedure.

For more information about managing the queue manager and queue database, see Section 5.3.1.

5.2.9.3 Defining Forms and Characteristics You Have Planned

Before creating queues, you must define any forms or characteristics you will associate with them. To define a form, use the DCL command DEFINE/FORM as explained in Section 5.2.8.3.2. To define a characteristic, use the DCL command DEFINE/CHARACTERISTIC as explained in Section 5.2.6.1.

Example 5-2 includes sample commands used to create forms and characteristics.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

5.2.9.4 Creating and Starting Queues

Before users can submit jobs, you must create queues. Before queues can begin processing jobs, you must start the queues. Section 5.2.9.4.1 and Section 5.2.9.4.2 provide information you need to create execution and generic queues. Section 5.2.9.4.3 provides a procedure of steps for creating and starting your queues.

5.2.9.4.1 Creating Execution Queues Before you create an execution queue, you should decide whether it will be an autostart queue. This section provides information about creating autostart and nonautostart execution queues.

Autostart Queues

To create an autostart queue, specify the /AUTOSTART_ON qualifier with the INITIALIZE/QUEUE command as follows.

For output queues:

```
INITIALIZE/QUEUE[/START]/AUTOSTART_ON=node::device queue-name
```

For *node*, specify the name of the node on which the queue is to run. For *device*, specify the name of the output device to which the queue's output is sent. The /START qualifier activates the queue for autostart.

For batch queues:

```
INITIALIZE/QUEUE/BATCH[/START]/AUTOSTART_ON=node:: queue-name
```

/BATCH is required for batch queues; *node* is the name of the node on which the queue is to run. The /START qualifier activates the queue for autostart.

For more information about this command, see DCL Help.

Caution

The node name you specify as *node* is not checked to determine if it is an existing node name. So, be sure to correctly specify the node name.

To increase the availability of execution queues in a cluster, you can set up an autostart queue to execute on one of several nodes (for batch queues) or nodes and devices (for output queues) in a list. If the node on which an autostart queue is running leaves the cluster, the queue will automatically fail over to the next available node in the list on which autostart is enabled. For information about enabling autostart, see Section 5.3.2.5.

To specify the list of nodes (or nodes and devices) to which an autostart queue can fail over, include the list with the /AUTOSTART_ON qualifier as follows:

```
INITIALIZE/QUEUE[/BATCH][[/START]/AUTOSTART_ON=(node::[device] [...]) queue
```

For example:

```
$ INITIALIZE/QUEUE/BATCH START -  
_ $ /AUTOSTART_ON=(DATE::, XERF::, DEANNA::) ENTERPRISE_QUEUE
```

For more information, see the /AUTOSTART_ON qualifier for the INITIALIZE/QUEUE command in DCL Help.

You must include the ENABLE AUTOSTART/QUEUES command in the startup procedure for each node that can run the queue to start autostart queues when the node reboots, as shown in Example 5-3.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Nonautostart Queues

To create a nonautostart execution queue, specify the /ON qualifier with the INITIALIZE/QUEUE command as follows:

For output queues:

```
INITIALIZE/QUEUE[/START]/ON=node::device queue-name
```

For *node::*, specify the node on which the queue is to execute. For *device*, specify the device to which the queue's output is sent. The /START qualifier starts the queue.

For batch queues:

```
INITIALIZE/QUEUE/BATCH[/START]/ON=node:: queue-name
```

The /BATCH qualifier is required to create a batch queue; *node::* is the node on which the queue is to execute. The /START qualifier starts the queue.

For more information, see the INITIALIZE/QUEUE command in DCL Help.

For each nonautostart execution queue you create on a node, you must include a START/QUEUE command naming the queue in the node's startup command procedure, as shown in Example 5-3, to start the queue when the node reboots.

5.2.9.4.2 Creating Generic Queues You create a generic queue by specifying the /GENERIC qualifier with the INITIALIZE/QUEUE or START/QUEUE command.

You explicitly name the execution queues to which the generic queue sends jobs by including a list of execution queues with the /GENERIC qualifier as follows:

```
INITIALIZE/QUEUE[/START]/GENERIC=(execution-queue-list) generic-queue-name
```

For example:

```
$ INITIALIZE/QUEUE/START/GENERIC=(LPA0,LPB0,LPC0) LINE_PRINT
```

The /START qualifier starts the queue.

You can also set up a generic queue without explicitly naming the execution queues to which it may send jobs. Instead, you enable certain execution queues to receive jobs from generic queues by specifying the /ENABLE_GENERIC qualifier with the INITIALIZE/QUEUE, START/QUEUE or SET QUEUE command for the execution queue. This method is not normally recommended. However, if your queue configuration is simple, you can use this method. For more information, see the *VMS DCL Dictionary*.

Generic queues are not stopped when a node is shut down. For this reason, you do not need to include commands to start generic queues in your startup command procedure.

5.2.9.4.3 Procedure for Creating and Starting Queues Create and start your queues as follows:

1. Make sure your account has the OPER privilege.
2. If you are setting up output queues, make sure you have set up the hardware and software for the output devices as described in Section 5.2.9.1.
3. Make sure you have defined any forms and characteristics you will use with your queues, as explained in Section 5.2.8.3.2 and Section 5.2.6.1.
4. Create each queue you have planned as explained in Section 5.2.9.4.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Create execution queues before you create generic queues. With each INITIALIZE/QUEUE command, specify the qualifiers you chose to customize the queues as explained in Section 5.2.1 and Section 5.2.2. Table 5-1 lists the qualifiers you can specify for batch and output queues. The commands in Example 5-2 include sample commands to create and start queues.

5. For each node on which you want autostart queues to run (including those to which the queues can later fail over), you must enter the ENABLE AUTOSTART/QUEUES command. This command starts all stopped active autostart queues on a node and starts any active autostart queue that fails over to the node from another node in the cluster.

Later, you should add the ENABLE AUTOSTART/QUEUES command to your startup command procedure on each node that is to run autostart queues to ensure that autostart is enabled each time the node reboots. This step is described in Section 5.2.9.4.5.

For information about creating batch and output queues for a VAXcluster environment, see the *VMS VAXcluster Manual*.

5.2.9.4.4 Sample Queue Configuration Setup Example 5-2 includes commands to set up a queue configuration with the following components on a standalone system named LILITH:

- Two batch queues:
 - A default, nonautostart batch execution queue named SYS\$BATCH.
 - A special autostart batch execution queue named FAST_BATCH. This queue is designed to execute high priority jobs.
- Three line printers LPA0:, LPB0:, and LPC0: with the following queues:
 - Three autostart output execution queues LPA0, LPB0, and LPC0
 - A generic output queue named SYS\$PRINT.
- Four LN03 printers on the LAT ports LTA3331:, LTA3332: LTA3333:, and LTA3334: with the following queues:
 - Four autostart output execution queues, LN03_1, LN03_2, LN03_3, and LN03_4
 - A generic output queue named LN03_PRINT.
- Four characteristics, EAST, WEST, NORTH, SOUTH, to be used with the LN03 printers.
- Five forms: DEFAULT, LPA0, MEMO, LN03_LANDSCAPE, and LN03_LANDSCAPE_INDENTED.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Example 5-2 Sample Commands for Setting Up Queue Configuration

```
①
$ SET PRINTER /TAB/PAGE=66 /WIDTH=132 /LOWER /FF /NOCR -
$ /FALLBACK /NOWRAP /NOTAB LPA0:
$ SET PRINTER /TAB/PAGE=66 /WIDTH=132 /LOWER /FF /NOCR -
$ /FALLBACK /NOWRAP /NOTAB LPB0:
$ SET PRINTER /TAB/PAGE=66 /WIDTH=132 /LOWER /FF /NOCR -
$ /FALLBACK /NOWRAP /NOTAB LPC0:

②
$ SET TERMINAL/SPEED=9600/PAGE=100/WIDTH=200/DEVICE=LN03/NOBROADCAST -
$ /NOECHO /HARDCOPY /NOTYPE_AHEAD /NOFORM /NOWRAP /PASTHRU /PERMANENT LTA3331:
$ SET TERMINAL/SPEED=9600/PAGE=100/WIDTH=200/DEVICE=LN03/NOBROADCAST -
$ /NOECHO /HARDCOPY /NOTYPE_AHEAD /NOFORM /NOWRAP /PASTHRU /PERMANENT LTA3332:
$ SET TERMINAL/SPEED=9600/PAGE=100/WIDTH=200/DEVICE=LN03/NOBROADCAST -
$ /NOECHO /HARDCOPY /NOTYPE_AHEAD /NOFORM /NOWRAP /PASTHRU /PERMANENT LTA3333:
$ SET TERMINAL/SPEED=9600/PAGE=100/WIDTH=200/DEVICE=LN03/NOBROADCAST -
$ /NOECHO /HARDCOPY /NOTYPE_AHEAD /NOFORM /NOWRAP /PASTHRU /PERMANENT LTA3334:

③
$ SET DEVICE /SPOOLED=(LPA0,SYSSSYSDEVICE) LPA0:
$ SET DEVICE /SPOOLED=(LPB0,SYSSSYSDEVICE) LPB0:
$ SET DEVICE /SPOOLED=(LPC0,SYSSSYSDEVICE) LPC0:
$ SET DEVICE /SPOOLED=(LN03_1,SYSSSYSDEVICE) LTA3331:
$ SET DEVICE /SPOOLED=(LN03_2,SYSSSYSDEVICE) LTA3332:
$ SET DEVICE /SPOOLED=(LN03_3,SYSSSYSDEVICE) LTA3333:
$ SET DEVICE /SPOOLED=(LN03_4,SYSSSYSDEVICE) LTA3334:

④
$ START/QUEUE/MANAGER/NEW_VERSION

⑤
$ INITIALIZE/QUEUE/BATCH/START/JOB_LIMIT=6 /BASE_PRIORITY=4 -
$ /ON=LILITH:: SYSSBATCH

⑥
$ INITIALIZE/QUEUE /BATCH/START/JOB_LIMIT=1 /BASE_PRIORITY=5 -
$ /NODISABLE_SWAPPING -
$ /AUTOSTART_ON=LILITH:: FAST_BATCH

⑦
$ DEFINE/FORM DEFAULT 0 -
$ /STOCK=DEFAULT/DESCRIPTION="DEFAULT"
$ /LENGTH=66 /MARGIN=(BOTTOM=6) /TRUNCATE /WIDTH=132
$
$ DEFINE/FORM LPA0 4 -
$ /STOCK=DEFAULT/DESCRIPTION="LINE PRINTER" -
$ /LENGTH=66 /MARGIN=(BOTTOM=6) /TRUNCATE /WIDTH=132
$
$ DEFINE /FORM MEMO 110 -
$ /DESCRIPTION="LN03 INDENTED MEMO FORMAT" -
$ /WIDTH=80/TRUNCATE/LENGTH=66/STOCK=DEFAULT -
$ /MARGIN=(TOP=2,LEFT=5,BOTTOM=2)
$
$ DEFINE /FORM LN03SLANDSCAPE 111 -
$ /STOCK=DEFAULT /DESCRIPTION="132 BY 66 (LANDSCAPE)" -
$ /NOWRAP /NOTRUNCATE /WIDTH=132 /LENGTH=66 -
$ /SETUP=LANDSCAPE /MARGIN=(BOTTOM=0)
$
$ DEFINE/FORM LNC3SLANDSCAPE INDENTED 112 -
$ /STOCK=DEFAULT/DESCRIPTION="132 BY 55 (LANDSCAPE INDENTED)" -
$ /NOWRAP/NOTRUNCATE/WIDTH=132 /LENGTH=66 -
$ /SETUP=LANDSCAPE /MARGIN=(TOP=5,BOTTOM=6)
```

(continued on next page)

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

Example 5-2 (Cont.) Sample Commands for Setting Up Queue Configuration

```

8 $ DEFINE/CHARACTERISTIC EAST 1
  $ DEFINE/CHARACTERISTIC WEST 2
  $ DEFINE/CHARACTERISTIC NORTH 3
  $ DEFINE/CHARACTERISTIC SOUTH 4
9 $ SINITIALIZE /QUEUE /START /DEFAULT=(NOBURST, FLAG=ALL, TRAILER=ONE) -
  $ /AUTOSTART_ON=LILITH::LPA0: LPA0
  $ SINITIALIZE /QUEUE /START /DEFAULT=(NOBURST, FLAG=ALL, TRAILER=ONE) -
  $ /AUTOSTART_ON=LILITH::LPB0: LPB0
  $ SINITIALIZE /QUEUE /START /DEFAULT=(NOBURST, FLAG=ALL, TRAILER=ONE) -
  $ /AUTOSTART_ON=LILITH::LPC0: LPC0
10 $ INITIALIZE /QUEUE /START -
  $ /GENERIC=(LPA0:,LPB0:,LPC0:) SYSS$PRINT
11 $ INITIALIZE /QUEUE /START /DEVICE=TERMINAL /AUTOSTART_ON=LILITH::LTA3331: -
  $ /RECORD_BLOCKING /BLOCK_LIMIT=600 /CHARACTERISTICS=(EAST) -
  $ /SEPARATE=(NOBURST,NOTRAILER,NOFLAG,RESET=ANSI$RESET) -
  $ /DEFAULT=(NOFEED,NOBURST,FLAG=ONE,NOTRAILER,FORM=MEMO) -
  $ /LIBRARY=LN03LIBRARY /PROCESSOR=LATSYM LN03_1
  $ INITIALIZE /QUEUE /START /DEVICE=TERMINAL /AUTOSTART_ON=LILITH::LTA3332: -
  $ /RECORD_BLOCKING /BLOCK_LIMIT=600 /CHARACTERISTICS=(WEST) -
  $ /SEPARATE=(NOBURST,NOTRAILER,NOFLAG,RESET=ANSI$RESET) -
  $ /DEFAULT=(NOFEED,NOBURST,FLAG=ONE,NOTRAILER,FORM=MEMO) -
  $ /LIBRARY=LN03LIBRARY /PROCESSOR=LATSYM LN03_2
  $ INITIALIZE /QUEUE /START /DEVICE=TERMINAL /AUTOSTART_ON=LILITH::LTA3333: -
  $ /RECORD_BLOCKING /BLOCK_LIMIT=600 /CHARACTERISTICS=(NORTH) -
  $ /SEPARATE=(NOBURST,NOTRAILER,NOFLAG,RESET=ANSI$RESET) -
  $ /DEFAULT=(NOFEED,NOBURST,FLAG=ONE,NOTRAILER,FORM=MEMO) -
  $ /LIBRARY=LN03LIBRARY /PROCESSOR=LATSYM LN03_3
  $ INITIALIZE /QUEUE /START /DEVICE=TERMINAL /AUTOSTART_ON=LILITH::LTA3334: -
  $ /RECORD_BLOCKING /BLOCK_LIMIT=600 /CHARACTERISTICS=(SOUTH) -
  $ /SEPARATE=(NOBURST,NOTRAILER,NOFLAG,RESET=ANSI$RESET) -
  $ /DEFAULT=(NOFEED,NOBURST,FLAG=ONE,NOTRAILER,FORM=MEMO) -
  $ /LIBRARY=LN03LIBRARY /PROCESSOR=LATSYM LN03_4
12 $ INITIALIZE /QUEUE /START -
  $ /GENERIC=(LN03_1, LN03_2, LN03_3, LN03_4) LN03_PRINT
13 $ ENABLE AUTOSTART/QUEUES

```

- 1 Set up line printer devices.
- 2 Set up LN03 devices.
- 3 Spool all output devices.
- 4 Create the queue database and start the queue manager.
- 5 Initialize and start the default batch queue (nonautostart) named SYSS\$BATCH.
- 6 Initialize and start the special batch queue (autostart) named FAST_BATCH.
- 7 Define forms.
- 8 Define characteristics.
- 9 Initialize and start the autostart execution queues for line printers.

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

- ⑩ Initialize and start the generic queue for line printers.
- ⑪ Initialize and start the autostart execution queues for LN03 printers.
- ⑫ Initialize and start the generic queue for LN03 printers.
- ⑬ Enable autostart to start all autostart queues on node LILITH.

For more information about these commands, see the *VMS DCL Dictionary*.

5.2.9.4.5 Creating a Command Procedure to Restart Execution Queues

on Reboot Once you define forms and characteristics and create queues, information about the forms, characteristics, and queues is stored in the queue database. You do not need to perform these tasks each time the node or cluster reboots.

However, on each node, you must start certain execution queues each time the node reboots. To do so, create a command procedure. If your configuration is simple, you can add the commands to SYSTARTUP_V5.COM. If your configuration requires a large number of commands, create a separate command procedure and execute it from SYSTARTUP_V5.COM. Example 5-3 illustrates a sample queue configuration command procedure that might be executed as part of a node's SYSTARTUP_V5.COM procedure.

Generic queues are not automatically stopped when a node is shut down. Therefore, you do not need to include commands to start generic queues in your startup command procedure.

Starting Nonautostart Queues

For each queue that is not an autostart queue, you must include a START /QUEUE command naming the queue in your startup command procedure.

Starting Autostart Queues

You can start all stopped active autostart queues on a node with the single command ENABLE AUTOSTART/QUEUES. You do not need to include a separate START/QUEUE command in order to start an active autostart queue.

When a node reboots, autostart is disabled until you enter the ENABLE AUTOSTART/QUEUES command. Add the ENABLE AUTOSTART/QUEUES command to your startup command procedure following the commands that configure printer devices and mount important disks. Digital recommends you add the ENABLE AUTOSTART/QUEUES command to your startup procedure on all of your nodes. This eliminates the need for you to add the command later, if you need to add autostart queues or add nodes to autostart queue failover lists.

Example 5-3 provides sample commands you might include in SYSTARTUP_V5 to restart the queues created in Example 5-2.

Example 5-3 Sample Queue Startup Commands

```
$! Start the nonautostart batch queue
$ START/QUEUE SYSSBATCH
$! Start all autostart queues
$ ENABLE AUTOSTART/QUEUES
```

Batch and Print Operations

5.2 Setting Up the Batch and Print Queuing System

5.2.9.5 Maintaining Your Queue Configuration

Setting up queues is not restricted to startup time. During normal operation, you can enter the commands to create and start queues as needs dictate.

If you create additional output queues at a later time, make sure you add commands to set device characteristics to the startup command procedure on the node on which the device is located. For devices attached to a LAT port, add commands to the startup procedures on all nodes with queues for the device.

If you create any additional nonautostart queues at a later time, you must also add appropriate START/QUEUE commands to the startup command procedure on the node on which the queue will run. If you do not add appropriate START/QUEUE commands, the queues will not be started when the system reboots.

5.3 Maintaining the Queuing System

The following sections provide information about maintaining your batch and print queuing system. For information about setting up your queuing system, see Section 5.2.

Once your queuing system is set up, you might need to perform the following types of maintenance on your queuing system:

Maintenance Task	Section
Managing the queue manager and queue database	See Section 5.3.1.
Managing queues	See Section 5.3.2.
Managing jobs in queues	See Section 5.3.3.
Troubleshooting	See Section 5.3.4.

5.3.1 Maintaining the Queue Manager and Queue Database

The following sections describe tasks for maintaining the queue manager and queue database:

Task	Section
Starting the queue manager	See Section 5.3.1.1.
Stopping the queue manager	See Section 5.3.1.2.
Saving the queue database	See Section 5.3.1.3.
Restoring the queue database	See Section 5.3.1.4.
Moving the master file of the queue database	See Section 5.3.1.5.1.
Moving the queue and journal files of the queue database	See Section 5.3.1.5.2.
Submitting an Software Performance Report (SPR) for problems related to the queuing system	See Section 5.3.1.6.

5.3.1.1 Starting the Queue Manager

The START/QUEUE/MANAGER command starts the queue manager. The command has the following format:

```
START/QUEUE/MANAGER[/ON=(node-list)] [dirspec]
```

For more information about the command, see the DCL Help for the command.

Batch and Print Operations

5.3 Maintaining the Queuing System

Section 5.2.9.2 explains how to start your queue manager the first time. After you initially enter the `START/QUEUE/MANAGER` command with the `/NEW_VERSION` qualifier, you do not normally need to enter the `START/QUEUE/MANAGER` command again. The queue manager is automatically restarted, unless you enter the `STOP/QUEUE/MANAGER/CLUSTER` command.

Because the command you enter to start the queue manager is stored in the queue database, if you restart the queue manager, you need to specify the `/ON=(node-list)` qualifier or the `dirspec` parameter only if you want to change the value you specified for the qualifier or parameter when you started the queue manager previously. Otherwise, the queue manager is started using the node-list and location (if any) stored in the queue database.

After starting the queue manager initially, you might need to re-enter the `START/QUEUE/MANAGER` command for one of the following reasons:

- If the `STOP/QUEUE/MANAGER/CLUSTER` command has been executed, enter the `START/QUEUE/MANAGER` command to restart the queue manager.
- In a VAXcluster environment, enter the `START/QUEUE/MANAGER` command with the `/ON` qualifier to modify the list of preferred nodes on which the queue manager can run.
- In a VAXcluster, enter the `START/QUEUE/MANAGER` command to ensure that the queue manager process is executing on the most preferred, available node. If the queue manager is not running on the most preferred, available node, the queue manager will be moved to that node without interruption of service. If you are using the default node list (*), the queue manager will not move. For more information, see the description of the `/ON` qualifier.

If the queue manager does not start, see Section 5.3.4.1 for troubleshooting information.

5.3.1.2 Stopping the Queue Manager

To stop the queue manager, enter the DCL command `STOP/QUEUE/MANAGER/CLUSTER`. This command shuts down the queue manager on a standalone node or cluster. The queue manager performs the following tasks:

- Aborts all current jobs that cannot be restarted and requeues all current restartable jobs
- Stops all execution queues and disables autostart on all nodes
- Closes all queue database files

Once the `STOP/QUEUE/MANAGER/CLUSTER` command is entered, the queue manager process remains stopped, and requests to the queuing system are denied until you restart the queue manager by entering `START/QUEUE/MANAGER` command as explained in Section 5.3.1.1. Cluster transitions do not change the state of the queue manager. Newly available nodes do not attempt to start the queue manager (unless the `START/QUEUE/MANAGER` command is entered.)

The `/CLUSTER` qualifier is required to stop the clusterwide queue manager. If you enter the obsolete command `STOP/QUEUE/MANAGER` (without the `/CLUSTER` qualifier), the command performs the same function as the DCL command `STOP/QUEUES/ON_NODE`.

To stop all queues on a single node without stopping the queue manager, enter the `STOP/QUEUES/ON_NODE` command.

Batch and Print Operations

5.3 Maintaining the Queuing System

5.3.1.3 Saving the Queue Database

To save a record-by-record copy of the queue database *while the queuing system is functioning*, perform the following steps:

1. Enter the following VMS Convert Utility (CONVERT) command to save the master file:

```
CONVERT/SHARE QMAN$MASTER.DAT master-filename
```

For *master-filename*, specify the name of the file to which QMAN\$MASTER.DAT is to be copied.

2. Enter the following CONVERT command to save the queue file:

```
CONVERT/SHARE SYS$QUEUE_MANAGER.QMAN$QUEUES queue-filename
```

For *queue-filename*, specify the name of the file to which SYS\$QUEUE_MANAGER.QMAN\$QUEUES is to be copied.

3. Use the VMS Backup Utility (BACKUP) to save the files created with the CONVERT. You might use a command similar to the following:

```
BACKUP/LOG masterfile-name.queuefile-name device:saveset-name/LABEL=label
```

For example:

```
$ SET DEFAULT SYS$COMMON:[SYSEXE]
$ CONVERT/SHARE QMAN$MASTER.DAT MASTERFILE_9SEP.KEEP;
$ CONVERT/SHARE SYS$QUEUE_MANAGER.QMAN$QUEUES QFILE_9SEP.KEEP;
$ INITIALIZE MUA0: QDB
$ MOUNT/FOREIGN MUA0:
%MOUNT-I-MOUNTED, QDB mounted on _LILITH$MUA0:
$ BACKUP/LOG MASTERFILE_9SEP.KEEP,QFILE_9SEP.KEEP MUA0:QDB_9SEP.SAV/LABEL=QDB
%BACKUP-S-COPIED, copied SYS$COMMON:[SYSEXE]MASTERFILE_9SEP.KEEP;
%BACKUP-S-COPIED, copied SYS$COMMON:[SYSEXE]QFILE_9SEP.KEEP;
$ DISMOUNT MUA0:
```

This procedure saves definitions of queues, forms, and characteristics. No job information is preserved.

You should save a copy of the queue database each time you want to preserve changes to your queue configuration. If, for some reason, your queue database files are not accessible, you can restore the queue database so you do not need to redefine forms and characteristics and reinitialize each queue.

For more information about the VMS Convert Utility, see *VMS Convert and Convert/Reclaim Utility Manual*. For more information about the Backup Utility, see the *VMS Backup Utility Manual*.

5.3.1.4 Restoring the Queue Database

If, for some reason, you lose your the queue database, you can restore your queue configuration from copies you saved as explained in Section 5.3.1.3. All queue, form, characteristic and queue manger information will be restored. Information about jobs in the queues will not be restored.

To restore previously saved queue database files, perform the following steps:

1. If the queue manager is running, stop it by entering the DCL command STOP /QUEUE/MANAGER/CLUSTER.
2. Delete all three queue database files. You must delete all three files, even if only one or two of them is lost.

Batch and Print Operations

5.3 Maintaining the Queuing System

3. Use the VMS Backup Utility (BACKUP) to restore queue file and master file from the saveset you created in Step 3 of Section 5.3.1.3. If the master file or queue file is stored in a location other than the default, make sure you restore them to the correct location or specify the new location when starting the queue manager.

Note

When you restore your queue database, you must always restore both the master and queue file of the database, even if you lost only one of those files.

4. Start the queue manager with the START/QUEUE/MANAGER command as you would normally. Do not enter the /NEW_VERSION qualifier. A new (empty) journal file will automatically be created.

For example:

```
$ STOP/QUEUE/MANAGER/CLUSTER
$ SET DEFAULT SYSSCOMMON:[SYSEXE]
$ DELETE SYSSQUEUE_MANAGER.QMAN$JOURNAL;,,SYSSQUEUE_MANAGER.QMAN$QUEUES;,-
_$ QMAN$MASTER.DAT;
$ MOUNT/FOREIGN MUAC:
%MOUNT-I-MOUNTED, QDB mounted on LILITH$MUA0:
$ BACKUP/LOG MUAC:QDB_9SEP.SAV/SELECT=[SYSEXE]MASTERFILE_9SEP.KEEP; -
_$ QMAN$MASTER.DAT;
%BACKUP-S-CREATED, created SYSSCOMMON:[SYSEXE]QMAN$MASTER.DAT;1
$ SET MAGTAPE/REWIND MUAC:
$ BACKUP/LOG MUAC:QDB_9SEP.SAV/SELECT=[SYSEXE]QFILE_9SEP.KEEP; -
_$ SYSSQUEUE_MANAGER.QMAN$QUEUES
%BACKUP-S-CREATED, created SYSSCOMMON:[SYSEXE]SYSSQUEUE_MANAGER.QMAN$QUEUES;1
$ DISMOUNT MUAC:
$ START/QUEUE/MANAGER
```

5.3.1.5 Moving Queue Database Files

By default, the queue database files are created in the default location of SYSSCOMMON:[SYSEXE] when you start the queue manager with the /NEW_VERSION qualifier. The following sections provide instructions for moving the files from their default location.

5.3.1.5.1 Moving the Master File To move the master file from the default location, perform the following steps:

1. Shut down the queue manager by entering the DCL command STOP/QUEUE/MANAGER/CLUSTER.
2. Copy the file QMAN\$MASTER.DAT to its new location.
3. Define the logical name QMAN\$MASTER by entering the following command:

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE QMAN$MASTER location
```

In a VAXcluster environment, use the System Management Utility (SYSMAN) to execute the command on all nodes in the cluster.

4. To make sure the logical name QMAN\$MASTER is defined each time the system reboots, edit the file SYSSCOMMON:[SYSMGR]SYLOGICALS.COM to add the command you entered in step 3. In a VAXcluster environment, add the command to the SYLOGICALS.COM procedure on every node in the cluster. In a VAXcluster environment, QMAN\$MASTER must be identically defined on all nodes in the cluster.

Batch and Print Operations

5.3 Maintaining the Queuing System

5. Restart the queue manager by entering the DCL command `START/QUEUE/MANAGER`.

5.3.1.5.2 Moving the Queue and Journal Files The queue and journal files are not required to reside in the same directory as the master file; however, if you move the queue and journal files, they must reside together in the same directory. The master file contains the location of these files.

To move the queue file (`SYS$QUEUE_MANAGER.QMAN$QUEUES`) and the journal file (`SYS$QUEUE_MANAGER.QMAN$JOURNAL`) to a new location, perform the following steps:

1. Shut down the queue manager by entering the DCL command `STOP/QUEUE/MANAGER/CLUSTER`.
2. Copy the files to their new location. They must reside together in the same directory.
3. Restart the queue manager by entering the following DCL command, where *dirspec* is the device and directory specification for the directory containing the queue and journal files:

```
START/QUEUE/MANAGER dirspec
```

In a VAXcluster environment, if the string substituted for *dirspec* in the `START/QUEUE/MANAGER` command is a concealed logical name, it must be identically defined on all nodes in the cluster.

Once you enter the `START/QUEUE/MANAGER` command, the location you enter is stored in the master file of the queue database. If you need to restart the queue manager, you do not need to respecify the location.

5.3.1.6 Submitting a Software Performance Report (SPR)

If you encounter problems with the queuing system and need to submit a Software Performance Report (SPR), make sure you provide as much information as possible with the report by performing the following steps:

1. Search the operator log file that was active at the time of the problem for any messages related to the queuing system by entering the following command:

```
SEARCH/WINDOW=5/OUTPUT=filename SYS$MANAGER:OPERATOR.LOG;n - QUEUE_MANAGE,JOB_CONTROL
```

For *n*, specify the version number for the operator log file that was active at the time of the problem. For *filename*, specify the name of the file where the results of search are to be written. Include this file with your SPR.

2. Use the `DIRECTORY` command to determine whether either of the following two files were created in your specific `[SYSEXE]` directory:

```
QMAN$QUEUE_MANAGER.DMP  
JBC$JOB_CONTROL.DMP
```

If one or both of these files was created, include a copy of the files with your SPR.

3. Use the VMS Backup Utility (`BACKUP`) with the `/IGNORE=INTERLOCK` qualifier to create a copy of the journal file `SYS$QUEUE_MANAGER.QMAN$JOURNAL` of the queue database. By default, this file is located in `SYS$COMMON:[SYSEXE]`. Include the copy of the journal file with your SPR.

5.3.2 Managing Queues

Once you set up queues, you must monitor, control, and, if necessary, change them. The following sections introduce the basic DCL commands used to perform routine queue management operations.

The following tables list the basic commands for creating and controlling queues. Table 5-3 lists commands that have a different effect on autostart and nonautostart queues. Table 5-4 lists commands that have the same effect on all queues.

Table 5-3 Effects of Queue Commands on Autostart and Nonautostart Queues

Command	Effect on Autostart Queues	Effect on Nonautostart Queues
INITIALIZE/QUEUE	Create the queue. The /AUTOSTART_ON qualifier specifies one or more nodes or nodes and devices on which the queue can run.	Creates the queue. The /ON qualifier specifies a single node or node and device on which the queue is to run.
INITIALIZE/QUEUE/START	Creates the queue and activates it for autostart.	Creates and starts the queue.
START/QUEUE	Activates the queue for autostart.	Starts the queue.
STOP/QUEUE/NEXT	Stops a queue after allowing the current jobs to complete, and deactivates the queue for autostart.	Stops a queue after allowing the current jobs to complete.
STOP/QUEUE/RESET	Stops a queue abruptly and deactivates the queue for autostart.	Stops a queue abruptly.
ENABLE AUTOSTART/QUEUES	Starts all stopped, active autostart queues capable of running on the node.	No effect.
DISABLE AUTOSTART/QUEUES	After allowing jobs to complete, fails all autostart queues over to the next available node in each queue's node list. If a queue has no node list specified, the queue is stopped.	No effect.

Table 5-4 Effects of Queue Commands on All Queues

Command	Description
SET QUEUE	Modifies a queue.
SHOW QUEUE	Displays information about a queue.
STOP/QUEUE	Pauses a queue.
STOP/QUEUES/ON_NODE	Aborts current jobs and stops all queues on a node without stopping the queue manager.
DELETE/QUEUE	Deletes a queue.
ASSIGN/MERGE	Moves jobs from one queue to another.

(continued on next page)

Batch and Print Operations

5.3 Maintaining the Queuing System

Table 5-4 (Cont.) Effects of Queue Commands on All Queues

Command	Description
ASSIGN/QUEUE	Redirects jobs in a logical queue to an execution queue.

5.3.2.1 Initializing Queues

Typically, you create your queues when you first set up your system. However, you might want to create queues later as the need arises.

Use the INITIALIZE/QUEUE command to create a queue, as explained in Section 5.2.9.4.

5.3.2.2 Starting Nonautostart Queues

To start a queue that is not an autostart queue, you use the START/QUEUE command as follows:

```
START/QUEUE[/qualifier,...] queue-name
```

Queue-name is the name of the queue you want to start. For information about the qualifiers you can specify, see DCL Help.

5.3.2.3 Starting Autostart Queues

To start all stopped, active autostart queues on a node, enter the following command:

```
$ ENABLE AUTOSTART/QUEUES
```

For more information, see Section 5.3.2.5.

This command only starts *active* autostart queues. If an autostart queue does not start when you enter the ENABLE AUTOSTART/QUEUES command, the queue might not be active for autostart. Section 5.3.2.4 provides information about activating an autostart queue.

5.3.2.4 Activating an Autostart Queue

Because the ENABLE AUTOSTART/QUEUES command only starts *active* autostart queues capable of running on a node, you must initially activate an autostart queue in one of the following ways:

- Specify the /START qualifier in the INITIALIZE/QUEUE command used to create the queue, as follows:

```
INITIALIZE/QUEUE/START[/qualifiers,...] queue-name
```

- Enter a START/QUEUE command after you create the queue, as follows:

```
START/QUEUE[/qualifiers,...] queue-name
```

Once an autostart queue is activated, it remains active unless the queue is stopped with the STOP/QUEUE/NEXT or STOP/QUEUE/RESET command. Shutting down a node does not deactivate autostart queues on the node.

To start an autostart queue that has been deactivated by the STOP/QUEUE /NEXT or STOP/QUEUE/RESET command, enter the START/QUEUE command. The queue will then be automatically started by the queue manager. The queue will be started immediately if a node on which it can run is enabled for autostart. Otherwise, the queue will be started as soon as a node on which it can run is enabled for autostart.

5.3.2.5 Enabling Autostart on a Node

To notify the queue manager to automatically start all stopped, active autostart queues on a node and to start any active autostart queue that fails over to the node, enter the following command:

```
$ ENABLE AUTOSTART/QUEUES
```

By default, the command affects the node from which it is entered. However, you can specify the `/ON_NODE` qualifier to enable autostart on a different node. For more information, see DCL Help for the `ENABLE AUTOSTART/QUEUES` command.

When a node boots, autostart is disabled until you enter the `ENABLE AUTOSTART/QUEUES` command. Typically, you should add this command to your site-specific startup command procedure or your queue startup command procedure to start a node's autostart queues each time the node boots.

5.3.2.6 Monitoring Queue Information

You use the DCL command `SHOW QUEUE` to monitor the status of queues. To display queue information, enter the `SHOW QUEUE` command in the following format:

```
SHOW QUEUE [/qualifier,...] [queue-name]
```

If you do not specify a qualifier or a queue name, the system displays the status of all queues on the system and all jobs owned by you. The `SHOW QUEUE` qualifiers let you select the type of queue and the amount of information you want to display.

Use the following qualifiers to select the information you want to display:

Qualifier	Description
<code>/BY_JOB_STATUS[=<i>keyword</i>,...]</code>	Displays queues that contain jobs of the specified status. You can specify one or more of the following keywords: executing holding pending retained timed_release If no keyword is specified, by default the jobs of all status are displayed. For more information about job status, see Table 5-6.
<code>/BATCH</code>	Displays batch execution queues.
<code>/DEVICE[=<i>keyword</i>,...]</code>	Displays output execution queues. You can select a specific type of execution queue by entering one or more of the following keywords: printer terminal server If no keywords are specified, all types of output queue are displayed.
<code>/GENERIC</code>	Displays the status of generic queues.

Batch and Print Operations

5.3 Maintaining the Queuing System

Use the following qualifiers to select the amount of information you want to display:

Qualifier	Description
/ALL_JOBS	Displays information about all jobs for the selected queues.
/BRIEF	Displays a brief listing of information about job entries in the queue. The brief listing is the default when no qualifier is specified with the SHOW QUEUE command.
/FILES	Adds a list of files associated with each job to the display.
/FULL	Displays complete queue and job information (also displays any ACLs set for the queues).
/SUMMARY	Displays the total number of executing, pending, holding, retained, and timed release jobs. The jobs themselves are not displayed.

You can also combine certain qualifiers to further delineate the queue information you want to display, as shown in the following examples.

The following example displays summary information for all printer and terminal queues:

```
$ SHOW QUEUE/SUMMARY/DEVICE=(PRINTER,TERMINAL)
Printer queue HERA_LPA0, busy, on HERA::LPA0, mounted form DEFAULT
  <Printer queue on node HERA for a line printer>
  Job summary: 1 executing
Printer queue HERA_LPB0, busy, on HERA::LPB0, mounted form DEFAULT
  <Printer queue on node HERA for a line printer>
  Job summary: 1 executing
Generic printer queue CLUSTER PRINT
  <Generic printer queue for LPA0: and LPB0:>
  Job summary: 1 holding
Terminal queue LQ_PRINT, stopped, on HERA::TXA7:,
  <Letter quality printer in Bob's office>
  mounted form PORTRAIT_INDENTED (stock=DEFAULT)
  Job summary: 2 pending (445 blocks), 1 holding
```

The next example displays the full status and options of all executing jobs.

```
$ SHOW QUEUE/FULL/ALL/BY_JOB_STATUS=EXECUTING
Batch queue HERA_BATCH, available, on HERA::
  /AUTO_START_ON=(HERA::) /BASE_PRIORITY=3 /JOB_LIMIT=25 /OWNER=[SYSTEM]
  /PROTECTION=(S:E,O:D,G:R,W:W)
Entry  Jobname      Username      Status
-----
  700  VUE             SMITH        Executing
      Submitted 25-JUL-1991 14:46 /KEEP /NOLOG /NOPRINT /PRIORITY=100
      File: _$333$DISK1:[SMITH.COM]VUE.COM;19 (executing)

Batch queue ZZ_BATCH, available, on ZZ::
  /AUTO_START_ON=(ZZ::) /BASE_PRIORITY=3 /JOB_LIMIT=25 /OWNER=[SYSTEM]
  /PROTECTION=(S:E,O:D,G:R,W:W)
```

Batch and Print Operations

5.3 Maintaining the Queuing System

```

Entry  Jobname      Username      Status
-----  -
      874  PIPE           FITZGERALD   Executing
          Submitted 26-JUL-1991 11:25 /KEEP /NOTIFY /NOPRINT /PRIORITY=100
          /RESTART=CLUSTER_BATCH /RETAIN=UNTIL="0 01:00"
          File:  _$333$DISK1:[FITZGERALD]PIPE.COM;2 (executing)

Server queue NM$QUE01, available, on HERA::, mounted form DEFAULT
  /BASE PRIORITY=4 /DEFAULT=(FEED,FORM=DEFAULT) /OWNER=[DOC,SMITH]
  /PROCESSOR=NM$DAEMON /PROTECTION=(S:E,O:D,G:R,W:R) /RETAIN=ERROR

Entry  Jobname      Username      Blocks  Status
-----  -
      236  NM           ROSENBERG    12      Processing
          Submitted 23-JUL-1991 08:42 /FORM=DEFAULT /PRIORITY=100
          File:  _$5$DISK3:[FOLK$.NM]NM$J1991072308340647.WRK;1

```

Table 5-5 defines queue statuses returned by the SHOW QUEUE command:

Table 5-5 Queue Status

Queue Status	Description
Aligning	The queue manager is processing a START/QUEUE/ALIGN command.
Autostart inactive	Queue was stopped and needs to be activated. For more information, see Section 5.3.4.5.
Device unavailable	Device to which the queue is assigned is not available.
Available	Queue is processing at least one job but is capable of processing additional concurrent jobs.
Busy	Queue cannot process additional jobs because of one or more jobs in progress.
Closed	Queue is closed and will not accept new jobs until it is set open. For more information, see Section 5.3.2.9.
Idle	Queue is not processing any jobs and is capable of doing so.
Paused	A STOP/QUEUE command has been executed.
Pausing	The queue manager is processing a STOP/QUEUE command.
Remote	Queue is assigned to a physical device that is not connected to the local system.
Resuming	The queue manager is processing a START/QUEUE command on a paused queue.
Server	Queue processing is directed to a server symbiont.
Stalled	Symbiont processing temporarily halted due to device-related problem.
Starting	Queue has been started, but the symbiont process is not yet active.
Stopped	The queue is stopped and will not process work until started.
Stop pending	Queue will be in the Stopped state when current jobs have finished executing.
Stopping	The queue is being stopped.

To display the forms or characteristics available on a system, use the DCL command SHOW QUEUE/FORM or SHOW QUEUE/CHARACTERISTIC.

Batch and Print Operations

5.3 Maintaining the Queuing System

You can further customize the type of queue information you want to monitor by writing a command procedure that uses the F\$GETQUI lexical function. F\$GETQUI invokes the \$GETQUI system service to return information stored in the queue database.

You can use the F\$GETQUI lexical function to obtain information about the following types of objects:

- Characteristics
- Forms
- Queues
- Jobs contained in queues
- Files of jobs contained in queues

For example, you could write a command procedure to display the total number of blocks of jobs in a pending state in all printer queues. You must have READ access to the job or SYSPRV or OPER privilege to obtain job and file information.

See the *VMS DCL Dictionary* for detailed information about F\$GETQUI. For more information about the system service invoked by the F\$GETQUI lexical function, refer to the description of the \$GETQUI system service in the *VMS System Services Reference Manual*.

5.3.2.7 Modifying a Queue

You can use the INITIALIZE/QUEUE, START/QUEUE, and SET QUEUE commands to change queue options; as you change queue options, information about the queue in the queue database is updated. You can use the INITIALIZE and START commands only on stopped queues.

The SET QUEUE command lets you change many queue options without having to stop the queue, initialize it, and restart it. For example, the following command modifies the running batch queue, SYS\$BATCH:

```
$ SET QUEUE/JOB_LIMIT=4/DISABLE_SWAPPING SYS$BATCH
```

The command in this example changes the job limit for the queue and disables swapping for all jobs processed in SYS\$BATCH. All other options of the queue remain the same. The changed options do not affect the execution of current jobs; however, all subsequent jobs are executed with the new options in effect.

To change queue options that cannot be altered with the SET QUEUE command, use the following procedure:

1. Stop the queue with the STOP/QUEUE/NEXT command.
2. Restart the queue with the START/QUEUE or the INITIALIZE/QUEUE /START command, specifying the appropriate qualifiers for the options you desire.

Any qualifiers that you do not specify remain as they were when the queue was previously initialized, started, or set.

Note that initializing an existing queue does not delete any current jobs in that queue. Any new queue settings established by the new INITIALIZE/QUEUE command affect all jobs waiting in the queue or subsequently entering the queue.

See Table 5–1 for a list of the options that can be used for batch and output queues.

5.3.2.8 Pausing a Queue

The DCL command `STOP/QUEUE` (when used without qualifiers) temporarily suspends the execution of all current jobs in the queue and places the queue in a paused state. Pausing an output queue lets you enter print job positioning and alignment commands to the print symbiont. (See Section 5.3.3.9 for more information about using the `STOP/QUEUE` command to control print jobs.)

To resume the execution of a paused queue, enter the DCL command `START/QUEUE`.

5.3.2.9 Closing a Queue

When a queue is not available for an extended period of time (for example, when a printer needs to be serviced), you can prevent new jobs from entering the queue by specifying the `/CLOSE` qualifier with the `SET QUEUE`, `INITIALIZE/QUEUE`, or `START/QUEUE` command. The `/CLOSE` qualifier prevents users from entering jobs in the queue with `PRINT` or `SUBMIT` commands. When a user attempts to print or submit a job to a closed queue, the job is rejected, and the user is notified that the queue is closed. For example:

```
$ PRINT/QUE=$PRINTER_1 REPORT.TXT;  
%PRINT-F-CREJOB, error creating job  
-JBC-E-QUE_CLOSED, queue closed, jobs not accepted
```

Jobs currently in the queue are not affected.

When the queue is available again, use the `/OPEN` qualifier to open the queue for incoming jobs.

5.3.2.10 Stopping a Queue

To stop a queue, enter one of the following commands, according to the desired stop queue operation:

Command	Description
<code>STOP/QUEUE/NEXT</code>	Lets all currently executing jobs complete and then stops the queue. Once you enter this command, all new jobs are prevented from executing.
<code>STOP/QUEUE/RESET</code>	Abruptly stops the queue and returns control to the system. Any jobs that are currently executing are stopped immediately. If the queue is not set to retain jobs completed with an error status, you might want to use the <code>SET QUEUE/RETAIN=ERROR</code> command to do so before stopping the queue with the <code>STOP/QUEUE/RESET</code> command. This causes the queue to retain information about aborted jobs. For print jobs retained on error, you can use the <code>SET ENTRY/RELEASE/NOCHECKPOINT</code> command to restart the interrupted jobs from the beginning. Print jobs are restartable by default; batch jobs are not restartable unless submitted with the <code>/RESTART</code> qualifier.

For autostart queues, these commands deactivate a queue for autostart as explained in Section 5.3.2.11.

To restart a stopped nonautostart queue or to reactivate a deactivated autostart queue, enter the `START/QUEUE` command.

Batch and Print Operations

5.3 Maintaining the Queuing System

5.3.2.11 Preventing Autostart Queues from Starting

With autostart queues, the STOP/QUEUE/NEXT or STOP/QUEUE/RESET command stops a queue and marks it inactive for autostart until the START/QUEUE command is entered. The STOP/QUEUE/NEXT or STOP/QUEUE/RESET command prevents an autostart queue from being automatically restarted.

You might use this feature to prevent an autostart output queue from accidentally restarting when a printer is being serviced.

5.3.2.12 Disabling Autostart on a Node

The DISABLE AUTOSTART/QUEUES command notifies the queue manager to perform the following tasks on the affected node:

- Prevent autostart queues from failing over to the node.
- Mark all autostart queues on the node as “stop pending” in preparation for a planned shutdown. This lets jobs currently executing on the queues complete.
- Upon completion of any jobs currently executing on one of the node’s autostart queues, force the queue to fail over to the next available node in the queue’s failover list on which autostart is enabled. (An autostart queue can only fail over if you have set it up to run on more than one node.)

The DISABLE AUTOSTART/QUEUES command only affects autostart queues. By default, the command affects the node from which it is entered. However, you can specify the /ON_NODE qualifier to disable autostart on a different node.

The DISABLE AUTOSTART/QUEUES command should be used prior to shutting down a node. For more information, see Section 5.3.2.14.

For more information, see DCL Help for the DISABLE AUTOSTART/QUEUES command.

5.3.2.13 Stopping All Queues on a Node

To stop all queues on a node without stopping the queue manager, enter the DCL command STOP/QUEUES/ON_NODE. By default, this command affects the node on which the command is entered. To stop queues on a different node specify the name of the node on which queues are to be stopped as follows:

```
STOP/QUEUES/ON_NODE=node-name
```

When you enter the STOP/QUEUES/ON_NODE command, nonautostart queues and autostart queues without a failover list are stopped. Autostart queues created or started with a failover list fail over to the next available node in that list that has autostart enabled. In all cases, currently executing jobs are aborted.

However, you can allow jobs executing on autostart queues to complete by entering the DISABLE AUTOSTART/QUEUES command and waiting for jobs to complete before entering the STOP/QUEUES/ON_NODE command. For more information, see Section 5.3.2.14.

The STOP/QUEUES/ON_NODE command is included in the shutdown command procedure SYS\$SYSTEM:SHUTDOWN.COM. For more information, see Section 5.3.2.14.

5.3.2.14 Stopping Queues Before Shutting Down a System

The following commands are included in the shutdown command procedure `SYSSYSTEM:SHUTDOWN.COM` and are automatically executed when you shut down a node using `SHUTDOWN.COM`:

- `DISABLE AUTOSTART/QUEUES`
- `STOP/QUEUES/ON_NODE`

Allowing Jobs to Complete Before Stopping Autostart Queues

The `STOP/QUEUES/ON_NODE` command aborts jobs and stops all queues on a node; the `DISABLE AUTOSTART` command allows jobs on autostart queues to finish processing before failing over or stopping autostart queues. If your configuration uses autostart queues, you might want to allow jobs on those queues to complete before stopping your queues.

In `SHUTDOWN.COM`, the `STOP/QUEUES/ON_NODE` command is executed shortly before the node is shut down. When using `SHUTDOWN.COM`, you can ensure that jobs on autostart queues have time to complete before the queues are stopped by specifying the time interval between the `DISABLE AUTOSTART /QUEUES` command and the shutdown. Use one of the following methods:

- Before executing `SHUTDOWN.COM`, define the logical name `SHUTDOWN$DISABLE_AUTOSTART` to be the number of minutes as follows:

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE SHUTDOWN$DISABLE_AUTOSTART number-of-minutes
```

- When running `SHUTDOWN.COM`, specify the number of minutes as a shutdown option as follows:

```
Shutdown options [NONE]: DISABLE_AUTOSTART=number-of-minutes
```

Determine an appropriate number of minutes for your configuration, based on the number and type of jobs in the autostart queues.

If you shut down a node without using `SHUTDOWN.COM`, you might want to enter the `DISABLE AUTOSTART/QUEUES` command and wait a few minutes to allow jobs on autostart queues to finish processing before you enter `STOP /QUEUES/ON_NODE`.

5.3.2.15 Assigning a Logical Queue

When a problem occurs with a print device, you can reroute the queue associated with that device to another queue associated with a functioning device. By creating a logical queue out of a broken execution queue, all of the jobs submitted to that queue will execute in another queue. Use the following procedure to create a logical queue that redirects its jobs to another queue:

1. Stop the queue associated with the malfunctioning print device by entering a command in the following format:

```
STOP/QUEUE/NEXT source_queue
```

This command inhibits new jobs from processing but lets the current job finish processing, unless the print device is inoperable. If the device is inoperable, use the `STOP/QUEUE/RESET` command to halt the queue and immediately cancel all output from the device.

2. Take the device off line.

Batch and Print Operations

5.3 Maintaining the Queuing System

3. Reroute existing jobs from the malfunctioning print device to another print device by entering the following command:

```
ASSIGN/QUEUE target_queue source_queue
```

Check to be sure that the options of the new print device are appropriate for processing the new jobs.

To deassign the logical queue, enter the following command:

```
DEASSIGN/QUEUE source_queue
```

5.3.2.16 Moving All Jobs From One Queue to Another

Before you delete a queue, you might want to requeue all jobs in the queue to another queue. To do so, enter a command in the following format:

```
ASSIGN/MERGE source-queue target-queue
```

Source-queue is the queue to be deleted. *Target-queue* is the queue to which you are moving the jobs.

The ASSIGN/MERGE command moves all jobs currently in the source queue. If new jobs are entered into the source queue before it is deleted, those new jobs remain in the source queue, and are not transferred to the target queue. You might want to close the queue to prevent new jobs from being entered in the queue, as explained in Section 5.3.2.9, before entering the ASSIGN/MERGE command.

For ongoing redirection of jobs, use the ASSIGN/QUEUE command as explained in Section 5.3.2.15.

5.3.2.17 Deleting a Queue

Perform the following steps to delete a queue:

1. Stop the queue by entering the STOP/QUEUE/NEXT command. (Use STOP/QUEUE/RESET if you want to abort all executing jobs.)
2. Wait for executing jobs to complete.
3. Requeue the entries still pending in the queue. If you do not perform this step, jobs will be deleted along with the queue.
4. Remove all references to the queue from generic queues or jobs. See Section 5.3.4.6 for more information about removing references to queues.
5. Delete the queue by entering the DELETE/QUEUE command.

For more information, see the DELETE/QUEUE command in the *VMS DCL Dictionary*. See Section 5.3.4.6 for help solving problems deleting a queue.

5.3.3 Managing Jobs

As system manager, you are responsible for controlling the flow of batch and print jobs and for maintaining efficient job processing performance. Some of the routine tasks for managing jobs include the following:

Task	Section
Monitoring jobs	See Section 5.3.3.1.
Modifying job processing options.	See Section 5.3.3.2.

Batch and Print Operations

5.3 Maintaining the Queuing System

Task	Section
Holding and releasing a job.	See Section 5.3.3.3.
Changing the scheduling priority of a job.	See Section 5.3.3.4.
Requeuing an executing job.	See Section 5.3.3.5.
Requeuing a pending job.	See Section 5.3.3.6.
Retaining jobs in a queue.	See Section 5.3.3.7.
Deleting a job.	See Section 5.3.3.8.
Specifying the position at which a job should begin printing.	See Section 5.3.3.9.1.
Aligning print forms.	See Section 5.3.3.9.2.

5.3.3.1 Monitoring Jobs

You use the DCL command `SHOW ENTRY` to monitor the status of batch and print jobs. (For information about job status, see Table 5–6.) Use the following format to specify the `SHOW ENTRY` command:

```
SHOW ENTRY [entry-number[,...]], [[job-name[,...]]]
```

If you do not specify an entry number or job name, the system displays all jobs owned by you or by the user specified with the `/USER_NAME` qualifier. If you specify a job name, the system displays all jobs owned by you or by the user specified with `/USER_NAME` that match the specified character string. You can also display a group of jobs by entering a list of entry numbers or job names or both on the command line.

The `SHOW ENTRY` qualifiers let you specify the type of job information you want to display. The following command displays jobs owned by user `GARDNER`:

```
$ SHOW ENTRY/USER_NAME=GARDNER
Entry  Jobname      Username      Blocks  Status
-----  -----
      4  TEST          GARDNER              Holding
      On available batch queue OPAL_BATCH
     611  SET          GARDNER          140  Pending
      On stopped printer queue LQPRINT
```

The `/FULL` qualifier displays job status information, the time the job was submitted, the file specification, and the job processing options. For example:

```
$ SHOW ENTRY/FULL 4,611
Entry  Jobname      Username      Blocks  Status
-----  -----
      4  TEST          GARDNER              Holding
      On available batch queue OPAL_BATCH
      Submitted 15-JUN-1991 16:12 /LOG=_$SDUAL:[GARDNER]TEST.LOG;
      /PRIORITY=100
      File: _$SDUAL:[GARDNER]TEST.COM;8
     611  SET          GARDNER          140  Pending (queue stopped)
      On stopped printer queue LQPRINT
      Submitted 21-JUN-1991 16:23 /FORM=DEFAULT /PRIORITY=200
      File: _$SDUAL:[GARDNER]SET.TXT;5
      File: _$SDUAL:[GARDNER]WAIT.TXT;1
```

For a complete description of the `SHOW ENTRY` command and qualifiers, see the *VMS DCL Dictionary*.

Batch and Print Operations

5.3 Maintaining the Queuing System

Table 5–6 describes the job statuses returned by the SHOW ENTRY command.

Table 5–6 Job States

Status	Description
Aborting	Executing job is terminating abnormally.
Executing	Job is executing from a batch queue.
Holding	Job is being held until explicitly released. For more information, see Section 5.3.3.3.
Holding until	Job is being held until a specified time. For more information, see Section 5.3.3.3.
Pending	Job is waiting to execute. For more information, see Section 5.3.4.3.
Printing	Job is executing from a printer or terminal queue.
Processing	Job is executing from a server queue.
Retained on Completion	Job remains in the queue upon completion. For more information, see Section 5.3.3.7.
Retained on Error	Job remains in the queue upon encountering an error. For more information, see Section 5.3.3.7.
Retained Until	Job remains in the queue until the specified time. For more information, see Section 5.3.3.7.
Stalled	Job is stalled because the physical device on which the queue is located is stalled.
Waiting	Symbiont refuses the job.

5.3.3.2 Modifying Job Processing Options

You can modify many job processing options by specifying qualifiers with a command in the following format:

```
SET ENTRY/qualifier[...] entry-number
```

The following table lists some qualifiers that are frequently used to change jobs. For a list of all the job processing options you can change with the SET ENTRY command, see the *VMS DCL Dictionary*.

Qualifier	Description
/[NO]AFTER= <i>time</i>	Controls whether a job is held until after a specified time. For more information, see Section 5.3.3.3.
/CHARACTERISTICS=(<i>characteristic</i> [,...])	Specifies the name or number of one or more characteristics associated with a batch or print job. For more information, see Section 5.2.6.
/FORM= <i>form-name-or-number</i>	Specifies the name or number of the form to be associated with a print job. For more information, see Section 5.2.8.3.
/[NO]HOLD	Controls whether a job is available for immediate processing or held until it is released for processing. For more information, see Section 5.3.3.3.

Batch and Print Operations

5.3 Maintaining the Queuing System

Qualifier	Description
<code>/PRIORITY=<i>n</i></code>	Specifies the scheduling priority of the job. For more information, see Section 5.3.3.4.
<code>/RELEASE</code>	Releases a previously held job. For more information, see Section 5.3.3.3.
<code>/REQUEUE=<i>queue-name</i>[:]</code>	Requests that the job be moved from the original queue to the specified queue; this qualifier can also be used with the STOP /QUEUE/ENTRY command. For more information, see Section 5.3.3.6.
<code>/RESTART</code>	Specifies whether a batch or print job is restarted after a system failure or a STOP /QUEUE/REQUEUE command. Print jobs are restartable by default. Batch jobs are restartable only if submitted or modified with the /RESTART qualifier.

5.3.3.3 Holding and Releasing a Job

Users can specify that a job be held in a queue before processing by specifying one of the following qualifiers with the PRINT, SUBMIT, or SET ENTRY command:

- `/AFTER=time-value` holds a job until the specified time.
- `/HOLD` holds a job indefinitely.

You can use the following commands to hold and release jobs:

Command	Purpose
<code>SET ENTRY/HOLD</code>	Holds a job in a queue indefinitely before processing.
<code>SET ENTRY/AFTER=<i>time-value</i></code>	Holds a job in a queue for processing after a specified time.
<code>SET ENTRY/NOHOLD</code>	Releases a job that is held in a queue for any of the following reasons: <ul style="list-style-type: none"> • Releases a job with a status of "Holding". • Releases a job in "Retained" state (a job held in a queue by the /RETAIN qualifier). For more information see Sections 5.2.5 and 5.3.3.7 • Releases a job that was refused by a user-written symbiont.
<code>SET ENTRY/NOAFTER</code>	Releases a job with a status of "Holding until <i>time-value</i> ", and does not wait until the specified time to schedule the job.

Batch and Print Operations

5.3 Maintaining the Queuing System

Command	Purpose
SET ENTRY/RELEASE	<p>Releases a job that is held in a queue for any of the following reasons:</p> <ul style="list-style-type: none">• Releases a job with a status of "Holding".• Releases a job with a status of "Holding until <i>time-value</i>", and does not wait until the specified time to schedule the job.• Releases a job in "Retained" state (a job held in a queue by the /RETAIN qualifier). For more information see Sections 5.2.5 and 5.3.3.7• Releases a job that was refused by a user-written symbiont.

The following example holds a job until the specified time and subsequently releases the job before that time:

```
$ SET ENTRY 1121/AFTER=12-JUL-1990:17:30
$ SET ENTRY/NOAFTER
```

The following example holds a job until the end of the current day (at 00:00:00 0 o'clock) and subsequently releases the job before that time:

```
$ SET ENTRY 1121/AFTER=TODAY
$ SET ENTRY/NOAFTER
```

The following example holds a job indefinitely and subsequently releases it:

```
$ SET ENTRY 1234/HOLD
$ SET ENTRY 1234/RELEASE
```

5.3.3.4 Changing the Scheduling Priority of a Job

You can change the scheduling priority of a job by using the /PRIORITY qualifier with the SET ENTRY command as follows:

```
SET ENTRY/PRIORITY=n entry-number
```

Do not confuse the job scheduling priority with the process base priority on a queue.

The job scheduling priority value must be in a range of 0 through 255, where 0 is the lowest priority and 255 is the highest. The default value for /PRIORITY is the value of the SYSGEN parameter DEFQUEPRI (usually set at 100). The following example changes the scheduling priority of a job to 50:

```
$ SET ENTRY/PRIORITY=50 1131
```

5.3.3.5 Requeuing an Executing Job

To stop and requeue an executing print job, enter the STOP/QUEUE/REQUEUE command. This command suspends a currently executing job and requeues it to the specified queue. Other jobs remain pending in the queue until they are processed.

Note

The STOP/QUEUE/REQUEUE command stops only the job currently executing in the queue. The queue is not stopped.

Batch and Print Operations

5.3 Maintaining the Queuing System

For example, suppose a job is executing in output execution queue BETA_LPBO when the printer on which the queue is running jams. If no other jobs are pending in the queue, you might want to stop and requeue the job to a queue running on another printer. Because the printer in this example is jammed, you might also want to stop the queue. To do so, you would enter commands similar to the following:

```
$ STOP/QUEUE/REQUEUE=BETA_LPA0 BETA_LPBO ①  
$ STOP/QUEUE/RESET BETA_LPBO ②
```

- ① This command causes the executing print job on BETA_LPBO to be stopped and requeued to BETA_LPA0.
- ② This command stops queue BETA_LPBO.

If you are requeuing a job on a batch queue, you must include the /ENTRY=*n* qualifier. For example:

```
$ STOP/QUEUE/ENTRY=1251/REQUEUE=FRED_BATCH WILMA_BATCH
```

To hold an aborted job, specify the /HOLD qualifier as follows:

```
STOP/QUEUE/REQUEUE[=queue-name]/HOLD[/ENTRY=entry-number] queue-name
```

When you specify /HOLD, the aborted job is placed in a hold state for later release with the SET ENTRY/RELEASE or SET ENTRY/NOHOLD command.

To change the scheduling priority of the aborted job, specify the /PRIORITY qualifier as follows:

```
STOP/QUEUE/REQUEUE[=queue-name]/PRIORITY=n[/ENTRY=entry-number] queue-name
```

Specify the new priority as *n*.

For more information about the STOP/QUEUE/REQUEUE command, see the *VMS DCL Dictionary*.

5.3.3.6 Requeuing a Pending Job

To requeue a job that is pending in a queue, enter the SET ENTRY/REQUEUE command. For example:

```
$ SET ENTRY/REQUEUE=LN03$PRINT 196
```

This command moves job 196 to the queue LN03\$PRINT.

For more information about the SET ENTRY/REQUEUE command, see the *VMS DCL Dictionary*.

5.3.3.7 Retaining Jobs in a Queue

Users can specify a retention option for a batch or print job by specifying the /RETAIN=*option* qualifier with the SUBMIT or PRINT command.

To change the user-specified retention policy for a job, specify the /RETAIN=*option* qualifier with the SET ENTRY command in the following format:

```
SET ENTRY/RETAIN=option entry-number
```

Batch and Print Operations

5.3 Maintaining the Queuing System

Option is one of the following:

Option	Description
ALWAYS	Holds the job in the queue regardless of the job's completion status.
DEFAULT	Holds the job in the queue as specified by the queue's retention policy. For more information, see Section 5.2.5. If no policy has been set on the queue, the job is not retained.
ERROR	Holds the job in the queue only if the job completes unsuccessfully.
UNTIL= <i>time-value</i>	Holds the job in the queue for a specified length of time, regardless of the job's completion status. This lets you retain the job in the queue only as long as the job is needed and eliminates the need to delete the job from the queue later. The time value you specify is interpreted first as a delta time, then as a combination time, and finally as an absolute time. For information about specifying time values, see the <i>VMS User's Manual</i> .

For example, the following command retains job 172 in the queue until 3 hours after the job completes. At that time, the job will automatically be deleted from the queue:

```
$ SET ENTRY/RETAIN=UNTIL="+3:00" 172
```

For more information about user-specified job retention, see the /RETAIN qualifier for the PRINT, SUBMIT, or SET ENTRY command in DCL Help.

You can set a job retention policy on a queue by specifying the /RETAIN qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command. For information see Section 5.2.5.

5.3.3.8 Deleting a Job

Under certain circumstances, it is necessary to terminate an executing batch or print job. For example, you might need to terminate a batch job that has entered an endless loop.

Follow this procedure to delete either a pending or an executing job:

1. Determine the entry number of the job by entering one of the following DCL commands:

```
SHOW ENTRY/USER_NAME=username [job-name]
```

```
SHOW QUEUE/ALL [queue-name]
```

If you do not know the job name, username, or queue name, enter the following command:

```
SHOW QUEUE/BATCH/ALL/BY_JOB_STATUS=EXECUTING
```

2. Delete the job by entering a command in the following format:

```
DELETE/ENTRY=(entry-number){,...}
```

For example, suppose a user has noticed that a job is processing in an endless loop. The user is not the owner of the job and lacks sufficient privilege to stop it. The user enlists your aid as the system manager. You might enter the following commands:

Batch and Print Operations

5.3 Maintaining the Queuing System

```
$ SHOW QUEUE/BATCH/ALL/BY_JOB_STATUS=EXECUTING
Batch queue JADE_BATCH, available, on JADE::
  Entry  Jobname      Username      Status
  -----
    312  ARTWORK      HUNTER       Executing
Batch queue OPAL_BATCH, available, on OPAL::
  Entry  Jobname      Username      Status
  -----
    317  STOCKS      CHANDLER     Executing
Batch queue RUBY_BATCH, available, on RUBY::
  Entry  Jobname      Username      Status
  -----
    888  TEMPO      ENGLISH      Executing
$ DELETE/ENTRY=317
```

5.3.3.9 Pausing an Output Queue to Control Print Job Position and Alignment

Pausing an output queue lets you communicate with the print symbiont interactively. You enter the `STOP/QUEUE` command (without any qualifiers) to pause a queue. Once a queue is paused, you can perform operations such as the following:

- Specify the position at which a suspended job is to resume printing. For example, suppose a printer has a paper jam and the first several pages of a print job are destroyed. You can pause the queue and restart the job, resuming printing at the beginning of the file.
For more information, see Section 5.3.3.9.1.
- Specify the number of pages and the type of data for aligning printer forms. For example, suppose a printer uses a paper stock that is a preprinted continuous form paper. When you begin printing a job, you notice the paper is not correctly aligned, so output does not print in the correct space on the preprinted form. You could pause the queue and print sample data to help you correct the paper alignment.
For more information, see Section 5.3.3.9.2.

Note

In order to perform these tasks, you must pause the queue with the `STOP/QUEUE` command *after* the job has begun printing.

5.3.3.9.1 Specifying the Position of Print By default, when you pause a queue and restart it, printing resumes in the current job at a check point near (though not exactly) where it left off. To specify the position at which the current job is to resume printing, pause the queue, then enter the `START/QUEUE` command with any of the following qualifiers:

Batch and Print Operations

5.3 Maintaining the Queuing System

Qualifier	Description
<code>/[NO]BACKWARD[=<i>pages</i>]</code>	Specifies the number of pages the file is to be backspaced before printing is resumed. A value of 1 indicates that printing is to resume at the top of the current page. If you omit the page value, the file is backspaced one page.
<code>/[NO]FORWARD[=<i>pages</i>]</code>	Specifies the number of pages the file is to be advanced before printing is resumed. A value of 1 indicates that printing is to resume at the top of the next page. If you omit the page value, the file is advanced one page.
<code>/[NO]SEARCH=<i>string</i></code>	Specifies that printing is to resume at the page containing a particular search string. The search for the string moves forward, beginning on the page following the current page. During the search, consecutive tabs and spaces are treated as a single space, and character case is ignored.
<code>/TOP_OF_FILE</code>	Specifies that printing is to resume at the beginning of the file.

When you must use more than one positioning qualifier with the same `START/QUEUE` command, file positioning is performed in the following order:

1. `/TOP_OF_FILE`
2. `/FORWARD`
3. `/BACKWARD`
4. `/SEARCH`

The `STOP/QUEUE` command in the following example suspends the job that is currently printing on the printer queue `JADE_PRINT` and places that queue in the paused state. The `START/QUEUE` command releases the queue from the paused state. The `/TOP_OF_FILE` qualifier causes the job that was suspended to resume printing at the beginning of the file rather than at where it was interrupted.

```
$ STOP/QUEUE JADE_PRINT
$ START/QUEUE/TOP_OF_FILE JADE_PRINT
```

The command in the following example resumes output on printer `SYS_LPA0` after advancing 15 pages from the beginning of the file:

```
$ START/QUEUE/TOP/FORWARD=15 SYS_LPA0
```

5.3.3.9.2 Aligning Print Forms To print alignment data to aid in aligning printer forms, pause the queue, then enter the `START/QUEUE` command along with the `/ALIGN` qualifier.

```
/ALIGN[=(option[,...])]
```

The following options control the number of alignment pages and type of alignment data:

Batch and Print Operations

5.3 Maintaining the Queuing System

Option	Description
MASK	Specifies that input data is masked by replacing alphabetic characters with the character X and numeric characters with the number 9. Mask characters let you prevent the printing of sensitive information. If you omit the MASK option, data is printed unaltered.
<i>n</i>	A decimal number in the range 1 through 20 that specifies the number of alignment pages to print. By default, one page of alignment data is printed.

You can use the /ALIGN qualifier along with any of the file positioning qualifiers described in the previous section. File positioning is performed before alignment data is printed. After the alignment is complete, the queue enters a paused state until you restart it by reentering the START/QUEUE command. Printing resumes from the point that alignment data started; that is, the task is backspaced over the pages printed for alignment.

The command in the following example requests masked alignment for four pages of output:

```
$ START/QUEUE/BACKWARD=2/ALIGN=(MASK,4) SYS_LPAC
```

In this example, the file for the job that was being printed when the queue was paused is backspaced two pages before alignment is performed. Four pages of alignment mask characters are printed. Then the output for the current job is positioned backward four pages, and the queue pauses.

5.3.4 Troubleshooting

The following sections can help you solve common queue problems:

Problem	Section
Queue manager problems	See Section 5.3.4.1
General printer problems	See Section 5.3.4.2
Pending jobs	See Section 5.3.4.3
Stock mismatch problems	See Section 5.3.4.3.1
Characteristics mismatch problems	See Section 5.3.4.3.2
Stalled output queues	See Section 5.3.4.4
Autostart queues that will not start	See Section 5.3.4.5
Problems deleting queues, forms, and characteristics	See Section 5.3.4.6
Problems adding or deleting a module from a device control library	See Section 5.3.4.7

5.3.4.1 Solving Queue Manager Problems

Sections 5.3.4.1.1 and 5.3.4.1.2 provide help for solving typical queue manager problems.

Batch and Print Operations

5.3 Maintaining the Queuing System

5.3.4.1.1 If the Queue Manager Becomes Unavailable If the queue manager will not start or has stopped running, an insufficient failover node list might have been specified for the queue manager, such that none of the nodes in the failover list is available to run the queue manager. Make sure the queue manager node list contains a sufficient number of nodes by entering the `START/QUEUE/MANAGER` command and specifying a node list appropriate for your configuration with the `/ON` qualifier. For more information, see the `START/QUEUE/MANAGER` command in DCL Help.

If in doubt, Digital recommends you specify an asterisk (*) wildcard character as the last node in the list to specify that any remaining node in the cluster can run the queue manager. Specifying the asterisk wildcard prevents your queue manager from becoming unavailable because of an insufficient node list.

5.3.4.1.2 If the Queue Manager Does Not Start If the queue manager does not start when you enter the `START/QUEUE/MANAGER` command, the system displays the following message:

```
%JBC-E-QMANNOTSTARTED, queue manager could not be started
```

If you see this message, search the operator log file `SYSS$MANAGER:OPERATOR.LOG` (or look on the operator console) for messages from the facilities `QUEUE_MANAGE` and `JOB_CONTROL` for information about the problem, as follows:

```
$ SEARCH SYSS$MANAGER:OPERATOR.LOG WINDOW=5 QUEUE_MANAGE,JOB_CONTROL
```

For example:

```
$ START/QUEUE/MANAGER DUA55:[SYSQUE] ①
%JBC-E-QMANNOTSTARTED, queue manager could not be started ②
$ SEARCH SYSS$MANAGER:OPERATOR.LOG WINDOW=5 QUEUE_MANAGE,JOB_CONTROL ③
%%%%%%%%%% OPCOM 14-AUG-1991 18:55:18.23 %%%%%%%%%%%
Message from user QUEUE_MANAGE on QMUNGR
%QMAN-E-OPENERR, error opening DUA55:[SYSQUE]SYSS$QUEUE_MANAGER.QMAN$QUEUES;
%%%%%%%%%% OPCOM 14-AUG-1991 18:55:18.29 %%%%%%%%%%%
Message from user QUEUE_MANAGE on QMUNGR
-RMS-F-DEV, error in device name or inappropriate device type for operation
%%%%%%%%%% OPCOM 14-AUG-1991 18:55:18.31 %%%%%%%%%%%
Message from user QUEUE_MANAGE on QMUNGR
-SYSTEM-W-NOSUCHDEV, no such device available ④
$ START/QUEUE/MANAGER DUA5:[SYSQUE] ⑤
```

- ① This command attempts to start the queue manager specifying `DUA55:[SYSQUE]` as the location of the queue and journal files.
- ② The error message indicates that the queue manager did not start.
- ③ This command searches the operator log file for relevant messages.
- ④ One of the messages reveals that the queue file could not be opened because device `DUA55:` does not exist.
- ⑤ This command successfully starts the queue manager, specifying `DUA5:[SYSQUE]` as the correct location for the queue and journal files

5.3.4.2 Determining the Cause of General Printer Problems

If you are experiencing general printer problems, the following hints are helpful in determining the cause of the problem:

1. Enter a command in the following format to determine the status of the queue with which the printer is associated:

SHOW/QUEUE/FULL queue name

Table 5-5 lists and describes queue statuses.

2. Determine whether the logical name SMBSRVSHR is assigned by entering the command `SHOW LOGICAL SMBSRVSHR`. If the logical name is defined, deassign it by entering the command `DEASSIGN SMBSRVSHR`. This logical name should normally not be defined.
3. If a print request returns a fatal error or does not print, perform the following steps:
 - a. Stop the queue by entering the `STOP/QUEUE/RESET` command.
 - b. If the output device is spooled, despool it using the `SET DEVICE /NOSPOOLED` command.
 - c. Copy the file you are attempting to print by entering the `COPY` command in the following format:

`COPY filename output-device:`

Be sure to include a colon (:) after the device name.

If the `COPY` command does not produce output, a `PRINT` request will not work.

4. If the problem is on a queue using the LATSYP symbiont, try using the default symbiont PRTSMB and see if the problem persists. You can use the PRTSMB symbiont for printers on LAT ports. However, only one queue will be able to send jobs to the printer at a time.

To determine whether a queue is using the LATSYP symbiont, enter the `SHOW QUEUE/FULL` command. If the queue is using the LATSYP symbiont, the display shows the following words: `/PROCESSOR=LATSYP`.

To change the queue's symbiont to PRTSMB, perform the following steps:

- a. Stop the queue by entering the `STOP/QUEUE/RESET` command.
- b. Restart the queue using the `START/QUEUE/NOPROCESSOR` command.

If a `PRINT` request succeeds when the queue is using PRTSMB, the problem is with LATSYP or with the LAT driver LTDRIVER.

5. Attach a terminal at the end of the cable that is being used for the printer. If the data shows up on the terminal, the hold screen button works, and no data is lost, the problem is probably with the printer (an incorrect setting, for example). If the problem exists with the terminal, it might be caused by the cable, the VAX interface port, or a VAX port setting.

Batch and Print Operations

5.3 Maintaining the Queuing System

5.3.4.3 Making Pending Jobs Eligible for Scheduling

If a job does not execute when expected, the job might have a pending or holding status. The `SHOW QUEUE/FULL/ALL_JOBS` command displays the status for all jobs on a queue.

If the job is in a holding status, see Section 5.3.3.3 for information about holding and releasing a job.

If the job is in the pending state, the `/FULL` qualifier enables you to see the reason why the job is ineligible to execute. (Use a 132-character wide display to make sure that all the information is displayed.) For example:

```
$ SHOW/QUEUE/FULL/ALL_JOBS/BY_JOB_STATUS=PENDING
Generic printer queue REGR$GENERIC
/GENERIC=(REGR$Q1,REGR$Q2,REGR$Q3) /OWNER=[SYSTEM]/PROTECTION=(S:E,O:D,G:R,W:W)
Entry  Jobname      Username      Blocks  Status
-----  -----
 684  PROBLEMS        CHURCHILL    3118   Pending (check execution queues)
      Submitted 7-AUG-1991 17:49 /FORM=DEFAULT /NOTIFY /PRIORITY=100
      File:  _$5DUA174:[CHURCHILL]PROBLEMS.TXT;2

Printer queue REGR$Q1, stopped, on LONDON::NPA1, mounted form DEFAULT
/BASE_PRIORITY=4 /DEFAULT=(FEED,FORM=DEFAULT) /OWNER=[SYSTEM]
/PROTECTION=(S:E,O:D,G:R,W:W)
Entry  Jobname      Username      Blocks  Status
-----  -----
 687  PMS$SPEECH     CHURCHILL    3558   Pending (queue stopped)
      Submitted 7-AUG-1991 17:51 /FORM=DEFAULT /NOTIFY /PRIORITY=100
      File:  _$5DUA174:[CHURCHILL]PMS$SPEECH.TXT;1 (checkpointed)
```

A job enters the pending status whenever the job is not eligible to execute. The following are common reasons for a pending status and solutions:

Problem	Solution
The queue is currently processing as many jobs as it can	Wait until intervening jobs complete.
The queue is stopped or stalled.	If a <code>SHOW QUEUE/FULL</code> command shows the queue status as stopped or stalled, determine why the queue is stopped or stalled. If the queue is stopped, start it with the <code>START/QUEUE</code> command (for autostart queues, use <code>START/QUEUE</code> to activate the queue and <code>ENABLE AUTOSTART/QUEUES</code> to start the queue.). If the queue is stalled, follow the steps in Section 5.3.4.4.
The stock of the print job's form does not match the stock of the queue's mounted form.	Perform the steps in Section 5.3.4.3.1.
The job was submitted or modified with characteristics not associated with the queue.	Perform the steps in Section 5.3.4.3.2.
The queue has a block limit set and the size of the print job does not fall within the specified range.	Use the <code>SET ENTRY/REQUEUE</code> command to move the job to another queue, or use the <code>SET QUEUE/(NO)BLOCK_LIMIT</code> command to change or remove the block limit on the queue.

Batch and Print Operations

5.3 Maintaining the Queuing System

Problem	Solution
The owner of the job does not have WRITE access to the execution queue.	Use the SET ENTRY/REQUEUE command to move the job to another queue, or change the access to the queue as explained in Section 5.2.4.
The print job is in a logical queue that is assigned to a stopped execution queue.	Use the SET ENTRY/REQUEUE command to requeue the job to another queue or start the execution queue to which the logical queue is assigned.
An output job requires an output device that is enabled for lowercase printing	If the printer supports lowercase printing, use the /LOWERCASE qualifier with the SET PRINTER or SET TERMINAL command enable lowercase printing on the device. Otherwise, use the SET ENTRY/REQUEUE command to move the job to an execution queue that sends its output to a printer with lowercase printing enabled.

5.3.4.3.1 Fixing Print Jobs That Are Pending Due to Stock Mismatch When monitoring jobs, you might see a print job that is pending in a queue because the stock does not match that of the mounted form. For example, you might see a SHOW ENTRY display similar to the following:

```
$ SHOW ENTRY 133/FULL
Entry  Jobname      Username      Blocks  Status
-----  -----
  133  SET              RANDOM              74  Pending (stock type mismatch)
      On idle printer queue SUE$PRINT
      Submitted 21-JUN-1991 16:14 /FORM=MANUAL (stock=HQ) /PRIORITY=100
      File: _$5$DUA1:[RANDOM]SET.TXT;5
```

To fix jobs that are pending because of a stock mismatch, do one or more of the following:

- Make the stocks match by mounting a different form on the queue (for example, using the SET QUEUE/FORM_MOUNTED command) or by specifying a different form with the job (for example, using the SET ENTRY /FORM command).
- Move the job to a queue on which the stock of the mounted form matches the stock of the job's form (for example, using the SET ENTRY/REQUEUE command).
- Delete the job (for example, using the DELETE/ENTRY command).

See Section 5.2.8.3 for more information about forms.

5.3.4.3.2 Fixing Jobs That Are Pending Because of Characteristics Mismatch

When monitoring jobs, you might see a batch or print job that is pending in a queue because the characteristics do not match those assigned to the queue. For example, you might see a SHOW ENTRY display similar to the following:

```
$ SHOW ENTRY 882/FULL
Entry  Jobname      Username      Blocks  Status
-----  -----
  882  SETHOST      RANDOM              5  Pending (characteristics mismatch)
      On idle printer queue $PRINTER 1
      Submitted 28-AUG-1991 15:21 /CHAR=(5) /FORM=DEFAULT /PRIORITY=100
      File: _$5$DUA1:[RANDOM]SETHOST.LOG;5
```

Batch and Print Operations

5.3 Maintaining the Queuing System

To fix jobs that are pending because of a characteristics mismatch, do one or more of the following:

- Move the job to a queue on which the characteristics match those of the job (for example, using the SET ENTRY/REQUEUE command).
- Delete the job (for example, using the DELETE/ENTRY command).
- Make the characteristics match by assigning new characteristics to the queue (for example, using the SET QUEUE/CHARACTERISTIC command)

See Section 5.2.6 for more information about characteristics.

5.3.4.4 Fixing a Stalled Output Queue

If an output queue is in the stalled state, the device on which the queue is running is malfunctioning. Check the device and fix the problem. Once the problem is fixed, the queue will leave the stalled state.

If you cannot fix the problem immediately, stop the queue by entering the command STOP/QUEUE/RESET. While the queue is stopped, you might want to reroute the jobs in the queue to a functioning queue, as explained in Section 5.3.2.15. When the problem is fixed, deassign the logical queue and start the queue by entering the START/QUEUE command.

5.3.4.5 Determining Why an Autostart Queue Will Not Start

If you attempt to start an autostart queue with the ENABLE AUTOSTART /QUEUES command and the queue does not start, the queue might not be active for autostart. The ENABLE AUTOSTART/QUEUES command starts only *active* autostart queues capable of running on a node.

For example:

```
$ ENABLE AUTOSTART/QUEUES/ON_NODE=KATY①
$ SHOW/QUEUE KATY_BATCH
Batch queue KATY_BATCH, stopped, autostart inactive, on KATY::②
$ START/QUEUE KATY_BATCH③
%JBC-I-AUTONOTSTART, queue is autostart active, but is not started④
$ SHOW QUEUE KATY_BATCH/ALL
Batch queue KATY_BATCH, stopped, autostart, on KATY::⑤
$ ENABLE AUTOSTART/QUEUES/ON_NODE=KATY⑥
$ SHOW QUEUE KATY_BATCH/ALL
Batch queue KATY_BATCH, idle, on KATY::⑦
```

- ① The ENABLE AUTOSTART/QUEUES command attempts to start autostart queues on node KATY.
- ② The SHOW QUEUE display shows that the autostart batch queue KATY_BATCH did not start. The display also reveals that the queue is inactive for autostart; the queue was either not activated initially, or it was deactivated with a STOP/QUEUE/NEXT or STOP/QUEUE/RESET command.
- ③ The START/QUEUE command activates the queue for autostart.
- ④ The informational message informs you that the queue is active for autostart but is not started.
- ⑤ The SHOW QUEUE command also provides this information.
- ⑥ Once the autostart queue is active, reentering the ENABLE AUTOSTART /QUEUES command starts the queue on node KATY.

- ⑦ The SHOW QUEUE command informs you that the queue is started.

5.3.4.6 Solving Problems Deleting a Queue, Form, or Characteristic

If you are having problems deleting a queue, form, or characteristic, make sure you have met the following requirements:

- When deleting a form or characteristic with the DELETE/FORM or DELETE/CHARACTERISTIC command, you must specify the *name* assigned to the form or characteristic. (The form or characteristic *number* cannot be used with the DELETE command). To determine the name of a form or characteristic, enter the SHOW QUEUE/FORM or SHOW QUEUE/CHARACTERISTIC command.
- A queue must be stopped before being deleted.
- All references to a queue, form, or characteristic must be removed before you can delete the queue, form, or characteristic.

If you see a message similar to the following, a reference to the queue, form, or characteristic still exists:

```
%DELETE-E-NOTDELETED, error deleting object-name  
-JBC-E-REFERENCED, existing references prevent deletion
```

For example, the queue you are attempting to delete might be named as a target for a generic queue, or the form you are attempting to delete might be specified for a print job. All references to a queue, form, or characteristic must be removed before you can delete the queue, form, or characteristic. Perform the following steps to find and remove references to a queue, form, or characteristic:

1. Enter the command SHOW/QUEUE/FULL/ALL/OUTPUT=*filename*, where *filename* is a name of a file to which the display output of the command is to be sent.
2. Use the SEARCH command to search the output file for the *name* of the form or queue, or the *number* of the characteristic to be deleted. The result of your search will include all references to the queue, form, or characteristic.
3. If the SEARCH command reveals any queues with references to the queue, form, or characteristic you are trying to delete, perform the following steps:
 - a. Use the STOP/QUEUE/NEXT command to stop each queue with an offending reference.
 - b. Use the START/QUEUE command with the appropriate qualifiers to restart each queue without the reference.
4. If the SEARCH command reveals any jobs with references to the queue, form, or characteristic you are trying to delete, perform *one* of the following steps to eliminate the job reference:
 - Wait for the job to complete. (You might want to raise the priority of the job as explained in Section 5.3.3.4, so the job will be scheduled for processing sooner.)
 - Delete the job as explained in Section 5.3.3.8.
 - Modify the options for the job as explained in Section 5.3.3.2, or ask the owner of the job to do so.

Batch and Print Operations

5.3 Maintaining the Queuing System

The following example includes several commands used to fix problems preventing the deletion of a queue:

```
$ DELETE/QUEUE JADE_BATCH ❶
%DELETE-E-NOTDELETED, error deleting JADE_BATCH
-JBC-E-QUEENOTSTOP, queue must be stopped to perform operation ❷
$ STOP/QUEUE/NEXT JADE_BATCH ❸
$ DELETE/QUEUE JADE_BATCH ❹
%DELETE-E-NOTDELETED, error deleting JADE_BATCH
-JBC-E-REFERENCED, existing references prevent deletion ❺
$ SHOW QUEUE/FULL
.
.
.
Generic batch queue CLUSTER_BATCH
  /GENERIC=(JADE_BATCH,RUBY_BATCH,OPAL_BATCH) /OWNER=[SYSTEM]
  /PROTECTION=(S:E,O:D,G:R,W:W) ❻
.
.
.
$ STOP/QUEUE/NEXT CLUSTER_BATCH ❼
$ START/QUEUE CLUSTER_BATCH/GENERIC=(RUBY_BATCH,OPAL_BATCH) ❽
$ DELETE/QUEUE JADE_BATCH ❾
```

- ❶ The DELETE/QUEUE command attempts to delete the queue.
- ❷ The message indicates that the queue is not stopped.
- ❸ The STOP/QUEUE/NEXT command stops the queue after allowing the current job to complete.
- ❹ The DELETE/QUEUE command again attempts to delete the queue.
- ❺ This time, the message indicates references to the queue exist.
- ❻ The SHOW QUEUE/FULL command shows information about all queues. However, the only reference to JADE_BATCH names the queue as a target queue for the generic queue CLUSTER_BATCH.
- ❼ The STOP/QUEUE/NEXT command stops the generic queue that targets JADE_BATCH.
- ❽ The START/QUEUE command eliminates the reference to JADE_BATCH by restarting the generic queue without specifying JADE_BATCH as a target.
- ❾ The DELETE/QUEUE command successfully deletes the queue.

5.3.4.7 Adding or Deleting a Device Control Library Module

When attempting to add or delete a module from a device control library, you might see the following message:

```
$LIBRAR-F-OPENIN, error opening module-name
-RMS-E-FLK, file currently locked by another user
```

To add or delete a module from a library, you must stop all output queues to which the library is assigned. To determine which queues the library is assigned to, perform the following steps:

1. Enter a command similar to the following, where *filename* is the name of a file to which the display output of the command is to be sent:

```
$ SHOW QUEUE/FULL OUTPUT=filename
```


Batch and Print Operations

5.3 Maintaining the Queuing System

2. Use the `SEARCH` command to search the output file for the name of the library.

The result of your search will include all queues to which that library is assigned. Stop the queues and reenter the command to add or delete the module from the library.

Note

The `SHOW QUEUE/FULL` display only shows the library assigned to a queue if you explicitly assigned a library for the queue by including the `/LIBRARY` qualifier with the `INITIALIZE/QUEUE` or `START/QUEUE` command. If you do not explicitly assign a library to a queue, the default library `SYSDEVCTL` is used.

If the module you are trying to delete is in the default library `SYSDEVCTL`, you must stop all queues for which the `SHOW QUEUE/FULL` command displays no library. To make sure the `SYSDEVCTL` library appears in the `SHOW QUEUE/FULL` display in the future, specify `/LIBRARY=SYSDEVCTL` when you restart the queue.

If you cannot stop the queues immediately, perform the following steps:

1. Use the `COPY` command to copy the library to be modified into your own directory.
2. Add or delete the module to the copy.
3. Use the `COPY` command to copy the library back to the directory `SYS$COMMON:[SYSLIB]`. As long as you use the same name for the modified library, the library will be picked up by each queue when the queue is stopped and restarted.

If your site has a large number of different printers, you can help prevent this problem by using more libraries, such that each library is assigned to fewer queues. For example, you should create and assign a different library for each type of printer, as explained in Section 5.2.8.5.3.

5.4 Managing the Card Reader

Information in this section applies to the `DIGITAL CR-11` card reader, which is used to read computer card decks. Users can submit the two following types of card decks for processing:

- Batch job card decks
- Data card decks

To ensure that card decks are processed efficiently, you must understand their characteristics and the use of the card reader. The following sections describe which cards you should check before processing a deck through a card reader and how to determine which cards are damaged. Section 5.4.2.2 outlines a procedure for processing a card deck through the card reader.

Batch and Print Operations

5.4 Managing the Card Reader

5.4.1 Distinguishing the Type of Card Deck

Before loading a card deck into the card reader, you should determine:

- Whether the deck is a batch job or a data deck, because their processing requirements differ
- Whether the card reader is set to the correct translation mode

The following sections describe how to make these determinations.

5.4.1.1 Batch Job Card Deck

A batch job card deck consists of three segments:

- Initial cards
- Program cards
- Last card

The initial two cards in a batch job card deck are the \$JOB and the \$PASSWORD cards. These cards log in the user and the batch job to the system. Following the initial two cards are program cards. Program cards contain instructions that direct the system to libraries, routines, and data needed to complete the batch job. The last card must be either an end-of-job command (\$EOJ) card or an end-of-file (EOF) card. Either of these cards tells the system that this is the end of the job.

Checking Input

The system cannot execute the job without \$JOB and \$PASSWORD cards. If you are given a card deck with these cards omitted, you should return the deck so that the user can insert them.

Since the card deck contains the user's password, you must ensure that it is always handled with care to preserve the security of the user's account.

The last card in the deck must be either an \$EOJ or an EOF card.

If the last card is not one of these end cards, you can type an end card on the card punch (12-11-0-1-6-7-8-9 overpunch in column 1) and place it at the end of the deck.

Checking Output

The log file produced by a card reader batch job is queued for printing to the default system printer queue, SYS\$PRINT. To have the log file queued to a different queue, the user can specify the /PRINTER qualifier on the \$JOB card.

If an error occurs while the system is attempting to validate the \$JOB and \$PASSWORD cards, OPCOM sends to the card operator an error message that reports the job card and the error.

5.4.1.2 Data Card Deck

A data deck contains data that will be either read by a program or copied to a file for later use. The process that will read the data deck is usually associated with an interactive user at a terminal or else with a batch job that is submitted by an interactive user. Since the user and process already are logged in to the system, the first card can contain any data the user wants to specify. Then, either the program must read the exact number of cards supplied, or the last card should be an EOF card to inform the program that this is the end of the data deck.

When a user wants a data deck to be read, you 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 just flushes the deck through the reader, rejecting the job.

If the program does not read the exact number of cards (as with the COPY command), the EOF card must be the last card in the deck, to inform the program that this is the end of the deck. Without this card, the program waits indefinitely for more cards, and the system prints “card reader off line” messages on the operator’s terminal. If the card deck lacks an EOF card, you can type one on a card punch and insert it at the end of the deck.

5.4.1.3 Setting Card Reader Translation Modes

For the system to read input properly, the card reader must be set to the correct translation mode—the same as the translation mode of the card punch used to prepare the deck. The VMS system supports 026 and 029 card punches.

Therefore, these conditions must exist for you to be able to set the card reader to the correct translation mode:

- The first card in the deck must be the translation mode card.
- You must know the mode in which the cards were punched.

To set the translation mode of the card reader for many decks of the same type, use the SET_CARD_READER command. See the *VMS DCL Dictionary* for more information about the SET_CARD_READER command. By default, when the system is booted, the translation mode is set to 029.

5.4.2 Tending the Card Reader

This section describes how to tend the DIGITAL CR-11 card reader. The job of tending the card reader includes:

- Replacing physically defective cards
- Operating the card reader

5.4.2.1 Replacing Physically Defective Cards

Even when the card deck contains all the required cards, the card reader might not be able to read the deck. This problem is usually caused by one or more physically defective 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 is read. The card reader goes off line, and operator intervention is required to put it back on line. Table 5-7 lists the error indicators, the reasons they might light up, and the operator action required to correct the situation.

5.4.2.2 Operating the Card Reader

The following steps describe how to load and process a card deck through a card reader:

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

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

Batch and Print Operations

5.4 Managing the Card Reader

2. Press the **RESET** button. The **HOPPER CHECK** error indicator and the **STOP** light will go out and the cards will be read.
If the card deck is too large to fit in the input hopper, you can load the excess cards while the reader is operating if you maintain tension on the front portion of the deck.
3. Remove the cards from the output stacker when either the **HOPPER CHECK** error indicator or the bold light is lit.
If you accidentally press the **STOP** button while the card deck is being read, return the last card in the output hopper to the bottom of the input hopper and press the **RESET** button.
4. If the cards are not read properly after the **RESET** button has been pressed, refer to Table 5-7 for recovery procedures.

Table 5-7 Card Reader Errors: Causes and Corrective Actions

Error	Causes	Corrective Action
READ CHECK	Card edges torn Punch in column 0 or 81	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, the read logic of the card reader is malfunctioning.
PICK CHECK	Damage to leading edge Torn webs Cards stapled together	Remove the card from the input hopper, duplicate the faulty card, place the card back in the input hopper, and press the RESET button. If there is no evidence of card damage, check for excessive warping of the card deck or a buildup of ink glaze on the picker face.
STACK CHECK	Jam in the card track Badly mutilated card	Correct the jam 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.

5.4.3 Running the Input Symbiont Interactively

You can run the input symbiont interactively, taking card image input from a VAX RMS file, by entering the following commands:

```
$ DEFINE/USER SYSSINPUT filename  
$ RUN SYSS$SYSTEM:INPSME
```

Running the input symbiont interactively requires the following:

- **CMKRNL** privilege
- Read access to the **UAF**
- Write access to the default directory of the user

All messages are entered to the terminal instead of the card operator.

Gathering System Information

In maintaining your system, you need to collect and review information about system events. This chapter describes how to gather some of the information you need.

The VMS operating system provides several log files that record information about the use of system resources, error conditions, and other system events. These files include the following:

- **Accounting log file**

The accounting log file records the use of system resources and is the source of the accounting reports generated by the ACCOUNTING command (see Section 6.1).

- **Error log file**

VMS automatically records device and CPU error messages in the error log file. The Error Log Utility invokes the Error Log Report Formatter (ERF), which selectively reports the contents of an error log file (see Section 6.2).

- **Operator log file**

The Operator Communication Manager (OPCOM) records system events in the operator log file (see Section 6.3).

- **Security audit log file**

The security audit log file is a clusterwide file that contains a record of security events on the system. Using the ANALYZE/AUDIT command, you can produce reports and summaries of security events from the security audit log file (see Section 6.4).

The VMS operating system also provides a **system dump file** to assist you in analyzing the cause of the system failure. In the event of a severe system failure, VMS automatically shuts down and produces a crash dump of the state of the system at the time that the error was detected. The DCL command ANALYZE/CRASH_DUMP invokes the System Dump Analyzer (SDA) for analysis of a system dump file (see Section 6.5).

You might also need information on special environments like symmetric multiprocessing and vector processing. Section 6.6 describes how to collect information about a multiprocessing system. Section 6.7 describes how to collect information about a vector processing system.

If you find errors in the software, submit a Software Performance Report (SPR), along with detailed information about the error condition.

Gathering System Information

6.1 The Accounting Log File

6.1 The Accounting Log File

The Accounting Utility (ACCOUNTING) collects statistics on the use of system resources in an accounting log file SYS\$MANAGER:ACCOUNTNG.DAT. This information is used to monitor system activity and charge for the use of system resources. On most VAX computers (with the exception of a standalone MicroVAX computer), the Accounting facility is enabled by default when the system is started. You can modify the SET ACCOUNTING command in the site-specific startup template (SYS\$MANAGER:SYSTARTUP_V5.COM) to change the default setting.

READ privilege is required to gain access to the accounting log file. Only a user who has the ACNT privilege can create subprocesses or detached processes in which accounting is disabled. The DCL command RUN/NOACCOUNTING disables all accounting in a created process.

A user with the OPER privilege can selectively disable various kinds of accounting throughout the system by using the DCL command SET ACCOUNTING/DISABLE. See the *VMS DCL Dictionary* for a full description of the SET ACCOUNTING command.

By default, the accounting log file records each of the following activities for all users:

- Batch job termination (BATCH)
- Detached job termination (DETACHED)
- Image activation (IMAGE)
- Interactive job termination (INTERACTIVE)
- Login failures (LOGIN_FAILURE)
- User messages (MESSAGE)
- Network job termination (NETWORK)
- Print jobs (PRINT)
- Process termination (PROCESS)
- Subprocess termination (SUBPROCESS)

Use the SHOW ACCOUNTING command to display which, if any, of these activities are currently being recorded in the accounting log file.

To enable or disable the logging of one or more activities, specify the corresponding keyword in the preceding list with the /ENABLE or /DISABLE qualifier of the SET ACCOUNTING command. (If you do not specify any keywords, the /DISABLE and /ENABLE qualifiers by default disable and enable all the activities previously listed.) For example, to enable the recording of login failures, specify the following:

```
$ SET ACCOUNTING/ENABLE=LOGIN_FAILURE
```

To disable the recording of print jobs, specify the following:

```
$ SET ACCOUNTING/DISABLE=PRINT
```

The following list summarizes the characteristics of the accounting log file:

- File name: ACCOUNTNG.DAT (this file is not an ASCII file; hence, it must be formatted before it is printed)

- Residence and directory: SYS\$MANAGER
- File organization: sequential
- Record length: variable
- Record types: eight

The eight types of records correspond to the conditions that cause records to be written to the file. These record types are shown in the list that follows. Note that their corresponding codes, as defined in the macro \$ACRDEF in SYS\$LIBRARY:STARLET.MLB, are shown in parentheses:

- Records written when processes are deleted (ACR\$K_PRCDEL)
- Records written when an image terminates (ACR\$K_IMGDEL)
- Records written when the system was initialized (ACR\$K_SYSINIT)
- Records written when print jobs are queued (ACR\$K_PRINT)
- Records written when login failures occurred (ACR\$K_LOGFAIL)
- Records written when users' messages are sent to the accounting log file (ACR\$K_USER)
- Records that point to the next accounting file (ACR\$K_FILE_FL)
- Records that point to the previous accounting file, if any (ACR\$K_FILE_BL)

As records are entered in the accounting log file, all but image termination records are immediately flushed to disk. This precaution guarantees the integrity of the file and the completeness of accounting data even if the system fails.

Usually, the current version of the accounting log file is closed at the end of a billing period, and a new version is created and opened. Because the accounting file is always growing, you may want to begin a new accounting file and purge the old version regularly. To begin a new accounting file, enter the DCL command SET ACCOUNTING/NEW_FILE.

If an attempt to write to the accounting log file results in an error, the file is closed automatically and a new copy is created and opened.

6.1.1 Accounting Records

Accounting records contain cumulative accounts of the resources used either by processes or images set up for users, or by print symbionts that print out files for users. Each accounting record contains three fields—user name, UIC, and account name—that identify the user and establish the connection between the accounting record and a user of the system. These fields correspond to similar fields of the user's account record in the user authorization file (UAF).

As system manager, you can use the Accounting Utility to sort, select, and report the accounting records. The reports can provide valuable system management tools. (See the *VMS Accounting Utility Manual*.) Alternatively, by using the detailed accounting records provided by the system, you or perhaps a system programmer can devise programs for reporting on the use of system resources and for billing for their use.

Gathering System Information

6.1 The Accounting Log File

6.1.2 Accounting Report Formats

The Accounting Utility (ACCOUNTING) uses the data from the accounting log file to produce accounting reports. Using ACCOUNTING qualifiers, you can produce a variety of report formats, choose how the reports are organized, and select specific report items. Accounting reports can serve as system management tools to learn more about how the system is used, how it performs and, in some cases, how particular individuals use the system. The reports also provide a means of billing users for system resources.

The basic forms of accounting output are listing and binary output. By default, the output is directed to SYS\$OUTPUT. However, you can specify an output file with the /OUTPUT qualifier. You can further specify whether the output should be in binary or ASCII format with the /BINARY qualifier. If you specify /BINARY, a binary accounting file is produced that can later be processed with other accounting commands.

The three basic output formats used for displaying data are: brief, full, and summary listings. If you specify /FULL or /SUMMARY, or assume the default (/BRIEF), an ASCII file is produced. The following example illustrates a summary output for the following ACCOUNTING command line:

```
$ ACCOUNTING/SUMMARY=(ACCOUNT,USER)/REPORT=(RECORDS,ELAPSED,PROCESSOR)
```

From:	5-APR-1990 16:33	To:	23-APR-1990 14:18	
Account	Username	Total Records	Elapsed Time	Processor Time
ADMIN	JFUSCIA	128	5 19:43:47.22	0 10:03:58.09
ADMIN	JGREEN	56	0 23:14:23.01	0 00:14:55.17
DECMail	POSTOFFICE	2	0 00:04:01.10	0 00:00:02.89
DECNET	NETMGR	1	0 00:01:31.17	0 00:00:02.81
DECNET	NETNONPRIV	2443	2 09:01:15.10	0 01:09:42.61
FIELD	FIELD	31	0 05:18:16.50	0 00:09:41.59
MANUF	BPURPLE	37	1 02:38:45.03	0 02:23:35.42
MANUF	JBROWN	227	4 04:35:07.25	0 04:30:40.60

For more information about the Accounting Utility, see the *VMS Accounting Utility Manual*.

6.2 The Error Log File

The system automatically writes error messages to the latest version of a file named SYS\$ERRORLOG:ERRLOG.SYS. You can display the information in this file by entering the DCL command ANALYZE/ERROR_LOG.

The Error Logging 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 process called ERRFMT, which 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. (The ERRFMT process is started when the system is booted.)
- The Error Log Utility, which you invoke by entering the DCL command ANALYZE/ERROR_LOG; it is used to selectively report the contents of an error log file. (See the *VMS Error Log Utility Manual*.)

The executive routines and the ERRFMT process operate continuously without user 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 automatically writes the buffers to ERRLOG.SYS.

Sometimes a burst of errors can cause the buffer to fill up before ERRFMT can empty them. You can detect this condition by noting a skip in the error sequence number of the records reported in the error log reports. As soon as ERRFMT frees the buffer space, the executive routines resume preserving error information in the buffers.

The ERRFMT process displays an error message on the system console and deletes itself if it encounters excessive errors while writing the error log file. To restart the ERRFMT process, first log in to the system manager's account so that you have the required privileges to perform the operation. Then execute the startup command procedure (STARTUP.COM) specifying ERRFMT as the command parameter, as follows:

```
$ @SYS$SYSTEM:STARTUP ERRFMT
```

6.2.1 Using Error Reports

The error reports generated by the Error Log Utility are useful in two basic ways:

- They aid preventive maintenance by identifying areas within the system that show potential for failure.
- They aid the diagnosis of a failure by documenting the errors and events that led up to it.

The detailed contents of the reports are most meaningful to field service personnel. However, you can use the reports as an important indicator of the system's reliability. For example, using the DCL command SHOW ERROR, you may see that a particular device is producing a relatively high number of errors. You can then use the Error Log Utility to obtain a more detailed report and decide whether you should consult the Digital Services Group. In that case, field service personnel can run diagnostic programs to investigate the device and attempt to isolate the source of the errors.

If a system component does fail, a representative from the Digital Services Group can study the error reports of the system activity leading up to and including the failure. For example, if a device fails, you can generate error reports immediately after the failure. One report might describe in detail all errors associated with the device that occurred within the last 24 hours; another report might summarize all types of errors for all devices that occurred within the same time period. The summary report can put the device errors into a systemwide context. The Services Group representative can then run the appropriate diagnostic program for a thorough analysis of the failed device. Using the combined error logging and diagnostic information, the Services Group representative can proceed to correct the device.

Error reports allow you to anticipate potential failures. In turn, Digital Services Group personnel rely on the reports as an aid to both preventive and corrective maintenance. Overall, effective use of the Error Log Utility in conjunction with diagnostic programs can significantly reduce the amount of system downtime.

Gathering System Information

6.2 The Error Log File

6.2.2 Maintaining the Error Log File

Because the error log file (SYS\$ERRORLOG:ERRLOG.SYS) is a shared file, ERRFMT can write new error log entries while other entries in the same file are being read and reported by the Error Log Utility.

ERRLOG.SYS will increase in size and remain on the system disk until it is explicitly renamed or deleted. Therefore, you must devise a plan for regular maintenance of the error log file.

One method is to rename ERRLOG.SYS on a daily basis. This action causes a new error log file to be created and allows the old file (which was renamed) to be copied to a backup volume where it can be kept as long as needed. For example, you could rename the current copy of ERRLOG.SYS to ERRLOG.OLD every day at 9:00 a.m. To free space on the system disk, you could then back up the renamed version of the error log file on a different volume and delete the file from the system disk. Note that you should exercise caution to ensure that error log files are not deleted inadvertently. You may also want to adopt a naming convention for your files that incorporates in the file name a beginning or ending date for the data.

6.2.3 Printing the Error Log File

The following steps describe how to generate an error log report for all entries in the error log file and how to print the report:

1. Ensure that you have the SYSPRV privilege. You need this privilege to access the error log file.
2. Set your default disk and directory to SYS\$ERRORLOG.
3. Examine the error log directory to see which error log file you want to analyze.
4. To obtain a full report of the current error log file, enter the following command:

```
$ ANALYZE/ERROR_LOG/OUTPUT=ERRORS.LIS
```

5. Print a copy of the report, using the file name specified with the /OUTPUT qualifier:

```
$ PRINT ERRORS.LIS
```

Example

```
$ SET PROCESS/PRIV=SYSPRV
$ SET DEFAULT SYS$ERRORLOG
$ DIRECTORY
```

```
Directory SYS$SYSROOT:[SYSERR]
```

```
ERRLOG.OLD;2 ERRLOG.OLD;1 ERRLOG.SYS;1
```

```
Total of 3 files.
```

```
$ ANALYZE/ERROR_LOG/OUTPUT=ERRORS.LIS ERRLOG.OLD
$ PRINT ERRORS.LIS
```

The DIRECTORY command lists all the files in the SYS\$ERRORLOG directory. In this example the directory contains three files, two old error log files and the current error log file, ERRLOG.SYS. The ANALYZE/ERROR_LOG command requests that a full report be written to a file called ERRORS.LIS, using the most recent ERRLOG.OLD file as input.

6.3 The Operator Log File

The operator log file (SYS\$MANAGER:OPERATOR.LOG) records system events and user requests sent to the operator terminal by the operator communication manager (OPCOM), even when all operator terminals have been disabled. By default, OPCOM is started when your system is booted. You use the operator log file to anticipate and prevent hardware and software failures, and to monitor user requests for disk and magnetic tape operations. The following message types appear in the operator log file:

- Initialization of the operator log file
- Status reports for devices attached to the system
- Operator terminals enabled and disabled
- Volume mounts and dismounts
- User requests and operator replies
- Changes to system parameters through the System Generation Utility (SYSGEN)
- Security alarm messages
- DECnet-VAX status messages

Example 6-1 illustrates some typical messages found in the operator log file. By regularly examining the operator log file, you can often detect potential problems and take corrective action.

6.3.1 Types of OPCOM Messages

This section describes some of the messages you might find in the operator log file. See the *VMS Networking Manual* for information about DECnet-VAX status messages.

6.3.1.1 Initialization Messages

When you enter the REPLY/LOG command, the current operator 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.

When a new log file is created, the first message recorded in it is an initialization message that tells when and by whom the log file was initialized. This message appears in the following format:

```
%OPCOM, dd-mmm-yyyy hh:mm:ss.cc, logfile initialized by operator  
operator-name logfile is SYS$MANAGER:OPERATOR.LOG
```

6.3.1.2 Device Status Messages

Some I/O drivers send messages to OPCOM concerning changes in the status of the devices they control. For example, when a line printer goes off line, an OPCOM message is written into the operator log file at periodic intervals until the device is explicitly returned to online status.

The device status message appears in the operator log file in the following format:

```
%OPCOM, dd-mmm-yyyy hh:mm:ss.cc, device device-name is offline
```

The devices for which this message can appear are card readers, line printers, and magnetic tapes.

Gathering System Information

6.3 The Operator Log File

Example 6-1 Sample Operator Log File (SYS\$MANAGER:OPERATOR.LOG)

```
%%%%%%%%%% OPCOM, 15-APR-1990 22:33:54.07 %%%%%%%%%%%
Operator ' ZEUS$VT333:' has been disabled, user JONES
%%%%%%%%%% OPCOM, 15-APR-1990 22:34:15.47 %%%%%%%%%%%
Operator ' ZEUS$VT333:' has been enabled, user SMITH
%%%%%%%%%% OPCOM, 15-APR-1990 22:34:15.57 %%%%%%%%%%%
operator status for ' ZEUS$VT333:'
PRINTER, TAPES, DISKS, DEVICES
%%%%%%%%%% OPCOM, 15-APR-1990 22:38:53.21 %%%%%%%%%%%
request 1, from user PUBLIC
Please mount volume KLATU in device MTA0:
The tape is in cabinet A
%%%%%%%%%% OPCOM, 15-APR-1990 22:39:54.37 %%%%%%%%%%%
request 1 was satisfied.
%%%%%%%%%% OPCOM, 15-APR-1990 22:40:23.54 %%%%%%%%%%%
message from user SYSTEM
Volume "KLATU      " mounted, on physical device MTA0:
%%%%%%%%%% OPCOM, 15-APR-1990 22:40:38.02 %%%%%%%%%%%
request 2, from user PUBLIC
MOUNT new relative volume 2 () on MTA0:
%%%%%%%%%% OPCOM, 15-APR-1990 22:41:07.54 %%%%%%%%%%%
message from user SYSTEM
Volume "KLATU      " dismantled, on physical device MTA0:
15-APR-1990 22:42:14.81, request 2 completed by operator OPA0
%%%%%%%%%% OPCOM, 15-APR-1990 22:46:47.96 %%%%%%%%%%%
request 4, from user PUBLIC
_TTB5:, This is a sample user request with reply expected.
%%%%%%%%%% OPCOM, 15-APR-1990 22:47:38.50 %%%%%%%%%%%
request 4 was canceled
%%%%%%%%%% OPCOM, 15-APR-1990 22:48:21.15 %%%%%%%%%%%
message from user PUBLIC
_TTB5:, This is a sample user request without a reply expected.
%%%%%%%%%% OPCOM, 15-APR-1990 22:49:07.90 %%%%%%%%%%%
Device DMA0: is offline.
Mount verification in progress.
%%%%%%%%%% OPCOM, 15-APR-1990 22:49:20.22 %%%%%%%%%%%
Mount verification completed for device DMA0:
%%%%%%%%%% OPCOM, 15-APR-1990 22:49:37.64 %%%%%%%%%%%
Device DMA0: has been write locked.
Mount verification in progress.
%%%%%%%%%% OPCOM, 15-APR-1990 23:33:54.07 %%%%%%%%%%%
message from user NETACP
DECnet shutting down
```

6.3.1.3 Terminal Enable and Disable Messages

You designate a terminal as an operator terminal by entering the **REPLY/ENABLE** command from the desired terminal. OPCOM confirms the request by displaying the following message at the operator terminal and in the operator log file:

```
%OPCOM, dd-mmm-yyyy hh:mm:ss.cc, operator enabled, operator terminal-name
```

This message tells you which terminal has been established as an operator terminal and when it was established.

If a terminal has been designated as an operator terminal for a particular function, OPCOM displays the name of that function or operator class. For example, if you enter the command **REPLY/ENABLE=TAPES**, OPCOM displays the following message:

Gathering System Information

6.3 The Operator Log File

```
%OPCOM, 14-JUN-1990 10:25:35.74, operator enabled, operator TTE1
%OPCOM, 14-JUN-1990 10:25:38.82, operator status for operator TTE1
TAPES
```

OPCOM confirms that the terminal is established as an operator terminal and indicates that the terminal can only receive and respond to requests concerning magnetic tape-oriented events, such as the mounting and dismounting of tapes.

A terminal that has been designated as an operator terminal is automatically returned to nonoperator status when the operator logs out. To return the terminal to normal (nonoperator) status without logging out, enter the `REPLY/DISABLE` command from the terminal. OPCOM confirms that the terminal is no longer an operator terminal by displaying a message in the following format both at the operator terminal and in the operator log file:

```
%OPCOM, dd-mmm-yyyy hh:mm:ss.cc, operator disabled, operator terminal-name
```

This message tells you which terminal has been restored to nonoperator status and when the transition occurred.

If a terminal is designated as an operator terminal and only partial operator status is disabled, OPCOM displays a status message. This message lists which requests the terminal can still receive and respond to. This message is displayed at the operator terminal and in the operator log file in the following format:

```
%OPCOM, dd-mmm-yyyy hh:mm:ss.cc, operator status for operator terminal-name
status-report
```

For example, suppose you designate a terminal as an operator terminal that receives messages concerning magnetic tapes and disks, as well as messages intended for the special site-specific operator class known as `OPER10`. Later, you relinquish the terminal's ability to receive messages concerning tapes. When you enter the `REPLY/DISABLE=TAPES` command, OPCOM returns the following message:

```
%OPCOM, 14-JUN-1990 09:23:45.32, operator status for operator TTA3
DISKS, OPER10
```

This message tells you that terminal `TTA3` still receives and can respond to messages about disks and messages directed to `OPER10`.

6.3.1.4 Volume Mount and Dismount Messages

Perhaps the widest range of operator messages occurs with volume mounts and dismounts. See Example 6-1 for examples of messages relating to mount verification and operator-assisted mounts.

6.3.1.5 User Request and Operator Reply Messages

To communicate with you, the user enters the `REQUEST` command, specifying either the `/REPLY` or `/TO` qualifier.

If the user enters a `REQUEST/REPLY` command, the request is recorded in the operator log file in the following format:

```
%OPCOM, dd-mmm-yyyy hh:mm:ss.cc, request request-id from user user-name
__terminal-name:, "message-text"
```

This message tells you which user sent the message, the time the message was sent, the request identification number assigned to the message, the originating terminal, and the message itself.

Gathering System Information

6.3 The Operator Log File

If the user enters a REQUEST/TO command, the request is recorded in the operator log file in the format described above, but without a request identification number, as follows:

```
%OPCOM,dd-yyyymm-dd hh:mm:ss.cc, request from user user-name  
__terminal-name:, "message-text"
```

For examples of OPCOM messages that result from requests to mount magnetic tapes through the magtape ACP using the REQUEST/BLANK_TAPE and REQUEST/INITIALIZE_TAPE commands, see Chapter 3.

When you respond to a users' request and specify the /TO qualifier, the response is recorded in the operator log file in the following format:

```
response message  
%OPCOM, dd-yyyymm-dd hh:mm:ss.cc, request request-id completed by  
operator operator-name
```

This message indicates how the operator responded to the user's request, as well as when the response was entered and which operator responded.

When you respond to a user's request and specify the /ABORT qualifier, the response is recorded in the operator log file in the following format:

```
%OPCOM, dd-yyyymm-dd hh:mm:ss.cc, request request-id was canceled.
```

When you respond to a user's request using the /PENDING qualifier, the response is not recorded in the operator log file because the request has not yet been completed (that is, the request has not been fulfilled or aborted).

When a user enters a REQUEST/REPLY command and you have disabled all terminals as operator terminals, OPCOM records all subsequent users' requests in the log file in the format previously shown, but returns a message to the user indicating that no operator coverage is available.

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 log file.

6.3.1.6 SYSGEN Utility Messages

Users with CMKRNL privilege can use the System Generation Utility (SYSGEN) to change system parameters in the running (active) system. Users with the SYSPRV privilege can use SYSGEN to change system parameters in the current system. OPCOM logs all changes made to current system parameters with messages in the following format:

```
%OPCOM, dd-yyyymm-dd hh:mm:ss.cc, message from user user-name  
%SYSGEN-I-WRITExxx, system-mode system parameters modified by process ID n  
into file y
```

6.3.1.7 Security Alarm Messages

Security alarm messages are included in the operator log file. OPCOM also forwards alarm messages to the audit server process. The following example shows a security alarm OPCOM message:

Gathering System Information

6.3 The Operator Log File

```
%%%%%%%%% OPCOM 19-APR-1990 15:15:03.21 %%%%%%%%%%
Message from user AUDIT$SERVER on LASSIE
Security alarm (SECURITY) and security audit (SECURITY) on LASSIE, system id: 19611
Auditable event:      Attempted file access
Event time:           19-APR-1990 15:15:03.20
PID:                  20200283
Username:             WILSON
Image name:           LASSIE$DMA0:[SYS0.SYSCOMMON.] [SYSEXE]DIRECTORY.EXE
Object name:          LASSIE$DMA0:[000000]000000.DIR;1
Object type:          file
Access requested:     READ
Status:               %SYSTEM-F-NOPRIV, no privilege for attempted operation
```

If security auditing is enabled, OPCOM security messages are also written to the security audit log file (see Section 6.4). Use the VMS Audit Analysis Utility to selectively report the contents of the security audit log file, as described in the *Guide to VMS System Security*.

See the *Guide to VMS System Security* for more information on security auditing.

6.3.2 Setting Up the Operator Log File

The operator log file normally resides on the system disk in the [SYSMGR] directory. You can, however, maintain the log file in a different location by defining the logical name OPC\$LOGFILE_NAME.

Because this file is in ASCII format, you can print it. You should print copies regularly and retain these copies for reference. Section 6.3.4 describes how to print copies of the operator log file.

A new version of OPERATOR.LOG is created each time the system is rebooted (except on workstations in a VAXcluster, where the log file is not opened by default). Note that there is one operator log file per node; it is not a shared file.

You can use the DCL command REPLY/LOG to create a new version of the file at any time. The highest version is always the one in use and is inaccessible. By default, messages of all operator classes are included in the log file. You can use the REPLY/LOG/ENABLE=(*list-of-classes*) and REPLY/LOG/DISABLE=(*list-of-classes*) commands to specify which operator classes will be included in the log file. For example, the following command opens a log file to include messages about mounting and dismounting disks and tapes:

```
$ REPLY/LOG/ENABLE=(DISKS,TAPES)
```

When you add the /LOG qualifier to the REPLY/ENABLE and REPLY/DISABLE commands, the classes you select are enabled or disabled for the log file rather than the terminal. For more information, see the REPLY/LOG, REPLY/ENABLE, and REPLY/DISABLE commands in the *VMS DCL Dictionary*.

In addition, you can specify the default state of the operator log files by defining some logical names in the command procedure SYS\$MANAGER:SYLOGICALS.COM. The following table lists these logical names and their function. For more information on SYLOGICALS.COM, see the *Guide to Setting Up a VMS System*.

Gathering System Information

6.3 The Operator Log File

Logical Name	Function
OPC\$LOGFILE_ENABLE	Specifies whether or not an operator log file is opened. If defined to be true, an operator log file is opened. If defined to be false, no operator log file is opened. By default, a log file is opened on all systems except workstations in a VAXcluster.
OPC\$LOGFILE_CLASSES	Specifies the operator classes that are enabled for the log file. By default, a log file is opened for all classes. The logical name can be a search list of the allowed classes, a comma-separated list, or a combination of the two. Note that OPC\$LOGFILE_CLASSES can be defined even if OPC\$LOGFILE_ENABLE is not defined. In that case, the classes are used for any log files that are opened, but the default is used to determine whether or not to open the log file.
OPC\$LOGFILE_NAME	Specifies the name of the log file. By default, the log file is named SYS\$MANAGER:OPERATOR.LOG. If you specify a disk other than the system disk, commands to mount that disk should be included in the command procedure SYLOGICALS.COM.

By default, OPCOM logs all SECURITY class messages in the operator log file. However, these entries duplicate the entries written by the audit server process (AUDIT_SERVER) to the system security audit log. To conserve disk space you might want to disable the SECURITY class in the operator log file. The following example shows how to disable SECURITY class messages from being saved in the operator log file:

```
$ DEFINE/SYSTEM OPC$LOGFILE_CLASSES CENTRAL, PRINTER, TAPES,  
DISKS, DEVICES, NETWORK, CLUSTER, LICENSE,  
OPER1, OPER2, OPER3, OPER4, OPER5, OPER6, OPER7,  
OPER8, OPER9, OPER10, OPER11, OPER12
```

6.3.3 Maintaining the Operator Log File

You should devise a plan for regular maintenance of these files. One way is to rename the second-highest version on a daily basis. The procedure for renaming the operator log file is the same as that described in Section 6.2.2 for renaming the error log file. You may want to purge outdated versions of the operator log file on a regular basis. However, you should not delete versions that have not been backed up.

If OPCOM is inadvertently deleted or suspended, perform the following steps to start it manually.

1. Log in to the SYSTEM account so that you have the required privileges to perform the operation.
2. Enter the following command to execute the startup command procedure (STARTUP.COM), specifying OPCOM as the command parameter:

```
$ @SYS$SYSTEM:STARTUP OPCOM
```


6.3.4 Printing the Operator Log File

Perform the following operation from an enabled operator terminal to produce a printed copy of the most recent version of the operator log file:

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

```
$ REPLY/LOG
```

2. Set the default to `SYS$MANAGER` and enter the following command to list all versions of the file:

```
$ DIRECTORY OPERATOR.LOG
```

3. Rename the second-highest version to `OPERATOR.OLD`:

```
$ RENAME OPERATOR.LOG;-1 OPERATOR.OLD
```

The version number, `-1`, specifies that the second-highest version of this file is to be renamed. The highest version number is the current operator log file.

4. Obtain a printed copy of the operator log file by entering the following command from an enabled operator terminal:

```
$ PRINT OPERATOR.OLD
```

In the following example, 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 default disk to the system disk, thus enabling you to examine the files contained in the directory `[SYSMGR]`. You can rename the second-highest version of the operator log file to `OPERATOR.OLD` and then enter the `PRINT` command to request that this version of the operator log file (`OPERATOR.OLD`) be printed.

```
$ REPLY/LOG
%%%%%%%%%% OPCOM, 15-APR-1990 12:29:24.52  %%%%%%%%%%%
logfile initialized by operator _MARS$VTA2:
logfile is SYS$MANAGER:OPERATOR.LOG

$ SET DEFAULT SYSSMANAGER
$ DIRECTORY OPERATOR.LOG

Directory SYS$SYSROOT:[SYSMGR]
OPERATOR.LOG;582          OPERATOR.LOG;581

Total of 2 files.

$ RENAME OPERATOR.LOG;-1 OPERATOR.OLD
$ PRINT OPERATOR.OLD
```

6.4 The Security Audit Log File

Security auditing is the act of recording security-relevant events as they occur on the system. By default, security auditing is enabled at system startup for the following situations (or classes of security events):

- All uses of the `SET AUDIT` command
- All break-in attempts

The audit server process (created at system startup) records these events in the security audit log file (`SYS$MANAGER:SECURITY_AUDIT.AUDIT$JOURNAL`).

Gathering System Information

6.4 The Security Audit Log File

The usefulness of the security audit log file depends upon the procedures you adopt to review the file on a regular basis. For example, you might implement the following as part of your site audit review policy:

1. Create a new version of the security audit log file each morning.
2. Review the previous version of the log file for suspicious system activity. Depending on the number of security events that you are auditing on your system, it might be impractical to review every audit record written to the audit log file. In that case, you might want to select a specific set of records from the log file (for example, all AUTHORIZATION and BREAKIN records, or all events created outside normal business hours).
3. If, during your review, you find any security events that appear suspicious or out of place (login attempts outside normal business hours, for example), perform a more detailed inspection of the security audit log file, as described in the *VMS Audit Analysis Utility Manual*. Otherwise, delete it.

Use the VMS Audit Analysis Utility to selectively report the contents of the security audit log file, as described in the *Guide to VMS System Security*.

6.4.1 Creating a New Version

Use the following SET AUDIT command to create a new version of the clusterwide security audit log file:

```
$ SET AUDIT/SERVER=NEW_LOG
```

A new version of the audit log file is opened by the audit server process on each node in the cluster. To prevent the loss of audit messages, the previous version of the audit log file is not closed until all audit messages stored in memory are written to the file.

In some cases, the nodes in a cluster might not share the same system security audit log file. In a multiple-environment cluster, you can specify the following SET AUDIT command to create a new version of the system security audit log file on the local node only:

```
$ SET AUDIT/SERVER=CREATE_SYSTEM_LOG
```

System security audit log files on other nodes in the cluster are unaffected.

For more information about the security audit log file, see the *Guide to VMS System Security*.

6.5 The System Dump File

The following requirements must be met before the VMS operating system can write a complete dump file:

- Do not halt the system until the console dump messages have been printed in their entirety, and the memory contents have been written to the system dump file. Be sure to allow sufficient time for these events to take place, or make sure that all disk activity has stopped before using the console terminal to halt the system.
- There must be a dump file in the SYS\$SYSTEM directory that is named either SYSDUMP.DMP or PAGEFILE.SYS. AUTOGEN automatically creates the SYSDUMP.DMP file if there is enough disk space available.

If SYS\$SYSTEM:SYSDUMP.DMP is present, the system writes dumps to SYSDUMP.DMP. If SYS\$SYSTEM:SYSDUMP.DMP is not present, the system writes dumps to SYS\$SYSTEM:PAGEFILE.SYS. In this case, PAGEFILE.SYS must be at least 1004 blocks larger than physical memory, and the system parameter SAVEDUMP must be set to 1 (the default is 0). If neither file exists, the system will not generate any dumps.

The size of SYSDUMP.DMP is equal to the physical memory size plus the number of error log buffers plus 1. The number of error log buffers is controlled by the value set for the system parameter ERRORLOGBUFFERS. The range for ERRORLOGBUFFERS is from 2 to 64 with the default set to 4.

- The system parameter DUMPBUG must be set to 1 (the default is 1).

See the *VMS System Dump Analyzer Utility Manual* for more information on the System Dump Analyzer Utility.

6.6 Obtaining Information About a Multiprocessing System

Use the SHOW CPU command to display information about the status and configuration of a VMS multiprocessing system. The SHOW CPU command displays three levels of information:

Level	Command Example	Display Contents
Summary	SHOW CPU	Indicates which processor is the primary, which are configured, and which are active; displays the minimum revision levels for processors in the system and the setting of the MULTIPROCESSING SYSGEN parameter; and indicates whether multiprocessing is enabled and whether the presence of a uniprocessing device driver in the system prohibits its functioning as a multiprocessor.
Brief	SHOW CPU/BRIEF	Produces information from the summary display; lists the current CPU state and the current process (if any) for each configured processor.
Full	SHOW CPU/FULL	Produces information from the summary display; lists the current CPU state, current process (if any), revision levels, and capabilities for each configured processor; indicates which processes can be executed only on certain processors; lists the names of uniprocessing device drivers present in the system (if any) that prohibit its function as a multiprocessor.

6.7 Obtaining Information About a Vector Processing System

You can obtain information about the status of the vector processing system and the use of the system by individual processes through various means, including:

- The DCL lexical functions F\$GETJPI and F\$GETSYI
- The DCL command SHOW CPU
- The DCL commands SHOW PROCESS and LOGOUT/FULL
- The Accounting Utility
- The Monitor Utility

Gathering System Information

6.7 Obtaining Information About a Vector Processing System

6.7.1 DCL Lexical Functions F\$GETJPI and F\$GETSYI

The DCL lexical function F\$GETJPI accepts the following items and returns the corresponding information regarding the vector status of a specified process:

Item	Return Type	Information Returned
FAST_VP_SWITCH	Integer	Number of times this process has issued a vector instruction that resulted in an inactive vector processor being enabled without the expense of a vector context switch
SLOW_VP_SWITCH	Integer	Number of times this process has issued a vector instruction that resulted in an inactive vector processor being enabled with a full vector context switch
VP_CONSUMER	Boolean	Flag indicating whether the process is a vector consumer
VP_CPUTIM	Integer	Total amount of time the process has accumulated as a vector consumer

The DCL lexical function F\$GETSYI accepts the following items and returns the corresponding information regarding the status of the vector processing system:

Item	Return Type	Information Returned
VP_NUMBER	Integer	Number of vector processors in the system
VP_MASK	Integer	Mask indicating which processors in the system have vector coprocessors
VECTOR_EMULATOR	Integer	Flag indicating the presence of the VAX vector instruction emulator facility (VVIEF) in the system

See the *VMS DCL Dictionary* for additional information about the DCL lexicals F\$GETJPI and F\$GETSYI.

6.7.2 SHOW CPU Command

The SHOW CPU/FULL command lists the capabilities of the specified CPU. The manager of a VAX vector processing system may issue this command to determine the presence of the vector capability in the system prior to executing a STOP/CPU command.

See the *VMS DCL Dictionary* for additional information about the SHOW CPU command.

6.7.3 SHOW PROCESS and LOGOUT/FULL Commands

If the target process has accrued any time as a vector consumer scheduled on a vector-present processor, the DCL commands SHOW PROCESS and LOGOUT /FULL display the elapsed vector CPU time and the charged vector CPU time, respectively.

To accumulate vector CPU time, a process must be a vector consumer (that is, require the system vector capability) and be scheduled on a vector-present processor. VMS still charges the vector consumer vector CPU time even if, when scheduled on the vector-present processor, it does not actually use the vector CPU. Note that, because scalar consumers and marginal vector consumers do not use the vector CPU, they do not accrue vector CPU time even when scheduled on a vector-present processor.

Gathering System Information

6.7 Obtaining Information About a Vector Processing System

See the *VMS DCL Dictionary* for additional information about the SHOW PROCESS and LOGOUT/FULL commands.

6.7.4 Vector Processing Support Within the VMS Accounting Utility

In its full listing format, the VMS Accounting Utility displays the vector CPU time accumulated by a process or an image during its lifespan.

A process accumulates vector CPU time while it is a vector consumer (that is, requiring the system vector capability) and it is scheduled on a vector-present processor. VMS still charges the vector consumer vector CPU time even if, when scheduled on the vector-present processor, it does not actually use the vector CPU. Note that, because scalar consumers and marginal vector consumers do not use the vector CPU, they do not accrue vector CPU time even when scheduled on a vector-present processor.

An image accrues vector CPU time while it is executing within the context of a vector consumer on a vector-present processor. Because VMS marks all processes, including vector consumers, as scalar consumers at image rundown, it is impossible for an image that issues only scalar instructions to accumulate vector CPU time.

The ACCOUNTING command's /SORT qualifier accepts the VECTOR_PROCESSOR keyword and sorts the accounting records according to ascending or descending vector CPU time. Its /REPORT qualifier also accepts the VECTOR_PROCESSOR keyword and produces a summary report of vector CPU usage.

The *VMS Accounting Utility Manual* provides a complete description of the VMS Accounting Utility.

6.7.5 Vector Support Within the VMS Monitor Utility

The VMS Monitor Utility, in its VECTOR class output, displays the number of 10-millisecond clock ticks per second in which one or more vector consumers have been scheduled on each currently configured vector processor in the system. Because VMS schedules vector consumers only on those processors identified as "vector present," the VECTOR class output never displays vector CPU time for those processors that are "vector absent."

Note that, because vector consumers may use either the vector CPU or scalar CPU components, or both, of a vector-present processor, the vector CPU time in the VECTOR class display is not a strict measure of the actual usage of the processor's vector CPU component. Rather, it indicates the time during which a scheduled vector consumer has reserved both vector CPU and scalar CPU components of the vector-present processor for its own exclusive use. The MONITOR MODES command initiates monitoring of the VECTOR class, and displays the number of 10-millisecond clock ticks per second in which one or more vector consumers have been scheduled on each currently configured vector processor.

A

Aborting job state, 5-78
Access Control Lists
 See ACLs
Accounting
 See Accounting Utility (ACCOUNTING)
Accounting Utility (ACCOUNTING), 6-2
 disabling, 6-2
 enabling, 6-2
 log file, 6-1, 6-2
 closing, 6-3
 error, 6-3
 purging, 6-3
 records
 sorting, selecting, and reporting, 6-3
 record types, 6-3
 report formats, 6-4
 vector processing support, 6-17
ACCOUNTNG.DAT file, 6-2
ACLs
 on public volumes, 2-2
 on queues, 5-23
ACP cache parameters, 3-9
ACP_DINDXCACHE parameter, 2-6
Activating an autostart queue, 5-56, 5-68
 relationship with starting an autostart queue,
 5-5
Active set
 displaying, 6-15
Aligning preprinted forms, 5-83, 5-84
Aligning queue status, 5-71
Alignment data, 5-84
Analyze/Audit Utility, 6-1
Analyze/Disk_Structure Utility, 4-4
ANALYZE/ERROR_LOG command, 6-6
Assigning characteristics to a queue, 5-26
ASSIGN/MERGE command, 5-76
Audit log file
 See Security audit log file
Audit server process, 6-13
Automatic start
 See also Autostart
 of queue manager, 5-55
Automatic volume labeling (AVL), 3-5
Automatic volume recognition (AVR), 3-5

Automatic volume switching, 3-5
Autostart
 description, 5-5
 disabling, 5-63, 5-74
 before shutting down a node, 5-74
 enabling, 5-5, 5-61, 5-68, 5-69
 enabling when creating queues, 5-58
 recommended use, 5-11, 5-13, 5-17
 using with LAT queues, 5-13
Autostart inactive queue status, 5-71
Autostart queue
 activating, 5-5, 5-56, 5-68
 creating, 5-56
 creating and activating, 5-56
 preventing from starting, 5-74
 recommended use, 5-11, 5-13, 5-17
 relationship between activating and starting,
 5-5
 starting, 5-68, 5-69
 starting in startup command procedure, 5-61
 troubleshooting, 5-90
 using with LAT printers, 5-13
Availability
 See also Reliability
 of devices
 OPCOM message, 3-10
 of queue manager, 5-55, 5-86
 of queues, 5-5, 5-56
Available queue status, 5-71

B

Backup Utility (BACKUP)
 batch mode, 3-6
 CONTINUE option, 3-8
 operator assistance requests, 3-6
 QUIT option, 3-8
 recovering from errors, 3-7
 relationship with OPCOM, 3-6
 RESTART option, 3-8
Bad Block Locator Utility, 4-5
Banner pages
 commands used with, 5-32
 file, 5-32
 job, 5-32
Batch execution queue
 See also Execution queue

Batch execution queue (Cont.)
 description, 5-4

Batch job
 accounting, 6-2
 allowing to complete before stopping a queue,
 5-75
 changing scheduling priority, 5-80
 controlling, 5-76
 deleting, 5-82
 executing, 5-4
 holding and releasing, 5-79
 job card, 5-94
 modifying, 5-78
 monitoring, 5-77
 requeuing an executing, 5-80
 requeuing a pending, 5-81
 retaining in a queue, 5-81
 scheduling, 5-5
 states, 5-78

Batch queue
 allowing jobs to complete before stopping, 5-75
 control commands, 5-67
 creating, 5-56
 deleting, 5-76
 for memory constrained systems, 5-30
 optimizing for SORT, 5-30
 options, 5-17
 controlling job performance and resources,
 5-27
 qualifiers for specifying, 5-18
 restricting access, 5-21
 retaining jobs, 5-24
 using characteristics, 5-25
 pausing, 5-73
 starting, 5-56
 status, 5-71
 stopping, 5-73, 5-74
 stopping before shutting down a node, 5-75

Batch queuing system
 sample configurations, 5-7 to 5-11
 single queue, 5-8
 steps for setting up, 5-7
 using generic queues in a VAXcluster, 5-9
 with specialized queues, 5-8

Break-in attempts
 auditing, 6-13

Burst bar, 5-32

Burst page, 5-32
 file, 5-34
 job, 5-32

Busy queue status, 5-71

C

Cache
 disk volume information, 3-9

Cancelling characteristics on a queue, 5-26

Card
 decks, 5-94
 defective, 5-95

Card reader
 operating, 5-93, 5-95
 tending, 5-95
 translation modes, 5-95

Changing scheduling priority for a batch or print
 job, 5-80

Changing the DEFAULT form, 5-43

Characteristics
 See Queue characteristics, 5-25

Closed queue status, 5-71

Closing a queue, 5-73

Closing the accounting log file, 6-3

Closing the operator log file, 6-7

Command procedure
 to invoke EXCHANGE, 2-9

Configurations
 sample batch queuing system, 5-7 to 5-11
 sample print queuing system, 5-11 to 5-17

Console volumes
 transferring information to and from, 2-8

Continuation volumes
 in volume set, 3-5

CONTINUE option, 3-8

Controlling of page and line, 5-51

Creating a generic queue, 5-57

Creating an execution queue, 5-56
 autostart, 5-56
 nonautostart, 5-57

Creating a public volume, 2-8

Creating a queue database, 5-54

Creating a volume set, 2-6

Creating log files
 operator log file, 6-7, 6-11
 security audit log file, 6-14

D

Database
 queue
 See Queue database

Data card deck, 5-94

Default form, 5-43

DEFAULT form, 5-43

DEFINE/CHARACTERISTIC command, 5-26

DEFINE/FORM command, 5-41
 for controlling line overflow, 5-45

Defining a form, 5-41

DELETE/CHARACTERISTIC command, 5-27

DELETE/ENTRY command, 5-90

- DELETE/FORM command, 5-44
- DELETE/QUEUE command, 5-76
- Deleting
 - a characteristic, 5-27
 - problems, 5-91
 - a form, 5-44
 - problems, 5-91
 - a job, 5-82
 - a queue, 5-76
 - problems, 5-91
- Despooling a spooled output device, 5-53
- Detached jobs
 - accounting, 6-2
- Device
 - See also Output device
 - availability
 - OPCOM message, 3-10
 - mounting volumes, 3-3
 - setting characteristics, 5-51
 - in command procedure, 5-54
 - setting up output, 5-50
 - spooled, 5-52
 - testing, 5-53
 - status of, 6-7
 - status report, 6-7
- Device control library, 5-45 to 5-50
 - assigning to a queue, 5-47
 - order of module output, 5-49
 - procedure for using, 5-46
 - sample commands, 5-49
 - setting up, 5-45
- Device control module
 - adding, 5-47, 5-92
 - creating, 5-47
 - creating a form for, 5-48
 - deleting, 5-47, 5-92
 - inserting into a library, 5-47
 - listing, 5-48
 - naming, 5-48
 - order of output, 5-49
 - page setup, 5-46, 5-48
 - requesting with PRINT command, 5-48
 - reset, 5-46, 5-48
 - when queue is started, 5-48
 - sample commands, 5-49
 - setting up, 5-46
 - setup, 5-46, 5-48
 - specifying, 5-46
 - storing, 5-47
 - troubleshooting, 5-92
 - types, 5-45
 - using with forms, 5-46
- Device unavailable queue status, 5-71
- DISABLE AUTOSTART/QUEUES command, 5-74
 - entering before shutting down a system, 5-75
 - relationship to STOP/QUEUES/ON_NODE, 5-75
- Disabling accounting, 6-2
- Disabling autostart, 5-63, 5-74
 - before shutting down a node, 5-74
- Disabling OPAO
 - by default, 3-1
- Disabling operator classes, 3-2
- Disk
 - mounting
 - for queue database files, 5-54
 - See also Disk volumes, 2-6
 - system
 - See System disk
- Disk commands
 - INITIALIZE, 3-4
 - /CLUSTER_SIZE qualifier, 2-4
 - /HEADERS qualifier, 2-4
 - /INDEX qualifier, 2-5
 - /MAXIMUM_FILES qualifier, 2-5
 - /SYSTEM qualifier, 2-4
 - /WINDOWS qualifier, 2-5
 - MOUNT
 - /ACCESSED qualifier, 2-6
 - /ASSIST qualifier, 2-7
 - /BIND qualifier, 2-2, 2-6
 - /CLUSTER qualifier, 2-5, 2-6
 - /COMMENT qualifier, 2-7, 3-3
 - /EXTENSION qualifier, 2-7
 - /MOUNT_VERIFICATION qualifier, 2-7
 - /SHARE qualifier, 2-8
 - /SYSTEM qualifier, 2-5, 2-8
 - /WINDOWS qualifier, 2-8
 - SET DIRECTORY
 - /VERSION_LIMIT qualifier, 2-1
- Disk quota, 2-1, 4-2
 - ensuring accuracy, 4-3
 - exceeding, 4-2
 - operation, 4-2
 - suspension, 4-3
- Disk space
 - conserving, 4-1
 - management, 4-1 to 4-6
 - managing on small systems, 2-1
- Disk structure errors
 - analyzing, 4-4
- Disk volumes, 3-4
 - analyzing disk structure errors, 4-4
 - analyzing media errors, 4-5
 - conserving space, 4-1
 - console, 2-8
 - creating Files-11 structure, 2-4
 - disk quota operations, 4-2
 - dismounting, 3-9
 - formatting, 2-3
 - handling error conditions, 4-4
 - initializing, 2-4, 3-4
 - guidelines, 2-4
 - integrity, 3-9
 - load balancing, 2-2

- Disk volumes (Cont.)
 - loading, 3-4
 - monitoring operations, 6-7
 - mounting, 3-4
 - mounting clusterwide, 2-6
 - performance, 2-6
 - public, 2-1, 2-2
 - public access to, 2-1
 - removing before dismounting, 3-9
 - setting file expiration date, 4-3
 - transferring information to and from foreign volumes, 2-8
 - write-locked
 - dismounting, 3-9
 - write-locking, 3-9
- DISMOUNT command
 - mount verification cancellation, 3-12
- Displaying a form assigned to a queue, 5-44
- Displaying characteristics assigned to a queue, 5-26
- Displaying defined characteristics, 5-26
- Displaying defined forms, 5-42
- Displaying information about a queue, 5-69
- Distributed printing, 5-7
- DOS-11
 - volume, 2-8
- DQS
 - See VAX Distributed Queuing System (DQS), 5-7
- Drivers
 - device status messages, 6-7
- DUMPBUG system parameter, 6-15
- Dump file
 - job controller, 5-66
 - queue manager, 5-66
 - system, 6-14
 - automatic creation of, 6-14
 - size, 6-15
 - use of, 6-1

E

- ENABLE AUTOSTART/QUEUES command, 5-58, 5-61, 5-69
 - in startup command procedure, 5-58
 - recommended use, 5-61
- Enabling accounting, 6-2
- Enabling an operator terminal
 - OPCOM message, 3-1
- Enabling autostart, 5-5, 5-61, 5-68, 5-69
 - in startup command procedure, 5-58
 - when creating queues, 5-58
- Enabling OPA0
 - by default, 3-1
- Enabling operator classes, 3-2
- End of tape (EOT) marker
 - when reached, 3-5
- ERRFMT process, 6-4

- ERRFMT process (Cont.)
 - restarting, 6-5
 - starting, 6-4
- ERRLOG.SYS file, 6-4
- ERRORLOGBUFFERS parameter, 6-15
- Error log file, 6-1, 6-4
 - maintaining, 6-6
 - messages, 6-4
 - printing, 6-6
 - writing report, 6-6
- Error Logging Facility, 6-4
- Error Log Utility, 6-4
- Error report
 - use of, 6-5
- Errors
 - BACKUP
 - using to recover from, 3-7
 - disk structure
 - analyzing, 4-4
 - repairing, 4-4
 - excessive, 6-5
 - handling on disk volumes, 4-4
 - in writing to the accounting log file, 6-3
 - media
 - analyzing, 4-5
 - mounting disk, 2-5
- Exchange Utility (EXCHANGE), 2-8
- DIRECTORY command, 2-9
 - invoking
 - in command procedure, 2-9
 - interactively, 2-9
 - MOUNT command, 2-9
- Executing job state, 5-69, 5-78, 5-82
- Execution queue
 - activating an autostart, 5-5, 5-56, 5-68
 - creating, 5-56
 - description, 5-4
 - relationship to generic queue, 5-4
 - starting
 - autostart, 5-5, 5-61, 5-68, 5-69
 - nonautostart, 5-57, 5-61, 5-68
- Expiration date
 - file, 4-3
- Extension size
 - of disk, 2-7

F

- F\$GETJPI lexical function
 - using to gather vector processing information, 6-16
- F\$GETQUI lexical function
 - using to gather queue information, 5-72
- F\$GETSYI lexical function
 - using to gather vector processing information, 6-16
- Failover
 - forcing for queue manager, 5-63

Failover (Cont.)

- of queue manager, 5-2, 5-3
- of queues, 5-5, 5-56

Failover list

- queue manager, 5-2
 - insufficient, 5-86
- specifying for an autostart queue, 5-56
- specifying for queue manager, 5-55, 5-86

Failure

- login
 - accounting, 6-2
- mounting disk, 2-5
- notification for MOUNT command, 2-7
- system
 - anticipating, 6-5, 6-7
 - using dump file to determine cause of, 6-1

File

- expiration date, 4-3
- on public volumes, 2-1
- public, 2-2

File banner pages, 5-32

- See also Job banner pages

File protection

- applying to public disk volumes, 2-2

Files-11 disk

- using Exchange Utility (EXCHANGE) to transfer data, 2-8

Files-11 structure

- creating, 2-4

Flag page, 5-32

- file, 5-34
- job, 5-32

Format

- of accounting reports, 6-4

Formatting disks, 2-3

Forms

- See also Preprinted forms
- assigning default for a queue, 5-43
- associating with jobs and queues, 5-41
- changing DEFAULT, 5-43
- commands used with, 5-40
- default, 5-43
- DEFAULT, 5-43
- defining, 5-41, 5-55
- deleting, 5-44
- description, 5-40
- displaying, 5-42
- displaying assigned, 5-44
- mounting, 5-43
- problems deleting, 5-91
- procedure for using, 5-41
- using to control line overflow, 5-40
- using to control page width, length and margin size, 5-40
- using to control paper stock, 5-40
- using to format jobs, 5-40
- using to specify setup modules, 5-40
- using to specify sheet feed paper, 5-40

G

Generic queue

- batch, 5-4
 - recommended use, 5-9
- creating, 5-57
- description, 5-3, 5-4
- in a VAXcluster environment, 5-4
- output, 5-4, 5-14, 5-15
 - recommended use, 5-14
- relationship to execution queue, 5-4
- specifying target execution queues, 5-57

H

Holding a job, 5-79

Holding job state, 5-69, 5-78, 5-88

I

I/O drivers

- device status messages, 6-7

Idle queue status, 5-71

Image activation

- accounting, 6-2

INITIALIZE command, 2-4, 3-4

- See also Disk commands, 2-4

INITIALIZE/QUEUE command, 5-68

- /AUTOSTART_ON qualifier, 5-56
- /BLOCK_LIMIT qualifier, 5-30
- CHARACTERISTICS qualifier, 5-26
- /DEFAULT qualifier

- for banner pages, 5-32

- for controlling page overflow, 5-45

- for forms, 5-43

/FORM_MOUNTED qualifier, 5-43

/GENERIC qualifier, 5-57

/NOCHARACTERISTICS qualifier, 5-26

/NORETAIN qualifier, 5-25

/ON qualifier, 5-57

/RETAIN qualifier, 5-24

/SCHEDULE qualifier, 5-30

/SEPARATE qualifier

- for banner pages, 5-32

- for reset modules, 5-46

specifying queue options with, 5-18

/START qualifier

- for activating an autostart queue, 5-56, 5-68

- for starting a nonautostart queue, 5-57

/WSEXTENT qualifier, 5-30

Initializing a disk volume, 2-4

- guidelines, 2-4

Initializing a queue, 5-56

Initializing a tape volume, 3-6

Input symbiont

- running interactively, 5-96

- Interactive jobs
 - accounting, 6-2
- IPC (Interprocess communication)
 - use with queue manager, 5-1
- IPC (Interrupt Priority C)
 - using to cancel mount verification, 3-13

J

Job

- See also Batch job
- See also Output job
- accounting, 6-2
- changing scheduling priority, 5-80
- controlling print position and alignment, 5-83, 5-84
- deleting, 5-82
- holding, 5-88
- holding and releasing, 5-79
- merging, 5-75
- modifying, 5-78
- moving from one queue to another, 5-76
- pending, 5-88
- requeuing an executing, 5-80
- requeuing a pending, 5-81
- retaining in a queue, 5-81
- suspending, 5-83
- Job banner pages, 5-32
 - See also File banner pages
- JOB card, 5-94
- Job controller
 - and batch jobs, 5-1
 - communication with queue manager, 5-1
 - dump file, 5-66
 - function of, 5-3
- Job retention
 - changing for a job, 5-81
 - specifying for a job, 5-24
 - specifying for a queue, 5-24
- Job scheduling, 5-5
 - batch, 5-5
 - output, 5-5
 - based on job size, 5-30
- Job scheduling priority, 5-6
- Job states, 5-78
- Job status
 - holding, 5-88
 - pending, 5-88
- Journal file, 5-3
 - See also Queue database
 - changing location, 5-66
 - location, 5-54, 5-66
 - where location is stored, 5-3

L

- LAT\$SYSTARTUP command procedure, 5-51
- LATCP commands
 - for setting up ports, 5-51
- LAT printers
 - sample configuration, 5-13
 - setting up, 5-51
 - spooling, 5-6
 - troubleshooting, 5-87
 - using PRTSMB symbiont, 5-87
- LAT site-specific command procedure, 5-51
- LAT symbiont, 5-4, 5-87
- LATSYM symbiont, 5-4, 5-87
- Library
 - See Device control library
- Line overflow
 - controlling, 5-44
- Location of master file, 5-55
- Location of queue and journal files, 5-3
- Log files
 - accounting, 6-1, 6-2
 - closing, 6-3
 - error, 6-3
 - purging, 6-3
 - error, 6-1, 6-4
 - maintaining, 6-6
 - printing, 6-6
 - operator, 3-1, 6-1, 6-7
 - closing, 6-7
 - creating, 6-7, 6-11
 - device status message, 6-7
 - enabling and disabling classes, 6-11
 - information about system parameter changes, 6-10
 - initialization message, 6-7
 - location, 6-11
 - maintaining, 6-12
 - printing, 6-13
 - purging, 6-12
 - sample, 6-7
 - security alarm messages, 6-10
 - setting up, 6-11
 - use of, 6-7
 - using to troubleshoot queue manager, 5-86
 - security audit, 6-1, 6-13
 - creating, 6-14
 - review policy, 6-14
- Logical names
 - defining in SYLOGICALS.COM, 3-1
 - OPC\$LOGFILE_CLASSES, 6-12
 - OPC\$LOGFILE_ENABLE, 6-12
 - OPC\$LOGFILE_NAME, 6-11, 6-12
 - OPC\$OPA0_CLASSES, 3-2
 - OPC\$OPA0_ENABLE, 3-2
 - QMAN\$MASTER, 5-55, 5-65
 - requirement in a VAXcluster, 5-66

Logical names (Cont.)
SHUTDOWN\$DISABLE_AUTOSTART, 5-75
Logical queue
 assigning, 5-75
 description, 5-5
 recommended use, 5-5, 5-75
Login failure
 accounting, 6-2
LOGOUT/FULL command, 6-16
Lost file
 recovering, 4-4 to 4-5

M

Magnetic tape
 See Tape volume
Manager
 queue
 See Queue Manager
Mapping pointers
 resetting when mounting a disk, 2-8
Margin size
 specifying in forms, 5-40
Master file, 5-3
 See also Queue database
 changing location, 5-65
 how location is defined, 5-3
 information stored in, 5-3
 QMAN\$MASTER logical name, 5-55
 saving, 5-64
 specifying location, 5-55
Media errors
 analyzing, 4-5
Message
 OPCOM
 See OPCOM message
 user
 accounting, 6-2
Module
 device control
 See Device control module
Monitoring disk and tape operations, 6-7
Monitoring system activity, 6-2
Monitoring user requests, 6-7
Monitor Utility (MONITOR)
 analyzing disk use with, 2-2
 obtaining vector processing information with,
 6-17
MOUNT command, 2-5, 2-6, 2-7
 See also Disk commands
 /ACCESSED qualifier, 2-6
 /ASSIST qualifier, 2-7
 /BIND qualifier, 2-2, 2-6
 /CLUSTER qualifier, 2-5, 2-6
 /COMMENT qualifier, 2-7
 example, 3-3
 /EXTENSION qualifier, 2-7

MOUNT command (Cont.)
 /MOUNT_VERIFICATION qualifier, 2-7
 /SHARE qualifier, 2-8
 /SYSTEM qualifier, 2-5, 2-8
 /WINDOWS qualifier, 2-8
Mounted form, 5-41
Mounting a form, 5-43
Mounting disk volumes, 3-4
 holding queue database files, 5-54
 passing information to the operator, 2-7
Mounting public volumes, 2-5
Mounting tape volumes, 3-4
 volume sets, 3-5
Mounting volumes
 if device is unavailable, 3-3
 operator assistance, 2-7, 3-3
 substituting, 3-3
Mounting volume sets
 with automatic volume switching, 3-5
 without automatic volume switching, 3-5
Mount verification
 abort by dismount, 3-12
 aborted
 OPCOM message, 3-13
 canceling, 3-12, 3-13
 description, 3-9
 device off line, 3-10
 device write-lock, 3-11
 enabling and disabling on a volume, 2-7
 messages, 3-10
 time out, 3-12
 OPCOM message, 3-12
Moving jobs from one queue to another, 5-76
Moving queue database, 5-65
 master file, 5-65
 queue and journal files, 5-66
MTAACP process
 mount request, 3-5
Multiprocessing
 displaying information about, 6-15
MVTIMEOUT parameter, 3-12

N

Network job
 accounting, 6-2

O

OPA0, 3-1
 changing default setup, 3-1
 default setup, 3-1
OPC\$LOGFILE_CLASSES logical name, 6-12
OPC\$LOGFILE_ENABLE logical name, 6-12
OPC\$LOGFILE_NAME logical name, 6-11, 6-12
OPC\$OPA0_CLASSES logical name, 3-2
OPC\$OPA0_ENABLE logical name, 3-2
OPCOM (operator communication manager)

OPCOM (operator communication manager)
(Cont.)

- log file, 6-1, 6-7
 - sample, 6-7
 - use of, 6-7
 - mount verification, 3-9
 - messages, 3-10
 - recovering from BACKUP errors, 3-7
 - relationship with BACKUP, 3-6
 - restarting, 3-2, 6-12
 - using, 3-1
- OPCOM message
- appearing in operator log file, 6-7
 - continuation volume request, 3-5
 - device status message, 6-7
 - enabling and disabling of operator terminal, 6-8
 - initialization message, 6-7
 - mount request message, 3-4
 - mount verification aborted message, 3-13
 - mount verification time out message, 3-12
 - operator reply message, 6-9
 - operator terminal enabled message, 3-1
 - security alarm message, 6-10
 - system parameter change message, 6-10
 - types of, 6-7
 - user request message, 3-2, 6-9
 - if operator terminals are disabled, 3-3
 - volume mount and dismount message, 6-9
- Operator
- passing information to when mounting a disk, 2-7
 - requesting assistance from, 3-3
 - sending requests to, 3-1
- OPERATOR.LOG file, 6-7
- Operator assistance
- requesting, 3-2
 - with MOUNT, 2-7, 3-3
- Operator classes
- changing default setup, 3-2
 - default setup, 3-2
 - enabling and disabling for log file, 6-11
- Operator commands
- INITIALIZE, 3-4
 - REPLY
 - /ABORT qualifier, 3-3
 - /BLANK_TAPE qualifier, 3-6
 - /DISABLE qualifier, 3-2
 - /ENABLE qualifier, 3-1, 6-8
 - /INITIALIZE_TAPE qualifier, 3-6
 - /LOG qualifier, 6-7, 6-11
 - /PENDING qualifier, 3-3
 - /TO qualifier, 3-3, 3-5, 3-8
 - REQUEST
 - /REPLY qualifier, 3-2
 - /TO qualifier, 3-2
- Operator function
- assisting batch-oriented backup jobs, 3-6

Operator function (Cont.)

- mounting a device, 3-4
 - mounting and dismounting volumes, 3-3
 - mounting volume sets, 3-5
 - user request, 3-1, 3-3, 3-4
- Operator log file, 6-1, 6-7
- See also OPCOM, OPCOM message
- closing, 6-7
 - creating, 6-7, 6-11
 - device status message, 6-7
 - enabling and disabling classes, 6-11
 - example, 6-7
 - information about system parameter changes, 6-10
 - initialization message, 6-7
 - location, 6-11
 - maintaining, 6-12
 - messages appearing in, 6-7
 - printing, 6-13
 - purging, 6-12
 - recording messages in, 3-1
 - sample, 6-7
 - security alarm messages, 6-10
 - setting up, 6-11
 - use of, 6-7
 - using to troubleshoot queue manager, 5-86
- Operator terminal
- See also OPAO
- disabling, 3-2
 - enabled
 - OPCOM message, 3-1
 - enabling, 3-1
 - enabling and disabling, 6-8
- Optimizing batch queues for SORT, 5-30
- Options
- See Backup Utility (BACKUP)
- See Queue options
- Output device
- setting characteristics, 5-51
 - in command procedure, 5-54
 - setting up, 5-50
 - spooled, 5-6
 - recommended use, 5-6
 - sample configuration, 5-16
- Output execution queue
- See also Execution queue
- description, 5-4
 - printer, 5-4
 - server, 5-4
 - terminal, 5-4
- Output job
- accounting, 6-2
 - aligning forms, 5-84
 - allowing to complete before stopping a queue, 5-75
 - changing scheduling priority, 5-80
 - controlling, 5-76

Output job (Cont.)

- controlling print position and alignment, 5-83, 5-84
- deleting, 5-82
- holding and releasing, 5-79
- modifying, 5-78
- monitoring, 5-77
- requeuing an executing, 5-80
- requeuing a pending, 5-81
- resuming printing, 5-83, 5-84
- retaining in a queue, 5-81
- scheduling, 5-5
- scheduling based on job size, 5-30
- states, 5-78
- suspending, 5-83

Output queue

- allowing jobs to complete before stopping, 5-75
- assigning a default form, 5-43
- cancelling assigned characteristics, 5-26
- changing DEFAULT form, 5-43
- control commands, 5-67
- controlling line overflow in forms, 5-40
- creating, 5-56
- defining a form, 5-41
- deleting, 5-76
- mounting a form on, 5-43
- options, 5-17, 5-30
 - controlling page and line overflow, 5-44
 - qualifiers for specifying, 5-18 to 5-20
 - restricting access, 5-21
 - retaining jobs, 5-24
 - scheduling jobs based on size, 5-30
 - using banner pages, 5-31
 - using characteristics, 5-25
 - using device control libraries, 5-45
 - using forms, 5-40
- order of device control module output, 5-49
- pausing, 5-73, 5-83
 - to align position of print for preprinted forms, 5-83, 5-84
 - to change position of print, 5-83
- rerouting jobs in, 5-75
- specifying page and margin size in forms, 5-40
- starting, 5-56
- status, 5-71
- stopping, 5-73, 5-74
- stopping before shutting down a node, 5-75
- troubleshooting a stalled, 5-90

Output queuing system

- for LAT printers, 5-13
- for mixed printers, 5-12
- for multiple printers of the same kind, 5-14
- in VAXcluster environment, 5-15
- sample configurations, 5-11 to 5-17
- single printer, 5-11
- steps for setting up, 5-7
- using spooled printers, 5-16

Overflow

Overflow (Cont.)

- controlling of page and line, 5-44
- line
 - using forms to control, 5-40
- setting devices to control, 5-51

P

PAGEFILE.SYS file, 6-14

Page overflow

- controlling, 5-44

Page setup module, 5-46

- specifying forms, 5-40

Page width and length

- specifying in forms, 5-40

Paper stock

- See Stock

PASSWORD card, 5-94

Paused queue status, 5-71

Pausing an output queue, 5-83

- to align position of print for preprinted form, 5-83, 5-84

- to change position of print, 5-83

Pausing queue status, 5-71

Pending jobs

- requeuing, 5-81
- troubleshooting, 5-88

Pending job state, 5-69, 5-78, 5-80, 5-81, 5-82, 5-88

Performance

- disk, 2-6
- load balancing on public volumes, 2-2
- testing for disks, 2-2

Ports

- setting up LAT, 5-51

Position of print

- aligning for preprinted forms, 5-83, 5-84
- changing, 5-83

PostScript printing, 5-6

Preprinted forms

- aligning, 5-83, 5-84

Preventing autostart queues from starting, 5-74

PRINT command

- /BURST qualifier, 5-32
- /CHARACTERISTIC qualifier, 5-25
- /FEED qualifier, 5-45
- /FLAG qualifier, 5-32
- /FORM qualifier, 5-40
 - for specifying setup and page setup modules, 5-48
- /NOPASSALL qualifier, 5-45
- preventing users from executing, 5-73
- /PRIORITY qualifier, 5-6
- processing of, 5-1
- /RETAIN qualifier, 5-24, 5-81
- /SETUP qualifier, 5-48
- /TRAILER qualifier, 5-32

Printer

Printer (Cont.)

- controlling functions, 5-45
- LAT, 5-51
- setting characteristics, 5-51
- setting up, 5-50
- spooled, 5-6
 - recommended use, 5-6
- troubleshooting, 5-87

Print forms

- See Forms

Printing

- printing, 5-6
- remotely, 5-7
- resuming at a specified position, 5-83, 5-84

Printing job state, 5-78

Printing log files

- the error log file, 6-6
- the operator log file, 6-13

Print job

- See Output job

Print queue

- See Output queue

Print symbiont

- See Symbiont

Priority

- job scheduling, 5-6
- scheduling
 - changing for a batch or print job, 5-80

Problems

- See also Troubleshooting
- detecting in error log file, 6-4
- detecting in operator log file, 6-7, 6-10
- detecting in security audit log file, 6-13
- detecting in system dump file, 6-1

Processing job state, 5-78

Process termination

- accounting, 6-2

Protection

- applying to public disk volumes, 2-2
- applying to queues, 5-21 to 5-23

PRTSMB symbiont, 5-4, 5-87

Public files and volumes, 2-2

Public volumes

- access to, 2-1
- creating, 2-8
- creating Files-11 structure, 2-4
- initializing, 2-4
 - guidelines, 2-4
- load balancing, 2-2
- mounting, 2-5
- on large configurations, 2-2
- on small configurations, 2-1
- planning, 2-1
- testing disk performance, 2-2
- using volume sets, 2-2

Purging the accounting log file, 6-3

Purging the operator log file, 6-12

Q

QMAN\$MASTER.DAT

- See Master file

QMAN\$MASTER logical name, 5-55, 5-65

- defining in a VAXcluster environment, 5-65

Queue

- activating an autostart, 5-56, 5-68
- activating and starting an autostart, 5-5
- allowing jobs to complete before shutdown, 5-75
- assigning a default form, 5-43
- assigning device control libraries, 5-47
- autostart
 - recommended use, 5-11, 5-13, 5-17
 - availability of, 5-5, 5-56
- batch execution
 - See Batch queue
- cancelling characteristics
 - assigned to a queue, 5-26
- changing DEFAULT form, 5-43
- changing options on, 5-72
- closing, 5-73
- command
 - See Queue command
- creating, 5-56, 5-68
 - autostart execution, 5-56
 - generic, 5-57
 - nonautostart execution, 5-57
- creating and activating an autostart, 5-56
- defining a characteristic, 5-26
- defining a form, 5-41
- deleting, 5-76
- deleting a job in, 5-82
- displaying information about, 5-69
- execution
 - See Execution queue
- failover of, 5-56
- generic batch, 5-9
 - See Generic Queue
- generic output, 5-14
 - See Generic Queue
- holding and releasing jobs, 5-79
- initializing, 5-68
- merging, 5-75
- modifying, 5-72
- monitoring, 5-69
- mounted form, 5-41
- mounting a form on, 5-43
- moving jobs from one queue to another, 5-76
- output execution
 - See Output execution queue
- pausing, 5-73
- print
 - See Queue, output
- problems deleting, 5-91

Queue (Cont.)

- protection, 5-21 to 5-23
 - reinitializing an existing, 5-72
 - requeuing an executing job, 5-80
 - requeuing a pending job, 5-81
 - spooling output devices, 5-52
 - starting, 5-56
 - autostart, 5-68, 5-69
 - starting an autostart in startup command procedure, 5-61
 - starting a nonautostart, 5-57, 5-68
 - starting a nonautostart in startup command procedure, 5-61
 - startup command procedure
 - sample, 5-61
 - stopping, 5-73
 - abruptly, 5-73
 - before shutdown, 5-75
 - smoothly, 5-73
 - stopping all, 5-74
 - types, 5-3
 - types of output execution, 5-4
 - using characteristics, 5-25
 - using F\$GETQUI, 5-72
 - using forms, 5-40
- ## Queue characteristics
- assigning to a queue, 5-26
 - cancelling assigned, 5-26
 - commands used with, 5-25
 - defining, 5-26, 5-55
 - deleting, 5-27
 - description, 5-25
 - displaying, 5-26
 - displaying assigned, 5-26
 - mismatch problems
 - troubleshooting, 5-89
 - problems deleting, 5-91
 - sample use of, 5-25
- ## Queue commands
- ASSIGN/MERGE, 5-76
 - DEFINE/CHARACTERISTIC, 5-26
 - DEFINE/FORM, 5-41
 - for controlling line overflow, 5-45
 - DELETE/CHARACTERISTIC, 5-27
 - DELETE/ENTRY, 5-90
 - DELETE/FORM, 5-44
 - DELETE/QUEUE, 5-76
 - DISABLE AUTOSTART/QUEUES
 - entering before shutting down a system, 5-75
 - relationship to STOP/QUEUES/ON_NODE, 5-75
 - ENABLE AUTOSTART/QUEUES, 5-58, 5-61, 5-68, 5-69
 - in startup command procedure, 5-58
 - INITIALIZE/QUEUE, 5-68
 - /BLOCK_LIMIT qualifier, 5-30
 - /CHARACTERISTICS qualifier, 5-26

Queue commands

INITIALIZE/QUEUE (Cont.)

- /DEFAULT qualifier
 - for banner pages, 5-32
 - for controlling page overflow, 5-45
 - for forms, 5-43
 - /FORM_MOUNTED qualifier, 5-43
 - /NOCHARACTERISTICS qualifier, 5-26
 - /NORETAIN qualifier, 5-25
 - /RETAIN qualifier, 5-24
 - /SCHEDULE qualifier, 5-30
 - /SEPARATE qualifier
 - for banner pages, 5-32
 - for reset modules, 5-46
 - specifying queue options with, 5-18
 - /START qualifier
 - for activating an autostart queue, 5-56, 5-68
 - for starting a nonautostart queue, 5-57
 - /WSEXTENT qualifier, 5-30
- ## PRINT
- /BURST qualifier, 5-32
 - /CHARACTERISTIC qualifier, 5-25
 - /FEED qualifier, 5-45
 - /FLAG qualifier, 5-32
 - /FORM qualifier, 5-40
 - for specifying setup and page setup modules, 5-48
 - preventing users from executing, 5-73
 - /PRIORITY qualifier, 5-6
 - /RETAIN qualifier, 5-24, 5-81
 - /SETUP qualifier, 5-48
 - /TRAILER qualifier, 5-32
- sample setup, 5-58
- ## SET ENTRY, 5-78
- /AFTER qualifier, 5-80
 - /FORM qualifier, 5-89
 - /HOLD qualifier, 5-79
 - /PRIORITY qualifier, 5-80
 - /RELEASE qualifier, 5-79
 - /REQUEUE qualifier, 5-81, 5-90
 - /RETAIN qualifier, 5-81
- ## SET QUEUE, 5-72
- /BLOCK_LIMIT qualifier, 5-30
 - /CHARACTERISTIC qualifier, 5-90
 - CHARACTERISTICS qualifier, 5-26
 - /DEFAULT qualifier
 - for banner pages, 5-32
 - for controlling page overflow, 5-45
 - for forms, 5-43
 - /FORM_MOUNTED qualifier, 5-43
 - /NOCHARACTERISTICS qualifier, 5-26
 - /NORETAIN qualifier, 5-25
 - /PROTECTION qualifier, 5-23
 - /RETAIN qualifier, 5-24
 - /SCHEDULE qualifier, 5-30
 - /SEPARATE qualifier

Queue commands

SET QUEUE
 /SEPARATE qualifier (Cont.)
 for banner pages, 5-32
 for reset modules, 5-46
 specifying queue options with, 5-18
 /WSEXTENT qualifier, 5-30

SHOW ENTRY, 5-77

SHOW QUEUE, 5-69
 /CHARACTERISTIC qualifier, 5-26
 /FORM qualifier, 5-42
 for showing defined forms, 5-42
 /FULL qualifier
 for showing assigned form, 5-44
 for showing characteristics, 5-26
 for showing protection, 5-22, 5-23

START/QUEUE, 5-68
 /BACKWARD qualifier, 5-83
 /BLOCK_LIMIT qualifier, 5-30
 /CHARACTERISTICS qualifier, 5-26
 /DEFAULT qualifier
 for banner pages, 5-32
 for controlling page overflow, 5-45
 for forms, 5-43
 for activating an autostart queue, 5-68
 FORM_MOUNTED qualifier, 5-43
 /FORWARD qualifier, 5-84
 /NOCHARACTERISTICS qualifier, 5-26
 /NORETAIN qualifier, 5-25
 /RETAIN qualifier, 5-24
 /SCHEDULE qualifier, 5-30
 /SEARCH qualifier, 5-84
 /SEPARATE qualifier
 for banner pages, 5-32
 for reset modules, 5-46
 specifying queue options with, 5-18
 /TOP_OF_FILE qualifier, 5-84
 using to resume printing of a suspended job, 5-83
 /WSEXTENT qualifier, 5-30

START/QUEUE/MANAGER, 5-54, 5-62

STOP/QUEUE, 5-73, 5-83

STOP/QUEUE/MANAGER/CLUSTER, 5-63

STOP/QUEUE/NEXT, 5-68, 5-73

STOP/QUEUE/RESET, 5-68, 5-73

STOP/QUEUES/ON_NODE, 5-74
 entering before shutting down a system, 5-75
 relationship with DISABLE AUTOSTART /QUEUES, 5-75

SUBMIT
 /CHARACTERISTIC qualifier, 5-25
 preventing users from executing, 5-73
 /PRIORITY qualifier, 5-6
 /REATAIN qualifier, 5-81
 /RETAIN qualifier, 5-24
 used for device control libraries, 5-46
 used to manage queues, 5-67

Queue commands (Cont.)

 used to modify batch and print jobs, 5-78
 used to specifying queue options, 5-18
 used with banner pages, 5-32
 used with characteristics, 5-25
 used with forms and stock, 5-40

Queue configuration
 sample batch queuing system, 5-7 to 5-11
 sample commands for setting up, 5-58
 sample output queuing system, 5-11 to 5-17

Queue database
 See also Journal file
 See also Master file
 See also Queue file
 closing, 5-63
 creating, 5-54
 default location, 5-3
 description, 5-3
 determining location of master file, 5-54
 determining location of queue and journal file, 5-54
 function of, 5-3
 moving, 5-65
 moving master file, 5-65
 moving queue and journal files, 5-66
 restoring a damaged, 5-64
 saving, 5-64

Queue failover, 5-5

Queue file, 5-3
 See also Queue database
 changing location, 5-66
 location, 5-54, 5-66
 saving, 5-64
 where location is stored, 5-3

Queue files
 See Queue file or Queue database

Queue forms
 See Forms

Queue manager
 automatic restart, 5-2, 5-3, 5-55
 availability of, 5-55, 5-86
 description, 5-1
 dump file, 5-66
 failover, 5-2, 5-3
 forcing, 5-63
 list, 5-55, 5-86
 function of, 5-1
 moving to another node in a VAXcluster, 5-63
 restarting, 5-63, 5-65, 5-66
 role in processing print jobs, 5-4
 role in queuing process, 5-1
 role in starting active autostart queues, 5-5
 starting initially, 5-54
 stopping, 5-63
 submitting a Software Performance Report (SPR), 5-66
 troubleshooting, 5-85

Queue manager (Cont.)
 use of IPC (Interprocess Communication), 5-1

Queue name
 default for batch queue, 5-8
 default for print queue, 5-12
 defining, 5-56

Queue options, 5-17
 controlling job performance and resources, 5-27
 controlling page and line overflow, 5-44
 qualifiers for specifying, 5-18 to 5-21
 restricting access, 5-21
 retaining jobs, 5-24
 scheduling jobs based on size, 5-30
 using banner pages, 5-31
 using characteristics, 5-25
 using device control libraries, 5-45
 using forms, 5-40

Queue startup
 simplifying, 5-5

Queue status, 5-71
 determining, 5-87

Queuing process, 5-1

Queuing system
 components, 5-1
 sample configurations, 5-7 to 5-17
 steps for setting up, 5-7

QUIT option, 3-8

Quota
 disk
 See Disk quota

QUOTA.SYS file, 4-3

R

Read operation
 continuation volumes, 3-6

Rebuilding a disk volume, 4-3

Releasing a job, 5-79

Reliability
 using error reports to analyze, 6-5

Remote printing, 5-7

Remote queue status, 5-71

REPLY command
 /ABORT qualifier, 3-3
 /BLANK_TAPE qualifier, 3-6
 /DISABLE qualifier, 3-2
 /ENABLE qualifier, 3-1, 6-8
 /INITIALIZE_TAPE qualifier, 3-6
 /LOG qualifier, 6-7
 using to create a log file, 6-11
 /PENDING qualifier, 3-3
 /TO qualifier, 3-3, 3-5, 3-6, 3-8

Request
 for mounting volumes, 3-3
 OPCOM message, 3-2
 if operator terminals are disabled, 3-3

REQUEST command, 6-9

REQUEST command (Cont.)
 /REPLY qualifier, 3-2
 /TO qualifier, 3-2

Requeuing
 an executing job, 5-80
 a pending job, 5-81

Requirements
 defining logical name in a VAXcluster environment, 5-66
 location of queue and journal file, 5-66

Reset modules, 5-46

Resources
 system
 tracking, 6-2

Restarting OPCOM, 6-12

Restarting queue manager, 5-65

Restarting the ERRFMT process, 6-5

Restarting the queue manager, 5-63

RESTART option, 3-8

Restoring the queue database, 5-64

Resuming printing of an output job, 5-83, 5-84

Resuming queue status, 5-71

Retained job states, 5-24, 5-69, 5-78, 5-80, 5-81

Retaining jobs in a queue
 changing retention of a job, 5-81

Retention
 job, 5-24

RT-11 volume
 block-addressable, 2-8

S

Saving the queue database, 5-64

Scheduling
 of batch jobs, 5-5
 of print jobs, 5-5
 based on job size, 5-30
 priority
 changing for a batch or print job, 5-80

Security, 6-13
 messages in operator log file, 6-10
 protecting public volumes, 2-2
 protecting queues, 5-21 to 5-23

Security audit log file, 6-1, 6-13
 closing, 6-14
 creating, 6-14
 review policy, 6-14

SECURITY_AUDIT.AUDIT\$JOURNAL file, 6-13

Server queue, 5-4

Server queue status, 5-71

SET ACCOUNTING command
 /NEW_FILE qualifier, 6-3

SET AUDIT command, 6-13
 /SERVER=NEW_LOG qualifier, 6-14

SET ENTRY command, 5-78
 /AFTER qualifier, 5-80
 /FORM qualifier, 5-89
 /HOLD qualifier, 5-79

SET ENTRY command (Cont.)

- /PRIORITY qualifier, 5-80
- /RELEASE qualifier, 5-79
- /REQUEUE qualifier, 5-81, 5-90
- /RETAIN qualifier, 5-81

SET PRINTER command, 5-51

SET QUEUE command, 5-72

- /BLOCK_LIMIT qualifier, 5-30
- /CHARACTERISTIC qualifier, 5-90
- /CHARACTERISTICS qualifier, 5-26
- /DEFAULT qualifier
 - for banner pages, 5-32
 - for controlling page overflow, 5-45
 - for forms, 5-43
- FORM_MOUNTED qualifier, 5-43
- /NOCHARACTERISTICS qualifier, 5-26
- /NORETAIN qualifier, 5-25
- /PROTECTION qualifier, 5-23
- /RETAIN qualifier, 5-24
- /SCHEDULE qualifier, 5-30
- /SEPARATE qualifier
 - for banner pages, 5-32
 - for reset modules, 5-46
- specifying queue options with, 5-18
- /WSEXTENT qualifier, 5-30

SET TERMINAL command, 5-51

Setting up LAT printers, 5-51

Setting up printers, 5-50

Setting up terminals and output devices, 5-51

Setup module

- specifying in forms, 5-40

Setup modules, 5-46

Sharing a disk volume, 2-8

Sheet feed paper

- specifying in forms, 5-40

SHOW ACCOUNTING command, 6-2

SHOW CPU command, 6-15, 6-16

SHOW ENTRY command, 5-77

SHOW PROCESS command, 6-16

SHOW QUEUE command, 5-69

- /ALL_JOBS qualifier, 5-70
- /BATCH qualifier, 5-69
- /BRIEF qualifier, 5-70
- /BY_JOB_STATUS qualifier, 5-69
- /CHARACTERISTIC qualifier, 5-26
- /DEVICE qualifier, 5-69
- /FILES qualifier, 5-70
- /FORM qualifier, 5-42
 - for showing defined forms, 5-42
- /FULL qualifier, 5-70
 - for showing assigned form, 5-44
 - for showing characteristics, 5-26
 - for showing protection, 5-22, 5-23
- /GENERIC qualifier, 5-69
- /SUMMARY qualifier, 5-70

Shutdown

- allowing batch and print jobs to complete before, 5-75

Shutdown (Cont.)

- options
 - specifying time interval between DISABLE AUTOSTART/QUEUES and STOP /QUEUES/ON_NODE commands., 5-75
 - stopping queues before, 5-75
- SHUTDOWN\$DISABLE_AUTOSTART logical name, 5-75
- Size
 - scheduling jobs based on, 5-30
- Software errors
 - reporting, 6-1
- Software Performance Report (SPR)
 - submitting, 6-1
 - submitting for queue manager, 5-66
- SORT
 - optimizing on batch queues, 5-30
- Spooled output device, 5-6, 5-52
 - despooling, 5-53
 - recommended use, 5-6
 - sample configuration, 5-16
 - testing, 5-53
- Stalled job state, 5-78
- Stalled queue
 - troubleshooting, 5-90
- Stalled queue status, 5-71, 5-88, 5-90
- Starting queues
 - autostart, 5-68, 5-69
 - in startup command procedure, 5-61
 - nonautostart, 5-57, 5-68
 - in startup command procedure, 5-61
 - relationship with activating an autostart queue, 5-5
- Starting queue status, 5-71
- Starting the ERRFMT process, 6-4
- Starting the queue manager
 - initially, 5-54
 - restarting, 5-63
- START/QUEUE command, 5-61
 - /AUTOSTART_ON qualifier, 5-56
 - /BACKWARD qualifier, 5-83
 - /BLOCK_LIMIT qualifier, 5-30
 - /CHARACTERISTICS qualifier, 5-26
 - /DEFAULT qualifier
 - for banner pages, 5-32
 - for controlling page overflow, 5-45
 - for forms, 5-43
 - for activating an autostart queue, 5-68
 - /FORM_MOUNTED qualifier, 5-43
 - for starting a nonautostart queue, 5-68
 - /FORWARD qualifier, 5-84
 - /GENERIC qualifier, 5-57
 - /NOCHARACTERISTICS qualifier, 5-26
 - /NORETAIN qualifier, 5-25
 - /RETAIN qualifier, 5-24
 - /SCHEDULE qualifier, 5-30
- SEARCH qualifier, 5-84

START/QUEUE command (Cont.)

- /SEPARATE qualifier
 - for banner pages, 5-32
 - for reset modules, 5-46
- specifying queue options with, 5-18
- TOP_OF_FILE qualifier, 5-84
- using to resume printing of a suspended job, 5-83

- /WSEXTENT qualifier, 5-30

START/QUEUE/MANAGER command, 5-54, 5-62

- /NEW_VERSION qualifier, 5-54
- /ON qualifier, 5-55
- storage of, 5-55

Startup

- of queue manager, 5-54, 5-62, 5-63
- of queues, 5-5

See also Starting queues

STARTUP.COM procedure

- operator-assisted mount, 2-7

Startup command procedure

- to set up output devices, 5-54
- to start queues, 5-61

States

- of jobs, 5-78

Status

- of queues, 5-71

Stock

See also Forms

- commands used with, 5-40
- mismatch, 5-41
- mismatch problems
 - troubleshooting, 5-89
- specifying, 5-40

Stopped queue status, 5-71, 5-88

Stop pending queue status, 5-71

Stopping all queues on a node, 5-74

Stopping a queue, 5-73

- abruptly, 5-73
- smoothly, 5-73

Stopping queue status, 5-71

Stopping the queue manager, 5-63

STOP/QUEUE command, 5-73, 5-83

STOP/QUEUE/MANAGER/CLUSTER command, 5-63

STOP/QUEUE/NEXT command, 5-68, 5-73

- with autostart queues, 5-74

STOP/QUEUE/RESET command, 5-68, 5-73

- with autostart queues, 5-74

STOP/QUEUES/ON_NODE command

- entering before shutting down a system, 5-75
- relationship with DISABLE AUTOSTART /QUEUES, 5-75

SUBMIT command

- /CHARACTERISTIC qualifier, 5-25
- preventing users from executing, 5-73
- /PRIORITY qualifier, 5-6
- processing of, 5-1

SUBMIT command (Cont.)

- /RETAIN qualifier, 5-24, 5-81

Subprocess termination

- accounting, 6-2

Substituting volumes, 3-3

Suspending a job, 5-83

SYLOGICALS.COM command procedure, 3-1

- defining logical names in, 3-1
- use for specifying master file location, 5-55

Symbiont, 5-73

- bypass formatting, 5-45
- communicating with, 5-83
- default, 5-4
- determining, 5-87
- function of, 5-1
- LAT, 5-4, 5-87
- role in processing print jobs, 5-4
- user-written, 5-4
- using PRTSMB on LAT printers, 5-87

SY\$BATCH

- default queue name, 5-8

SY\$ERRORLOG directory, 6-6

SY\$PRINT

- default queue name, 5-12

SY\$QUEUE_MANAGER.QMAN\$JOURNAL

See Journal file

SY\$QUEUE_MANAGER.QMAN\$QUEUES

See Queue file

SYSDUMP.DMP file, 6-14

System

- accounting, 6-2

System activity

- monitoring, 6-2

System disk

- and volume sets, 2-3

System Dump Analyzer (SDA), 6-1

System dump file, 6-14

- automatic creation of, 6-14
- size, 6-15
- use of, 6-1

System failure

- anticipating, 6-5, 6-7

System files

- on public volumes, 2-1

System Generation Utility (SYSGEN)

- operator log messages, 6-10

System Management Utility (SYSMAN)

- establishing disk quotas with, 4-2

System parameters

- ACP cache parameters, 3-9
- ACP_DINDXCACHE, 2-6
- DUMPCACHE, 6-15
- ERRORLOGBUFFERS, 6-15
- MVTIMEOUT, 3-12
- TAPE_MVTIMEOUT, 3-12

System resources

See Resources

System parameters

System parameters (Cont.)

VIRTUALPAGECNT, 5-30

T

Tape commands

MOUNT, 3-3, 3-4

 /COMMENT qualifier, 3-3

REPLY

 /ABORT qualifier, 3-7

 /BLANK_TAPE qualifier, 3-6

 /INITIALIZE_TAPE qualifier, 3-6

 /TO qualifier, 3-5, 3-6, 3-7, 3-8

Tape volumes

 automatic volume labeling (AVL), 3-5

 automatic volume recognition (AVR), 3-5

 end of tape (EOT) marker

 when reached, 3-5

 ensuring correct volume is mounted, 3-6

 identifying in MOUNT command, 3-6

 identifying in REPLY command, 3-6

 initializing, 3-6

 loading, 3-4

 monitoring operations, 6-7

 mounting, 3-4

 mounting volume sets, 3-5

 mounting with automatic switching, 3-5

 mounting without automatic switching, 3-5

 unformatted

 writing to, 3-6

 volume checking, 3-6

 if failed, 3-6

 write-locking, 3-5

 write rings, 3-5

TAPE_MVTIMEOUT parameter, 3-12

Terminal

 operator, 6-8

Terminal queue, 5-4

Testing spooled output devices, 5-53

Time out

 mount verification, 3-12

 OPCOM message, 3-12

Trailer page, 5-32

 file, 5-36

 job, 5-36

Translation modes

 card reader, 5-95

Troubleshooting

 adding or deleting a device control library
 module, 5-92

 autostart queues, 5-90

 general printer problems, 5-87

 holding jobs, 5-88

 jobs that will not execute, 5-88

 jobs with characteristic mismatch, 5-89

 pending jobs, 5-88

 print jobs with stock mismatch, 5-89

Troubleshooting (Cont.)

 problems deleting a queue, form, or
 characteristic, 5-91

 queue manager, 5-85

 queue problems, 5-85

 stalled output queue, 5-90

 system failure, 6-1

 using dump file for, 6-1

U

UIC protection

 of public volumes, 2-2

 of queues, 5-21

User files

 on public volumes, 2-1

 placement, 2-2

User messages

 accounting, 6-2

User request

 See also Operator function

 monitoring, 6-7

 OPCOM message, 3-2

 if operator terminals are disabled, 3-3

 responding to, 3-4

Users

 handling requests from, 3-1

User-specified job retention

 changing, 5-81

Using banner pages, 5-31

Using characteristics with queues, 5-25

Using forms with queues, 5-40

V

VAXcluster environment

 defining QMAN\$MASTER in, 5-65

 generic queues in, 5-4

 local security audit log files in, 6-14

 mounting public volumes in, 2-6, 2-8

 using autostart queues in, 5-5, 5-11

 using generic batch queues in, 5-9

 using generic output queues in, 5-15

VAX Distributed Queuing System (DQS), 5-7

Vector capability

 determining availability within a system, 6-16

Vector consumer

 determining the identity of, 6-16

 obtaining information about, 6-15 to 6-18

Vector context switch

 obtaining information about, 6-16

Vector CPU time

 definition, 6-17

 obtaining information regarding image, 6-17

 obtaining information regarding process, 6-16,
 6-17

 obtaining information regarding processor,
 6-17

- Vector CPU time (Cont.)
 - obtaining information regarding system, 6-17
- Vector-present processor
 - identifying, 6-16
- Vector processing system
 - obtaining information about, 6-15 to 6-18
 - obtaining number of vector processors in, 6-16
- Verification
 - mount
 - See Mount verification
- Version limit
 - setting on public volumes, 2-1
- VIRTUALPAGECNT system parameter
 - use for optimizing batch queues for SORT, 5-30
- Volume integrity, 3-9
- Volumes
 - See also Disk volumes, Public volumes, Tape volumes, Volume sets
 - availability
 - OPCOM message, 3-10
 - canceling mount verification, 3-12
 - controlling cache size, 3-9
 - dismounting, 3-9
 - message in operator log file, 6-9
 - foreign
 - transferring information to and from, 2-8
 - mounting
 - if device is unavailable, 3-3
 - message in operator log file, 6-9
 - operator assistance, 3-3
 - operator functions, 3-3
 - mounting continuation, 3-5
 - mounting public, 2-5
 - mounting with EXCHANGE, 2-9
 - mount verification aborted
 - OPCOM message, 3-13
 - mount verification time out, 3-12
 - OPCOM message, 3-12
 - planning public, 2-1
 - rebuilding, 4-3
 - recovering from error, 3-9
 - substituting, 3-3
 - switching
 - automatic, 3-5
 - non-automatic, 3-5
- Volume sets
 - creating, 2-2, 2-6
 - mounting with automatic switching, 3-5
 - mounting without automatic switching, 3-5
 - restriction for system disk, 2-3
 - tape
 - end of tape (EOT) marker, 3-5
 - mounting, 3-5
- VVIEF (VAX Vector Instruction Emulation Facility)
 - determining presence of, 6-16

W

- Waiting job state, 5-78
- Write-lock
 - disk volumes, 3-9
 - mount verification, 3-11
 - tape volumes, 3-5
- Write operation
 - continuation volumes, 3-6
- Write rings, 3-5



How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-DEC-DEMO (800-332-3366) using a 1200- or 2400-baud modem. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	—————	Local Digital subsidiary or approved distributor
Internal ¹	—————	USASSB Order Processing - WMO/E15 <i>or</i> U.S. Area Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

¹For internal orders, you must submit an Internal Software Order Form (EN-01740-07).



Reader's Comments

Guide to Maintaining
a VMS System

AA-LA34B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completeness (enough information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clarity (easy to understand)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization (structure of subject matter)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Figures (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Examples (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index (ability to find topic)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page layout (easy to find information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

I would like to see more/less _____

What I like best about this manual is _____

What I like least about this manual is _____

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

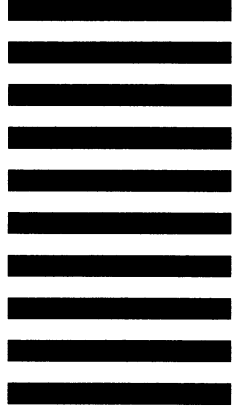
I am using **Version** _____ of the software this manual describes.

Name/Title _____ Dept. _____
Company _____ Date _____
Mailing Address _____
_____ Phone _____

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



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

DIGITAL EQUIPMENT CORPORATION
Corporate User Information Products
ZK01-3/J35
110 SPIT BROOK RD
NASHUA, NH 03062-9987



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



Reader's Comments

Guide to Maintaining
a VMS System

AA-LA34B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completeness (enough information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clarity (easy to understand)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization (structure of subject matter)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Figures (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Examples (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index (ability to find topic)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page layout (easy to find information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

I would like to see more/less _____

What I like best about this manual is _____

What I like least about this manual is _____

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

I am using **Version** _____ of the software this manual describes.

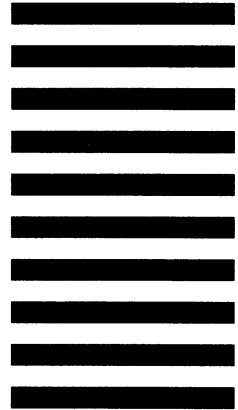
Name/Title _____ Dept. _____
Company _____ Date _____
Mailing Address _____
_____ Phone _____

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

digitalTM



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

DIGITAL EQUIPMENT CORPORATION
Corporate User Information Products
ZK01-3/J35
110 SPIT BROOK RD
NASHUA, NH 03062-9987



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

