

# VMS System Manager's Manual

Order Number: AA-LA00B-TE

**June 1989**

This manual provides the basic concepts and procedures for VMS system management; it is especially intended for managers of small clusters and systems.

**Revision/Update Information:** This manual supersedes the *VMS System Manager's Manual*, Version 5.0.

**Software Version:** VMS Version 5.2

**digital equipment corporation  
maynard, massachusetts**

**June 1989**

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 1989.

All Rights Reserved.  
Printed in U.S.A.

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:

CDA	MASSBUS	VAX RMS
DDIF	PrintServer 40	VAXstation
DEC	Q-bus	VMS
DECnet	ReGIS	VT
DECUS	ULTRIX	XUI
DECwindows	UNIBUS	
Digital	VAX	
LN03	VAXcluster	

The following is a third-party trademark:

PostScript is a registered trademark of Adobe Systems, Inc.

ZK3388

## **Production Note**

This book was produced with the VAX DOCUMENT electronic publishing system, a software tool developed and sold by Digital. In this system, writers use an ASCII text editor to create source files containing text and English-like code; this code labels the structural elements of the document, such as chapters, paragraphs, and tables. The VAX DOCUMENT software, which runs on the VMS operating system, interprets the code to format the text, generate a table of contents and index, and paginate the entire document. Writers can print the document on the terminal or line printer, or they can use Digital-supported devices, such as the LN03 laser printer and PostScript printers (PrintServer 40 or LN03R ScriptPrinter), to produce a typeset-quality copy containing integrated graphics.



# Contents

## Preface

xiii

## Chapter 1 Introduction

- 1.1 Who Should Use This Manual? ..... 1-2
- 1.2 System Management Concepts and Terms ..... 1-6

## Chapter 2 Starting Up the System

- 2.1 Starting Up Your System for the First Time ..... 2-1
- 2.2 Booting the System ..... 2-2
- 2.3 Logging In to the New System ..... 2-2
- 2.4 Startup Command Procedure for Your Site  
(SYSTARTUP\_V5.COM) ..... 2-4
  - 2.4.1 Mounting Public Disks ..... 2-5
  - 2.4.2 Setting Device Characteristics ..... 2-6
  - 2.4.3 Printers and Batch Processing: Initializing and  
Starting Queues ..... 2-6
  - 2.4.4 Installing Known Images ..... 2-7
  - 2.4.5 Starting Up the DECnet Network ..... 2-8
  - 2.4.6 Running the System Dump Analyzer ..... 2-8
  - 2.4.7 Purging the Operator's Log File ..... 2-9
  - 2.4.8 Submitting Batch Jobs That Are Run at Startup  
Time ..... 2-9
  - 2.4.9 Defining the Number of Interactive Users ..... 2-10
  - 2.4.10 Starting Up the LAT Network ..... 2-10
  - 2.4.11 Creating Systemwide Announcements ..... 2-11
- 2.5 Defining a System Login Command Procedure .... 2-12
  - 2.5.1 Sample System Login Command Procedure .... 2-13
- 2.6 Backing Up the System ..... 2-14
- 2.7 Building and Copying a VMS System Disk ..... 2-14

<b>2.8</b>	<b>System Startup Procedures</b> .....	<b>2-15</b>
2.8.1	Startup Command Procedure for the System (STARTUP.COM) .....	2-17
2.8.2	Setting Up Logical Names for Your Site (SYLOGICALS.COM) .....	2-19
<b>2.9</b>	<b>Emergency Startup Procedures</b> .....	<b>2-20</b>
2.9.1	Bypassing the User Authorization File .....	2-20
2.9.2	Emergency Startup After Modifying System Parameters .....	2-22
2.9.3	Bypassing Startup and Login Procedures .....	2-22
2.9.4	Startup Problems .....	2-23
<b>2.10</b>	<b>Shutdown Procedures</b> .....	<b>2-23</b>
2.10.1	Orderly Shutdown .....	2-24
2.10.2	Emergency Shutdown .....	2-29
<b>2.11</b>	<b>Summary</b> .....	<b>2-30</b>

## Chapter 3 Installing Software

<b>3.1</b>	<b>Preparing Your System for VMSINSTAL</b> .....	<b>3-1</b>
3.1.1	Starting the VMSINSTAL Procedure .....	3-3
3.1.2	When the Installation Is Complete .....	3-6
3.1.3	Choosing VMSINSTAL Options .....	3-7
3.1.4	Recovering from a System Failure .....	3-10
<b>3.2</b>	<b>Summary</b> .....	<b>3-11</b>

## Chapter 4 Managing Users

<b>4.1</b>	<b>The User Authorization File (UAF)</b> .....	<b>4-1</b>
4.1.1	System-Supplied UAF Records .....	4-3
4.1.2	General Maintenance of the UAF .....	4-5
<b>4.2</b>	<b>Adding a User Account</b> .....	<b>4-6</b>
<b>4.3</b>	<b>Setting Up an Automatic Login Account</b> .....	<b>4-8</b>
<b>4.4</b>	<b>Modifying a User Account</b> .....	<b>4-10</b>
<b>4.5</b>	<b>Listing User Accounts</b> .....	<b>4-10</b>
<b>4.6</b>	<b>Deleting a User Account</b> .....	<b>4-11</b>
<b>4.7</b>	<b>Summary</b> .....	<b>4-13</b>

## Chapter 5 Performing Batch and Print Operations

5.1	Generic Queues and Execution Queues .....	5-1
5.2	Setting Up Queues .....	5-2
5.3	Maintaining Batch and Print Queues .....	5-3
5.4	Monitoring Jobs .....	5-4
5.4.1	Deleting a Job .....	5-5
5.4.2	Retaining Jobs in a Queue .....	5-6
5.4.3	Modifying Job Processing Attributes .....	5-6
5.5	Summary .....	5-8

## Chapter 6 Setting Up and Maintaining a Network

6.1	Getting Started with Networks .....	6-1
6.2	Joining a Network .....	6-2
6.2.1	Preparing Your VMS System for the Network Environment .....	6-3
6.2.2	Using DECnet-VAX on Your System .....	6-4
6.2.3	Configuring the Network Environment .....	6-5
6.3	Keeping the Network Running .....	6-30
6.3.1	Monitoring the Network .....	6-30
6.3.2	Common Problems Encountered on the Network .....	6-36
6.4	Summary .....	6-43

## Chapter 7 Setting Up a Local Area VAXcluster Environment

7.1	What Is a Cluster? .....	7-1
7.1.1	VAXcluster Types .....	7-1
7.2	Shared Resources .....	7-3
7.3	Preparing a System for a Local Area VAXcluster Environment .....	7-3
7.3.1	Building a VAXcluster Configuration .....	7-4
7.4	DECnet-VAX Connections .....	7-6

<b>7.5</b>	<b>Clusterwide Tasks Using SYSMAN</b> .....	<b>7-6</b>
7.5.1	Setting a Clusterwide Environment .....	7-6
7.5.2	Executing Commands on a Cluster .....	7-7
<b>7.6</b>	<b>Summary</b> .....	<b>7-8</b>

## Chapter 8 Backing Up and Restoring Files

<b>8.1</b>	<b>Making Backup Copies of Files</b> .....	<b>8-2</b>
8.1.1	Image (Full) and Incremental Backups .....	8-2
8.1.2	Save Sets .....	8-3
8.1.3	Using the BACKUP Command to Save Files ...	8-4
8.1.4	Making Image Backups of a Disk .....	8-4
8.1.5	Making Incremental Backups of a Disk .....	8-5
8.1.6	Using Command Procedures for Backups .....	8-6
<b>8.2</b>	<b>Preparing Your System for Efficient Backups</b> .....	<b>8-10</b>
<b>8.3</b>	<b>Restoring Files from Backup Copies</b> .....	<b>8-13</b>
8.3.1	Restoring All of the Files on a Disk .....	8-14
8.3.2	Restoring an Individual Directory Structure ...	8-18
8.3.3	Restoring an Individual File .....	8-19
8.3.4	Listing the Contents of a Save Set .....	8-19
<b>8.4</b>	<b>Standalone Backup</b> .....	<b>8-20</b>
<b>8.5</b>	<b>Backup and Magnetic Tape</b> .....	<b>8-22</b>
8.5.1	Automatic Tape Unloading .....	8-22
8.5.2	Tape Label Processing .....	8-22
8.5.3	Assigning Volume Labels to Magnetic Tapes ...	8-22
<b>8.6</b>	<b>Summary</b> .....	<b>8-23</b>

## Chapter 9 Maintaining Acceptable System Performance

<b>9.1</b>	<b>Knowing Your Work Load</b> .....	<b>9-2</b>
9.1.1	Using the Monitor Utility (MONITOR) .....	9-3
9.1.2	Using the Accounting Utility (ACCOUNTING) .....	9-3
9.1.3	Managing Work Load .....	9-4
9.1.4	Distributing Work Load .....	9-5
9.1.5	Installing Known Images .....	9-6
9.1.6	Tuning a System .....	9-7



9.1.7	Predicting When Tuning Is Required .....	9-8
9.1.8	Evaluating Tuning Success .....	9-8
9.1.9	Performance Options .....	9-9
<b>9.2</b>	<b>Summary .....</b>	<b>9-11</b>

## Chapter 10 Operator Tasks

<b>10.1</b>	<b>Performing Backups .....</b>	<b>10-1</b>
<b>10.2</b>	<b>Maintaining System Log Files .....</b>	<b>10-1</b>
10.2.1	The System Dump File .....	10-2
10.2.2	The Error Log File .....	10-3
10.2.3	The Operator Log File .....	10-6
10.2.4	The Accounting Log File .....	10-12
<b>10.3</b>	<b>Summary .....</b>	<b>10-14</b>

## Chapter 11 System Security Issues

<b>11.1</b>	<b>Types of Computer Security Problems .....</b>	<b>11-1</b>
11.1.1	User Irresponsibility .....	11-1
11.1.2	User Probing .....	11-2
11.1.3	User Penetration .....	11-2
<b>11.2</b>	<b>Levels of Security Requirements .....</b>	<b>11-2</b>
<b>11.3</b>	<b>The Secure System Environment .....</b>	<b>11-4</b>
<b>11.4</b>	<b>Managing Passwords .....</b>	<b>11-5</b>
11.4.1	Initial Passwords .....	11-5
11.4.2	System Passwords .....	11-6
11.4.3	Primary and Secondary Passwords .....	11-7
11.4.4	Enforcing Minimum Password Standards .....	11-8
11.4.5	Requiring the Password Generator .....	11-10
11.4.6	Protecting Passwords .....	11-10
<b>11.5</b>	<b>Controlling Break-In Detection .....</b>	<b>11-11</b>
11.5.1	Controlling the Number of Retries on Dialups .....	11-11
11.5.2	Controlling Break-In Detection and Evasion ...	11-12
11.5.3	Displaying the Break-In Database .....	11-15
<b>11.6</b>	<b>Protecting Files and Directories with ACLs .....</b>	<b>11-15</b>
11.6.1	Creating and Maintaining ACLs .....	11-16
11.6.2	Identifiers .....	11-16
11.6.3	Access Control List Entries .....	11-18
11.6.4	Summary of ACLs .....	11-22

<b>11.7</b>	<b>Creating a Project Account</b> .....	<b>11-23</b>
<b>11.8</b>	<b>Security Auditing</b> .....	<b>11-24</b>
11.8.1	Enabling Security Alarms .....	11-25
11.8.2	Enabling an Operator Terminal .....	11-27
11.8.3	Enabling Alarm Messages .....	11-27
<b>11.9</b>	<b>The Audit Analysis Utility—A Security Auditing Tool</b> .....	<b>11-28</b>
<b>11.10</b>	<b>ANALYZE/AUDIT Command Line Format</b> .....	<b>11-29</b>
<b>11.11</b>	<b>ANALYZE/AUDIT Output</b> .....	<b>11-29</b>
<b>11.12</b>	<b>Using ANALYZE/AUDIT</b> .....	<b>11-31</b>
11.12.1	Recognizing Common System Events .....	11-32
11.12.2	Performing a Periodic Audit Analysis .....	11-32
11.12.3	Performing a Detailed Audit Analysis .....	11-33
11.12.4	Using Interactive Mode Commands .....	11-34
<b>11.13</b>	<b>Summary</b> .....	<b>11-35</b>
	<b>Accounting Utility</b>	<b>ACC-1</b>
	<b>Audit Analysis Utility</b>	<b>AUD-1</b>
	<b>Analyze/Disk_Structure Utility</b>	<b>ADSK-1</b>
	<b>Authorize Utility</b>	<b>AUTH-1</b>
	<b>Backup Utility</b>	<b>BCK-1</b>
	<b>Bad Block Locator Utility</b>	<b>BAD-1</b>
	<b>Error Log Utility</b>	<b>ERR-1</b>
	<b>Exchange Utility</b>	<b>EXCH-1</b>
	<b>Install Utility</b>	<b>INS-1</b>
	<b>LAT Control Program Utility</b>	<b>LAT-1</b>
	<b>Mount Utility</b>	<b>MOUNT-1</b>
	<b>NCP Utility</b>	<b>NCP-1</b>
	<b>System Generation Utility</b>	<b>SGN-1</b>
	<b>SYSMAN Utility</b>	<b>SM-1</b>
	<b>Terminal Fallback Utility</b>	<b>TFU-1</b>

## Index

### Examples

2-1	Orderly System Shutdown with SHUTDOWN.COM .....	2-27
2-2	Emergency Shutdown Using OPCCRASH .....	2-29
4-1	Sample UAF Record Display .....	4-3
4-2	Command Procedure Template for Deleting an Account's Files .....	4-12
6-1	Sample NETCONFIG.COM Dialogue .....	6-11
10-1	Sample Operator Log File (SYS\$MANAGER:OPERATOR.LOG) .....	10-7
11-1	Sample Brief Listing .....	11-30
11-2	Sample Full Listing .....	11-30
11-3	Sample Summary Output .....	11-31
11-4	Spotting Suspicious Activity in the Audit Analysis Report .....	11-33
11-5	An Example of a Full Format Audit Analysis Report .....	11-34
11-6	Entering Interactive Command Mode .....	11-35

### Figures

6-1	DECnet-VAX Software Design as Based on DNA Layers .....	6-39
-----	---	------

### Tables

5-1	Queue Management Commands .....	5-4
6-1	DECnet Event Classes .....	6-35
7-1	Installation Questions for Local Area VAXcluster Configurations .....	7-4
8-1	Recommended Process Quotas for Efficient Backups .....	8-12
8-2	Sample Process Quotas for Efficient Backups ..	8-12
11-1	Event Tolerance as a Measure of Security Requirements .....	11-3

**xii Contents**

11-2	System Files Benefiting from ACL-Based File Access Auditing .....	11-26
AUTH-1	Summary of Qualifiers for the ADD, COPY, DEFAULT, and MODIFY Commands .....	AUTH-2
NCP-1	Object Type Codes .....	NCP-26
SGN-1	Device Type Codes .....	SGN-3
SGN-2	SYSGEN Device Table .....	SGN-27

# Preface

The *VMS System Manager's Manual* provides system managers with the concepts and procedures needed to manage daily operations on a VMS system. This manual contains a subset of the information included in the Extended VMS System Management documentation subkit.

## Intended Audience

This manual can be used by anyone who performs the functions of a system manager or operator on a VMS system. It is especially intended for managers of small systems and clusters.

## Document Structure

The *VMS System Manager's Manual* is divided into two main sections: System Management Tasks and Reference.

Chapters 1 through 11 describe how to perform the tasks that are generally assigned to system managers. The Reference Section documents the utilities that serve as system management tools on a VMS system.

Chapter 1 describes each chapter in some detail. Read Chapter 1 to determine which of the remaining chapters in the book are appropriate for your needs.

The Reference Section contains quick reference information on the VMS system management utilities. Each utility chapter includes a usage summary and a subset of frequently used commands and qualifiers.

The Reference Section includes information about the following utilities:

- Accounting Utility
- Analyze/Disk\_Structure Utility
- Audit Analysis Utility
- Authorize Utility
- Backup Utility
- Bad Block Locator Utility

## xiv Preface

- Error Log Utility
- Exchange Utility
- Install Utility
- LAT Control Program Utility
- Mount Utility
- Network Control Program (NCP) Utility
- Security Audit Utility
- SYSGEN Utility
- SYSMAN Utility
- Terminal Fallback Utility

## Associated Documents

In the VMS Base documentation set:

- For guidance in finding additional system management information, see the *Overview of VMS Documentation*.
- For general information about how to use a VMS system, see the *VMS User's Manual*.
- For information about the License Management Utility, see the *VMS License Management Utility Manual*.

In the Extended VMS Documentation Set:

(The Extended VMS Documentation Set is the complete set of software manuals for the VMS operating system. For information about ordering any of the manuals in the Extended VMS Documentation Set, see the *Overview of VMS Documentation* or contact your Digital representative.)

- For general background information about the operating system, see the *Introduction to VMS*.
- For more information about setting up the system for operation, see the *Guide to Setting Up a VMS System*.
- For more information about maintaining the system, see the *Guide to Maintaining a VMS System*.
- For information about security management, see the *Guide to VMS System Security*.
- For more information about networking, see the *Guide to DECnet-VAX Networking*.

- For more information about VMS clusters, see the *VMS VAXcluster Manual*.
- For more information about performance tuning, see the *Guide to VMS Performance Management*.
- For more information about utilities, see the individual VMS utility manuals.
- For complete descriptions of DCL commands, see the *VMS DCL Dictionary*.
- For explanations of system messages, see the *VMS System Messages and Recovery Procedures Reference Volume*.

Other related documentation:

- For information about system installation and other processor-specific procedures, see your VAX processor installation and operations guide.
- If you have purchased the volume shadowing option, see the *VAX Volume Shadowing Manual* for information about creating and maintaining volumes using volume shadowing.
- If you have purchased the RMS journaling option, see the *VAX RMS Journaling Manual* for information about RMS journaling.
- For hardware operating instructions, see the appropriate hardware owner's 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.
PF1 x	A sequence such as PF1 x indicates that you must first press and release the key labeled PF1, then press and release another key or a pointing device button.
<span style="border: 1px solid black; padding: 2px;">Return</span>	A key name is shown enclosed to indicate that you press a key on the keyboard.

...	In examples, a horizontal ellipsis indicates one of the following possibilities:
	<ul style="list-style-type: none"><li>• Additional optional arguments in a statement have been omitted.</li><li>• The preceding item or items can be repeated one or more times.</li><li>• Additional parameters, values, or other information can be entered.</li></ul>
.	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 is optional; you can select none, one, or all of the choices.
{ }	In format descriptions, braces surround a required choice of options; you must choose one of the options listed.
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, user input is shown in <b>bold</b> .
<b>boldface text</b>	Boldface text represents the introduction of a new term or the name of an argument, an attribute, or a reason.
UPPERCASE TEXT	Uppercase letters indicate the name of a command, the name of a file, the name of a file protection code, a command or a qualifier, 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.



# Chapter 1

## Introduction

The VMS operating system and the other software products that run on your computer provide you and the other users on your system with a wide range of computing capabilities. In order to create and maintain a proper and efficient computing environment, certain administrative tasks must be undertaken. These tasks are called **system management**, and they include the following:

- Setting up the system
- Giving individual users access to the system
- Installing software (and software updates)
- Maintaining acceptable performance levels
- Preventing the loss of important information that you keep on your system
- Making sure that the system is secure
- Handling media (such as disks or magnetic tapes)
- Setting up the software to allow for printers and for batch jobs
- Setting up a cluster
- Setting up a network

As system manager, you may need to do some of these tasks only once (for example, setting up software to allow for printers or batch jobs, or setting up a network); others are done on a continuing basis (for example, maintaining system security and preventing the loss of data). At some sites, one or more people are designated as system managers, and other individuals are designated as **operators**. In these cases, operators are responsible for tasks such as physically mounting magnetic tapes and disks, monitoring printers, responding to emergencies or security alarms, and maintaining system log files.

Not all of the tasks described in this manual may be necessary for your site. This chapter provides an overview of the information that this manual contains. You should read this introductory chapter to determine which parts of the manual may be applicable to your site.

## 1.1 Who Should Use This Manual?

This manual is divided into two parts. Chapters 1 through 11 describe the tasks that managers of small standalone systems and Ethernet-based (low end) VAXcluster configurations are likely to encounter. Each chapter is divided into a series of topics, and each chapter concludes with a summary of the topics that have been discussed. The second part of the manual is a condensed reference section describing the system management tools that the VMS operating system provides.

If you are a manager of a small system or cluster, you can use this manual for most or all of your system management tasks. Managers of all types of systems can use the Reference Section of this manual as a centralized source of information for system management utilities. You should also be aware that expanded documentation exists for all of the topics discussed in this manual. See the *Overview of VMS Documentation*, included in the Base Documentation Set, for a complete list of the technical manuals for the VMS operating system.

The next sections describe the remaining chapters in this manual. Read these sections to determine which parts of this manual are applicable to your site.

### Chapter 2 — Starting Up the System

Chapter 2 describes the procedure for starting up your system, for the first time and for subsequent system startups. It discusses how to customize your startup procedure, so that your system automatically provides the proper environment each time that the system is started. The chapter also tells you how to shut down the system in an orderly manner.

All managers of small systems and clusters should read this chapter. It describes the procedures that are needed to boot your system and to create a proper environment for the users on your system.

### Chapter 3 — Installing Software

Software such as the VMS operating system and any layered products that you use must be installed on your system. You must also use a similar procedure when you upgrade software (that already exists on your system) to a more recent version. Chapter 3 describes the procedures that you should follow when you install or upgrade software. This chapter also tells you how to remove a software product that has previously been installed.

All managers of small systems or clusters should read this chapter, because it contains information that system managers need when installing the VMS operating system and layered products. (In addition to the information in this chapter, you will also need the specific installation instructions for the software you want to install.)

## Chapter 4 — Managing Users

Chapter 4 describes how to give access to users on your system. It tells you how to add and maintain user accounts, and it describes the privileges that you can give and the resources that you can allocate to the users on your system.

If you are a manager of a system with more than one user, or if you are the manager of a single-user system and would like more than one user account on your system, you should read this chapter.

## Chapter 5 — Print and Batch Queues

Chapter 5 tells you how to set up and maintain queues for printers and for batch jobs. If one or more printers are connected to your system, then you must have a print queue in order to use them. (You do not, however, need a print queue to use a terminal that has its own printing capabilities.) If you want batch processing to be available to users, then you must also establish one or more batch queues.

If you have a printer connected to your system, or if you want to use batch processing capabilities, you should read this chapter.

## Chapter 6 — Setting Up a Network

A computer **network** allows you to exchange information between two or more individual computers. In the VMS operating system, the DECnet-VAX product provides networking capabilities. In order to use the DECnet-VAX functions, you must have the appropriate hardware and software.

Chapter 6 tells you how to set up a basic network using DECnet-VAX. The chapter describes how to set up the basic network control functions that allow you to communicate with other systems, and it also tells you how to control certain network functions such as stopping and restarting the network, monitoring network activity, and so on.

If you are a manager of a small system that is part of a network of computers or if you are a manager of an Ethernet-based (low end) cluster, then you should read this chapter.

## Chapter 7 — Cluster Configurations

A **cluster** is a group of two or more processors that share some or all of their resources. When a group of VAX processors share resources in a VAXcluster environment, the storage and computing resources of all of the processors are combined, which can increase the processing capability, communications, and availability of your computing system. Clusters also provide an environment in which additional computers can be added easily.

Chapter 7 tells you how to create a VAXcluster environment. It discusses the software and hardware that is required, the various types of VAXcluster configurations, how to use DECnet-VAX functions in your cluster, and the resources that you can share in the cluster.

If you manage a cluster, you should read this chapter.

### **Chapter 8 — Backup Procedures**

To help prevent the loss of important data on your system, make **backup** copies of your data at regular intervals. A backup copy is a reserve copy of the data that you keep in a safe place (for example, on a magnetic tape, or on a different disk). If the data that is on line is lost (for example, because of inadvertent deletion or a hardware failure), you can use the backup copy of the data.

All system managers should have a plan for regular backups of the data on their systems. Chapter 8 describes procedures for making full and incremental backups for your system, and it also tells you how to restore the data from a backup copy. This chapter contains essential information for all managers of small systems.

### **Chapter 9 — Maintaining Acceptable Performance**

The **performance** of a system refers to the speed of interactive and batch processing. Performance can be measured in the response time for interactive processing and the time that it takes to complete batch processing jobs.

Chapter 9 describes some of the actions that you can take to optimize your system's performance. This chapter tells you how to monitor the use of system resources; it shows you how to reset system parameters to optimize performance, and it provides some hints for making some other performance improvements.

In most cases, performance tuning is not necessary for small systems. The VMS operating system provides tools that automatically set system parameters that provide for optimum performance. This chapter is useful for acquiring background information about performance issues or for determining whether your system performance might benefit from additional tuning.

### **Chapter 10 — Operator Tasks**

Chapter 10 describes maintaining media, maintaining print devices, system problem diagnosis and recovery, error log issues, the operator console, sending system messages to interactive users, and other functions that may be assigned to an operator. When there is no individual designated as the operator, these tasks might be the responsibility of the system manager.

You should read this chapter if you are the manager or an operator in a small system environment.

### **Chapter 11 — System Security Issues**

Chapter 11 discusses security issues in the context of a small system or cluster. These issues include basic security for single-user and multi-user systems, network security, user privileges (including rights and proxies), system passwords, and ongoing security practices (such as security audits).

Security is important for any system, and no system manager should take security for granted. Although some parts of this chapter may not be applicable to all sites, all system managers should read this chapter in order to provide a secure data processing environment.

### **Reference Section**

The Reference Section provides information for VMS utilities that you can use for system management tasks. For each utility, the Reference Section provides a brief description of the utility, format statements for using the utility, and a description of the commands and qualifiers that you can use with the utility.

The following utilities are included in Part II of this manual:

- Accounting Utility
- Analyze/Disk\_Structure Utility
- Audit Analysis Utility
- Authorize Utility
- Backup Utility
- Bad Block Locator Utility
- Error Log Utility
- Exchange Utility
- Install Utility
- LATCP Utility
- Mount Utility
- NCP Utility
- SYSMAN Utility
- System Generation Utility
- Terminal Fallback Utility

## 1.2 System Management Concepts and Terms

Some concepts and terms are used frequently in system management, and you should become familiar with them. The following terms and concepts are used both in the context of everyday general use in a VMS system and in the context of system management; they are described in the *VMS User's Manual*:

- **Accounts and directories**
- **Command procedures**
- **DIGITAL Command Language (DCL)**

The following concepts and terms apply primarily to system management:

- **SYSTEM account and [SYSMGR] directory**

The SYSTEM account is reserved for use by the system manager. When you log in to the SYSTEM account, your default directory (which is also reserved for the system manager) is SYS\$SYSROOT:[SYSMGR].

Always be careful when you are logged in to the SYSTEM account. When you are logged in to the SYSTEM account, all privileges are enabled, by default. You need these privileges to perform many system management tasks; however, they can also produce unwanted or even destructive results if they are used incorrectly.

- **Console (Operator's) terminal**

You can perform most system management tasks from any terminal that is connected to the processor (or the cluster). However, certain tasks such as booting the system and communicating with the VAX processor's console subsystem must be performed at a special terminal called the **console terminal**.

The console terminal, which always has the designation OPA0, is also usually designated as the **operator's terminal**. You use the operator's terminal to send messages to system users and respond to user requests, using the operator communication process (OPCOM).

# Chapter 2

## Starting Up the System

The system startup procedure establishes the computing environment for your system.

This chapter covers three major topics:

- How to start your system for the first time
- How to create the proper computing environment whenever you restart your system
- How to shut down your system

Before you can use the procedures described in this chapter, you must first set up the hardware for each VAX computer. To set up the hardware and install the VMS operating system, refer to the instructions in your VAX processor installation and operations guide. After you have installed the operating system, you will be able to log into the SYSTEM account on your computer.

After the operating system has been successfully loaded, you can set up the proper computing environment for your system. The site-specific system startup file, SYSTARTUP\_V5.COM, is an essential aspect of establishing an environment specific to the needs of your site. Section 2.4 describes how to modify SYSTARTUP\_V5.COM to meet the needs of your site.

### 2.1 Starting Up Your System for the First Time

Instructions for installing the VMS operating system are included in the installation and operations guide for your processor. You must choose whether you are installing the VMS operating system as either a **new installation** or as an **upgrade**. If you are installing the VMS operating system for the first time, you must use the new installation procedure. If you already have a previous version of the VMS operating system on your processor, then you should use the upgrade procedure. Instructions for a new installation are found in your processor installation and operations guide; instructions for an upgrade procedure are found in the Upgrade and Installations Procedures manual.

## 2-2 Starting Up the System

When you install the VMS operating system using the new installation procedure, the disk on which you install the operating system is first erased, and then a directory structure and the operating system itself is put in place. When you use the upgrade procedure, the files for the VMS operating system are replaced (with files for the upgraded operating system), and all other files on your system disk (for example, data files, executable images that are not part of the operating system, and so on) remain as they are.

**CAUTION:** If you use the new installation procedure for a processor that already has a previous version of the VMS operating system, you will destroy the previous contents of the disk that you designate as the system disk.

## 2.2 Booting the System

**Booting** is the process of loading the operating system from the system disk into processor memory. You can perform either a *nonstop* boot or a *conversational* boot. A nonstop boot is the quickest and easiest method, and the operating system will automatically set system parameters that are appropriate for most computing activities for your system's hardware configuration. A conversational boot requires you to supply more information during the boot process, but it allows you to change system parameters during the boot procedure. See your VAX processor installation and operations guide for detailed booting instructions.

After a system shuts down, it must be rebooted before you can use it. Some processors provide the capability of an automatic reboot; when you enable this feature, the system automatically attempts to reboot itself after it has been shut down. For example, if your site experiences a power failure, a processor that has automatic reboot enabled restarts itself automatically after the power has been restored. See your VAX processor installation and operations guide for information about automatic rebooting.

In unusual cases, the normal startup procedures will not work properly and troubleshooting might be necessary. Section 2.9 describes procedures that you should follow if the normal startup procedure fails, or if you find yourself locked out of your system.

## 2.3 Logging In to the New System

When the boot procedure is complete, a message is displayed on the terminal from which the system is booted (except on workstations, where the message is displayed on the operator's window). The message is similar to the following:



```
VAX/VMS Version 5.2 <dd-mmm-yyyy hh:mm:ss.s>

%%%%%%%%%% OPCOM, <dd-mmm-yyyy hh:mm:ss.s> %%%%%%%%%%%
Logfile has been initialized by operator _OPA0:
Logfile is SYS$SYSROOT:[SYSMGR]OPERATOR.LOG;1

%SET-I-INTSET, login interactive limit = 64, Current interactive value = 0
SYSTEM          job terminated at <dd-mmm-yyyy hh:mm:ss.s>
```

After you see this display, you can then log in to the system manager's account, using the following procedure:

1. Press the RETURN key on the console terminal.
2. In response to the system's request for your *username*, type SYSTEM.
3. In response to the system's request for your *password*, type the password that you chose for the SYSTEM account during installation. You should change your system password immediately after logging in to the system for the first time. To change your password, enter the DCL command SET PASSWORD.

**CAUTION:** Digital recommends that you change the system manager's account password frequently to maintain system security. The system manager's account has full privileges by default; therefore, you should exercise caution when using it.

After you enter your password, the system prints a welcome message on the console terminal. If it is not your first time logging in, the system also prints the time of your last login, for example:

```
Welcome to VAX/VMS Version 5.2

Last interactive login at 19-APR-1990 15:13:21.07
```

The sample command procedure SYS\$EXAMPLES:MGRMENU.COM generates a menu that you can use to accomplish many system management tasks, such as adding a user account, building a standalone backup kit, or shutting down your system. To see and use the menu, enter the following command:

```
$ @SYS$EXAMPLES:MGRMENU
```

You can modify this procedure to serve your own site-specific needs. If you modify the procedure, Digital recommends that you first copy the procedure to another directory (for example, SYS\$MANAGER) so that an original version of MGRMENU.COM will always be available in the SYS\$EXAMPLES directory.

## 2.4 Startup Command Procedure for Your Site (SYSTARTUP\_V5.COM)

A command procedure that sets up a computing environment for the specific needs of your site is executed each time that your system starts up. This file is located in the system manager's directory, [SYSMGR], and it is called SYSTARTUP\_V5.COM. In order to customize SYSTARTUP\_V5.COM for the needs of your site, you must make the appropriate edits to the file. This section describes how to customize the SYSTARTUP\_V5 command procedure.

After you install the VMS operating system, the file SYSTARTUP\_V5.COM is placed in the [SYSMGR] directory. SYSTARTUP\_V5.COM is a template file, which means that it can be used as a basis or guide for creating a startup file that suits your own system. In particular, the SYSTARTUP\_V5.COM template includes sections that can perform the following tasks at startup time:

- Mounting public disks
- Setting the characteristics of terminals and other devices
- Initializing and starting queues
- Installing known images
- Starting up the DECnet network
- Running the System Dump Analyzer
- Purging the operator's log file
- Submitting batch jobs that are run at system startup time
- Limiting the number of interactive users
- Starting up the LAT network
- Site-specific LAT command procedure
- Creating systemwide announcements
- Defining a system login command procedure
- Backing up the system

To modify SYSTARTUP\_V5.COM, you can use any text editor. This file is a command procedure, so any changes that you make must conform to the rules for command procedures, as described in the *VMS User's Manual*. In order to be used as a site-specific startup file, be sure to keep the file in the [SYSMGR] directory and use the file name SYSTARTUP\_V5.COM.

To allow SYSTARTUP\_V5.COM to continue in the event of an error, include the DCL command SET NOON at the beginning of the file, as follows:

```
$ SET NOON
```

This command disables error checking after the execution of each command in SYSTARTUP\_V5.COM.

The following sections describe many of the elements of your user environment that you can establish with SYSTARTUP\_V5.COM.

### 2.4.1 Mounting Public Disks

A **public disk** is a disk that can be accessed by any system process. In order to make a public disk available for use, the disk must be physically mounted and you must then use the MOUNT command. You do not need to use the MOUNT command for the system disk, because the system disk is already mounted when the system starts up.

This section describes how to mount disks in the SYSTARTUP\_V5.COM file. If your system uses any disks that should be mounted whenever the system is running, you should read this section.

To mount your public disks for systemwide access, use the following MOUNT command syntax in SYSTARTUP\_V5.COM:

```
$ MOUNT/SYSTEM ddcu: volume_label logical_name
```

You use the /SYSTEM qualifier to mount the disk for systemwide access; this is called a **public volume**. If you are in a VAXcluster environment, then you should also use the /CLUSTER qualifier to make the volume accessible to any user in the cluster.

The expression **ddcu** represents the physical device name. You must always include a colon after the device name. The expression **volume\_label** is a name that you choose for the disk. For example, if you mount a disk with the physical device name DRA1, and you choose USERFILES as the volume label for this disk, then you would include the following command in the SYSTARTUP\_V5.COM file:

```
$ MOUNT DRA1: USERFILES
```

In the context of the MOUNT command, the expression **logical\_name** is a logical volume name that is associated with the volume that you mount. You can use the logical volume name (instead of the physical device name) in programs and procedures that are used on your system, and it is not necessary to know the physical drive on which the volume is mounted.

If you do not specify a logical volume name in the MOUNT command, then the logical volume name is in the form DISK\$volume\_label. In the previous example, where no logical name was specified and the volume label was USERFILES, the MOUNT command would automatically assign the logical name DISK\$USERFILES to the disk.

## 2-6 Starting Up the System

The following command produces the logical volume name `USER` and equates it to `DRA1`, the device name. Note that the logical volume name `USER` is equivalent to `DRA1` only while the disk is actually mounted; if the volume is dismounted, then the logical volume name no longer has any systemwide meaning.

```
$ MOUNT/SYSTEM DRA1: USERFILES USER
```

### 2.4.2 Setting Device Characteristics

On some systems, certain devices (such as terminals or printers) should have the same characteristics whenever the system is running. Characteristics include defining the device as a printer, setting the transmission speed for terminals, and so on. You can define these characteristics in the `SYSTARTUP_V5.COM` procedure. Read this section if you want to define certain characteristics for specific devices on your system.

To establish the characteristics of the terminals and other devices on the system, use a series of `SET` commands in `SYSTARTUP_V5.COM`. Use the `SET TERMINAL` command for terminals; you can include comments to remind yourself of the users to whom specific terminals are assigned.

Use the `SET PRINTER` command for printers. Printer characteristics must be set before you establish queues for the printers.

The following example shows how you could modify `SYSTARTUP_V5.COM` to set up characteristics for terminals and a printer:

```
$ SET TERMINAL TTC2: /SPEED=300 /DEVICE_TYPE=LA36 /PERMANENT !JONES
$ SET TERMINAL TTD1: /SPEED=9600 /PERMANENT !WRENS
$ SET TERMINAL TTD4: /SPEED=1200 /PERMANENT !JRSMITH
$ SET TERMINAL TTG4: /SPEED=1200 /MODEM /PERMANENT !DIALUP1
$ SET PRINTER /LA11 /PAGE=60 /WIDTH=80 LPA0:
```

For more information about the qualifiers available with the `SET TERMINAL` and `SET PRINTER` commands, see the *VMS User's Manual*.

### 2.4.3 Printers and Batch Processing: Initializing and Starting Queues

If your system has a printer that you want to make available for general use (that is, a printer that is not connected directly to an individual terminal), you must establish a **print queue**. Similarly, if you want to allow batch processing on your system, you must establish a **batch queue**. A queue allows users to submit requests for printing or batch processing, and the system prints or processes the users' jobs as resources allow.

If you want to include printing or batch processing capabilities on your system, you should include commands in `SYSTARTUP_V5.COM` that do the following:

1. Start the system job queue manager

## 2. Initialize and start each queue with a separate INITIALIZE/QUEUE/START command line

The following example shows how to start the system job queue manager and initialize and start queues in SYSTARTUP\_V5.COM:

```
$ !
$ !Start the system job queue manager
$ !
$ START/QUEUE/MANAGER/RESTART
$ !
$ !Set printers spooled and establish printer queues
$ !
$ SET PRINTER/LOWER LPA0:
$ SET DEVICE/SPOOLED=SYS$PRINT LPA0:
$ INITIALIZE/QUEUE/START/DEFAULT=FLAG/NOENABLE_GENERIC LPA0:
$ !
$ SET PRINTER/LOWER LPB0:
$ SET DEVICE/SPOOLED=SYS$PRINT LPB0:
$ INITIALIZE/QUEUE/START/DEFAULT=FLAG/NOENABLE_GENERIC LPB0:
$ !
$ INITIALIZE/QUEUE/START/GENERIC=(LPA0,LPB0) SYS$PRINT
$ !
$ !Establish batch queues
$ !
$ INITIALIZE/QUEUE/START/BATCH/JOB_LIMIT=2/BASE_PRIORITY=3 SYS$BATCH
```

**NOTE:** Digital recommends using the /RESTART qualifier with the START/QUEUE/MANAGER command. This qualifier causes the queue manager to restart automatically if the job controller should abort.

A **spooled device** directs the output of an application to an intermediate file until the application program finishes. When the application completes, the file is submitted for printing. A spooled device can help balance the workload demand on line printers if you are running applications on a time-shared system. Use the SET DEVICE /SPOOLED command to establish spooled devices.

### 2.4.4 Installing Known Images

You can **install** commonly used programs as **known images** to reduce the I/O overhead in activating those images and to assign attributes or privileges to the images. If you have programs on your system that meet any of the following conditions, you should read this section and install such programs as known images in the SYSTARTUP\_V5.COM startup file:

- Programs that are frequently run
- Programs that are usually run concurrently by several processes
- Programs that require special privileges

## 2-8 Starting Up the System

All known images must be reinstalled each time the system is rebooted, because known file lists are not saved if the system is shut down or fails. You include `INSTALL` commands in `SYSTARTUP_V5.COM` to install programs as known images. Chapter 9 includes a discussion about performance characteristics and known images.

The following example shows a command sequence that might appear in `SYSTARTUP_V5.COM` for installing additional known images:

```
$ INSTALL
ADD/OPEN/SHARED/HEADER_RESIDENT BLISS32
ADD/OPEN/SHARED MACRO32
ADD/OPEN DIRECTORY
```

### 2.4.5 Starting Up the DECnet Network

The DECnet software lets your system communicate with other computers. If you want to use the DECnet software on your system, you must include commands in `SYSTARTUP_V5.COM` that start up the DECnet network. Read this section if you use the DECnet software on your system.

**NOTE:** You must configure your network (usually by running `NETCONFIG.COM`) before starting up the network, or the system returns error messages stating that the database is not initialized. For information about starting the DECnet-VAX software for the first time, see Chapter 6.

If you have started a batch queue on your system (as described in an earlier section), then you should start the network using the following commands in `SYSTARTUP_V5.COM`:

```
$ IF F$SEARCH("SYSS$SYSTEM:NETACP.EXE") .NES. "" - !This is faster, if you
$ THEN SUBMIT SYS$MANAGER:STARTNET.COM           !have batch queues set up.
```

These commands submit the network startup procedure (`SYS$MANAGER:STARTNET.COM`) as a batch job, which reduces the amount of time it takes to boot your system. Alternatively, if you have not started a batch queue, use the following command in `SYSTARTUP_V5.COM` to start up the network:

```
$ IF F$SEARCH("SYSS$SYSTEM:NETACP.EXE") .NES. "" THEN @SYS$MANAGER:STARTNET
```

### 2.4.6 Running the System Dump Analyzer

In the event of a system failure, the System Dump Analyzer (SDA) can help you determine why the system failed. In order to use SDA for this purpose, you should make sure that the system dump file is available for analysis and not overwritten by a new crash. Read the rest of this section if you want to learn about using SDA with `SYSTARTUP_V5.COM`.

You can start SDA in your site-specific startup file by using the following lines in SYSTARTUP\_V5.COM:

```
$ ANALYZE/CRASH_DUMP SYS$SYSTEM:SYSDUMP.DMP
COPY SYS$ERRORLOG:SYSDUMP.DMP SYSDUMP.BACK
```

For further information, invoke the System Dump Analyzer for an interactive session upon completion of startup, or refer to the SDA documentation in the Extended VMS Documentation Set.

**CAUTION:** If you use the page file for the crash dump file, when the system reboots, you must enter the SDA command COPY to copy the dump from the page file to another file suitable for analysis. If you do not perform the copy operation, pages used to save the crash dump information are not released for paging, and your system hangs while executing STARTUP.COM in the rebooting process.

#### 2.4.7 Purging the Operator's Log File

Each time the system is rebooted, a new version of OPERATOR.LOG is created in the SYS\$MANAGER directory. You should devise a plan for regular maintenance of these files. The following command in SYSTARTUP\_V5.COM purges all except the last two versions of the operator's log file:

```
$ PURGE/KEEP=2 SYS$MANAGER:OPERATOR.LOG
```

See Chapter 10 for additional suggestions for maintaining the operator's log file.

#### 2.4.8 Submitting Batch Jobs That Are Run at Startup Time

Some sites have batch jobs that are submitted at system startup time. To submit such batch jobs, add SUBMIT commands to your SYSTARTUP\_V5 file, in the following format:

```
$ SUBMIT [/qualifier,...] file-spec
```

In the following example, a batch job is submitted to run a command procedure that rebuilds the disks each time the system is initialized:

```
$ SUBMIT SYS$MANAGER:SYSDISK_REBUILD
```

If you submit batch processing jobs in SYSTARTUP\_V5.COM, make sure that the batch processing jobs are submitted after the batch queues have been initialized. See Chapter 5 for more information on submitting batch jobs.

### 2.4.9 Defining the Number of Interactive Users

You can set a limit for the number of interactive users that are allowed to be logged in to your system at one time by defining a number to be equivalent to the logical name `STARTUP$INTERACTIVE_LOGINS`. For example, the following command in `SYSTARTUP_V5.COM` limits the number of interactive logins to 72:

```
$ STARTUP$INTERACTIVE_LOGINS == 72
```

Where  $n$  is the maximum number of interactive users that are permitted to log in at one time.

**NOTE:** The number of interactive users must be set to a value no greater than that which is authorized by your VAX computer license.

### 2.4.10 Starting Up the LAT Network

A LAT network is any local area network where terminal servers and operating systems use the Local Area Transport (LAT) protocol. A LAT network can coexist on the same Ethernet with other protocols. If your system uses a LAT network, you should read this section.

To configure your system as a service node within a LAT network, execute the command procedure `SY$MANAGER:LTLOAD.COM` from within `SYSTARTUP_V5.COM`. The `LTLOAD.COM` procedure starts up the LAT protocol. In the LAT protocol, a VMS operating system advertises its services over the Ethernet and responds to connection requests from terminal servers supporting user terminals and other asynchronous devices.

To start up the LAT network, add the following command line to `SYSTARTUP_V5.COM`:

```
$ @SYS$MANAGER:LTLOAD
```

This command in `SYSTARTUP_V5.COM` also configures a node as a service node that connects only to interactive terminals on a terminal server. However, if you want to use remote printers on a terminal server or to create dedicated application services on the VMS service node, you must modify `LTLOAD.COM`.

#### Supporting User Terminals on a Terminal Server

Creating a VMS service node on a LAT network that supports only interactive terminals is a one-step procedure. You insert the command `@SYS$MANAGER:LTLOAD` into `SYSTARTUP_V5.COM` and append any of the following arguments:

```
@SYS$MANAGER:LTLOAD "P1" "P2" "P3" "P4"
```



The arguments P1 through P4 have the following meanings:

Argument	Format	Meaning
P1	Service-name	Name of the VMS service. For clustered VMS service nodes, use the cluster name as the service name. For independent VMS service nodes, use the physical node name.
P2 - P4	Any of the following: /IDENTIFICATION="string"  /ENABLE=group-list  /DISABLE=group-list	Description of the node and its services that are advertised over the Ethernet. The default is the string defined by the logical name SYS\$ANNOUNCE.  Terminal server groups qualified to establish connections with the VMS service node. By default, Group 0 is enabled.  Removes previously enabled terminal server groups.

The argument **P1** assigns a service name to the node, using the LATCP command **CREATE SERVICE**. Arguments **P2** through **P4** can be any valid qualifier to the **SET NODE** command.

For example, the following command creates the service **OFFICE** on the VMS service node, **MOE**, which is part of the **OFFICE** cluster.

```
$ @SYS$MANAGER:LTLOAD OFFICE "/ENABLE=1" "/DISABLE=0"
```

### 2.4.11 Creating Systemwide Announcements

This section describes how to define the following types of systemwide announcements in your **SYSTARTUP\_V5.COM** file:

- A message to users informing them that the system is available for use (after a system boot)
- A message to users when first accessing the system for login
- A welcoming message when a user logs in

When your system has completed the startup procedure and is up and running, you can send a message to all connected terminals announcing the system's availability. To do this, include a line, with an appropriate message within the quotation marks, before the **\$EXIT** command in your **SYSTARTUP\_V5.COM** file:

```
$ REPLY/ALL/BELL "VMS Operating System at WUZNOT, INC., ready for use."
```

If you want to display a message at the beginning of each user's login procedure, include a line, with an appropriate message within the quotation marks, in **SYSTARTUP\_V5.COM**:

```
$ DEFINE/SYSTEM SYS$ANNOUNCE "WUZNOT, INC. -- Authorized Use Only"
```

You can also display a message to all interactive users immediately after they log in by including a line similar to the following in SYSTARTUP\_V5.COM:

```
$ DEFINE/SYSTEM SYS$WELCOME "Welcome to the VMS Operating System at WUZNOT, INC."
```

If you do not define SYS\$WELCOME, the following standard message is displayed:

```
Welcome to VMS Version 5.2
```

The SYSTARTUP\_V5 command file supplied as a template with Digital's distribution kit includes additional command examples for SYS\$ANNOUNCE and SYS\$WELCOME.

You can also display various system announcements to users at the time that they log in. You do this with a command in the systemwide login command procedure, SYLOGIN.COM, as explained in Section 2.5.

## 2.5 Defining a System Login Command Procedure

A system login command procedure is executed for each interactive user when the user logs in. With a system login command procedure, you can establish elements of a computing environment that are the same for all interactive users. To use a system login procedure, do the following:

1. Define the logical name SYS\$SYLOGIN, usually in your site-specific startup file (SYSTARTUP\_V5.COM).
2. Create a system login command procedure.

To define the logical name SYS\$SYLOGIN and point to a system login command file named SYS\$MANAGER:SYLOGIN.COM, include the following line in SYSTARTUP\_V5.COM:

```
$ DEFINE/SYSTEM/EXEC/NOLOG SYS$SYLOGIN SYS$MANAGER:SYLOGIN.COM
```

A template for a system login command procedure is found in SYS\$MANAGER:SYLOGIN.COM. This file includes commands that you can modify and add to according to the needs of your site.

You can use the system login command procedure to display announcements for your site. To do this, you would do as follows:

1. Create a text file that has current announcements, for example with the file name SYS\$MANAGER:ANNOUNCEMENTS.TXT. You could then update this file (adding and deleting announcements) as needed.
2. Include a line at the end of your system login command procedure that displays the announcements file, such as the following:

```
$ TYPE SYS$MANAGER:ANNOUNCEMENTS.TXT
```

In addition to a system login command procedure, users can also have their own login command procedures. User login command procedures are executed immediately after the system login command procedure.

### 2.5.1 Sample System Login Command Procedure

The following example shows some of the possibilities for a system login command procedure:

```
$ if (F$MODE() .eqs. "INTERACTIVE") then set control=t ①
$ if (F$MODE() .eqs. "INTERACTIVE") then set terminal/inquire ②
$
$ define /key pf1 "mail" /term ③
$ define /key pf2 "directory" /term ④
$
$ cop*y ::= copy /log /write_check /read_check ⑤
$ del*ete ::= delete /confirm /log ⑥
$ ed*it ::= edit /tpu /section=tpu$library:key_definitions ⑦
$exit
```

The lines in this procedure do the following:

- ① Allows interactive users to use `CTRLT` to determine current process status.
- ② Sets the proper terminal type for interactive users.
- ③ At DCL level (only), the PF1 key is defined as the MAIL command for all users. (That is, the Mail Utility is entered whenever a user presses `PF1` at DCL level.)
- ④ At DCL level, the PF2 key is defined as the DIRECTORY command.
- ⑤ Whenever a user types the command COP (or COPY), it is interpreted as the COPY command with the qualifiers /LOG, /WRITE\_CHECK, and /READ\_CHECK.
- ⑥ Whenever a user attempts to delete a file using the command DEL (or any expansion of that up to DELETE), the system asks the user for confirmation before actually deleting the file.
- ⑦ Defines the symbol ED (or expansions) as the EVE/TPU editor, using the section file KEY\_DEFINITIONS.TPU\$SECTION.

The logical name *tpu\$library* could be defined in the system login command. However, it is recommended that system logical names be defined elsewhere (for example, in SYSTARTUP\_V5.COM or in SYS\$MANAGER:SYLOGICALS.COM).

## 2.6 Backing Up the System

To limit the risk of losing your operating system environment, you should perform the following sequential operations after installing and customizing your system:

1. Back up the console volume
2. Build a standalone backup kit
3. Back up the system disk

If your computer has a console storage device, Digital recommends that you make a backup copy of your console volume; it is useful to have a backup copy in case your original becomes corrupted. The VMS operating system provides a command procedure called CONSCOPY.COM in the SYS\$UPDATE directory that copies your console volume to a blank one.

To back up your system disk, Digital recommends that you use standalone BACKUP, which uses a subset of Backup Utility qualifiers. If your system was not distributed on magnetic tape, you must build a standalone BACKUP kit either on console media or on disk. You can then boot standalone BACKUP from the console block storage device or from the alternate directory root SYSE on a Files-11 disk.

Installing and using standalone BACKUP in an alternate root on your system disk saves time when you are backing up your system disk, because you do not have to boot standalone BACKUP from your console volume.

**NOTE:** The procedures for backing up the console volume and backing up the system disk vary for different VAX computers. See your VAX computer installation and operations guide for the step-by-step procedures that apply to your computer.

## 2.7 Building and Copying a VMS System Disk

The command procedure SYS\$UPDATE:VMSKITBLD is used for building and copying a VMS system disk. The procedure provides you with the following options:

- **BUILD**—Destroys all previous information on the target disk and then builds the new system disk.
- **ADD**—Adds another copy of the operating system to an alternate system root directory on the same system disk.

- **COPY**—Copies the operating system files to a target disk without destroying the files that are currently on the target disk.
- **COMMON**—Initializes the target disk and builds it as a cluster-common system disk.

**CAUTION:** The **VMSKITBLD BUILD** and **COMMON** options initialize the target disk, deleting all of its previous contents.

In some cases, you might want the operating system to exist on another disk. The following paragraphs describe two such cases and the procedures that you would use.

You might want to move your operating system files to another disk. For example, assume that your operating system is initially stored on a disk together with some of your user files. Suppose that you want to move only the operating system files from the original disk to a smaller disk. You can build the operating system on the smaller disk (called the target, or destination, disk) without affecting the user files on the original disk (the source disk) by using the **BUILD** option of the **VMSKITBLD** command procedure.

You can create a separate test environment where you can modify the operating system without affecting current operations. You could use the **ADD** option to copy the operating system to an alternate system root directory and create a boot command procedure to select that version for testing sessions. In addition, you might want to preserve the current version of the operating system before upgrading your system to the next major version. To do so, use the **ADD** option to make a copy of the current operating system in an alternate system root directory (**SYSA**) and then upgrade and run the new version of the operating system in **SYS0**.

**CAUTION:** When you copy the system disk using **VMSKITBLD.COM**, **SYSUAF.DAT** and all user-modified command files are *not* copied to the target disk. **VMSKITBLD.COM** uses the site-specific template files with the **TEMPLATE** file type in building the new system disk.

## 2.8 System Startup Procedures

This section describes the process that the VMS operating system follows when you boot your system. This section is mostly informational—that is, you usually do not have to do anything during the booting process, but you might want to know how the operating system is set up.

Each time that your system is booted, the VMS operating system initiates a startup procedure. The startup procedure includes the execution of the following series of command procedures:

- **SYS\$SYSTEM:STARTUP.COM**—A file containing a series of procedures that must execute at system startup time in order for the system to

run properly. `STARTUP.COM` is the site-independent startup command procedure supplied by Digital. Do not modify this command procedure. The `STARTUP.COM` procedure invokes the site-specific procedures that are described in this section.

- **`SYS$MANAGER:SYCONFIG.COM`**—A template file supplied by Digital to which you can add site-specific device configuration commands.
- **`SYS$MANAGER:SYLOGICALS.COM`**—A template file supplied by Digital for defining logical names. This file contains a command procedure for adding system logical names for a MicroVAX that is not in a cluster. If your computer is not a standalone MicroVAX, you can ignore that section of the procedure that pertains only to MicroVAX systems and add any systemwide logical name assignments for your own system to the end of this file.
- **`SYS$MANAGER:SYLOGIN.COM`**—A template file supplied by Digital to which you can add commands that are executed whenever a user logs in.
- **`SYS$MANAGER:SYSTARTUP_V5.COM`**—A template file supplied by Digital to which you can add various commands for setting up site-specific operations that are executed at startup time. The template contains commands that you can modify to meet the needs of your processing environment.
- **`SYS$MANAGER:SYPAGSWPFILES.COM`**—A file supplied by Digital to which you can add commands to install page and swap files on any disk.

Two versions of the template files are included in your VMS distribution kit: an executable version with the file extension `COM` and a nonexecutable version with the file extension `TEMPLATE` (for example, `SYS$MANAGER:SYCONFIG.COM` and `SYS$MANAGER:SYCONFIG.TEMPLATE`). The files with the `COM` file type are executed at startup time, and those are the files that you should modify to meet the needs of your site. The files with the `TEMPLATE` file type should not be modified.

**CAUTION:** Do not delete the Digital-supplied template command files with the `TEMPLATE` file type. The `VMSKITBLD.COM` procedure uses the `TEMPLATE` versions to create a new system disk.

More information on `STARTUP.COM` and the site-specific command procedures is provided in the sections that follow.

## 2.8.1 Startup Command Procedure for the System (STARTUP.COM)

This section describes the system startup file (STARTUP.COM). STARTUP.COM is executed whenever the system is booted, and it creates the basic environment for the operating system and some software products. It is not a startup file that is customized for your site. You should not modify the STARTUP.COM file. Read this section if you are interested in learning about the startup process.

The file SYSTARTUP\_V5.COM, which is also executed each time the system is booted, is the startup file where you include features specific to your site. To learn how to customize the startup process for your site by modifying SYSTARTUP\_V5.COM, see Section 2.4.

The file SYS\$SYSTEM:STARTUP.COM executes immediately after the operating system is booted. It is a driver that uses a series of component files to perform the following startup tasks:

- Defines systemwide logical names required for the symbolic debugger, language processors, linker, image activator, and help processor.
- Starts processes that control error logging, SMISERVER (the system management server), the job controller, and the operator's log. (On a standalone workstation, the operator's log is not automatically started.)
- Connects and configures devices that are physically attached to the system and loads their I/O drivers by invoking the SYCONFIG.COM procedure.
- Installs known images to reduce I/O overhead in activating the most commonly run images or to identify images that must have special privileges.

**CAUTION:** Do not modify SYS\$SYSTEM:STARTUP.COM. This file is deleted and replaced each time you upgrade your system with the next version of the VMS operating system. Leaving STARTUP.COM intact prevents you from inadvertently altering any commands in the file, which in turn could cause the startup procedure to fail.

All of the component files used by STARTUP.COM are in the directories with the logical name **SYS\$STARTUP**. SYS\$STARTUP is actually a searchlist that includes both SYS\$SYSROOT:[SYSMGR] (the SYS\$MANAGER directory) and SYS\$SYSROOT:[SYS\$STARTUP].

The following three data files are involved in the startup process and located in SYS\$STARTUP:

- **VMS\$PHASES.DAT**—This file determines the order of the phases of the startup procedure. It is a sequential list of the phases that will be started by STARTUP.COM. It includes a series of four basic phases (INITIAL, CONFIGURE, SYSFILES, and BASEENVIRON) needed to bring the VMS operating system up to a basic working environment, followed by a series of phases for optional software products. This file must not be modified.

## 2-18 Starting Up the System

- **VMS\$VMS.DAT**—This is a component data file for starting the base VMS operating system environment. You should not modify this file.
- **VMS\$LAYERED.DAT**—This is a component file for optional software products that are installed using the callback procedure of **VMSINSTAL**. It is an indexed-sequential file, containing the following fields for each file:
  1. Name of the component file to be run (with either **EXE** or **COM** file type).
  2. Phase in which the component file is to be run. The valid phases are **LPBEGIN**, **LPMAIN** (default), **LPBETA**, and **END**.
  3. Method (or mode) by which the component file is to run. The valid choices are **DIRECT** (the default, where the command procedure or image is executed immediately), **BATCH** (valid only for command procedures), or **SPAWN**.
  4. Node restrictions for the component. This is either the node or nodes on which the component file should *only* be run, or the node or nodes on which the component file should *not* be run.
  5. Node restriction byte field. This field determines whether the nodes listed in the previous field are allowed or disallowed (for running the component).
  6. Parameters passed to the component file for execution. You can pass up to eight parameters, using the following format:

(P1:args,P2:args,...)

(The parentheses can be omitted if you pass only a single parameter.)

An important function of each phase is to meet the prerequisites of the following phase; therefore, the ordering of the phases is extremely important. Components that occur in a phase cannot have dependencies on components that are in the same phase or in subsequent phases. When installing optional software products as known images using the **STARTUP.COM** procedure, be sure that all requisite components occur in a previous phase.

If an optional software product can use the callback procedure included in **VMSINSTAL**, then you can install it as a known image at system startup using the method described earlier in this section, and you do not have to include the product in the site-specific startup file (**SYSTARTUP\_V5.COM**). In these cases, the component files must be in the **SYS\$STARTUP** directory. Software products that do not use the callback procedure should be installed as known images at system startup using **SYSTARTUP\_V5.COM**.

You can also use the System Management Utility (**SYSMAN**) to manage the new startup process. With the **STARTUP** command of **SYSMAN**, you can add, modify, display, or remove elements of existing component files, create a new startup file, and perform other startup functions. A description of **SYSMAN** commands is found in the Reference Section.



Several site-specific command procedures are executed from within `STARTUP.COM`. You can add commands to these files or modify the template files supplied in your VMS distribution kit. Remember, however, to modify only the executable version of the file (with the file extension `COM`) and not the template version (with the file extension `TEMPLATE`). If you have an existing `COM` file and you want to modify a version of the original `TEMPLATE` file, then you should first make a copy of the `TEMPLATE` file (giving the copy a file type of `COM`).

`STARTUP.COM` executes the site-specific command procedures in the following sequence:

1. `SYS$MANAGER:SYPAWSWPFILES.COM`
2. `SYS$MANAGER:SYCONFIG.COM`
3. `SYS$MANAGER:SYLOGICALS.COM`
4. `SYS$MANAGER:SYSTARTUP_V5.COM`

## 2.8.2 Setting Up Logical Names for Your Site (`SYLOGICALS.COM`)

A logical name is a name that is equivalent to a file specification, a directory, a device name, another logical name, or some other equivalence string. For example, when you have a logical name associated with a device name, you can use the logical name instead of the formal device name.

You can assign logical names that apply for the entire system; these are called systemwide logical names, and they can be used by any process on the system. For example, if a systemwide logical name equated the logical name `FINANCE_DISK` to the device `DRA2`, any user on the system (and any program running on the system) could use the name `FINANCE_DISK` in place of `DRA2`.

The file `SYS$MANAGER:SYLOGICALS.COM` can be used for assigning systemwide logical names. `SYLOGICALS.COM` is executed as part of the `STARTUP.COM` procedure whenever your system is booted. The logical names defined in `SYLOGICALS.COM` (as well as the logical names assigned automatically in `STARTUP.COM`) are always included in the system logical name table.

If your system is a MicroVAX that is *not* in a cluster, you should use the file `SYLOGICALS.COM` as a template for assigning systemwide logical names. If you have a MicroVAX that is not in a VAXcluster environment and you want to have systemwide logical names, you should read this section.

Unless your computer is a MicroVAX that is not in a VAXcluster environment, the template procedure that is found in `SYLOGICALS.COM` has no effect. However, if your computer is one where the template procedure does not apply, you can use `SYLOGICALS.COM` to assign systemwide logical names by adding them to `SYLOGICALS.COM` before the `EXIT` command, as indicated in the `SYLOGICALS.COM` template.

During VMS system operations when the integrity of the system could be compromised by incorrect logical names, such as the activation of privileged images (LOGINOUT, MAIL, and so forth), only executive-mode and kernel-mode logical names are used; supervisor-mode and user-mode names are ignored. Digital therefore recommends that logical names for system components (for example, public disks and directories) be defined in executive mode, for example:

```
$ DEFINE/SYSTEM/EXECUTIVE/NOLOG SYSDSK SYS$SYSDEVICE:
```

See the *VMS User's Manual* for information about logical name assignments and the privilege modes.

## 2.9 Emergency Startup Procedures

The startup and login procedures provided by Digital should always work; however, certain user interventions can cause them to fail. For example, if you modify the startup or login procedures, or modify the login accounts, you might accidentally lock yourself out of the system. A very simple way to lock yourself out of the system is to set passwords and forget them. Another way to lock yourself out is to introduce an error condition or an infinite loop into a startup or login procedure. Under such circumstances, use the emergency startup procedure described in this section.

### 2.9.1 Bypassing the User Authorization File

The preferred method of breaking into a locked system is to set the alternate user authorization file. This method requires setting the system parameter UAFALTERNATE, which defines the logical name SYSUAF to refer to the file SYS\$SYSTEM:SYSUAFALT.DAT. If this file is found during a normal login, the system uses it to validate the account and prompts you for the user name and password.

If this file is not located, the system assumes that the UAF is corrupt and accepts any user name and password to log you into the system from the system console. Logins are prohibited from all other locations.

**NOTE:** You can use this method to log into the system only from the console terminal; you cannot use other terminal lines.

To set the alternate user authorization file, use the following procedure:

1. Perform a conversational boot by following the instructions in your VAX computer installation and operations guide.
2. When the SYSBOOT> prompt appears, enter the following command:  

```
SYSBOOT> SET UAFALTERNATE 1
```
3. Type CONTINUE and press RETURN.

4. When the startup procedure completes, log in on the console terminal by entering any user name and password in response to the *Username:* and *Password:* prompts.

The system assigns the following values to your user account:

- Name—User name
- UIC—[001,004]
- Command interpreter—DCL
- Login flags—None
- Priority—Value of the system parameter DEFPRI
- Resources—Values of the PQL system parameters
- Privileges—All

The process name is usually the name of the device on which you logged in (for example, \_OPA0).

5. Fix the problem that caused you to be locked out of the system. That is, make the necessary repairs to the UAF or to the startup or login procedures. (If you modify a login or startup procedure and the problem is still not solved, you should restore the procedure to its previous state.)

If the problem is a forgotten password, reset the UAFALTERNATE system parameter to 0, as explained in the next step. Then invoke the Authorize Utility and type HELP MODIFY for information on modifying passwords.

6. Clear the UAFALTERNATE parameter by running SYSGEN and giving appropriate SYSGEN commands. To run SYSGEN, enter the following command at the DCL prompt:

```
$ RUN SYS$SYSTEM:SYSGEN
```

The SYSGEN> prompt is displayed, and you should enter the following commands:

```
SYSGEN> SET UAFALTERNATE 0
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
```

7. Shut down and reboot the system.

## 2.9.2 Emergency Startup After Modifying System Parameters

In some cases, modifying system parameters might cause the system to become unbootable. If this occurs, use the following emergency startup procedure to restore normal operation:

1. Perform a conversational boot by following the instructions in your VAX computer installation and operations guide.
2. When the SYSBOOT> prompt appears, enter the following commands:  

```
SYSBOOT> USE DEFAULT.PAR  
SYSBOOT> CONTINUE
```
3. When the system finishes booting, review any changes you made to SYSGEN parameters, modify SYS\$SYSTEM:MODPARAMS.DAT as necessary, and reexecute AUTOGEN.

## 2.9.3 Bypassing Startup and Login Procedures

If the system does not complete the startup procedures or does not allow you to log in, bypass the startup and login procedures by following these steps:

1. Perform a conversational boot operation by following the instructions in your VAX computer installation and operations guide.
2. Define the console to be the startup procedure by entering the following command at the SYSBOOT> prompt:

```
SYSBOOT> SET/STARTUP OPAO:
```

Type CONTINUE and press RETURN in response to the next SYSBOOT> prompt. Wait for the DCL prompt to return.

3. Correct the error condition that caused the login failure. That is, make the necessary repairs to the startup or login procedures, or to the UAF. You might want to enter the following DCL commands because bypassing the startup procedures leaves the system in a partially initialized state:

```
$ SET NOON  
$ SET DEFAULT SYS$SYSROOT:[SYSEXE]
```

Invoke a text editor to correct the startup or login procedure file. Note that some system consoles might not supply a screen-mode editor. You can also copy a corrected file and delete the incorrect version by using the RENAME and DELETE commands.

4. Reset the startup procedure by invoking SYSGEN and entering the following commands:

```
$ RUN SYS$SYSTEM:SYSGEN  
SYSGEN> SET/STARTUP SYS$SYSTEM:STARTUP.COM  
SYSGEN> WRITE CURRENT  
SYSGEN> EXIT
```

5. Perform a normal startup by entering the following command:

```
$ @SYS$SYSTEM:STARTUP
```

## 2.9.4 Startup Problems

Sometimes the operating system does not boot after you enter the `BOOT` command. This can be caused by either a hardware or software malfunction.

A read error on a disk drive or console medium, or a machine check error, might indicate a hardware malfunction. When a hardware problem occurs, a question mark (?) usually precedes the error message that is displayed on the system console terminal. You should then do the following:

1. Consult the hardware manual for your VAX computer.
2. If you still cannot correct the problem, contact your Digital Field Service representative.

When the operating system is loaded into memory but the `STARTUP.COM` command procedure does not execute, a software malfunction has probably occurred. You should suspect this condition if the usual message specifying the number of interactive users does not appear.

Perform one or both of the following actions to correct the situation:

- Try again, by repeating the boot procedure to restart the system (see the installation guide for your VAX computer).
- Leave the system disk in the original drive. Restore a backup copy of the system disk using Standalone Backup.

## 2.10 Shutdown Procedures

The VMS operating system provides the following types of shutdown procedures:

- **An orderly shutdown with `SYS$SYSTEM:SHUTDOWN.COM`.** This procedure shuts down the system while performing housekeeping functions such as disabling future logins, stopping the batch and output queues, dismounting mounted volumes, and stopping user processes.

`SHUTDOWN.COM` optionally invokes a site-specific command procedure named `SYS$MANAGER:SYSHUTDWN.COM`, which you can modify to meet the needs of your specific installation. An empty `SYSHUTDWN.COM` file is included in your VMS distribution kit.

- **An emergency shutdown with `SYS$SYSTEM:OPCCRASH.EXE`.** If you are unable to perform an orderly shutdown with `SHUTDOWN.COM`, run the `OPCCRASH` emergency shutdown program.

## 2-24 Starting Up the System

- **An emergency shutdown with CRASH.** Use this emergency shutdown procedure if OPCCRASH fails. Note that not all VAX computers have the CRASH emergency shutdown program. If your VAX computer has the CRASH procedure, it is located on the console media, and it can be executed only from the console terminal. See your VAX computer installation and operations guide for a description of the CRASH procedure or for the equivalent commands to use to force an abrupt emergency shutdown.

### 2.10.1 Orderly Shutdown

Use SHUTDOWN.COM to shut down the system in an orderly fashion. Do not modify SHUTDOWN.COM. Add commands to the SYS\$MANAGER:SYSHUTDWN.COM command procedure to perform site-specific functions.

To execute SHUTDOWN.COM, you must have either the SETPRV privilege or all the following privileges: CMKRNL, EXQUOTA, LOG\_IO, OPER, SYSNAM, SYSPRV, TMPMBX, and WORLD. Usually, you shut down the system from the SYSTEM account, which includes these privileges by default.

#### SHUTDOWN.COM Sequence of Prompts and Messages

To perform an orderly shutdown of the system, invoke SHUTDOWN.COM from any terminal and any privileged account with the following DCL command:

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

The procedure then prompts you with a series of questions and messages. The default responses appear in brackets at the end of each question. Press the RETURN key to select the default response. A summary of the questions follows:

- **Minutes until shutdown:**

```
How many minutes until final shutdown [0]?
```

Enter an integer. If the system logical name SHUTDOWN\$MINIMUM\_MINUTES is defined, its integer value is the minimum value that you can enter. Therefore, if the logical name is defined as 10, you must specify at least 10 minutes to final shutdown or an error message is returned. If you do not enter a value, the logical name value is used. If the logical name is not defined, and you do not enter a value, 0 minutes is the default.

- **Reason for shutdown:**

```
Reason for shutdown [standalone]:
```

Enter a one-line reason for shutting down the system.

- **Decide if you want to spin down the disk volumes:**

```
Do you want to spin down the disk volumes [No]?
```

Enter YES or NO (Y or N). Note, however, that the system disk cannot be spun down.

- Decide if you want to invoke a site-specific shutdown procedure:

Do you want to invoke the site-specific shutdown procedure [Yes]?

Enter YES or NO. This refers to SYS\$MANAGER:SYSHUTDWN.COM.

- Decide if you want an automatic system reboot:

Should an automatic system reboot be performed [No]?

By default, the system does not automatically reboot. However, if you respond with YES, the system attempts to reboot automatically when the shutdown is complete.

- Message broadcast to users about rebooting the system:

When will the system be rebooted [later]?

Enter the expected reboot time in the format you want printed in the message that will be broadcast to users. For example, you could specify IMMEDIATELY, or IN 10 MINUTES, or a time such as 2 P.M. or 14:00. If you do not know when the system will be available again, press RETURN to specify "later" as the time when the system will reboot.

- Shutdown options:

Shutdown options (enter as a comma-separated list):

SAVE_FEEDBACK	Saves feedback data for AUTOGEN calculations
REMOVE_NODE	Remaining nodes in the cluster should adjust quorum
CLUSTER_SHUTDOWN	Entire cluster is shutting down
REBOOT_CHECK	Check existence of basic system files

Shutdown options [NONE]

The procedure prompts you to specify one or more shutdown options.

Entering the SAVE\_FEEDBACK option records feedback data collected from the system since it was last booted. This option creates a new version of the AUTOGEN feedback data file, which can be used when you next run AUTOGEN.

If your system is a cluster member, all options are listed. When the REMOVE\_NODE option is specified on one cluster member system, users on all member systems are notified. Clusterwide notification is required, because users logged in to any member system might be affected by the shutdown of another system, for example:

- Users might have batch jobs running on other systems.
- If terminal servers are in operation, users might have alternate terminal sessions in progress on the system being shut down.

Otherwise, only the REBOOT\_CHECK and SAVE\_FEEDBACK options are listed. Enter REBOOT\_CHECK to verify the presence of a subset of files necessary to reboot the system after shutdown completes. (If you are certain that the files exist, press RETURN.)

## 2-26 Starting Up the System

If you select the `REBOOT_CHECK` option, the procedure checks for the necessary files and notifies you if any are missing. Replace missing files before proceeding. If all files are present, the following success message appears:

```
%SHUTDOWN-I-CHECKOK, Basic reboot consistency check completed.
```

The following events occur as the shutdown procedure continues, and the corresponding messages are printed on the terminal:

1. A message requesting users to log out is broadcast at decreasing time intervals to all users on the system.
2. The system logical name `SHUTDOWN$TIME` is defined as the absolute time of shutdown. For example, if the value 10 is specified at 12:00 in response to the first question, the logical name is assigned the absolute time value 12:10 along with the date. By requesting the logical name definition for `SHUTDOWN$TIME` (with the `SHOW LOGICAL` command), you can see if a shutdown is in progress or determine the actual time of shutdown. This feature is useful if a user missed the shutdown broadcast message.
3. At six minutes or less until system shutdown, the terminal from which shutdown was invoked is made an operator's console and all future nonoperator logins are disabled. If the DECnet network is running, it is shut down.
4. When there is one minute left until system shutdown, batch and device queues and the system job queue manager are stopped.
5. At zero minutes before system shutdown, the site-specific command procedure `SYS$MANAGER:SYSHUTDWN.COM` is invoked.
6. All user processes are stopped; however, system processes continue. Ancillary Control Processes (ACPs) might delete themselves when their mounted volumes are finally dismounted.
7. For dual-processor systems, the secondary processor is stopped.
8. All installed images are removed.
9. All mounted volumes are dismounted and, if you request it, the disks are spun down. Note, however, that the system disk cannot be spun down. Also, the quorum disk (if present on your system) is not dismounted or spun down.
10. The operator's log file is closed.
11. The program `SYS$SYSTEM:OPCCRASH` is invoked to shut down the system.
12. If you did not request an automatic reboot, the following message appears on the system console:

```
SYSTEM SHUTDOWN COMPLETE - USE CONSOLE TO HALT SYSTEM
```

If you requested an automatic reboot, the system reboots, provided the necessary controls are set.



13. If you are not automatically rebooting, halt the system after the previous message is printed at the console terminal.

Example 2-1 demonstrates an orderly system shutdown on standalone node AVALON.

### Example 2-1: Orderly System Shutdown with SHUTDOWN.COM

---

```
$ @SYSS$SYSTEM:SHUTDOWN

      SHUTDOWN -- Perform an Orderly System Shutdown

How many minutes until final shutdown [0]: 10
Reason for shutdown: [Standalone] MONTHLY PREVENTIVE MAINTENANCE.
Do you want to spin down the disk volumes [No]?  RET
Do you want to invoke the site-specific shutdown procedure [Yes]?  RET
Should an automatic system reboot be performed [No]?  RET
When will the system be rebooted [later]? 12:30
Shutdown options:
  REBOOT_CHECK      Check existence of basic system files
  SAVE_FEEDBACK     Save AUTOGEN feedback information from this boot
Shutdown options [NONE]  RET

SHUTDOWN message on AVALON, from user SYSTEM at _AVALON$OPA0: 12:00:00.20
AVALON will shut down in 10 minutes; back up 12:30. Please log off node AVALON.
MONTHLY PREVENTIVE MAINTENANCE

%SHUTDOWN-I-OPERATOR, This terminal is now an operator's console.
%%%%%%%%%%%%%%%% OPCOM, 19-APR-1990 12:01:00.15 %%%%%%%%%%%%%%%%%
Operator status for operator _AVALON$OPA0:
CENTRAL, PRINTER, TAPES, DISKS, DEVICES, CARDS, NETWORK, OPER1, OPER2,
OPER3, OPER4, OPER5, OPER6, OPER7, OPER8, OPER9, OPER10, OPER11,
OPER12

%SHUTDOWN-I-DISLOGINS, Interactive logins will now be disabled.
%SET-I-INTSET, login interactive limit = 0 current interactive value = 17
%SHUTDOWN-I-SHUTNET, The DECnet network will now be shut down.
%SHUTDOWN-I-STOPQUEMAN, The queue manager will now be stopped.

SHUTDOWN message on AVALON, from user SYSTEM at _AVALON$OPA0: 12:05:00.20
AVALON will shut down in 5 minutes; back up 12:30. Please log off node AVALON.
MONTHLY PREVENTIVE MAINTENANCE

17 terminals have been notified on AVALON.

SHUTDOWN message on AVALON from user SYSTEM at _AVALON$OPA0: 12:06:55.28
AVALON will shut down in 4 minutes; back up 12:30. Please log off node AVALON.
MONTHLY PREVENTIVE MAINTENANCE

%%%%%%%%%%%%%%%% OPCOM, 19-APR-1990 12:07:12.30 %%%%%%%%%%%%%%%%%
Message from user DECnet on AVALON
DECnet event 2.0, local node state change
From node 2.161 (AVALON), 19-APR-1990 12:07:22.26
Operator command, Old state = On, New state = Shut

SHUTDOWN message on AVALON from user SYSTEM at _AVALON$OPA0: 12:07:12.56
AVALON will shut down in 3 minutes; back up 12:30. Please log off node AVALON.
MONTHLY PREVENTIVE MAINTENANCE
```

---

(continued on next page)

### Example 2-1 (Cont.): Orderly System Shutdown with SHUTDOWN.COM

---

```
%SHUTDOWN-I-STOPQUEMAN, The queue manager will now be stopped.
SHUTDOWN message on AVALON user SYSTEM at _AVALON$OPA0: 12:08:12.56
AVALON will shut down in 2 minutes; back up 12:30. Please log off node AVALON.
MONTHLY PREVENTIVE MAINTENANCE
```

```
%%%%%%%%%% OPCOM, 19-APR-1990 12:08:12.30 %%%%%%%%%%%
Message from user JOB_CONTROL on AVALON
-SYSTEM-S-NORMAL, normal successful completion
```

```
%%%%%%%%%% OPCOM, 19-APR-1990 12:08:42.30 %%%%%%%%%%%
Message from user DECNET on AVALON
DECnet shutting down
```

```
SHUTDOWN message on AVALON from user SYSTEM at _AVALON$OPA0: 12:09:12.56
AVALON will shut down in 1 minute; back up 12:30. Please log off node AVALON.
MONTHLY PREVENTIVE MAINTENANCE
```

```
17 terminals have been notified on AVALON
%SHUTDOWN-I-SITESHUT, The site-specific shutdown procedure will now be invoked.
%SHUTDOWN-I-STOPUSER, All user processes will now be stopped.
%SHUTDOWN-I-REMOVE, All installed images will now be removed.
%SHUTDOWN-I-DISMOUNT, All volumes will now be dismounted.
%%%%%%%%%% OPCOM, 19-APR-1990 12:09:42.30 %%%%%%%%%%%
Message from user System on AVALON
_AVALON$OPA0:, AVALON shutdown was requested by the operator.
```

```
%%%%%%%%%% OPCOM, 19-APR-1990 12:10:02.44 %%%%%%%%%%%
Logfile was closed by operator _AVALON$OPA0:
Logfile was SYS$SYSROOT:[SYSMGR]OPERATOR.LOG;8
```

```
%%%%%%%%%% OPCOM, 19-APR-1990 12:10:32.20 %%%%%%%%%%%
Operator _AVALON$OPA0: has been disabled, username SYSTEM
```

---

SYSTEM SHUTDOWN COMPLETE - USE CONSOLE TO HALT SYSTEM

---

### Defining SHUTDOWN\$INFORM\_NODES

Before you execute `SYS$SYSTEM:SHUTDOWN.COM`, you can define the logical name `SHUTDOWN$INFORM_NODES` to be a list of cluster member nodes. The nodes listed in `SHUTDOWN$INFORM_NODES` will be notified when the system is shut down, as shown in the following example:

```
$ DEFINE SHUTDOWN$INFORM_NODES "NODE1,NODE2,NODE3"
$ @SYS$SYSTEM:SHUTDOWN
```

If you define `SHUTDOWN$INFORM_NODES` and then execute `SYS$SYSTEM:SHUTDOWN.COM`, all cluster member nodes included in the list are notified of the shutdown. Users on the node that is being shut down are always notified regardless of whether you define `SHUTDOWN$INFORM_NODES`. If you omit the name of the node that is being shut down from the list specified in the `DEFINE` command, the name is automatically added to the list by the `SHUTDOWN.COM` procedure.

## 2.10.2 Emergency Shutdown

This section describes how to halt the system immediately without performing any of the housekeeping functions that ensure an orderly shutdown. Usually, you shut down the system using the orderly shutdown procedure. You use the OPCCRASH procedure only if SHUTDOWN.COM fails. OPCCRASH performs only the following minimal housekeeping functions:

- Marks the system disk for dismount and empties all file system data caches.
- Writes the modified page list back to the disk. This ensures that all writable section files are updated to their correct state before the system crashes and all in-memory data is lost.

To perform this procedure, you must have the CMKRNL privilege. You can enter the commands from any terminal.

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

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

2. If the system fails to respond after a few minutes, use the CRASH procedure or, if your system does not have a CRASH procedure, enter the emergency shutdown commands described in your VAX computer installation and operations guide.
3. At the system console, the following message is displayed:

```
SYSTEM SHUTDOWN COMPLETE - USE CONSOLE TO HALT SYSTEM
```

4. Halt the system.

Example 2-2 illustrates an emergency shutdown using the OPCCRASH procedure.

### Example 2-2: Emergency Shutdown Using OPCCRASH

---

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

```
SYSTEM SHUTDOWN COMPLETE - USE CONSOLE TO HALT SYSTEM
```

```
CTRL/P
```

```
>>>HALT
```

```
HALTED AT 8000708A
```

---

## 2.11 Summary

### Booting the System

The process of starting up your system is called *booting*. When you boot your system for the very first time, you install the VMS operating system. After the operating system has been installed, you can begin to customize the system and its startup procedures to meet your site's needs.

### The Site-Specific Startup File—SYSTARTUP\_V5.COM

The file `SYS$MANAGER:SYSTARTUP_V5.COM` is the file that you can customize for your site's needs. Every time you boot the system, `SYSTARTUP_V5.COM` is executed.

Some of the system startup tasks that can be done in `SYSTARTUP_V5.COM` include:

- Mounting public disks
- Setting the proper characteristics for printers, terminals, and other devices
- Establishing print and batch queues
- Installing known images
- Starting up the DECnet network
- Starting up a LAT network
- Starting up DECwindows
- Executing startup files for software products
- Submitting batch jobs that should be run at startup time

(Note that this list is only a subset of the possible applications for `SYSTARTUP_V5.COM`.)

### System Login Command Procedure

A system login command procedure is executed for every user when they log in to the system. As system manager, you can use a system login command procedure to define site-specific symbols or any other functions appropriate for a login command.

To use a system login command procedure, first modify the file `SYS$MANAGER:SYLOGIN.COM` to meet your site's requirements. Then, define the logical name `SYS$SYLOGIN` to point to this system login command procedure file.

System logical names should be defined in the file `SYS$MANAGER:SYLOGICALS.COM`.

**Back Up Your Operating System!**

After you have installed and customized the VMS operating system, you should back up your system in order to limit the risk of losing your operating system environment.



# Chapter 3

## Installing Software

As was the case with the VMS operating system, any additional software products that you use on your system must be **installed**. By installing software, the software is made available every time that the system starts up. For each software product that you use, you must use an installation procedure to install the product for the first time, and you must also use an installation procedure each time that you upgrade the software (for example, to a newer version).

On the VMS operating system, you use the command procedure `SYS$UPDATE:VMSINSTAL.COM` to install most software products and any upgrades. Use `VMSINSTAL.COM` to upgrade your system to the most recent version of the VMS operating system and any subsequent maintenance updates, and to install many optional software products.

This chapter describes the following:

- How to prepare your system for running `VMSINSTAL.COM`
- How to start `VMSINSTAL.COM`
- How to select appropriate `VMSINSTAL.COM` options
- What to do if the system fails while running `VMSINSTAL.COM`

This chapter does not describe `VMSINSTAL.COM` procedures that are specific to any upgrade, update, or product. The examples used are for illustration only. For details of a particular product, refer to the `VMSINSTAL.COM` procedure described in the installation documentation for the specific product or update.

### 3.1 Preparing Your System for `VMSINSTAL`

This section provides guidelines for preparing your system for installing software such as the VMS operating system. Not all of the steps listed in this section are required for each software product that you install; refer to the installation documentation of each product to determine which of the following steps are needed.

## 3-2 Installing Software

Before you use VMSINSTAL.COM, do the following:

1. Back up your system disk, as described in Chapter 8. Use the backup copy as a working copy for the installation.

If you back up your system disk to magnetic tape, you must restore the tape to a Files-11 disk to get a working copy. The working copy has more contiguous space than the original system disk because of the way BACKUP creates the copy. You might need this additional contiguous space during the installation.

If the system fails during installation, VMSINSTAL.COM might delete the older version of the product before it installs the newer version. You might have to make a new working copy of the system disk and restart the installation.

2. Log in to the SYSTEM account at the console terminal. (If SYSGEN parameters MOUNTMSG or DISMOUMSG are set to 1, OPCOM displays a message each time a disk or tape is mounted or dismounted. If these messages are not disabled, and if you are installing from an operator's terminal, they appear within 30 seconds of each mount or dismount.)
3. Be sure that all users are logged out and that all batch jobs are completed.
4. Keep users off the system using the following command:

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

5. If your system includes it, you should shut down DECnet-VAX if DECnet-VAX would be affected by the software product that you are installing. To shut down DECnet-VAX, do the following:

- a. Start the Network Control Program (NCP):

```
$ RUN SYS$SYSTEM:NCP
```

- b. At the NCP prompt (NCP>), enter the following command:

```
NCP> SET EXECUTOR STATE SHUT
```

DECnet-VAX performs an orderly shutdown. The OPCOM facility notifies you when DECnet-VAX is shut down.

- c. At the NCP prompt, enter EXIT.

6. Make sure the limits in the SYSTEM account authorization record are equal to or greater than the following:



```

Buffered byte count quota (BYTLM) = 20480
Enqueue quota (ENQLM) = 200
Direct I/O limit (DIOLM) = 18
Buffered I/O limit (BIOLM) = 18
Open file quota (FILLM) = 40
AST limit (ASTLM) = 24

```

- a. To check these limits, run the Authorize Utility (AUTHORIZE). To run AUTHORIZE, enter the following commands:

```

$ SET DEFAULT SYSS$SYSTEM
$ RUN AUTHORIZE

```

- b. At the UAF prompt (UAF>), enter the following command:

```
UAF> SHOW SYSTEM
```

AUTHORIZE displays the current limits of the SYSTEM account's user authorization file.

7. If necessary, modify the SYSTEM account, using the AUTHORIZE utility as described in Section 4.4.

For example, to increase the DIOLM limit to 18, enter the following command:

```
UAF> MODIFY SYSTEM/DIOLM=18
```

8. When you have changed the limits, at the UAF prompt (UAF>) enter EXIT. This returns you to the dollar-sign prompt (\$). Your changes will not take effect until you log out and log in again. (See the Reference Section for a description of the commands in the Authorize Utility.)
9. Physically mount the first distribution medium that contains the software product.

### 3.1.1 Starting the VMSINSTAL Procedure

This section tells you how to start VMSINSTAL.COM and describes options that you can use.

When you start VMSINSTAL.COM, several prompts and messages direct and explain the installation. These prompts and messages differ, depending on the software product that you are installing. Before you begin, make sure that you read and understand the installation procedures for the specific product or update. If you need assistance during an installation, enter a question mark (?) for an explanation of acceptable responses.

When you start VMSINSTAL.COM, it asks if you are satisfied with the backup of your system disk. If you have a recent backup of your system disk, then you can continue; otherwise, you should back up your system disk before continuing with the installation.

## 3-4 Installing Software

If you have not satisfied all conditions required to start VMSINSTAL.COM, a warning message explaining the problem is displayed, and you are asked if you want to continue. Digital strongly recommends that you correct these conditions before you try to start VMSINSTAL.COM again.

**NOTE:** If you continue without making the required corrections, Digital might not support the resulting installation.

To start VMSINSTAL.COM, enter a command in the following format:

```
@SYS$UPDATE:VMSINSTAL.COM product-list source: [OPTIONS option-list] -  
[destination] [qualifiers]
```

For example, suppose you wanted to install the product CALENDAR, from a saveset named CALENDARV2.A on a magnetic tape in drive MUA0. In the simplest case, where you use no options or qualifiers, you would use the following command:

```
$ @SYS$UPDATE:VMSINSTAL.COM CALENDARV2 MUA0:
```

The following sections describe the parameters and the keyword that you must supply in this command.

### 3.1.1.1 Selecting a Product-List

This parameter lists the products that you want to install. If you use a wildcard character (\*), all versions and updates of all products from the distribution media will be installed in alphabetical order.

If you want to specify more than one item in the product list parameter, you must separate the items using commas and no intervening spaces. Specify the product list in the following format:

facvvu

fac	The product name code (1 to 36 alphanumeric characters)
vv	The major version number (2 digits)
u	The update number (1 digit)

For example, if you upgrade your operating system to VMS Version 5.2, the product name (*fac*) is VMS, the major version (*vv*) is 05, and the update number (*u*) is 2. Therefore the product-list parameter is VMS052.

Using this format, you can install a specific version and update of a product from distribution media containing several versions and updates. If you do not include a version or update number, all versions and updates from the source are installed in alphabetical order.

If you are installing from a distribution kit, the list of products on your distribution media is included with the bill of materials for the distribution kit. If the list is not available, enter a `DIRECTORY` command so that the distribution media will find the products that are included. To obtain the product list, enter commands in the following format:

```
MOUNT/OV=ID ddcu:
DIRECTORY ddcu:[0,0]
```

where *ddcu* is the drive that holds the distribution media.

If you are installing from a disk directory, obtain the product list by entering a `DIRECTORY` command, specifying the disk directory in the following format:

```
DIRECTORY node::drive:[directory]
```

If you are accessing a remote node, you must have `READ` and `EXECUTE` access (R,E) to the directory.

### 3.1.1.2 Selecting the Source

This parameter identifies the source of the optional software product as one of the following:

- A drive that holds the distribution media; for example, the RX50 drive designated CSA2: on a VAX 8200.
- A disk directory to which the product save set has been transferred from the distribution media for later installation.

You specify a disk directory as the source when you select the `GET SAVE SET` option. For more information about this option, see Section 3.1.3.

- A disk directory on another node.

You also can use a logical name to specify the source. If you do not specify the source, `VMSINSTAL.COM` asks you for it.

### 3.1.1.3 Selecting Options

This keyword is optional. It precedes the list of options to be installed. If you enter an option list without the `OPTIONS` keyword, `VMSINSTAL.COM` displays an informational error message and the installation ends.

The options-list parameter lists the options requested. The `VMSINSTAL.COM` command procedure permits the use of five options. You specify each option by entering the appropriate option letter after the keyword `OPTIONS` in the command that starts `VMSINSTAL.COM`.

If you specify more than one option, do not separate the letters with spaces or commas. For more information about `VMSINSTAL.COM` options, see Section 3.1.3.

## 3-6 Installing Software

### 3.1.1.4 Selecting the Destination

This parameter is optional. By default, VMSINSTAL.COM assumes that the product is to be installed in the system-specific directory root on the system disk. However, there are two instances in which you must use this parameter:

- If you want to install the product in an alternate root. The product is installed on a system root other than that on which the target system is running. Specify the alternate system root in the following format:

ddcu:[SYSn.]

ddcu                The device on which the alternate root resides.

SYSn.              The top-level directory of the alternate system root.

You can also specify a previously defined logical name for the alternate root.

- If you specify the GET SAVE SET option to copy the product kit save sets into a storage directory for later installation. For more information on the GET SAVE SET option, see Section 3.1.3.

Specify the destination directory in the following format:

[node::]ddcu:[directory]

node                A node remote from the target system. (DECnet software must be installed on your system to use this option.) If you do not specify a node, VMSINSTAL.COM assumes that the product save sets are to be stored on the local node.

ddcu                The destination disk device.

directory          Usually, a directory dedicated to product save sets on the specified disk.

### 3.1.1.5 Qualifying the BACKUP Command

You can specify this parameter along with the GET SAVE SET option to qualify the BACKUP command further. The BACKUP command copies the save sets to the destination directory.

The following example includes the GET SAVE SET option and BACKUP qualifiers:

```
$ @SYS$UPDATE:VMSINSTAL.COM TEST042 DUA0:[KITS] OPTIONS G DUB0:[KITS] -  
_ $ "/VERIFY/LOG/CONFIRM"
```

## 3.1.2 When the Installation Is Complete

When the installation is complete, VMSINSTAL.COM does one of the following, depending on the requirements of the product you have installed:

- Performs an automatic shutdown of the system and instructs you to reboot manually
- Returns you to the dollar-sign prompt (\$)

When the product is installed, back up the updated system disk.

### 3.1.3 Choosing VMSINSTAL Options

The VMSINSTAL.COM command procedure permits the use of five options:

- Auto-answer option (A)
- Get save set option (G)
- File log option (L)
- Release notes option (N)
- Alternate root option (R)

To specify each option, do the following:

- Type the appropriate option letter after the keyword `OPTIONS` in the command line that starts `VMSINSTAL.COM`.
- Press `RETURN`.

If you specify more than one option, do not separate the letters with commas or spaces. For example:

```
$ @VMSINSTAL.COM NEWAID021 MTA0: OPTIONS AN
```

The following sections describe the VMSINSTAL.COM options.

#### 3.1.3.1 Auto-Answer (A)

The auto-answer option makes it easier to *reinstall* a product by providing responses to VMSINSTAL.COM questions and prompts during the reinstallation. You use the auto-answer option most often to reinstall products after an upgrade.

If you specify the auto-answer option when you install a product, an answer file is created in the form `product.ANS` in the `SYS$UPDATE` directory, where *product* is the product-list parameter that you provide when you start VMSINSTAL.COM.

The file type of an answer file is `ANS`. The answer file contains a record of your responses to questions and prompts from VMSINSTAL.COM. For example, if you install the product, `NEWAID`, with the `AUTO-ANSWER` option, VMSINSTAL.COM creates an answer file designated `NEWAID.ANS`.

When you reinstall the product and specify the auto-answer option (typically after upgrading your operating system), VMSINSTAL.COM reads the answer file instead of asking you questions.

If you want to create a new answer file when you reinstall a product, you must delete the existing answer file first.

### 3.1.3.2 Get Save Set (G)

Installing products either from a distribution tape or from console media directly onto your system disk is time consuming. The GET SAVE SET option saves you time by allowing you to store product save sets temporarily on a magnetic tape or in a disk directory.

You might consider dedicating a user disk on a node that other licensed system users can access. You can store product save sets on this dedicated user disk to give other licensed system users fast access to the product save-set directory.

To store a product save set on a disk directory using the GET SAVE SET option, enter a command using following syntax:

```
@SYS$UPDATE:VMSINSTAL.COM product-list source OPTIONS G device:[directory]
```

**NOTE:** The directory that you specify must exist, and the device must be mounted; VMSINSTAL.COM does not perform any CREATE/DIRECTORY or MOUNT operations.

You specify the disk directory immediately after the OPTIONS G parameter. For example, if you are storing save sets for a product named NEWAID from the console drive into disk directory USER1:[PRODUCTS], enter the following command:

```
$ @SYS$UPDATE:VMSINSTAL.COM NEWAID CSA1: OPTIONS G USER1:[PRODUCTS]
```

VMSINSTAL.COM creates one or more files to store the product save set in the disk directory. The save set file name has the following format:

facvvu.x

fac	The product name code (1 to 36 alphanumeric characters).
vv	The major version number (2 digits).
u	The update number (1 digit).
x	A literal file type that is used to identify save set files, where A is the first save set, B the second, and so forth.

**NOTE:** Valid file types for save set files include the literal range A through Z and the numeric range VMI\_0027 through VMI\_9999.

For example, suppose you are storing update 1 to Version 2.0 of the product, NEWAID, and this process requires four save sets. VMSINSTAL.COM creates the following four files:

```
NEWAID021.A  
NEWAID021.B  
NEWAID021.C  
NEWAID021.D
```

When you want to install the product on your system, enter a command in the following format:

```
@SYS$UPDATE:VMSINSTAL.COM product-list device:[directory]
```

For the product NEWAID, enter this command:

```
$ @SYS$UPDATE:VMSINSTAL.COM NEWAID USER1:[PRODUCTS]
```

VMSINSTAL.COM installs the NEWAID product on your system disk.

### 3.1.3.3 File Log (L)

The FILE LOG option logs all file activity to the terminal during installation. File activity is defined as any action that alters the disposition of a file, such as creating a new file, updating a library, or deleting a file.

### 3.1.3.4 Release Notes (N)

Use the RELEASE NOTES option to display or print the online release notes file supplied by the layered product.

**NOTE:** Not all layered products provide online release notes.

The person who builds the product kit names the release notes file. The release notes file is given the file name *facvvu.release\_notes*, where *facvvu* represents the product name code, version, and update numbers.

If release notes are available and you specify option N, VMSINSTAL.COM asks you the following questions. (The default answers are indicated in brackets.)

Release Notes Options:

1. Display release notes
2. Print release notes
3. Both 1 and 2
4. Copy release notes to SYS\$HELP
5. Do not display, print, or copy release notes

\*Select option [2]:

\*Queue name [SYS\$PRINT]:

\*Do you want to continue the installation [N]:

- The first prompt (\*Select option:.) allows you to choose options 1 through 5.
- The second prompt (\*Queue name:.) is displayed only if you select option 2 or option 3. If you enter the name of a print queue, the system displays a message saying that the release notes have been queued successfully to the printer. If you do not specify a print queue, the release notes are sent to SYS\$PRINT by default.
- The third prompt (\*Do you want to continue the installation:.) allows you either to continue or to end the installation. The default is to end the installation.

## 3-10 Installing Software

If release notes are not supplied with the product, VMSINSTAL.COM displays two error messages. It also asks whether you want to continue or to end the installation, as follows:

```
%VMSINSTAL.COM-W-NOFILE, New File facvuu.RELEASE_NOTES does not exist.  
%VMSINSTAL.COM-W-NORELNNOTE, unable to locate release notes.
```

```
*Do you want to continue the installation [N]:
```

To continue the installation (whether or not release notes are available), type Y (for YES) and press RETURN.

### 3.1.3.5 Alternate Root (R)

The ALTERNATE ROOT option lets you install the product to a system root other than that of the running system. The VMS operating system in the alternate root must be complete and the same version or update level as the running system. All files and software products that the product installation refers to must be present in the alternate root.

**NOTE:** Not all optional software products allow a product to be installed to an alternate system root.

Consult the documentation of the specific product to determine whether the product can be installed to an alternate system root.

If you specify option R, the product is installed on the alternate root. However, you cannot create accounts or request a system reboot.

### 3.1.4 Recovering from a System Failure

If the system fails during installation of an update or optional software product, VMSINSTAL.COM attempts to continue the installation upon rebooting. Depending on when the system failed, one of three conditions exists:

- The system disk was not altered before the system failure. In this case, VMSINSTAL.COM instructs you to restart the installation.
- The system disk or a library used by the installation was corrupted. In this case, VMSINSTAL.COM instructs you to restore either the system disk or the corrupted library from the backup copy and to restart the installation.
- VMSINSTAL.COM continues the installation. In this case, VMSINSTAL.COM performs most of the installation. In addition, it might tell you that you must perform some tasks manually to complete the installation. If VMSINSTAL.COM instructs you to do so, do the following:



- a. Reboot the system
- b. Log in as system manager
- c. Purge all system files that have been replaced, even if you requested an automatic purge.

```
$ PURGE/LOG SYS$$SYSROOT:[*...]*.*
```

## 3.2 Summary

### Software Must Be Installed

Use the VMSINSTAL.COM procedure to install most software products on your system. You use this procedure both to install new products and to apply upgrades to products that have already been installed. By installing software, the software is made available every time that the system starts up.

For any software product that you install, be sure to read the installation instructions carefully.

### Using VMSINSTAL.COML

Before executing the VMSINSTAL.COM procedure, you must know the following:

- The location of the save sets for the software products that you will install (for example, a tape drive or a directory)
- The name of the product save set, as specified in the installation instructions for the product

Additionally, you can specify certain product options and a nondefault installation location for each product with the VMSINSTAL.COM procedure.

For some software installations, some system restrictions apply (for example, no interactive processes should be active except for the process installing the software). You should back up your system before installing a large software product, such as an upgrade to the VMS operating system. Also, some software installations might automatically shut down your system at the end of the installation procedure. Read the installation instructions for any software that you will install, and be sure that your system is properly prepared before you begin any installation.



# Chapter 4

## Managing Users

As a system manager, it is your job to create and maintain user accounts on the system. To create accounts for users and effectively manage the use of the system, you must determine which users need access to the system and what system resources they require.

Once you understand user needs, you can establish controls that customize the system appropriately.

The VMS operating system provides the Authorize Utility (AUTHORIZE) to authorize and control the use of system resources by individual users. This chapter describes the use of AUTHORIZE to do the following:

- Add a user account
- Modify a user account
- Remove a user account
- List the user accounts

See the Authorize Utility chapter in the Reference Section for more information about AUTHORIZE.

### 4.1 The User Authorization File (UAF)

You manage VMS users by creating and maintaining user accounts, which control who can log in to the system and how it can be used. Use the Authorize Utility (AUTHORIZE) to do the following:

- Create new records and modify existing records in the system user authorization file (SYS\$SYSTEM:SYSUAF.DAT) and the network user authorization file (SYS\$SYSTEM:NETPROXY.DAT)
- Create new records and modify existing records in the rights database file (SYS\$SYSTEM:RIGHTSLIST.DAT)

## 4-2 Managing Users

Whenever a user logs in, the system uses the information contained in the user authorization file (UAF) to validate the login attempt, establish the account's environment, and create a process with appropriate attributes. In this way, the system restricts users to the resources you assign to each account.

As system manager, you might want to create a private copy of SYSUAF.DAT in a directory other than SYS\$SYSTEM as an emergency backup for the system SYSUAF.DAT file. Note that, to have an effect on user processes, any private version of SYSUAF.DAT must be copied to the SYS\$SYSTEM directory and have the system user identification code (UIC).

Because certain images (such as MAIL and SET) require access to the system UAF and are normally installed with the SYSPRV privilege, make certain that you always grant system access to SYSUAF.DAT. The authorization files are created with the following default protection:

```
SYSUAF.DAT      S:RWED, O:RWED, G, W
NETPROXY.DAT   S:RWED, O:RWED, G:RWED, W
RIGHTSLIST.DAT S:RWED, O:RWED, G:RWE, W:R
```

If you need to maximize the protection for SYSUAF.DAT or NETPROXY.DAT, use the DCL-level SET PROTECTION command (note, however, that RIGHTSLIST.DAT must be world-readable). For example, to limit access to SYSUAF.DAT you could enter the following command:

```
$ SET PROTECTION=(S:RWED,O,G,W) SYS$SYSTEM:SYSUAF.DAT
```

Using AUTHORIZE, you create and maintain UAF records by assigning values to various **fields** within each record. The values you assign identify the user, define the user's work environment, and control use of system resources. Example 4-1 presents a typical UAF record for a nonprivileged user account.

To gain access to a specific user record, set the default directory to SYS\$SYSTEM, enter the command RUN AUTHORIZE to invoke AUTHORIZE, and enter the command SHOW *username* at the UAF> prompt. You can then enter AUTHORIZE commands such as ADD and MODIFY to create new user accounts or change the information in the fields of an existing UAF account.

**Example 4-1: Sample UAF Record Display**


---

```

$ SET DEFAULT SYSS$SYSTEM
$ RUN AUTHORIZE
UAF> SHOW WELCH

Username: WELCH                               Owner:  ROB WELCH
Account:  INVOICE                             UIC:    [21,51] ([INV,WELCH])
CLI:     DCL                                  Tables: DCLTABLES
Default:  USER3:[WELCH]
LGICMD:
Login Flags:
Primary days:  Mon Tue Wed Thu Fri
Secondary days:                               Sat Sun
No access restrictions
Expiration:      (none)      Pwdminimum:  6   Login Fails:    0
Pwdlifetime:    (none)      Pwdchange:  19-APR-1990 13:58
Last Login:     (none)      (interactive),      (none) (non-interactive)
Maxjobs:        0   Fillm:    20   Byt1m:    8192
Maxacctjobs:   0   Shrfillm:  0   Pbyt1m:    0
Maxdetach:     0   BIO1m:    10   JTquota:  1024
Prclm:         2   DIO1m:    10   WSdef:    150
Prio:          4   AST1m:    10   WSquo:    256
Queprio:       4   TQELm:    10   WSextent:  512
CPU:           (none) Enqlm:   100   Pgflquo:  10240
Authorized Privileges:
  TMPMBX NETMBX
Default Privileges:
  TMPMBX NETMBX

```

---

**4.1.1 System-Supplied UAF Records**

AUTHORIZE provides a set of commands and qualifiers to assign values to any field in a UAF record. The software distribution kit provided with a new VMS system contains a UAF of four records:

- **DEFAULT**—Serves as a template for creating user records in the UAF. A new user record is assigned the values of the DEFAULT record except where you explicitly override those values. Thus, whenever you add a new account, you need only specify values for fields that you want to be different.

For example, the following AUTHORIZE command creates a new record having the same values as the DEFAULT record, except that the password, UIC, and default directory fields are changed:

```

$ SET DEFAULT SYSS$SYSTEM
$ RUN AUTHORIZE
UAF> ADD MARCONI/PASSWORD=QLP6YT9A/UIC=[033,004] -
_UAF> /DIRECTORY=[MARCONI]

```

Section 4.2 gives an example of how to use AUTHORIZE to add a user account.

## 4-4 Managing Users

Examine the values for the DEFAULT account and determine if they are appropriate for your site. To examine the values for the DEFAULT account, enter the UAF command SHOW DEFAULT, as follows:

```
$ SET DEFAULT SYSS$SYSTEM
$ RUN AUTHORIZE
UAF> SHOW DEFAULT

Username: DEFAULT                               Owner:
Account:                                         UIC: [200,200] ([200,200])
CLI:      DCL                                    Tables: DCLTABLES
Default:  SYSS$SYSDEVICE:[USER]
LGICMD:   LOGIN
Login Flags:
Primary days:  Mon Tue Wed Thu Fri
Secondary days:                               Sat Sun
No access restrictions
Expiration:      (none)      Pwdminimum: 12      Login Fails:      2
Pwdlifetime:    30 00:00     Pwdchange:  19-APR-1990 15:58
Last Login:     (none) (interactive),      (none) (non-interactive)
Maxjobs:        0      Fillm:      20      Byt1m:      8192
Maxacctjobs:    0      Shrfillm:    0      Pbyt1m:      0
Maxdetach:      0      BIO1m:     18      JTquota:    1024
Prclm:          2      DIO1m:     18      WSdef:      150
Prio:           4      AST1m:     24      WSquo:     256
Queprio:        0      TQE1m:     10      WSextent:   512
CPU:            (none) Enq1m:     100     Pgflquo:    10240
Authorized Privileges:
  TMPMBX NETMBX
Default Privileges:
  TMPMBX NETMBX
```

If necessary, modify the DEFAULT account to meet the needs of your site. For example, to set the default minimum length of user passwords to 10 characters, enter the following command:

```
UAF> MODIFY DEFAULT/PWDMINIMUM=10
```

You cannot rename or delete the DEFAULT account in the user authorization file. See Section 4.4 for more information about modifying a user record in the user authorization file.

- **FIELD**—Permits Digital Field Service personnel to check out a new system. The FIELD record should be disabled once the system is installed.
- **SYSTEM**—Provides a means for you to log in with full privileges. The SYSTEM record can be modified but cannot be renamed or deleted from the UAF.

**CAUTION:** Do not change the SYSTEM account UAF record fields for the default device and directory, and privileges. Installation of VMS maintenance releases and optional software products depends on certain values in these fields.

- **SYSTEST**—Provides an appropriate environment for running the User Environment Test Package (UETP). The SYSTEST record should be disabled once the system is installed.

### 4.1.2 General Maintenance of the UAF

Usually, you use the UAF supplied with the distribution kit. (You can, however, rename the UAF with the DCL command `RENAME`, and then create a new UAF with `AUTHORIZE`.) You should limit any kind of access to this file to the `SYSTEM` account. Furthermore, each time you modify the file, create a backup copy so that in case of a system failure you do not lose the modifications. See Chapter 8 for procedures for backing up files.

The UAF is accessed as a shared file, and updates to the UAF are made on a per-record basis, which eliminates the need for both a temporary UAF and a new version of the UAF after each `AUTHORIZE` session. Updates become effective as soon as `AUTHORIZE` commands are entered. You should not enter temporary values with the intent of fixing them later in the session.

After installing the system, you should make the following modifications to the UAF:

- **SYSTEM, FIELD, and SYSTEST accounts**—If the passwords on these accounts are not secure or if they have not been changed recently, be sure to change the passwords. Use obscure passwords of ten characters or more and continue to change them on a regular basis. You should not permit general users access to these accounts.

In addition to changing the password, you can disable an account, especially if it is used infrequently. To disable an account, specify the following `AUTHORIZE` command:

```
UAF> MODIFY username /FLAGS=DISUSER
```

The login flag `DISUSER` disables the account and prevents anyone from logging into the account. To enable the account when it is needed, run `AUTHORIZE` and specify `MODIFY username /FLAGS=NODISUSER`. However, you should be cautious about disabling the `SYSTEM` account, because some optional software and some command procedures might not start up properly if the `SYSTEM` account is disabled.

**CAUTION:** Be careful not to disable all of your privileged system accounts. If you inadvertently do so, you can recover by setting the `UAFALTERNATE SYSGEN` parameter during a conversational bootstrap operation. See Chapter 2 for information about emergency startup procedures.

- **DEFAULT account**—You might want to change several fields in this account. For example:

```
UAF> MODIFY DEFAULT/DEVICE=DISK$USER/WSQUO=750
```

## 4-6 Managing Users

The default device is set to the name most commonly used for user accounts that will be added. Likewise, the working set value is set to a value appropriate for most users on the system.

Use the SYSTEM account only for system functions such as making backups and installing maintenance updates. The account comes to you with full privileges, so exercise caution in using it. For example, because you have BYPASS privilege, the system will allow you to delete any file no matter what its protection. If you type an incorrect name or spurious asterisk, you might destroy files that you or other users need to keep. For this reason, use another account with fewer privileges for day-to-day system management activities.

If you want to receive mail sent to the SYSTEM account, use the SET FORWARD command in MAIL to have any SYSTEM mail forwarded to any other account.

### 4.2 Adding a User Account

How you set up a user account depends on the needs of the individual user. In general, there are two types of accounts:

- **Interactive**—A person using an interactive account has access to the system software and can perform work of a general nature (program development, text editing, and so on). Usually, such an account is considered individual; that is, only one person can use it.
- **Captive**—A person using a captive account (also called a turnkey or application account) has access only to limited user software and can only perform work that is limited to a particular function. Access to a captive account is limited by function; that is, only those who perform a particular function can use it. For example, you might develop an inventory system. Anyone whose job entails inventory control can access your system, but that person cannot access other subsystems or the base software.

You should perform the following tasks in conjunction with adding a user account:

1. Determine a user name and password.
2. Determine a unique user identification code (UIC).
3. Decide where the account's files will reside (the device and directory).
4. Create a default directory on the appropriate volume, using CREATE/DIRECTORY command at DCL-level with the following syntax:  

```
CREATE/DIRECTORY directory-spec/OWNER_UIC= uic
```
5. Determine the security needs of the account (that is, the level of file protection, privileges, and access control).



After you analyze the purpose of a user account and decide which attributes and resources it requires, you can use `AUTHORIZE` to create the account. Give yourself the `SYSPRV` privilege. Then enter the following commands to set your default device and directory to that of `SYS$SYSTEM` and invoke `AUTHORIZE` as follows:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF>
```

When the utility responds with the `UAF>` prompt, use the `AUTHORIZE` command `ADD` to specify attributes in the `UAF` fields as shown in this example:

```
UAF> ADD JONES/PASSWORD=LPB57WM/UIC=[014,1] -
_UAF> /DEVICE=DISK$USER/DIRECTORY=[JONES] -
_UAF> /LGICMD=DISK$USER:[NEWPROD]GRPLOGIN -
_UAF> /OWNER="ROBERT JONES"/ACCOUNT=DOC
```

The `/OWNER` and `/ACCOUNT` entries are primarily for accounting purposes and can be omitted unless required by your site. The following unspecified qualifiers usually take their default values from the `DEFAULT` record:

- **Limits and Quotas** (`/ASTLM`, `/BIOLM`, `/CPUTIME`, `/DIOLM`, `/ENQLM`, `/FILLM`, `/JTQUOTA`, `/MAXACCTJOBS`, `/MAXDETACH`, `/MAXJOBS`, `/PGFLQUOTA`, `/PRCLM`, `/SHRFILLM`, `/TQELM`, `/WSDEFAULT`, `/WSEXTENT`, `/WSQUOTA`)—These qualifiers impose limits on the use of reusable system resources; the default values are adequate in most cases.
- **Priority** (`/PRIORITY`, `/QUEPRIORITY`)—The default values are usually adequate for accounts not running real-time processes.
- **Privileges** (`/DEFPRIVILEGES`, `/PRIVILEGES`)—The default privileges (`TMPMBX`, `NETMBX`) are usually adequate, depending on the purpose of the account.
- **Primary and Secondary Login Times; Login Functions** (`/ACCESS`, `/DIALUP`, `/FLAGS`, `/INTERACTIVE`, `/LOCAL`, `/PRIMEDAYS`, `/REMOTE`)—By default, users are allowed to log in at any hour of any day. To override the setting of a particular day, use the `DCL` command `SET DAY`. Use this command if a holiday occurs on a day that would normally be treated as a primary day and you want it treated as a secondary day.

The following example shows an `AUTHORIZE` command that adds a `UAF` record for a captive account:

```
UAF> ADD INVENTORY/PASSWORD=QRC7Y94A/UIC=[033,066] -
_UAF> /DEVICE=DISK$INVENT/DIRECTORY=[INV]/LGICMD=INVENTORY -
_UAF> /FLAGS=CAPTIVE/NOACCESS=(PRIMARY, 18-8, SECONDARY, 0-23)
```

In this example, the `/FLAGS` and `/NOACCESS` qualifiers restrict users from logging in to the captive account. The `/NOACCESS` qualifier limits logins to specific hours. The `/FLAGS=CAPTIVE` qualifier adds the login flag `CAPTIVE` to the captive account record. The `CAPTIVE` flag locks the person using the account into the application software by doing the following:

## 4-8 Managing Users

- Disabling the CTRL/Y function to prevent users from interrupting the execution of the command procedure and gaining access to the command interpreter
- Preventing the user from specifying an alternate command interpreter with the /CLI qualifier at login time
- Preventing the user from specifying an alternate default disk device with the /DISK qualifier at login time

The following examples summarize the steps for setting up an individual user account and a captive account:

### Setting Up an Individual User Account with AUTHORIZE

```
$ SET DEFAULT SYSS$SYSTEM
$
$ RUN AUTHORIZE
UAF>ADD JONES -                ! User name
_/PASSWORD=ROCKET -          ! Password
_/UIC=[014,1] -              ! UIC
_/ACCOUNT=DOC -              ! Accounting group name
_/OWNER="ROCKET JONES" -     ! Owner
_/DEVICE=$DISK1 -           ! Default directory
_/DIRECTORY=[JONES]
UAF>EXIT
$
$ ! Create top-level directory for individual
$ CREATE/DIRECTORY $DISK1:[JONES] -
_$ /OWNER_UIC=[DOC,JONES] -
_$ /PROTECTION=(S:RWE,O:RWE,G:RE,W:RE)
$
```

### Setting Up a Captive Account with AUTHORIZE

```
$ SET DEFAULT SYSS$SYSTEM
$
$ RUN AUTHORIZE
UAF>ADD INVENTORY -          ! User name
_/PASSWORD=RIZUPE -        ! Password
_/UIC=[033,066] -          ! UIC
_/ACCOUNT=INV -            ! Accounting group name
_/LGICMD=$DISK1:[INVTORY]LOGIN - ! Login file
_/FLAGS=(DEFCLI,DISCTLY, - ! Set flags
_/DISNEWMAIL,DISWELCOME,DISMAIL)
UAF>EXIT
```

## 4.3 Setting Up an Automatic Login Account

You use the automatic login facility (ALFMAINT) to set up a terminal that accepts automatic logins from authorized users. For example, a terminal might be set up for the account INVENTORY, which automatically logs a user into a captive account when INVENTORY is specified as the user name.

First, you must follow the steps described in the previous sections to create the top-level default directory and add the account. Once the account has been added, you set your default directory to SYS\$MANAGER and invoke the ALFMAINT.COM command procedure. ALFMAINT prompts you for the name of the terminal that you want associated with the user name of the automatic login account.

The following example summarizes the steps for setting up automatic logins for an individual user account and a captive account:

### Individual Account with Automatic Login

```

$ SET DEFAULT SYS$SYSTEM
$
$ RUN AUTHORIZE
UAF>ADD JONES -                ! Username
_/PASSWORD= -                 ! Null password
_/UIC=[014,1] -               ! UIC
-/ACCOUNT=DOC -               ! Accounting group name
-/OWNER="ROCKET JONES" -      ! Owner
_/DEVICE=$DISK1 -            ! Default directory
_/DIRECTORY=[JONES]
UAF>EXIT
$
$ ! Create top-level directory for individual
$ CREATE/DIRECTORY $DISK1:[JONES] -
_$/OWNER UIC=[DOC, JONES] -
_$/PROTECTION=(S:RWE,O:RWE,G:RE,W:RE)
$
$
$ SET DEFAULT SYS$MANAGER
$
$ @ALFMAINT
    
```

Enter the name of the terminal that you would like to set for automatic login, or a blank line or EXIT to exit.

```

Terminal (ddcu)? TTA1          ! Assigned terminal
Username? JONES
Terminal (ddcu)? EXIT
    
```

### Captive Account with Automatic Login

```

$ SET DEFAULT SYS$SYSTEM
$
$ RUN AUTHORIZE
UAF>ADD INVENTORY -           ! Username
_/PASSWORD= -                 ! Null password
_/UIC=[033,066] -            ! UIC
-/ACCOUNT=INV -               ! Accounting group name
_/LGICMD=$DISK1:[INVTORY]LOGIN - ! Login file
_/ACCESS=(PRIMARY,8-17) -     ! No off hours
_/FLAGS=CAPTIVE              ! All flags on
UAF>EXIT
$
$ SET DEFAULT SYS$MANAGER
$ @ALFMAINT
    
```

## 4-10 Managing Users

Enter the name of the terminal that you would like to set for automatic login, or a blank line or EXIT to exit.

```
Terminal (ddcu)? TTA0           ! All terminals
Username? INVENTORY            ! on automatic
Terminal (ddcu)? TTA1           ! login except
Username? INVENTORY            ! the console terminal
Terminal (ddcu)? TTA2           ! (The console terminal
Username? INVENTORY            ! for this system is TTA4:)
Terminal (ddcu)? TTA3
Username? INVENTORY
Terminal (ddcu)? EXIT
```

### 4.4 Modifying a User Account

To change a user account's quotas, default directory, password, authorized privileges, or any other characteristics assigned by AUTHORIZE, use the MODIFY command. You can use the MODIFY command to change any of the fields in an existing user account.

For example, when a user forgets a password and cannot log in, use the MODIFY command in AUTHORIZE to reset a user password. For example, the following command changes the password for user WELCH:

```
UAF> MODIFY WELCH/PASSWORD=newpassword
```

**NOTE:** Note that AUTHORIZE does not display user passwords.

Any changes that you make to a user's record will take effect *after* the user next logs in. For example, suppose that user JONES currently has an open file quota (FILLM) of 20. To increase user Jones' open file limit to 40, you would use the following command in AUTHORIZE:

```
UAF> MODIFY JONES /FILLM=40
```

Any process of user JONES that is logged in at the time that you modify the user authorization file continues to have a file limit of 20. In order to have an open file limit of 40, user JONES must log out and then log in again, after you have made the modification to the user authorization file using AUTHORIZE.

### 4.5 Listing User Accounts

Use the AUTHORIZE command LIST to create the file SYSUAF.LIS containing a summary of all user records in the UAF, as follows:

```
UAF> LIST
%UAF-I-LSTMSG1, writing listing file
%UAF-I-LSTMSG2, listing file SYSUAF.LIS complete
```

By default, the LIST command produces a brief report containing the following information from the UAF:

- Account owner
- User name
- UIC
- Account names
- Privileges
- Process priority
- Default disk and directory

Use the /FULL qualifier to create a full report of all the information (except user passwords) contained within the UAF, as follows:

```
UAF> LIST/FULL
%UAF-I-LSTMSG1, writing listing file
%UAF-I-LSTMSG2, listing file SYSUAF.LIS complete
```

## 4.6 Deleting a User Account

The main problem in deleting an account, especially an interactive account, is cleaning up the files used by the account. The following steps are suggested:

1. Copy (or have the outgoing user of the account copy) any files of value to the ownership of another account. Be sure to change the owner UIC of the files to match the owner UIC of the new owner. You can also use the Backup Utility (BACKUP) to copy the files to a backup tape or disk.
2. Change the password, and log in to the account that you want to delete. (By working from a nonprivileged account, you can avoid inadvertently deleting files that might be owned by an account other than the one that you want to delete.)
3. Delete the account's files and directories from the deepest level up to the top level using the following procedure:
  - a. Locate and examine all subdirectories using the DCL command DIRECTORY [default . . . ], where *default* is the name of the account's default directory.
  - b. Delete the files in each subdirectory and then delete the subdirectory. Note that directory files are protected against owner deletion; therefore, you must change the protection before deleting directory files.

## 4-12 Managing Users

- c. Delete the account's top-level directory. Example 4-2 illustrates a command procedure that deletes an account's files from the bottom level up.

**NOTE:** The command procedure in Example 4-2 should not be executed from a privileged account.

4. Remove the account, using AUTHORIZE.
5. Remove the user's disk quota entry from the disk quota file, if one existed, with the System Management Utility (SYSMAN).
6. Remove associated mail information by entering the MAIL command REMOVE *username*.

### Example 4-2: Command Procedure Template for Deleting an Account's Files

---

```
$ !      DELTREE.COM - deletes a complete directory tree
$ !
$ !      P1 = pathname of root of tree to delete
$ !
$ !      All files and directories in the tree, including
$ !      the named root, are deleted.
$ !
$ IF "'DELTREE'" .EQS. "" THEN DELTREE = "@SYS$LIBRARY:DELTREE"
$ ON CONTROL_Y THEN GOTO DONE
$ ON WARNING THEN GOTO DONE
$ DEFAULT = F$LOGICAL("SYS$DISK") + F$DIRECTORY()
$10:
$ IF P1 .NES. "" THEN GOTO 20
$ INQUIRE P1 "Root"
$ GOTO 10
$20:
$ IF F$PARSE(P1) .EQS. "" THEN OPEN FILE 'P1'
$ SET DEFAULT 'P1'
$LOOP:
$ FILESPEC = F$SEARCH("*.DIR;1")
$ IF FILESPEC .EQS. "" THEN GOTO LOOPEND
$ DELTREE [.'F$PARSE(FILESPEC,,, "NAME")']
$ GOTO LOOP
$LOOPEND:
$ IF F$SEARCH("*.*,*") .NES. "" THEN DELETE *.*;*
$ DIR = (F$DIRECTORY()-]"-"])-F$PARSE("[-]","",-
      "DIRECTORY")-]"-"])-".-]"-]<"
$ SET PROTECTION=WORLD:RWED [-]'DIR'.DIR;1
$ DELETE [-]'DIR'.DIR;1
$DONE:
$ SET DEFAULT 'DEFAULT'
```

---

If you never assign the same UIC to multiple users, you can use Backup to remove the user's files, even if the files are scattered throughout the directory structure. The following is an example of a BACKUP command used to remove files:

```
$ BACKUP/DELETE PUBLIC:[...]/OWNER=[21,103] MTA0:PUBLIC_UIC.SAV
```

This BACKUP command copies and deletes only those files owned by the specified UIC on disk PUBLIC. The files are copied into a save set named PUBLIC\_UIC on device MTA0. Note that the BACKUP/DELETE command does not delete the directory files (file extension DIR) for the account.

### Disabling a User Account

If you want to disable an account without deleting it, set the disable user flag (/FLAGS=DISUSER) using AUTHORIZE. If the user is logged in, the account is disabled only after the user logs out.

Disabling a powerful yet infrequently used account provides an extra security measure by eliminating the risk of guessed or stolen passwords.

## 4.7 Summary

### The Authorize Utility

To give a user access to the system, use the Authorize Utility (AUTHORIZE). AUTHORIZE creates and modifies records in the system user authorization file (SYSUAF.DAT). To use AUTHORIZE, set your default to SYS\$SYSTEM and then give the RUN AUTHORIZE command.

### Creating a User Account

Create a new user account with the ADD command in AUTHORIZE. When you create a user account, you should generally include at least the /PASSWORD, /UIC, /DEVICE, and /DIRECTORY qualifiers. These qualifiers do the following:

- /DIRECTORY—Defines the user's password.
- /UIC—Assigns a UIC to the user. Each user on your system must have a unique UIC.
- /DEVICE and /DIRECTORY—Assigns the device and directory where the user's files will reside on your system. In addition to defining a user's device and directory with AUTHORIZE, you must also create the directory for the user, using the DCL CREATE/DIRECTORY command.

Each user account has a set of characteristics that are associated with the account, such as the user name, various quotas, and authorized privileges. The DEFAULT account in SYSUAF.DAT contains the characteristics that are assigned by default to accounts that you create.

### **Modifying a User Account**

Change the characteristics of a user's account with the **MODIFY** command in **AUTHORIZE**. You can change the default characteristics for any new account that you create by modifying the **DEFAULT** account. When you change the characteristics of a user's account (for example, modifying a user's authorized privileges), the changes will apply only to processes created *after* you make the modifications with **AUTHORIZE**.



# Chapter 5

## Performing Batch and Print Operations

If you have a printer on your system, or if you want to use batch processing on your system, then you must use **queues**. A queue allows users to submit requests for printing or batch processing, and the system prints or processes the users' jobs as resources allow.

The system manager is responsible for setting up batch and print queues and making sure that they function properly. This chapter describes how to set up (initialize) and maintain batch and print queues for your system or cluster.

Setting up and maintaining batch and print queues are closely related system management tasks. However, you are not required to set up both types of queues if you need only one type. In a VAXcluster environment, queues can be accessed from any node on the cluster.

### 5.1 Generic Queues and Execution Queues

In the VMS operating system, batch and print operations support two types of queues: generic queues and execution queues.

An **execution queue** is a queue through which the job (either print or batch) is actually processed or executed. For printing, an execution queue is associated with a specific printer; for batch processing in a VAXcluster environment, an execution queue is associated with a specific node. When a print or batch job is submitted to an execution queue, the job is ultimately printed on the output device associated with that queue or processed on the associated batch queue.

You can also designate one or more individual terminals as execution queues for print jobs. You should set up a terminal as a queue when you want to allow users on your system to send output to a hardcopy terminal.

A **generic queue** is an intermediate queue that holds a job until an appropriate execution queue becomes available to initiate the job. Users can submit jobs to a generic queue, and the generic queue then directs the job to an appropriate execution queue; alternatively, users can submit jobs directly to an execution queue.

## 5-2 Performing Batch and Print Operations

For example, suppose you have a system with several printers. You would set up individual execution queues for each of the printers, and you could also set up a generic print queue. Users would then normally submit a print job to the generic queue, and the generic queue would subsequently direct the print job to an available printer.

For batch processing, generic queues are often used in clustered systems to distribute the workload across several nodes. For example, suppose you have a Local Area VAXcluster environment with each of the satellite nodes having a batch processing queue. You could then establish a generic batch queue for the cluster. When users submit batch jobs to the generic queue, the generic queue would direct individual batch jobs to the execution queue that is best able to handle the workload.

If you have only a single printer for your system or cluster, or if you establish only a single batch queue, then there is no value in establishing generic queues.

### 5.2 Setting Up Queues

Set up your queues by including the appropriate commands in your site-specific startup file, `SYS$MANAGER:SYSTARTUP_V5.COM`. Section 2.4.3 describes the commands for setting up queues that should be included in `SYSTARTUP_V5.COM`, and this section summarizes them.

To establish and use queues, you must first start the queue manager and identify a **queue file**. To do this, include the following command in `SYS$MANAGER:SYSTARTUP_V5.COM`, making sure that this command appears before any other queue commands:

```
$ START/QUEUE/MANAGER/RESTART SYS$COMMON:[SYSEXE]JBCSYSQUE.DAT
```

If you have a cluster, you should use only one queue file for the cluster. Make sure that the queue file is on a disk that is accessible to all the nodes in the cluster from which you might want to submit batch or print jobs.

When you create a generic queue, you specify a list of execution queues to which the generic queue ultimately directs jobs. In a VAXcluster environment, execution queues that you specify for a generic queue can be on the same node as the generic queue, and they can also be on different nodes within the cluster.

After you establish the queue file, you can set up individual execution queues and generic queues by using the `INITIALIZE/QUEUE` command in your `SYSTARTUP_V5.COM` file. Be sure to initialize execution queues before initializing the generic queues. For example, you could include the following series of commands to set up execution and generic queues for batch and print operations in a VAXcluster environment:

```

$ INITIALIZE /QUEUE /ON=BLUE::LPA0 /START BLUE_LPA0 ❶
$ INITIALIZE /QUEUE /ON=GREEN::LPA0 /START GREEN_LPA0 ❷
$ INITIALIZE /QUEUE /GENERIC=(BLUE_LPA0, GREEN_LPA0) /START SYS$PRINT ❸
$
$ INITIALIZE /QUEUE /BATCH /ON=BLUE:: /START BLUE_BATCH ❹
$ INITIALIZE /QUEUE /BATCH /ON=RED:: /START RED_BATCH ❺
$ INITIALIZE /QUEUE /BATCH /GENERIC=(BLUE_BATCH, RED_BATCH) /START SYS$BATCH ❻

```

This series of commands in SYSTARTUP\_V5.COM does the following:

- ❶ Sets up an execution printer queue associated with LPA0 on node BLUE with a queue name of BLUE\_LPA0.
- ❷ Sets up an execution printer queue associated with LPA0 on node GREEN with a queue name of GREEN\_LPA0.
- ❸ Sets up a generic print queue for the cluster. The generic print queue has the name SYS\$PRINT and directs print jobs either to BLUE\_LPA0 or to GREEN\_LPA0.
- ❹ Sets up an execution batch queue on node BLUE with the queue name BLUE\_BATCH.
- ❺ Sets up an execution batch queue on node RED with the queue name RED\_BATCH.
- ❻ Sets up a generic batch queue for the cluster. The generic batch queue has the name SYS\$BATCH. When a job is submitted to SYS\$BATCH, this generic queue directs the job either to BLUE\_BATCH or to RED\_BATCH.

If you want to set up a terminal as an execution queue, use exactly the same procedure as for setting up print queues and use the /DEVICE=TERMINAL qualifier in your INITIALIZE command line.

### 5.3 Maintaining Batch and Print Queues

Once you have modified SYS\$MANAGER:SYSTARTUP\_V5.COM to establish your queues properly, they will be set up and available every time that your system is booted. From time to time, however, some additional maintenance of your queues might be needed.

The VMS operating system provides several DCL-level commands that you can use to manage your queues. Table 5-1 shows some of the commands that are available for queue management. More information about these commands is available in the *VMS User's Manual*.

**Table 5-1: Queue Management Commands**

<b>Command</b>	<b>Description</b>
SET QUEUE	Changes the attributes of a queue (for example, the number of jobs that can execute simultaneously in a batch queue) without having to stop the queue, initialize it, and then restart it.
SHOW QUEUE	Provides the status of queues, listing the jobs that are currently executing, as well as the jobs that have not yet begun execution.
STOP /QUEUE	Allows you to pause a queue temporarily. Using the appropriate qualifiers, you can use the STOP/QUEUE command to stop jobs that are currently executing, to stop a queue after the completion of all jobs that are currently executing, to shut down the queue manager on the node from which you execute the command, and so on.
START /QUEUE	Resumes execution of a queue that has been temporarily halted by the STOP /QUEUE command.
STOP /QUEUE /MANAGER	Shuts down the queue manager on the node from which you execute the command.
START /QUEUE /MANAGER	Starts the queue manager on the node from which you execute the command.

## 5.4 Monitoring Jobs

As system manager, you use the SHOW QUEUE command to monitor the overall status of a queue. The SHOW QUEUE command displays the status of each queue selected, and it also shows the status of all jobs in each of the selected queues. With SHOW QUEUE, you can also obtain a summary of the status of jobs in each queue by using the /SUMMARY qualifier.

You can also use the SHOW ENTRY command to monitor the status of jobs belonging to a particular user, or to determine the status of individual batch and print jobs. SHOW ENTRY and SHOW QUEUE each provide complete information about jobs, but SHOW QUEUE also provides status information about the queues themselves. SHOW ENTRY generally provides a faster response time than SHOW QUEUE. For a full description of the SHOW ENTRY and SHOW QUEUE commands, see the Reference Section of the *VMS User's Manual*.

The following list describes the types of job status returned by the SHOW QUEUE and SHOW ENTRY commands:

Status	Description
Aborting	Executing job is terminating
Executing	Job is executing from a batch queue
Holding	Job is being held until explicitly released
Holding until	Job is being held until a specified time
Pending	Job is in a waiting state
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
Retained on Error	Job remains in the queue upon encountering an error
Waiting	Symbiont refuses the job

### 5.4.1 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 program that has entered an endless loop or a job that is executing on a faulty print device.

Follow this procedure to delete a job:

1. Determine the entry number of the job
2. Delete the job by entering the DELETE/ENTRY command

The DELETE/ENTRY command is restricted to users with either OPER privilege, execute access (E) to the queue, or delete access (D) to the specified job.

For example, assume that you observe a job that appears to be processing in an endless loop and is using an inappropriate amount of system resources. You can delete the job by entering the command DELETE/ENTRY=entry\_number. To determine the entry number, enter the command SHOW QUEUE/BATCH/ALL/BY\_JOB\_STATUS=EXECUTING. The following example shows how to determine the entry number and delete the job:

```
$ SHOW QUEUE/BATCH/ALL/BY_JOB_STATUS=EXECUTING
```

```
Batch queue ZEUS_BATCH, on ZEUS
```

Jobname	Username	Entry	Status
-----	-----	-----	-----
2307SMRCL	MARCO	1719	Executing
TEST	JONES	1720	Executing

```
$ DELETE/ENTRY=1719
```

## 5.4.2 Retaining Jobs in a Queue

To retain a job in a queue after it has been processed, specify the `/RETAIN` qualifier with the `INITIALIZE/QUEUE`, `START/QUEUE`, or `SET QUEUE` command. The `/RETAIN` qualifier has the following format:

```
/[NO]RETAIN[=option]
```

By using the `ERROR` option with the `/RETAIN` qualifier, you indicate that the jobs in a queue will be held if they do not successfully complete. A job that has been held by the `/RETAIN=ERROR` qualifier can later be released or requeued after the problem that caused the error has been resolved.

By default, jobs are *not* retained.

For example, to retain print jobs that do not successfully complete, you could include the following command in your `SYSTARTUP_V5.COM` file, after you have initialized and started the queue:

```
$ SET QUEUE /RETAIN=ERROR GREEN_LPA0
```

## 5.4.3 Modifying Job Processing Attributes

You can modify certain job processing attributes by specifying qualifiers with the command `SET ENTRY` entry-number, as shown in the following table:

Qualifier	Description
<code>/[NO]AFTER</code>	Controls whether a job is held until a specified time
<code>/[NO]HOLD</code>	Controls whether a job is available for immediate processing or held until it is released for processing
<code>/NAME</code>	Specifies a new name for a job
<code>/[NO]PASSALL</code>	Specifies whether the symbiont bypasses all formatting and sends the output directly to the device driver
<code>/PRIORITY</code>	Specifies the relative scheduling priority of the job, with respect to the priorities of other jobs in the queue
<code>/RELEASE</code>	Releases a previously held job
<code>/REQUEUE</code>	Requests that the job be moved from the original queue to the specified queue; this qualifier can also be used with the <code>STOP/QUEUE/ENTRY</code> command

### 5.4.3.1 Holding and Releasing a Job

The `/HOLD` qualifier of the `SET ENTRY` command controls whether a job is to be made available for immediate processing. To release a held job, use either the `/NOHOLD` qualifier or the `/RELEASE` qualifier.

To request that the job be held until after a specified time, use the /AFTER qualifier with the command SET ENTRY. The job is queued for immediate processing when the specified time arrives. The /AFTER=time qualifier accepts either absolute or delta time values in the format [dd-mmm-yyyy] [hh:mm:ss.cc]. You can also specify the following keywords:

TODAY  
 YESTERDAY  
 TOMORROW

The following command holds a print job until it is queued for processing at the specified date and time:

```
$ SET ENTRY 1121/AFTER=19-APR-1990:17:30
```

You can use the /NOAFTER qualifier to release immediately a job that is being held until a specified time.

The /RELEASE qualifier releases a job that is being held for any of the following reasons:

- A job was submitted with the /HOLD qualifier
- A completed job was held in a queue with the /RETAIN qualifier
- A job was submitted with the /AFTER qualifier

Use the SET ENTRY command with the /HOLD and /RELEASE qualifiers to hold and release a batch job. This procedure applies only to a batch job that is currently in a pending state (that is, a job that has not yet begun to execute). The following example shows how to hold and release a batch job that has not begun execution:

```
$ SET ENTRY 1234/HOLD
$ SET ENTRY 1234/RELEASE
```

#### 5.4.3.2 Requeuing a Job

To requeue a job that has not begun execution, use the SET ENTRY/REQUEUE COMMAND. If you want to requeue a job that has already begun execution, use the STOP/QUEUE/REQUEUE command. The STOP/QUEUE/REQUEUE command suspends the currently executing job and requeues it to the specified queue, for example:

```
$ STOP/QUEUE/REQUEUE=ALPHA_LPA0 ALPHA_LPB0
```

This command stops the executing print job on ALPHA\_LPB0 and requeues it to ALPHA\_LPA0. The queue does not stop; only the currently executing job is affected. Other jobs remain pending in the queue until they are processed.

You can hold an aborted print job, using the /HOLD qualifier. Enter the STOP/QUEUE/REQUEUE/HOLD command in the following format:

```
STOP/QUEUE/REQUEUE/HOLD [queue-name]
```

## 5-8 Performing Batch and Print Operations

When you specify `/HOLD`, the aborted job is placed in a hold state for later release with the `SET ENTRY/RELEASE` command. If you do not need to process a job that is being held in a queue, you can delete the job with the `DELETE/ENTRY` command.

**NOTE:** 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
```

### 5.4.3.3 Changing the Scheduling Priority of a Job

You can change the scheduling priority of a job by using the `/PRIORITY=n` qualifier with the `SET ENTRY` command. Do not confuse the job scheduling priority with the base priority of 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). You must have either `OPER` or `ALTPRI` privilege to raise the priority value above the value of the `SYSGEN` parameter `MAXQUEPRI`. No privilege is needed to set the priority of your own job lower than the `MAXQUEPRI` value. The following example changes the priority of a job to 50:

```
$ SET ENTRY 1131/PRIORITY=50
```

## 5.5 Summary

### Types of Queues

A batch queue lets you use batch processing, and a print queue lets you use system printers. You must establish one or more batch queues if you want to allow batch processing on your system, and you must also establish one or more print queues if you want to make printers available to system users.

*Execution queues* process jobs; an execution queue is associated with a specific printer or batch processing queue. *Generic queues* are intermediate queues, passing jobs on to execution queues based on the availability of resources.

### Starting Queues and the Queue Manager

You establish, or *initialize*, queues in the site-specific startup file, `SY$MANAGER:SYSTARTUP_V5.COM`. Before initializing any queues, you must first start the queue manager and identify a queue file using the `START/QUEUE/MANAGER` command.

After you start the queue manager, you can establish queues with the `INITIALIZE/QUEUE` command. If you establish both execution and generic queues, initialize the execution queues first, and then initialize the generic queues.



### **Monitoring and Modifying Queues**

After queues are operating, you can monitor the queues and modify their operation using the `SHOW QUEUE`, `SET QUEUE`, `START/QUEUE`, and `STOP/QUEUE` commands. You can manipulate individual jobs in queues with the `SET ENTRY` command, and you can delete a job in a queue by using the `DELETE/ENTRY` command.



# Chapter 6

## Setting Up and Maintaining a Network

A **network** includes two or more computers that are connected so that they can communicate with each other, and the hardware and software that makes those connections. Networking software is also always required for certain applications, such as DECwindows.

Your system must be connected a network when it meets any one (or more) of the following conditions:

- Your computer is part of a VAXcluster configuration
- Your computer communicates with other computers (for example, to share data, to run applications, or to send mail)
- Your computer is a workstation using DECwindows

You should read this chapter to become familiar with the network-related tasks that a system manager might be required to undertake. You should also read this chapter to obtain an overview of networks on a VMS system.

### 6.1 Getting Started with Networks

A network allows communication between computers; the computers can be halfway across a room from each other, or halfway around the world. Digital computers with the VMS operating system are connected to a computer network using the **DECnet-VAX** software.

To use DECnet-VAX software on your system, use the following procedure (described in detail in the sections that follow):

1. Prepare your system for the DECnet-VAX environment. Your hardware should be connected and in place, and you should understand how you want to configure your VMS system for network operations. Section 6.2.1 describes this process.
2. Install the DECnet-VAX software (see Section 6.2.2).

## 6-2 Setting Up and Maintaining a Network

3. Configure your networking environment with the NETCONFIG command procedure (see Section 6.2.3), and, if appropriate, establish asynchronous connections to other systems.
4. Start the DECnet-VAX software (see Section 6.2.3.3).

This manual discusses only the primary software element of a network, DECnet-VAX. For information about any hardware that is associated with your network, refer to the documentation for the hardware, or contact your Digital representative.

If you are manager of a workstation that is using DECwindows (or some other windowing product that requires DECnet-VAX), you must install DECnet-VAX, even if the workstation will not be part of any computer network. However, you can skip several of the steps that are required for other DECnet-VAX installations. Section 6.2.3.4 describes the considerations for DECnet-VAX with this type of configuration.

### Networks and System Management

A system manager is usually responsible for setting up and maintaining DECnet-VAX on individual VMS systems. The tasks can be grouped as follows:

- Starting up DECnet on your system for the first time
- Starting up DECnet each time that your system is booted
- Maintaining and monitoring the network while your system is running

## 6.2 Joining a Network

This section describes how to connect your system to an existing network. If there is no existing network to which you can connect, you can create a new network when a system is booted as a network node. A **node** is a computer that is connected to a DECnet network.

In order to join the network and communicate with other systems, your system must have communications lines. A **communications line** connects your computer to the DECnet network. A communications line can be a high-speed line such as an Ethernet cable or a synchronous point-to-point line, or it can be a low-speed, low-cost asynchronous line.

An asynchronous point-to-point connection can be established over any VMS terminal line between a VMS system and another system (which can be a non-VMS system) that supports the DECnet asynchronous Digital Data Communications Message Protocol (DDCMP). Alternatively, an asynchronous connection can be made over a dialup line (for example, a telephone line) if a modem is used at each end of the connection. A modem is a device that connects the terminal line to the telephone line.

## 6.2.1 Preparing Your VMS System for the Network Environment

Before you bring up DECnet-VAX on your system, you should take the following steps to prepare your system to function as part of the network:

- Be sure your process has the privileges needed to perform network operations. The minimum privileges that a system manager normally requires to configure and control the network and run network programs are SYSPRV, OPER, TMPMBX, NETMBX, and BYPASS.
- Decide how you want to allow access for users on remote systems. (A remote system is one that is connected to your DECnet network.) Users on remote systems need access to your system when they want to send mail to users on your system, copy or read files that reside on your system, or otherwise make a connection to your system.

To allow access to remote users, you can either:

- Establish a default DECnet account that allows remote users to access certain VMS utilities (such as MAIL or PHONE) for the purpose of communicating with users on your system. The default DECnet account also allows remote users to have access to the DECnet directory, and to any files on your system that allow world (W) access.
- Establish individual default accounts for the network objects (MAIL, FAL, PHONE, NML, MIRROR, and VPM).

If network security is an important consideration at your site, you should consider the second of these alternatives.

- Set up any **proxy accounts** that you want to establish for your node. A proxy account allows a specific user (or group of users) on a remote node to access data with the same privileges as if the user had logged in to your local system.
- If necessary, tune your VMS system to accommodate DECnet-VAX software. The network manager who establishes network configuration guidelines should provide you with any required information if you need to update VMS system parameters and quotas.
- You must establish the node name and network address for your system before you can connect your system to a DECnet network. Each node in the network is identified by a unique name and numeric address by which the node is known to other nodes in the network.

Obtain the node name and node address for your system from your site's network or cluster manager. (The VAXcluster node name must be set in the VMS system parameter SCSNODE and the node address in SCSSYSTEMID.)

## 6-4 Setting Up and Maintaining a Network

If your site does not have a network or cluster manager, then you must choose a node name and address for your system. The node name can be no more than six alphanumeric characters and must include at least one alphabetic character. The node address consists of an area number (in the range from 1 to 63, with a default value of 1) and a node number (in the range from 1 to 1023) separated by a period (for example, 2.2).

If your node is a member of a VAXcluster that uses an alias node identifier (an alias name and address), you can obtain the alias identifier from the VAXcluster manager. An alias node identifier, common to some or all nodes in a cluster, permits remote nodes to treat the cluster as though it were a single node. Individual nodes in a VAXcluster can optionally assume the alias, while retaining their individual node names. You can use the alias adopted by the cluster, as well as your own node name, to communicate with other nodes in the network.

- Obtain the node names and addresses of all other nodes in your network to which you want to connect. You will need this information to communicate with other systems. Your network manager should have this information, or, if your site has no network manager, you can obtain the information from the system managers of the other nodes in your network.
- Determine whether your system is to be a router or an end node. If you have a DECnet full function license and the accompanying DECnet-VAX Product Authorization Key (PAK), you have the option of configuring your system as either a router or an end node. If your DECnet license and PAK are for the end node capability, you can configure your system only as an end node.
- Determine the types of connections that will be made to the network: Ethernet, synchronous DDCMP, or asynchronous DDCMP connections. You can use the network configuration procedure NETCONFIG.COM to configure all circuits and lines automatically except for asynchronous circuits and lines.

### 6.2.2 Using DECnet-VAX on Your System

This section describes the procedure for using DECnet-VAX on your VMS operating system. Use this procedure to bring up your system as a node on an existing DECnet network.

To perform the installation procedure, log in to the SYSTEM account on your node and complete the following steps:

1. Purchase the DECnet-VAX license and the DECnet-VAX PAK and register the PAK on your system, using the VMS License Management Utility.
2. Configure your DECnet-VAX node and define the remote node names. You can configure your node and turn on the network at your node either automatically or manually.

3. If you plan to use an asynchronous DECnet connection, perform any steps needed to establish the connection.
4. Verify that your node is connected to the network.

### Getting a DECnet-VAX License and PAK

To permit your node to communicate with other nodes in the network, you must have a DECnet-VAX license and register a DECnet-VAX PAK on your system using the VMS License Management Utility. You can purchase either an end node or a full function license and the corresponding PAK. The end node PAK permits you to configure your node only as an end node. The full function PAK permits you to configure your node as either a routing node or an end node. You can also use the full function PAK to upgrade from end node to full function capability.

You can register the DECnet-VAX PAK as the initial step in bringing up the network, or you can register it after performing the automatic configuration of DECnet-VAX (using NETCONFIG.COM), as described in the following section. Be sure to determine whether the PAK you are registering is for the full function or end node DECnet capability. The full function DECnet-VAX PAK is DVNETRTG; the end node DECnet-VAX PAK is DVNETEND.

## 6.2.3 Configuring the Network Environment

You are now ready to configure your DECnet-VAX system. You can configure the node manually, or you can configure it automatically using the command procedure SYS\$MANAGER:NETCONFIG.COM. Use the automatic configuration procedure when you first join the network or when you reconfigure your node completely. Use the manual procedure if you want to modify an existing configuration.

**NOTE:** If you have an existing configuration and want to modify only default network access in order to provide the security enhancements that are available with default accounts for individual objects (such as MAIL), you can use the procedure SYS\$UPDATE:NETCONFIG\_UPDATE.COM to automatically reconfigure your node for this purpose. (See Section 6.2.3.1 for more information about this procedure.)

The system manager at each node in the network is responsible for the DECnet-VAX configuration database for the node. The database includes files that describe the local (executor) node and the other nodes in the network with which the local node can communicate, as well as the circuits and lines that connect the local node to the network. The network database also includes information on the logging collection points (such as the logging monitor) to which network events are reported. In addition, DECnet-VAX provides object databases describing objects (such as MAIL) known to the network. Each node in the network has such a database.

## 6-6 Setting Up and Maintaining a Network

The configuration database comprises the *volatile database* (the working copy of the database that reflects current network conditions) and the *permanent database* (which provides the initial values for the volatile database when you start the network). Modifications to the volatile database exist only while the network is running. Changes made to the permanent database remain after the network is shut down, but they do not affect the current system.

As system manager, you provide network component information, from the point of view of the local node, in the configuration database at the local node. Use the Network Control Program (NCP) to supply this information in the form of parameter values, which determine how the various components of the network function together. Use NCP DEFINE commands to establish the contents of the permanent database and SET commands to specify the contents of the volatile database. Use PURGE commands to delete permanent database entries and CLEAR commands to delete or reset volatile database entries.

### Configuring Your Node Manually

You can always configure your node manually; however, you have the option of doing it automatically (as described in the next section) if you are configuring a new node or completely reconfiguring a node.

If you decide to configure your node manually, you must enter NCP commands to establish the permanent configuration database and then turn on the network manually, causing the contents of the permanent database to be entered in the volatile database. A brief explanation of how to use NCP to establish your configuration database manually appears later in this section.

If you decide to configure your node manually, you can optionally create default accounts for network objects.

### Configuring Your Node Automatically

You can use the interactive command procedure NETCONFIG.COM to configure your system automatically. NETCONFIG.COM configures all required permanent database entries except for remote nodes, asynchronous circuits, and lines. You can also use the command procedure to set up DECnet accounts on your system.

Use NETCONFIG.COM only if you are bringing up your node for the first time, or if you want to reconfigure your node completely. The procedure purges any existing permanent database entries on your system (except for remote node entries). You need OPER and SYSPRV privileges to use NETCONFIG.COM to configure your node.

Before you configure your node, you need the following information:

- Your node name and address (available from your network manager).



- The means by which you will allow network access. The NETCONFIG.COM procedure provides two means of access—default accounts for network objects (such as MAIL or PHONE), or a default DECnet account that is used for all nonprivileged network access. By using default accounts for individual network objects (rather than a single DECnet default account), you can increase network security for your system.

If you do not create a default DECnet account, you must create a default account for each of the named network objects (MAIL, FAL, PHONE, NML, MIRROR, AND VPM) that you want to use. A brief description of these network objects follows:

- **MIRROR**—Used for loopback testing. If you do not create a default DECnet account and you want to test DECnet with the VAX UETP (User Environmental Test Package), use the NETCONFIG.COM procedure to create a default account for the MIRROR object.
- **VPM (VMS Performance Monitor Utility)**—Used by the MONITOR utility in VAXcluster configurations. If you do not create a default DECnet account and you want to use the MONITOR CLUSTER command to obtain performance information about VAXcluster members, use the NETCONFIG.COM procedure to create a default account for the VPM object.
- **MAIL**—Allows users on your system to receive mail from users on other nodes. If you do not create a default DECnet account and you want to allow mail from other nodes to be received on your system, use the NETCONFIG.COM procedure to create a default account for the MAIL object.
- **PHONE**—Allows users to communicate interactively. If you do not create a default DECnet account and you want to allow use of PHONE between users on your system and users on remote systems, use the NETCONFIG.COM procedure to create a default account for the PHONE object.
- **FAL (File Access Listener)**—A network image that receives and processes requests from remote nodes for file access on your local node. If enabled by the default DECnet account or a separate default account, the FAL object can make a system vulnerable to certain types of unauthorized access. Digital suggests that you consider the security implications for your site before creating a default account for FAL.
- **NML (Network Management Listener)**—Performs local control and information functions requested by remote nodes.

The second, less restrictive form of default access is to create a default DECnet account but to disable default access to type 0 objects (also known as TASK objects). In general, DECnet access to TASK should be denied because it allows arbitrary command files—including those which that be

## 6-8 Setting Up and Maintaining a Network

used in attempted breakins—to be executed on your system. Default access for system objects is still enabled.

You can still create an unrestricted default DECnet account that includes default access to TASK objects; this type of access is suitable for small systems with very low security requirements. To do so, you must override the defaults provided by NETCONFIG.COM.

### Using the NETCONFIG.COM Command Procedure

To use the NETCONFIG.COM command procedure to configure your node automatically, perform the following steps. Default values appear in brackets [] after certain questions in the interactive dialogue. To accept a default, press RETURN.

1. **Log in to the SYSTEM account on your node.**
2. **Invoke NETCONFIG.COM.** Enter the following command at the dollar sign (\$) prompt:

```
$ @SYS$MANAGER:NETCONFIG
```

The following message is displayed:

```
DECnet--VAX network configuration procedure
```

This procedure will help you define the parameters needed to get DECnet running on this machine. You will be shown the changes before they are executed, in case you want to perform them manually.

3. **Provide the node name.** You will be prompted as follows:

```
What do you want your DECnet node name to be?
```

Enter the node name you have selected (or have been assigned by the network manager). Your node name must have from one to six alphanumeric characters, it must include at least one alphabetic character, and it must be unique among all node names in the network.

(If you are on a VAXcluster node, press RETURN to accept the default node name that appears in brackets at the end of the prompt. This default node name is based on the SYSGEN parameter SCSNODE. If no default node name is displayed, exit the procedure and use SYSGEN to set up a value for SCSNODE, then restart the procedure. The DECnet node name of a VAXcluster node must match the value of SCSNODE.)

4. **Provide the node address.** You will be prompted as follows:

```
What do you want your DECnet address to be?
```

Enter the node address you selected (or the node name assigned by the network manager). The node address is a numeric value in the following format:

```
area-number.node-number
```

**Area-number** (1 to 63) designates the area in which the node is grouped and **node-number** (1 to 1023) designates the node's unique address within the area. If you do not specify an area number, the system supplies a default area number (the default value is 1).

(If you are on a VAXcluster node, press RETURN to accept the default node address that appears in brackets at the end of the prompt. This default node address is based on the SYSGEN parameter SCSSYSTEMID. If no default node address is displayed, exit the procedure and use SYSGEN to set up a value for SCSSYSTEMID, then restart the procedure. The DECnet node address of a VAXcluster node must match the value of SCSSYSTEMID.)

**5. Specify router or nonrouter status.** You will be prompted as follows:

Do you want to operate as a router? [NO (nonrouting)]

Press RETURN to operate as a nonrouter (that is, as an end node). Type YES and press RETURN if you want your system to be a router and if you have registered the DECnet-VAX full function PAK or will register it before you start up the network.

**6. Specify the default access accounts for system objects, or the default DECnet account.** You will be prompted as follows:

Do you want a default DECnet account? [NO]:

(The following question will be asked only if you said YES to a default DECnet account.)

Do you want default access to the TASK object disabled? [YES]:

(The following questions will be asked regardless of whether you said YES or NO to a default DECnet account.)

Do you want a default account for the MAIL object? [YES]:

Do you want a default account for the FAL object? [NO]:

Do you want a default account for the PHONE object? [YES]:

Do you want a default account for the NML object? [YES]:

(The following questions will be asked only if you said NO to a default DECnet account.)

Do you want a default account for the MIRROR object? [YES]:

Do you want a default account for the VPM object? [YES]:

**7. Apply the configuration.** The network configuration procedure displays the list of commands necessary to start up your network. (An example showing the commands appears later in this section.)

You will be prompted as follows:

Do you want these commands to be executed? [YES]

## 6-10 Setting Up and Maintaining a Network

Press RETURN to configure the network; type NO and press RETURN to cancel the configuration operation. If you choose to configure the network, the procedure displays a series of informational messages and the following statement:

The changes have been made.

### 8. Turn on the network. You will then receive the following messages, ending in a prompt:

If you have not already registered the DECnet--VAX PAK, then do so now. After the PAK has been registered, you should invoke the procedure SYS\$MANAGER:STARTNET.COM to start up DECnet--VAX with these changes. (If the PAK is already registered) Do you want DECnet started? [YES]:

You can respond to this prompt in one of the following ways:

- If you need to register the PAK on your system at this point, type NO and press RETURN in response to the prompt. Register the PAK using the VMS License Management Utility. Once the DECnet-VAX PAK is registered, you can then start up DECnet-VAX manually with these configuration changes by entering the following command:

```
$ @SYS$MANAGER:STARTNET
```

- If the PAK is already registered but you do not want to start the network until a later time, type NO and press RETURN in response to the prompt.
- If you want to start the network at this time and the PAK is already registered, press RETURN in response to the prompt. The procedure turns on the network and displays the identification numbers of the created processes. When the dollar sign (\$) prompt appears, you have successfully configured and turned on the DECnet-VAX network.

If you want the network to be started automatically each time the VMS operating system is booted, enable the following command in the SYS\$MANAGER:SYSTARTUP\_V5.COM command procedure (by deleting the exclamation point at the beginning of this command line in the command procedure):

```
$ @SYS$MANAGER:STARTNET
```

- If you want to use DECnet-VAX only at your local node, press RETURN to start the network without the PAK being registered. The PAK is required if you want to establish connections to other nodes in the network. This response is appropriate if you are using an application that requires DECnet-VAX (for example, DECwindows), and your system is neither a member of a VAXcluster configuration nor any other network.

Example 6-1 shows the interactive dialogue that is displayed when you invoke NETCONFIG.COM to configure node PURPLE with address 2.3 as an end node with DECnet accounts for the MAIL, PHONE, MIRROR, and VPM objects only. In this example, node PURPLE is connected to Ethernet circuit UNA-0.

**Example 6-1: Sample NETCONFIG.COM Dialogue**

DECnet--VAX network configuration procedure  
 This procedure will help you define the parameters needed to get DECnet running on this machine. You will be shown the changes before they are actually executed, in case you want to perform them manually.

```

What do you want your DECnet node name to be?           : PURPLE
What do you want your DECnet address to be?            : 2.3
Do you want to operate as a router? [NO (nonrouting)]  :  RET
Do you want a default DECnet account? [NO ]:          :  RET
Do you want a default account for the MAIL object? [YES]:  RET
Do you want a default account for the FAL object? [NO ]:  RET
Do you want a default account for the PHONE object? [YES]:  RET
Do you want a default account for the NML object? [YES]:  RET
Do you want a default account for the MIRROR object? [YES]:  RET
Do you want a default account for the VPM object? [YES]:  RET
  
```

Here are the commands necessary to set up your system.

```

$ RUN SYSS$SYSTEM:NCP
  PURGE EXECUTOR ALL
  PURGE KNOWN LINES ALL
  PURGE KNOWN CIRCUITS ALL
  PURGE KNOWN LOGGING ALL
  PURGE KNOWN OBJECTS ALL
  PURGE MODULE CONFIGURATOR KNOWN CIRCUITS ALL
$ DEFINE/USER SYSS$OUTPUT NL:
$ DEFINE/USER SYSS$ERROR NL:
$ RUN SYSS$SYSTEM:NCP ! Remove existing entry, if any
  PURGE NODE 2.3 ALL
  PURGE NODE PURPLE ALL
$ RUN SYSS$SYSTEM:NCP
  DEFINE EXECUTOR ADDRESS 2.3 STATE ON
  DEFINE EXECUTOR NAME PURPLE
  DEFINE EXECUTOR MAXIMUM ADDRESS 1023
  DEFINE EXECUTOR TYPE NONROUTING IV
  DEFINE OBJECT TASK NUMBER 0 USER ILLEGAL PASSWORD DISABLED
  DEFINE OBJECT MAIL NUMBER 27 USER MAIL$SERVER PASSWORD yadnifaj
$ RUN SYSS$SYSTEM:AUTHORIZE
  ADD MAIL$SERVER /OWNER="MAIL$SERVER DEFAULT" -
    /PASSWORD=yadnifaj -
    /UIC=[376,374] /ACCOUNT=DECNET -
    /DEVICE=SYSS$SPECIFIC: /DIRECTORY=[MAIL$SERVER] -
    /PRIVILEGE=(TMPMBX,NETMBX) -
    /DEFPRIVILEGE=(TMPMBX,NETMBX) -
    /FLAGS=(RESTRICTED,NODISUSER) /LGICMD=NL: -
    /NOBATCH /NOINTERACTIVE
  MODIFY MAIL$SERVER /PASSWORD=yadnifaj
$ CREATE/DIRECTORY SYSS$SPECIFIC:[MAIL$SERVER] /OWNER=[376,374]
$ RUN SYSS$SYSTEM:NCP
  DEFINE OBJECT PHONE NUMBER 29 USER PHONE$SERVER PASSWORD dogbasow
$ RUN SYSS$SYSTEM:AUTHORIZE
  ADD PHONE$SERVER /OWNER="PHONE$SERVER DEFAULT" -
  
```

(continued on next page)

## 6-12 Setting Up and Maintaining a Network

### Example 6-1 (Cont.): Sample NETCONFIG.COM Dialogue

---

```
/PASSWORD=dogbasow -
/UIC=[376,372] /ACCOUNT=DECNET -
/DEVICE=SYS$SPECIFIC: /DIRECTORY=[PHONE$SERVER] -
/PRIVILEGE=(TMPMBX,NETMBX) -
/DEFPRIVILEGE=(TMPMBX,NETMBX) -
/FLAGS=(RESTRICTED,NODISUSER) /LGICMD=NL: -
/NOBATCH /NOINTERACTIVE
MODIFY PHONE$SERVER /PASSWORD=dogbasow
$ CREATE/DIRECTORY SYS$SPECIFIC:[PHONE$SERVER] /OWNER=[376,372]
$ RUN SYS$SYSTEM:NCP
  DEFINE OBJECT NML NUMBER 19 USER NML$SERVER PASSWORD kenrooka
$ RUN SYS$SYSTEM:AUTHORIZE
  ADD NML$SERVER /OWNER="NML$SERVER DEFAULT" -
  /PASSWORD=kenrooka -
  /UIC=[376,371] /ACCOUNT=DECNET -
  /DEVICE=SYS$SPECIFIC: /DIRECTORY=[NML$SERVER] -
  /PRIVILEGE=(TMPMBX,NETMBX) -
  /DEFPRIVILEGE=(TMPMBX,NETMBX) -
  /FLAGS=(RESTRICTED,NODISUSER) /LGICMD=NL: -
  /NOBATCH /NOINTERACTIVE
  MODIFY NML$SERVER /PASSWORD=kenrooka
$ CREATE/DIRECTORY SYS$SPECIFIC:[NML$SERVER] /OWNER=[376,371]
$ RUN SYS$SYSTEM:NCP
  DEFINE OBJECT MIRROR NUMBER 25 USER MIRRO$SERVER PASSWORD ewxgamula
$ RUN SYS$SYSTEM:AUTHORIZE
  ADD MIRRO$SERVER /OWNER="MIRRO$SERVER DEFAULT" -
  /PASSWORD=ewxgamula -
  /UIC=[376,367] /ACCOUNT=DECNET -
  /DEVICE=SYS$SPECIFIC: /DIRECTORY=[MIRRO$SERVER] -
  /PRIVILEGE=(TMPMBX,NETMBX) -
  /DEFPRIVILEGE=(TMPMBX,NETMBX) -
  /FLAGS=(RESTRICTED,NODISUSER) /LGICMD=NL: -
  /NOBATCH /NOINTERACTIVE
  MODIFY MIRRO$SERVER /PASSWORD=ewxgamula
$ CREATE/DIRECTORY SYS$SPECIFIC:[MIRRO$SERVER] /OWNER=[376,367]
$ RUN SYS$SYSTEM:NCP
  DEFINE OBJECT VPM NUMBER 51 USER VPM$SERVER PASSWORD galesobu
$ RUN SYS$SYSTEM:AUTHORIZE
  ADD VPM$SERVER /OWNER="VPM$SERVER DEFAULT" -
  /PASSWORD=galesobu -
  /UIC=[376,370] /ACCOUNT=DECNET -
  /DEVICE=SYS$SPECIFIC: /DIRECTORY=[VPM$SERVER] -
  /PRIVILEGE=(TMPMBX,NETMBX) -
  /DEFPRIVILEGE=(TMPMBX,NETMBX) -
  /FLAGS=(RESTRICTED,NODISUSER) /LGICMD=NL: -
  /NOBATCH /NOINTERACTIVE
  MODIFY VPM$SERVER /PASSWORD=galesobu
$ CREATE/DIRECTORY SYS$SPECIFIC:[VPM$SERVER] /OWNER=[376,370]
$ RUN SYS$SYSTEM:NCP
  DEFINE LINE      UNA-0 STATE ON
  DEFINE CIRCUIT  UNA-0 STATE ON COST 3
  DEFINE LINE      DMC-0 STATE ON
  DEFINE CIRCUIT  DMC-0 STATE ON COST 5
```

---

(continued on next page)

**Example 6-1 (Cont.): Sample NETCONFIG.COM Dialogue**


---

```

DEFINE LOGGING MONITOR STATE ON
DEFINE LOGGING MONITOR EVENTS 0.0-9
DEFINE LOGGING MONITOR EVENTS 2.0-1
DEFINE LOGGING MONITOR EVENTS 4.2-13,15-16,18-19
DEFINE LOGGING MONITOR EVENTS 5.0-18
DEFINE LOGGING MONITOR EVENTS 128.0-4

```

---

Do you want these commands to be executed? [YES]:

The changes have been made.

If you have not already registered the DECnet--VAX PAK, then do so now. After the PAK has been registered, you should invoke the procedure SYS\$MANAGER:STARTNET.COM to start up DECnet--VAX with these changes.

(If the PAK is already registered) Do you want DECnet started?[YES]:

---

9. **Define the other node names.** At the dollar sign (\$) prompt, invoke the Network Control Program (NCP) by entering the following command:

```
$ RUN SYS$SYSTEM:NCP
```

For each remote node in the network that you want to identify by node name, enter an NCP command in the following format to define the node address and name in your permanent node database:

```
DEFINE NODE address NAME name
```

**Address** is the existing node address in the form *area-number.node-number*, and **name** is the node name. If you omit the area number from the node address, the area number of your local node is used. The network manager or the system manager of the remote node you want to define can provide you with the correct name and address.

If a node that you can access on your network has a node database that contains all the node names and addresses you want to define and you have the appropriate privileges to access that database, you can enter a command in the following format at the NCP prompt (provided the network is turned on):

```
COPY KNOWN NODES FROM node-id TO PERMANENT
```

In this command, **node-id** is the node name or address of the remote node from which you are copying the information. If you specify the node name, that name must be in your volatile database. All the node names and addresses are copied to your permanent node database from the volatile node database of the remote node.

If your node is a member of a VAXcluster that uses an alias node identifier (that is, an alias node name and address), your node can adopt the alias. Specify commands in the following format at the NCP prompt to define the alias node address and name in the permanent node database, and associate the alias identifier with your node:

```
DEFINE NODE address NAME name  
DEFINE EXECUTOR ALIAS NODE node-id
```

For the **node-id**, you can specify either the alias node address or name that you have defined. Your node can then be identified by the alias node name and address as well as by its unique node name and address when DECnet is running.

Then enter the following commands to create the volatile node database for your node:

```
NCP>SET KNOWN NODES ALL  
NCP>EXIT
```

The other nodes on the network should define your node name and node address in their node databases in order to be able to communicate with your node by node name. If a network manager assigned the unique node name and address to your node, the manager can define your node name in an overall network node database.

10. **Determine how to proceed.** You have completed the network startup procedure. If you plan to use asynchronous DECnet, continue to the next section, which describes how to establish asynchronous connections.

### 6.2.3.1 Modifying Default Access for an Existing Network

If you have an existing, configured network that uses a default [DECNET] account, you should consider modifying that access to provide the security enhancements that are available with default accounts for individual objects (such as MAIL). Use the command procedure SYS\$UPDATE:NETCONFIG\_UPDATE.COM to modify an existing configuration in this manner.

The NETCONFIG\_UPDATE procedure configures only default access; everything else in your network configuration remains the same. When you run NETCONFIG\_UPDATE.COM in a VAXcluster environment, the procedure automatically instructs you to run the secondary procedure SYS\$MANAGER:UPDATE\_CLUSTER\_MEMBERS.COM on each of the other VAXcluster members, in order to configure each cluster member identically. You can use the SET ENVIRONMENT/CLUSTER command in the System Management Utility (SYSMAN) to apply this procedure to all cluster members while executing the procedure just one time. See Chapter 7 for more information about using SYSMAN.



### 6.2.3.2 Establishing Asynchronous DECnet Connections to Other Systems

The automatic network configuration procedure described in the previous section does not configure asynchronous lines and circuits. As a VMS system manager, you have the option of connecting your VMS system to another system by means of a low-cost, low-speed asynchronous DECnet line. You can establish either of the following two types of asynchronous DECnet connections:

- A static asynchronous DECnet connection, which creates a permanent DECnet link to a single remote node.
- A dynamic asynchronous DECnet connection, which provides a temporary DECnet link. You can establish dynamic connections to different remote nodes at different times.

Note that non-VMS systems that support DECnet asynchronous DDCMP lines can make asynchronous DECnet connections to VMS systems. The asynchronous connection can be between two routers, a router and an end node, or two end nodes. If you are on an end node and want to make an asynchronous connection, it will be your only connection to the network, because an end node can only have one circuit active at a time.

#### Establishing a Static Asynchronous Connection

A static asynchronous DECnet connection is a permanent connection between two nodes. This type of connection can be made in one of two ways:

- The nodes can be connected by a physical line (a null modem cable) attached to a terminal port at each system. No modems are required. You can communicate with the other system at any time.
- The connection can be made over a dialup line using modems at both ends of the line. For example, your VMS system can establish a static asynchronous connection to a remote node over a telephone line.

You can configure your static asynchronous line as soon as you have executed `NETCONFIG.COM`, and then turn on the network manually. If your system is brought up as a routing node, you can establish a static asynchronous connection at any time, no matter how many network connections you already have.

Follow the steps outlined in this section to establish a static asynchronous connection. For the connection to be successful, the node with which you are creating a DECnet link must also establish an asynchronous DECnet connection with your node. (Note that the line speeds at each end of the connection must be the same.)

## 6-16 Setting Up and Maintaining a Network

1. Log in to the **SYSTEM** account on your VMS node.
2. Load the asynchronous DDCMP driver, **NODRIVER (NOA0)**. Enter the commands shown below at your terminal (or include them in the **SYS\$MANAGER:SYSTARTUP\_V5.COM** command procedure before you boot the system).

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> CONNECT NOA0/NOADAPTER
SYSGEN> EXIT
```

The asynchronous driver must be loaded before any asynchronous connection can be made.

3. To set up the terminal line to become a static asynchronous DECnet line, enter the DCL command **SET TERMINAL** at your terminal. If there is more than one terminal attached to your VMS system, you must specify a **SET TERMINAL** command for each terminal line that will be used for a static asynchronous DECnet connection.

- **Nondialup line**—For a nondialup configuration, enter the following **SET TERMINAL** command to convert a terminal line to a static asynchronous line:

```
$ SET TERMINAL/PERMANENT/PROTOCOL=DDCMP device-name:
```

In this command, **device-name** is the name of the terminal port that is connected to the line that you want to make a static asynchronous DECnet line. (All references to a device in this section refer to the terminal port.)

- **Dialup line**—For a dialup configuration, enter the **SET TERMINAL** command in the following format to convert the terminal line to a static asynchronous DECnet line with modem control.

```
SET TERMINAL/PERMANENT/MODEM/NOAUTOBAUD/NOTYPE_AHEAD-
/PROTOCOL=DDCMP device-name:
```

You can ensure that these **SET TERMINAL** commands will be executed automatically each time the network is started. Modify your **SYS\$MANAGER:SYSTARTUP\_V5.COM** command procedure to include all required **SET TERMINAL** commands before the command **@SYS\$MANAGER:STARTNET**.

4. After configuring your node, configure the asynchronous lines and circuits in the network database. Use NCP commands to define each asynchronous line and accompanying circuit as being in the **ON** state. (The line and circuit are turned on when **SYS\$MANAGER:STARTNET.COM** is executed.) Enter the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>DEFINE LINE dev-c-u STATE ON RECEIVE BUFFERS 4 -
      _LINE SPEED baud-rate
NCP>DEFINE CIRCUIT dev-c-u STATE ON
NCP>EXIT
```

**Baud-rate** is the speed at which the line sends and receives data. For an asynchronous line or circuit, **dev-c-u** is defined as follows:

- dev** The first two letters of the asynchronous device name (possible values are TT and TX).
- c** A decimal number (zero or a positive integer) designating a device's hardware controller. If the third letter of the device name is A, c equals zero. If the third letter of the device name is B, c equals 1, and so on.
- u** The unit number of the device name; u is always equal to zero or a positive integer.

(An example is the device identifier TT-0-0, which represents the asynchronous device name TTA0.)

A minimum of four buffers should be allocated for data reception over the line. An insufficient number of receive buffers on asynchronous DDCMP lines can cause such network problems as timeouts and loss of packets. If these problems occur, you can enter the NCP command SHOW CIRCUIT to confirm whether an insufficient number of receive buffers was the cause:

```
$ RUN SYS$SYSTEM:NCP
NCP> SHOW CIRCUIT TT-0-0 CHARACTERISTICS
```

If the counters show any Remote Buffer Errors that include the expression "buffer unavailable," you should increase the number of receive buffers for the line, using the following procedure:

- a. Determine the number of receive buffers for line line, using the following command (substituting the appropriate line number):

```
NCP> SHOW LINE TT-0-0 CHARACTERISTICS
.
.
.
Receive Buffers           = 4
.
.
.
```

- b. Use the NCP command SET LINE to increase the number of receive buffers:

```
NCP> SET LINE TT-0-0 STATE OFF
NCP> SET LINE TT-0-0 RECEIVE BUFFERS 6
NCP> SET LINE TT-0-0 STATE ON
```

- c. Use the NCP DEFINE LINE command to change the number of receive buffers in your permanent database.

## 6-18 Setting Up and Maintaining a Network

If the line speed at the other end of the connection is changed after the initial static asynchronous connection is made, you can use the **DEFINE LINE** command to change the line speed for your end of the connection to match the line speed at the other end. The line speed will be changed the next time the line is turned on.

5. For security over a dialup connection, you can run NCP and establish optional transmit and receive passwords for the local end of the static asynchronous dialup link. The transmit password is the password sent to the other node during connection startup; the receive password is the password expected from the other node during connection startup. You must also use NCP to specify that your asynchronous circuit verifies the password supplied by the other node. If the correct passwords are not supplied, the asynchronous connection cannot be made.

Although transmit and receive passwords are not mandatory for static asynchronous dialup links, they add to the security of your DECnet connection. Passwords can contain from one to eight alphanumeric characters and must be delimited with quotation marks if they contain spaces. Specify commands at the NCP prompt using the following format:

```
DEFINE CIRCUIT dev-c-u VERIFICATION ENABLED
DEFINE NODE node-id TRANSMIT PASSWORD transmit-password -
      RECEIVE PASSWORD receive-password
```

**Node-id** is the name of the remote node to which your node will be connected.

Note that if you have defined passwords for the local end of the link, you must notify the remote node system manager to establish transmit and receive passwords for the remote end of the static asynchronous DECnet dialup link.

6. If the network is not already on, turn on the network at your node by entering the following command:

```
$ @SYS$MANAGER:STARTNET
```

This command starts the network and causes the permanent database entries defined in the previous steps to be entered in the volatile database on the running network.

If the network was already running before you began the static asynchronous connection procedure, enter the following commands to cause the permanent database entries to be entered in the volatile database.

```
$ RUN SYS$SYSTEM:NCP
NCP>SET LINE dev-c-u ALL
NCP>SET CIRCUIT dev-c-u ALL
NCP>SET NODE node-id ALL
NCP>EXIT
```

If the line and circuit could not be set on in the volatile database, causing DECnet to fail to gain control of the line, the following error message is displayed:

```
% NCP-I-NMLRSP, LISTENER RESPONSE - Operation failure
```

See the solutions suggested in Section 6.3.2.3 if the static asynchronous connection cannot be made.

7. If you want to turn off the asynchronous lines temporarily, run NCP and enter the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET LINE dev-c-u STATE OFF
NCP>SET CIRCUIT dev-c-u STATE OFF
NCP>CLEAR LINE dev-c-u ALL
NCP>CLEAR CIRCUIT dev-c-u ALL
NCP>EXIT
```

To turn the static asynchronous DECnet line back on, enter the following NCP commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET LINE dev-c-u ALL
NCP>SET CIRCUIT dev-c-u ALL
NCP>EXIT
```

8. If you want to switch an asynchronous DECnet line back to a terminal line with DECnet running, you must clear the line and circuit entries from the network volatile database. To clear the entries, enter these commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET LINE dev-c-u STATE OFF
NCP>SET CIRCUIT dev-c-u STATE OFF
NCP>CLEAR LINE dev-c-u ALL
NCP>CLEAR CIRCUIT dev-c-u ALL
NCP>EXIT
```

To switch the line for which modem control was not enabled back to a terminal line, enter the following command:

```
$ SET TERMINAL/PERMANENT/PROTOCOL=NONE device-name:
```

To switch the line for which modem control was enabled back to a terminal line, enter the following command:

```
$ SET TERMINAL/PERMANENT/MODEM/AUTOBAUD -
_$/TYPE_AHEAD/PROTOCOL=NONE device-name:
```

### Establishing a Dynamic Asynchronous Connection

A dynamic asynchronous DECnet connection is a temporary connection between two nodes, normally over a telephone line through the use of modems. The line at each end of the connection can be switched from a terminal line to a dynamic asynchronous DECnet line. Configuration of dynamic asynchronous lines is performed automatically by DECnet during establishment of a dynamic connection. A dynamic asynchronous connection is normally maintained only for the duration of a telephone call.

**NOTE:** A dynamic asynchronous connection to a VMS node can be initiated from any VMS or non-VMS node that supports the DECnet asynchronous DDCMP protocol.

On a VMS node, you have the option of performing the initial steps of the dynamic asynchronous connection process (steps 1 and 2 as follows) before you turn on the network at your node (step 3). The later steps of the process (starting with step 4) must occur when the line is being switched to DECnet.

Follow the steps listed in this section to establish a dynamic asynchronous DECnet connection. This procedure assumes the local VMS node is originating the connection and switching on the terminal line for DECnet use. The connection must be to a VMS node on which you have an account with NETMBX privilege. The steps that the system manager at the remote VMS node must perform in order for the dynamic asynchronous DECnet link to be established successfully are also included in this section.

1. Log in to the SYSTEM account and enter the following commands interactively (or include them in the SYS\$MANAGER:SYSTARTUP\_V5.COM command procedure before you boot the system). These commands load the asynchronous driver NODRIVER (NOAO) and install DYN SWITCH software on your system.

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> CONNECT NOAO/NOADAPTER
SYSGEN> EXIT
$ INSTALL:=$SYS$SYSTEM:INSTALL
$ INSTALL/COMMAND
INSTALL> CREATE SYS$LIBRARY:DYN SWITCH/SHARE -
_/PROTECT/HEADER/OPEN
INSTALL> EXIT
```

The system manager of the remote VMS node must also enter these commands.

Additionally, the system manager at the remote VMS node must enter the commands that follow. These commands enable the use of *virtual terminals* for the terminal line that is to be switched, and set the DISCONNECT characteristic for the terminal line. (The virtual terminal capability permits the process to continue running if the physical terminal you are using becomes disconnected.)

```

$ RUN SYSS$SYSTEM:SYSGEN
SYSGEN> CONNECT VTA0/NOADAPTER/DRIVER=TTDRIVER
SYSGEN> EXIT
$ SET TERMINAL/EIGHT_BIT/PERMANENT/MODEM/DIALUP -
_ $ /DISCONNECT device-name:

```

**Device-name** is the name of the terminal port to which the dynamic asynchronous connection is made.

2. You must establish the required transmit password at the originating end of the dynamic asynchronous dialup link. The transmit password is the password sent to the remote node during connection startup. Use NCP to enter a command to define the transmit password for the remote node. The password can contain one to eight alphanumeric characters and should not contain any spaces. At the NCP prompt, use the following command format:

```
DEFINE NODE node-id TRANSMIT PASSWORD password
```

**Node-id** is the name of the remote node with which your node is forming a connection.

For each remote node with which you will create a dynamic asynchronous DECnet dialup link, you must define a transmit password in a separate command.

The system manager for the node at the other end of the connection must define that same password as a receive password for your node (the password expected to be received from your node). The remote system manager should also specify the parameter **INBOUND ROUTER** or **INBOUND ENDNODE**, to indicate the type of node (router or end node) that is expected to initiate the dynamic connection. The remote manager should enter a command in the following format at the NCP prompt:

```
DEFINE NODE node-id RECEIVE PASSWORD password INBOUND node-type
```

3. DECnet must be running on both nodes for the remaining steps. If you have not already done so, turn on the network by entering the following command (and request that the remote system manager do so also):

```
$ @SYS$MANAGER:STARTNET
```

If the network was already running before you began the dynamic asynchronous connection procedure, use a command in the following format at the NCP prompt to enter the permanent database entry into the volatile database:

```
SET NODE node-id ALL
```

4. The remaining steps can be performed by any VMS user with **NETMBX** privilege. Log in to your local VMS system and enter a DCL command using the following format to cause your process to function as a *terminal emulator* (which makes the remote terminal appear to be a local terminal connection):

## 6-22 Setting Up and Maintaining a Network

SET HOST/DTE device-name:

**Device-name** is the name of your local terminal port that is connected to the modem. If both systems use modems with autodial capabilities (for example, DF03, DF112 or DF224 modems that support an autodial protocol), you can optionally include the /DIAL qualifier on the SET HOST/DTE command to cause automatic dialing of the modem on the remote node, as in the following format:

SET HOST/DTE/DIAL=number device-name:

5. If you are not using automatic dialing, dial in to the remote node manually.
6. Once the dialup connection is made and you receive the remote VMS system welcome message, log in to your account on the remote node.
7. While logged in to your account on the remote node, enter the following command to cause the line to be switched to a DECnet line automatically:

```
$ SET TERMINAL/PROTOCOL=DDCMP/SWITCH=DECNET
```

The following message indicates that the DECnet link is being established:

```
%REM-S-END - control returned to local-nodename::  
$
```

To check whether the communications link has come up, specify the following command on the local system:

```
$ RUN SYSS$SYSTEM:NCP  
NCP>SHOW KNOWN CIRCUITS  
NCP>EXIT
```

The resulting display should list a circuit identified by the mnemonic TT or TX, depending on the asynchronous device installed on the line, and indicate that it is in the ON state.

When the DCL prompt (\$, by default) appears on your terminal screen, you can begin to communicate with the remote node over the asynchronous DECnet connection.

If the dynamic connection is not made successfully, refer to Section 6.3.2.3.

8. As an alternative to switching the terminal line to a DECnet line automatically (as described in the previous step), you can switch the line manually. If you originate a dynamic connection to a VMS node from a non-VMS system, manual switching is required; from a VMS system, it is optional. If you are originating the connection from a non-VMS node, follow system-specific procedures to log in to the remote VMS node by means of terminal emulation.

Once you are logged in to the remote node, two steps are required to perform manual switching:

- a. Using your account on the remote VMS node, specify the SET TERMINAL command described in Step 7, and add the /MANUAL qualifier:



```
$ SET TERMINAL/PROTOCOL=DDCMP/SWITCH=DECNET/MANUAL
```

You will receive the following message from the remote node indicating the remote system is switching its line to DECnet use:

```
%SET-I-SWINPRG The line you are currently logged over is becoming
a DECnet line
```

- b. You should exit from the terminal emulator and switch your line manually to a DECnet line. The procedure depends on the specific operating system on which you are logged in. The following example shows how a VMS user originating a dynamic connection would perform this procedure.
  1. Exit from the terminal emulator by pressing the backslash (\) key and the CTRL key simultaneously.
  2. Enter the following command to switch your terminal line to a DECnet line manually:

```
$ SET TERMINAL/PROTOCOL=DDCMP TTA0:
```

TTA0 is the name of the terminal port on the local node.

3. Enter NCP commands to turn on the line and circuit connected to your terminal port TTA0 manually, as in the following example:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET LINE TT-0-0 RECEIVE BUFFERS 4 LINE SPEED 2400 STATE ON
NCP>SET CIRCUIT TT-0-0 RECEIVE BUFFERS 4 STATE ON
NCP>EXIT
```

Asynchronous DECnet is then started on the local VMS node.

9. You can terminate the dynamic asynchronous link in one of two ways:
  - a. Break the telephone connection.
  - b. Run NCP and turn off either the asynchronous line or circuit. The two commands you can use are as follows:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET LINE dev-c-u STATE OFF
NCP>SET CIRCUIT dev-c-u STATE OFF
NCP>EXIT
```

If either of the above NCP commands is entered at the remote node, the line returns to terminal mode immediately. If the command is entered at the local (originating) VMS node, the remote line and circuit remain on for approximately 4 minutes and then the line returns to terminal mode.

### 6.2.3.3 Starting the DECnet-VAX Software

You must start the DECnet-VAX software each time you boot your system. The best way to do this is to include the following line in your site-specific system startup file, `SYS$MANAGER:SYSTARTUP_V5.COM`:

```
$ @SYS$MANAGER:STARTNET
```

The command procedure `SYS$MANAGER:STARTNET.COM` is supplied with your VMS system, and you can start the DECnet-VAX software by executing this procedure. Make sure that you execute `STARTNET.COM` in `SYSTARTUP_V5.COM` before you start up any software that requires DECnet-VAX.

### 6.2.3.4 DECnet-VAX on Workstations not Connected to a Network

If you manage a workstation that is using DECwindows (or another windowing product that requires DECnet-VAX) then you must install DECnet-VAX on your system, even if your workstation is not connected to any other computers. However, if your workstation is not part of a network, the procedure for installing DECnet-VAX involves fewer steps than other DECnet-VAX installations. Use the procedure shown below when you meet both of the following criteria:

- You have a workstation that uses DECwindows or other software that requires DECnet-VAX.
- The workstation is not part of any network or cluster. That is, the workstation is not able to communicate directly with any other computer.

When you have this type of configuration, use the following procedure to install and use DECnet-VAX on your system:

1. Configure your "network" using the `SYS$MANAGER:NETCONFIG.COM` procedure, as described in Section 6.2.3. You will have to select a node name and a network address, as described in that section. After selecting a node name and address, you can simply accept the default values and answers for the remainder of the `NETCONFIG` procedure, except for the values that create accounts for default access. Do not set up default accounts for remote access (for example, for mail or file access), because your system is not on a network that is available to remote users.
2. In your site-specific system startup file, `SYS$MANAGER:SYSTARTUP_V5.COM`, include a line that executes the procedure `SYS$MANAGER:STARTNET.COM`, in order to start the DECnet-VAX software each time your system is booted. Make sure you execute the `SYS$MANAGER:STARTNET.COM` procedure before executing any procedures that start up software products requiring DECnet-VAX (for example, `SYS$MANAGER:DECW$STARTUP.COM`).

### 6.2.3.5 Shutting Down and Restarting the Network

The network shuts down automatically as part of the normal VMS system shutdown procedure. If your VMS system is running, you can shut down the network at your local node without destroying any active logical links by entering the following commands:

```
$ RUN SYSSYSTEM:NCP
NCP>SET EXECUTOR STATE SHUT
NCP>EXIT
```

When you enter this command sequence, no new links are allowed; when all existing links are disconnected, the network is turned off.

While your VMS system is running, you can stop the network at your node by entering the following commands:

```
$ RUN SYSSYSTEM:NCP
NCP>SET EXECUTOR STATE OFF
NCP>EXIT
```

All logical links are disconnected immediately and the network is stopped.

To turn on the network manually, specify the following:

```
$ @SYS$MANAGER:STARTNET
```

To start the network if it is not currently active, you must be logged in to the SYSTEM account or have the privileges listed at the beginning of the STARTNET.COM command procedure.

To cause the network to be started each time the VMS operating system is booted, enable the following command in the SYS\$MANAGER:SYSTARTUP\_V5.COM command procedure:

```
$ @SYS$MANAGER:STARTNET
```

The command is supplied in the command procedure; to enable it, use a text editor to delete the exclamation point at the start of the command line. The network is turned on automatically as part of the VMS system startup. You do not have to turn on the network again unless you should explicitly shut down the network or remove the network startup invocation from the site-specific startup command procedure.

### 6.2.3.6 Using NCP to Create and Tailor the Configuration Database

The system manager is responsible for configuring the node for network operation by supplying information in the DECnet-VAX configuration database about the following network components:

- The local (executor) node
- Remote nodes with which the local node can communicate
- Local circuits
- Local lines

- Network objects
- Network event logging

The configuration database is actually two databases: a permanent database that establishes the default parameter values for node startup, and a volatile database that contains the current parameter values in a functioning network.

You can use the NCP to build the network configuration database manually or to modify its contents. If you are configuring the node for the first time, you can use the automatic configuration command procedure, `NETCONFIG.COM`, to establish parameters needed to start DECnet running. The procedure for using `NETCONFIG.COM` is described in an earlier section.

Use NCP `SET` commands to establish the contents of the volatile database. Use NCP `DEFINE` commands to establish the contents of the permanent database. You must have `OPER` privilege to change the volatile database and `SYSPRV` privilege to change the permanent database.

The permanent database information is supplied to the volatile database when the network is started (that is, the `STARTNET.COM` command procedure is executed). You can also use the `ALL` parameter with the `SET` command to cause all permanent database entries for a network component to be loaded into the volatile database.

The basic NCP commands required to define the network components in the permanent configuration database are as follows:

```
$ RUN SYSS$SYSTEM:NCP
NCP>DEFINE EXECUTOR
NCP>DEFINE NODE node-id
NCP>DEFINE CIRCUIT circuit-id
NCP>DEFINE LINE line-id
NCP>DEFINE OBJECT object-name
NCP>DEFINE LOGGING MONITOR STATE ON
NCP>DEFINE LOGGING MONITOR EVENTS event-list
NCP>EXIT
```

NCP commands also recognize the plural forms of the network component names: `KNOWN NODES`, `KNOWN CIRCUITS`, `KNOWN LINES`, `KNOWN OBJECTS`.

To modify the current configuration of your node, you can enter `SET` commands for any network component. For example, to add circuit and line entries for the Ethernet UNA device (the `DEUNA`), enter the following commands:

```
$ RUN SYSS$SYSTEM:NCP
NCP>SET LINE UNA-0 STATE ON
NCP>SET CIRCUIT UNA-0 STATE ON
NCP>EXIT
```

To determine the contents of your network configuration database, use the NCP commands `LIST` and `SHOW`. The `LIST` command displays information in the permanent database; the `SHOW` command displays volatile database entries. To delete entries from the configuration database, use the `PURGE` and `CLEAR`

commands. The PURGE command deletes permanent database entries; the CLEAR command deletes or resets volatile database entries.

For example, to list the permanent name and address of a node, enter the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>LIST NODE node-id
NCP>EXIT
```

To delete a node from the permanent database, enter the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>PURGE NODE node-id ALL
NCP>EXIT
```

**Node-id** can be either the node name or the node address. You can also delete an individual parameter for a node.

Because the PURGE command does not affect the volatile (memory-resident) copy of the DECnet database, you can access a node deleted with the PURGE command until DECnet is started again. If you use the CLEAR command to delete a node entry, the node entry will reappear in the volatile database after DECnet is started again.

### 6.2.3.7 Providing Security for Your DECnet-VAX Node

Some of the security measures that you can use to protect your files and system in a network environment are summarized in this section. You should also read Chapter 11 for more information about system security.

As system manager of a VMS node, you can protect your system against unauthorized access by users on other nodes in the network by setting passwords for any accounts that you might create. Otherwise, users on other nodes could gain full access to your system by using the SET HOST command to log in to one of the accounts on your node.

#### Protecting Files and Using Proxy Accounts

As a user on a VMS node, you can protect the files in your directory against access over the network. To set limits on who can access the files in your account, specify the DCL command SET PROTECTION. If your file is protected, a VMS user on a remote node who wants to access your file must be able to specify the user name and password of a local account that has the appropriate privileges to access the file. A remote user to whom you have given this information must then include the authorization information in the form of an **access control string**, "*username password*", in the VMS file specification used to access your file:

```
node"username password"::device:[directory]filename.type;version
```

## Establishing Proxy Accounts

As system manager of your node, you can maintain the security of passwords by preventing their transmission over the network. You can permit selected outside users to access particular nonprivileged accounts on your node without having to send any explicit access control information over the network. To do this, you must create a proxy account that allows a remote user to have access privileges on your node without having a private account on your node. If the remote user is assigned a proxy account on your local node that maps into a local user account, the remote user assumes the same access privileges as the owner of the local account.

The system manager controls the use of proxy accounts at the local node. Use the Authorize Utility (AUTHORIZE) to create and modify the permanent proxy database, NETPROXY.DAT, at your node. Each NETPROXY.DAT entry can map a single remote user to multiple proxy accounts on the local node (one default proxy account and up to 15 additional proxy accounts). The proxy database entry identifies the user by *nodename::username* or *nodename::(group,member)*.

For example, to create a NETPROXY.DAT file at local node BOSTON and add a default proxy entry mapping user MARTIN on remote node MIAMI to user ALLEN at the local node, enter the following commands:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> CREATE/PROXY
UAF> ADD/PROXY MIAMI::MARTIN ALLEN/DEFAULT
UAF> EXIT
```

(For information on using AUTHORIZE, refer to Chapter 4.)

When DECnet is started up, the information in NETPROXY.DAT is used to construct a volatile proxy database. If changes are made to the permanent proxy database with AUTHORIZE, the volatile proxy database is updated automatically.

Similarly, the system manager at a remote node can create and maintain a proxy database of network users having proxy access to specific accounts on that node.

## Controlling Proxy Login Access

For proxy login to be successful, one node must be able to initiate proxy login access and the other node must allow proxy login access. To control proxy login for your local (executor) node, use Network Control Program commands to modify the proxy parameters in the executor and object databases. The NCP parameters that specify whether a node can initiate proxy login are the outgoing proxy parameters; the parameters that specify whether a node allows proxy login access are the incoming proxy parameters. By default, both the local node and the remote node can initiate proxy logins and allow proxy access.

Defaults for objects supplied by Digital are set in the object database. For example, the object MAIL has outgoing proxy access set by default. To specify or modify the proxy parameter for a network object, use the NCP command SET OBJECT. Use this command to permit outgoing proxy access for a network object:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET OBJECT object-name PROXY OUTGOING
NCP>EXIT
```

### Controlling Access to Your Node

In general, the system manager can control access to the local node at three levels:

- **Circuit-level access control:** For point-to-point connections, especially over dialup lines, you can use passwords to verify that the initiating node is authorized to form a connection with your node. Passwords are usually optional for point-to-point connections but are required for dynamic asynchronous connections.

Each end of a point-to-point circuit can establish a password to transmit to the other node, and specify a password expected from the other node. Before the link is established, each node verifies that it received the expected password from the other node.

Added security is provided for a dynamic asynchronous connection (which is normally maintained only for the duration of a telephone call): the node requesting the dynamic connection is required to supply a password, but the node receiving the login request is prevented from revealing a password to the requesting node.

- **Node-level access control:** To control the establishment of logical links with remote nodes, you can specify in your network database access control parameters that indicate which of the following logical link connections are permitted: INCOMING, OUTGOING, BOTH, or NONE. Use the NCP commands that follow to specify access parameters for a specific node, and the executor parameter DEFAULT ACCESS that applies to any node for which a specific access parameter is not specified:

```
$ RUN SYS$SYSTEM:NCP
NCP>DEFINE NODE node-id ACCESS option
NCP>DEFINE EXECUTOR DEFAULT ACCESS option
NCP>EXIT
```

- **System-level access control:** When a remote user requests access to the system, the following means of authorization are checked:
  - Is an explicit access control string available?
  - Does the user have a proxy account on the local node?
  - Is there a default object account?
  - Is there a default nonprivileged DECnet account at the local node?

If no explicit access control information or proxy account is available, DECnet-VAX attempts to use a default nonprivileged DECnet account to access the system. The default DECnet account allows users to perform certain network operations, such as the exchange of electronic mail between users on different nodes, without having to supply a name and password. The default DECnet account is also used for file operations when an access control string is not supplied. For example, it permits remote users to access local files on which the file protection has been set to allow WORLD access. If you do not want remote users to access your node, do not create a default DECnet account.

You can request the DECnet-VAX configuration command procedure, NETCONFIG.COM, to establish the default nonprivileged DECnet account and directory for you automatically, or you can establish the account and directory manually, as follows:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF>ADD NETNONPRIV/PASSWORD=NONPRIV/DEVICE=device-name:-
_ /DIRECTORY=[NETUSER]/UIC=[200,200]/PRIVILEGE=(TMPMBX,NETMBX)-
_ /FLAGS=(CAPTIVE)/NOBATCH/NOINTERACTIVE/LGICMD=NL:
UAF>EXIT
$ CREATE/DIRECTORY device-name:[NETUSER]/OWNER_UIC=[200,200]
```

**Device-name** is the name of the device on which you have your directory.

If a remote node requests access to an object on the local node but does not supply access control information, any access control information specified for the object in the local network database is used.

## 6.3 Keeping the Network Running

After you bring up your system as a network node, you can use a variety of software techniques to monitor and test the network. You can also use troubleshooting techniques to resolve problems related to keeping the network running. The tools you can use to monitor the network and the types of tests you can perform on the network are summarized in the following sections. Common problems encountered during network operation are indicated, along with advice on troubleshooting.

### 6.3.1 Monitoring the Network

You can monitor network activity using software tools. Analyzing the information you collect can help you to determine whether the network is running properly or whether any changes are required to resolve problems or improve performance. Major network monitoring tools include the following:

- NCP display commands you can use to determine the status and characteristics of components in the network



- NCP counters you can read to obtain error and performance statistics on current network operations
- Network events logged by DECnet that can be reported to you as they happen
- Other software tools, such as the Ethernet configurator and the DECnet Test Sender/DECnet Test Receiver (DTS/DTR) Utility, that permit you to learn more about network operation

### 6.3.1.1 Using NCP to Display Information About Network Components

You can use the NCP commands `SHOW` and `LIST` to monitor network activity by displaying the following:

- Information about the current condition of network components (using `SHOW` commands) and the startup values assigned to the components (using `LIST` commands)
- Counter information about circuits, lines, remote nodes, and the local node (using `SHOW COUNTER` commands)
- Information about the range of network events being logged by the DECnet event logging facility (using `SHOW LOGGING` commands)

You do not need any privileges to enter `SHOW` commands, but you do need the privilege `SYSPRV` to enter `LIST` commands.

Use the `SHOW` command to monitor the operation of the running network. You can display the characteristics and current status of network circuits, lines and nodes, including the local (executor) node. This information is useful in detecting any changes in the network configuration or operation. For example, if a circuit failure causes some nodes to become unreachable, you can use `SHOW` commands to check the status of the circuit and the nodes.

In general, the `SHOW` and `LIST` commands permit you to indicate what type of information you want NCP to display about the particular component you specify. The display types include the following:

- **CHARACTERISTICS**—Static information that does not normally change during network operations (for example, the identification of the local node and the circuits connected to the local node, and relevant routing parameters such as circuit cost).
- **STATUS**—Dynamic information that usually indicates network operation for the running network (for example, the operational state of the local node, circuits, lines and remote nodes).
- **SUMMARY**—Only the most useful information from both static and dynamic sources; usually a condensed list of information provided for the **CHARACTERISTICS** and **STATUS** display types. **SUMMARY** is the default if you do not specify a display type.

## 6-32 Setting Up and Maintaining a Network

- **COUNTERS**—Counter information about circuits, lines, remote nodes, and the local node.
- **EVENTS**—Information about which network events are currently being logged to which logging collection point.

When you display information about network components, you can specify either the singular or plural form of the component in the NCP command. Plural forms of component names are **KNOWN** (all components available to the local node), **ACTIVE** (all circuits, lines and logging not in the **OFF** state), and **ADJACENT** (all nodes directly connected to the local node).

Typical examples of NCP display commands follow:

```
$ RUN SYSSYSTEM:NCP
NCP>SHOW EXECUTOR CHARACTERISTICS
NCP>SHOW KNOWN LINES STATUS
NCP>SHOW ACTIVE CIRCUITS
NCP>SHOW ADJACENT NODES STATUS
NCP>LIST KNOWN NODES
NCP>EXIT
```

All NCP display commands optionally allow you to direct the information displayed to an output file you specify.

You can display information about network components on remote nodes using the **TELL** prefix in the NCP command. The format of the command is **TELL node-id SHOW component**. For example, to look at remote node counters, enter a command in the following format at the NCP prompt:

```
TELL node-id SHOW EXECUTOR COUNTERS
```

### 6.3.1.2 Using NCP Counters

You can use NCP commands to display error and performance statistics about network components at any time while the network is running. DECnet software uses counters to collect statistics for the executor node, remote nodes, circuits and lines automatically. To display the contents of counters, use NCP **SHOW COUNTER** commands, as in the format shown in following typical NCP commands:

```
SHOW EXECUTOR COUNTERS
SHOW NODE node-id COUNTERS
SHOW KNOWN CIRCUITS COUNTERS
SHOW KNOWN LINES COUNTERS
SHOW LINE line-id COUNTERS
```

For the local node and remote nodes, counter statistics cover such subjects as connection requests, user data traffic, timeouts, and errors. Circuit counters cover such topics as the transmission of data packets over the circuit, timeouts, and errors. Line counters cover such information as the transmission of bytes and data blocks over the line and relevant errors.

Use NCP commands to control counter usage. You can reset counters to zero if you are establishing a controlled environment for test purposes. To reset counters to zero, use the NCP command ZERO COUNTERS (the ZERO command requires the OPER privilege). You can reset counters to zero for the executor node and individual nodes, circuits and lines, or all nodes, circuits and lines. Use NCP commands in the following format (and note that the word COUNTERS is optional):

```
ZERO EXECUTOR COUNTERS
ZERO NODE node-id
ZERO KNOWN CIRCUITS
ZERO LINE line-id COUNTERS
```

You can regulate the frequency with which specific counters are logged by entering a command in the following format at the NCP prompt:

```
SET component COUNTER TIMER nn
```

The variable *nn* is in seconds. Expiration of the counter timer causes the contents of the counter to be logged and the counter reset to zero. For example, use the following command to cause a node counter logging event to occur every 600 seconds for the local node:

```
$ RUN SYSS$SYSTEM:NCP
NCP>SET EXECUTOR COUNTER TIMER 600
NCP>EXIT
```

### 6.3.1.3 Using DECnet Event Logging

Use the DECnet event logging facility to monitor significant network events, such as circuit failures or lost packets, on a continuous basis. Whenever a network error or other meaningful event occurs, the DECnet event logger logs an event message to a terminal or file that you specify. Examples of network events that are logged as they happen include the following:

- Changes in circuit and line states (for example, a circuit failure)
- A node becoming reachable or unreachable
- Circuit and node counter values, logged before the counter is automatically reset to zero
- Errors in data transmission
- Use of invalid data link passwords

Collection and analysis of network events can provide insight into why a particular error condition exists or why network performance can vary.

As events are detected, the event logger sends them to a collection point for analysis. Collection points are called *logging sinks*; they can be located on the local node or at a remote node. Event data can go to one or more sinks. Each of the following types of event sinks handles event data in a slightly different way:

- **Logging monitor.** A program that receives and processes events. Events sent to the logging monitor are displayed on the screen of any terminal declaring itself a “network operator” by means of the Operator Communication (OPCOM) facility. Directing events to the OPCOM terminal is very useful for applications where the operator needs to know what is happening on the network as it happens. For example, it might be useful to know that a circuit is going down as it happens.

The automatic configuration command procedure NETCONFIG.COM enables the logging monitor. The OPCOM process is started when the command procedure SYS\$MANAGER:SYSTARTUP\_V5.COM is executed. You can enable a terminal as a network operator terminal by specifying the DCL command REPLY/ENABLE=NETWORK. Usually the operator console (OPA0) is one of the OPCOM terminals.

- **Logging console.** A terminal or file that receives events in a readable format. The default logging console is the operator console.
- **Logging file.** A user-specified file that receives events in binary format, possibly for later analysis.

In order for logging to occur at your node, logging must be enabled and the events to be logged must be identified. If you use the automatic configuration command procedure, NETCONFIG.COM, logging will be established automatically.

Otherwise, you can use the NCP command SET or DEFINE LOGGING to set the logging sink state to be ON. To identify a remote location for a logging sink, specify the SINK *node-id* parameter in the command. Use one or more separate commands to define the events to be logged. For example, enter the following commands to cause all network events to be logged to OPCOM and displayed at your network operator terminal:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET LOGGING MONITOR STATE ON
NCP>SET LOGGING MONITOR KNOWN EVENTS
NCP>EXIT
```

Alternatively, for each event class you can specify the specific events to be logged, as follows:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET KNOWN LOGGING EVENTS event-list
NCP>EXIT
```

Events in the event list are identified by class and type (in the form *class.type*). An event *class* refers to the DECnet software functional layer in which the event occurred. Event classes logged by DECnet include those listed in Table 6-1. The event *type* is a decimal number representing a unique event within the class. You

can use the asterisk (\*) wildcard character for event types, and you can specify a single event type or a range of event types.

**Table 6-1: DECnet Event Classes**

Event Class	DECnet Functional Layer
0	Network Management
1	Application
2	Session Control
3	End Communication
4	Routing
5	Data Link
6	Physical Link
7	X.25 packet-level events
128-159	VMS system-specific

An example of the command to specify event types 5 through 7 in event class 4 is as follows:

```
$ RUN SYS$SYSTEM:NCP
NCP>DEFINE LOGGING MONITOR EVENTS 4.5-7
NCP>EXIT
```

The event message displayed by OPCOM is in the following format:

```
EVENT TYPE class.type, event-text
From node-address (node name) Occurred (date and time)
component type and identifier
descriptive text
```

An example of a network event message display on the operator terminal at node RED is as follows:

```
%OPCOM, 29-APR-1988, 11:10:09.54, message from user DECnet
DECnet event 4.14, node reachability change
From node 2.5 (RED), 29-APR-1988 11:10:05.16
Node 2.4 (YELLOW), Reachable
```

You can use the SHOW LOGGING command to learn what logging is being performed. For example, to display complete information about all logging being conducted at all nodes, use these commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>SHOW ACTIVE LOGGING KNOWN SINKS
NCP>EXIT
```

To stop monitoring at the network operator terminal temporarily, enter the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET LOGGING MONITOR STATE OFF
NCP>CLEAR LOGGING MONITOR ALL
NCP>EXIT
```

Enter these commands to turn monitoring back on:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET LOGGING MONITOR STATE ON
NCP>EXIT
```

To disable logging at the network operator terminal permanently, enter the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>PURGE LOGGING MONITOR ALL
NCP>EXIT
```

## 6.3.2 Common Problems Encountered on the Network

Once you bring up your system as a network node, you might receive messages related to networking errors. Other problems that can occur at any time during network operation might not result in messages being displayed. This section explains the causes of error messages that might be displayed.

### 6.3.2.1 Common Error Messages and Meanings

When you are using DECnet-VAX, you might receive network-related messages indicating software or hardware problems, transient conditions, or errors in your input. The following list displays some common network-related messages, explains what condition might be causing each message, and suggests actions you can take.

- **NCP-I-INVPVA, invalid parameter value**

This message is displayed if you specify a parameter value in an NCP command that is not a valid value for the specified parameter. The name of the parameter for which the value was invalid is displayed at the end of the error message. Re-enter the command with the correct value for the parameter.

- **SYSTEM-I-LINKEXIT, network partner exited**

This message is displayed if the process on the remote node exited before confirming the logical link to your node. The remote process might have exited prematurely, a timeout might have occurred at the remote node, or there might be a problem in the log file on the remote node. You can either retry the operation or try to read the NETSERVER.LOG file in the directory of the account you are attempting to access at the remote node. (DECnet-VAX automatically creates a NETSERVER.LOG file and places it in the directory for the appropriate account when it receives a connect request.)

- **SYSTEM-F-UNREACHABLE, remote node is not currently reachable**

This message is displayed when you attempt to connect to a node that is unreachable. You can try to access the remote node again at a later time.

The message is also displayed even if the remote node does not exist, as long as you have indicated a node address or a node name that you previously defined in your node database.

You also receive notice that the node is unreachable if the value of the executor parameter **MAXIMUM ADDRESS** in your network database is lower than the address of the remote node you are attempting to access. Increase the value of the NCP executor parameter **MAXIMUM ADDRESS** in your database to be at least as high as the highest address of any node that you want to contact.

- **SYSTEM-F-INVLOGIN, login information invalid at remote node**

This message is displayed if you attempt to access a remote node using an access control string that contains an invalid user name or password, or if you do not specify any access control information and no default DECnet account or proxy account is available at the remote node. Retry the file operation with the correct login information.

- **SYSTEM-F-NOSUCHNODE, remote node is unknown**

This message is displayed if you attempt to enter a command to access a remote node and the remote node represented by **node-id** is not identified in the local volatile database. Verify that the node identifier is correct, enter the node name in your node database, and retry the operation.

- **SYSTEM-F-PATHLOST, path to network partner lost**

This message is displayed if you logged in to another node over the network (for example, using the DCL command **SET HOST**) and the path to the remote node is lost.

The path might be lost because of too much network activity or communications problems, or because DECnet was turned off at the remote node. Wait, then check to see if the node is still reachable. If so, try again to log in.

- **SYSTEM-F-SHUT, remote node no longer accepting connects**

This message is displayed if you attempt to access the remote node using a DCL command (such as the **SET HOST** command) under either of these conditions:

- The executor parameter **DEFAULT ACCESS** on the remote node has been set to **NONE**. The default access at the remote node must be set to permit incoming and outgoing access before you can connect to the node.
- The command **SET EXECUTOR STATE SHUT** was executed on the remote system. The network must be restarted on the remote node.

- **SYSTEM-F-NOLINKS, maximum network logical links exceeded**

This message is displayed if the maximum number of links that the remote node allows has been exceeded. Wait and try again later.

- **SYSTEM-F-NOSUCHOBJ, network object unknown at remote node**

This message is displayed if you attempt to access a network object at a remote node and the object is not specified in the remote node database. For example, if you attempt to use the Phone Utility to reach a node that does not have an entry for the network object PHONE in its configuration database, you receive this message.

### 6.3.2.2 Problems Related to Network Operation

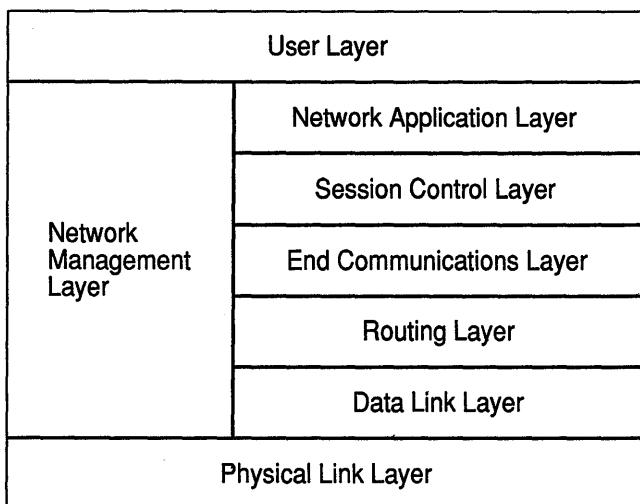
Problems in maintaining the proper functioning of the running network can be difficult to resolve. This section describes the technique for isolating a problem to a particular DECnet software functional layer or layers. As system manager of the local node, you might want to consult with the network manager (if one is available for your network) as necessary to resolve these problems.

#### Troubleshooting Techniques Based on DNA Layers

Techniques for troubleshooting DECnet-VAX problems are based on the layered network design of DECnet-VAX as specified in the Digital Network Architecture (DNA). The DNA layers are illustrated in Figure 6-1. Each layer performs particular services as part of the overall network capability provided at the node.

During troubleshooting, it is useful to distinguish among the network layers in localizing the cause of a particular problem. For example, some problems are characteristic of the Data Link layer, while others are related to the Routing layer or to the End Communications layer (which provides logical link services).



**Figure 6-1: DECnet-VAX Software Design as Based on DNA Layers**

ZK-6364-GE

### Network Problems and Suggested Actions

The following discussion of network difficulties identifies typical problems originating at the various layers, and it describes actions you can take to locate the source of the problem. The problems are grouped into those related to data links, routing, and logical links.

**Data link problems.** Inability to reach an active node is a common problem on the network. The problem could be either a data link problem or a routing problem.

To determine whether the problem is a data link problem, examine both the remote node and the circuit. The data link layer causes data to be moved over physical devices, so it affects only adjacent nodes (an adjacent node is connected to the local node by a single physical line). You can learn whether the unreachable node is an adjacent node and whether the node is available by checking with the network manager or the system manager of the unreachable node.

Also check the state of the circuits (the data link protocol causes a circuit to be in the ON-SYNCHRONIZING state). The problem might be with the data link if the circuit does not start up correctly or is up but the adjacent node is not reachable. (Note that circuit startup might also be affected by incorrect setting of the transmit and receive passwords, as described in the following section on routing problems.)

To locate a data link problem, examine the appropriate counters, line and circuit parameters, and network events.

- Use NCP SHOW commands to display the contents of the circuit and line counters to see if they are reporting errors.
- Use NCP SHOW commands to check the values of line and circuit parameters in the network configuration database.
- Then look at the network events DECnet logged for event class 4 (for the routing layer) and event class 5 (for the Data Link layer) to determine whether any events affecting the data link have occurred.

**Routing problems.** Routing layer problems can involve nodes that are not reachable or circuits that are not stable. The circuit might be up and the adjacent node might be reachable, but one or more intermediate nodes (along the communications path) that should be reachable are not.

To isolate such routing layer problems, examine the appropriate counters and passwords, and try to check the nodes along the routing path.

- Check the contents of the node and circuit counters at your node and, if possible, arrange for the node and circuit counters at the remote node to be examined.
- Examine network events logged for event class 4 (for the routing layer).
- Check the settings of the transmit and receive passwords for the local node and the adjacent node to see if they match (these passwords affect circuit startup).
- Finally, you can use NCP commands with the TELL prefix to try to trace the routing path from one node to another, to determine if an intermediate node is down and to examine the parameter values for all nodes on the communications path.

If erratic routing behavior occurs (for example, constant changes in the reachability of nodes, or connection to a node other than the one you expect to reach), check whether two or more nodes with the same node address are connected to the network. If routing seems to be functioning properly, you can look at executor parameters related to routing (such as cost and hops).

**Logical link problems.** The end communications layer, which provides logical link services, can also be the source of common problems. Logical link problems typically show the following symptoms:

- Link timeout
- Network partner exited
- Invalid account
- Problems with performance and response time

In general, for logical link problems, you can examine the following:

- The default DECnet nonprivileged account and directory on the remote node, to determine if they have been created properly.
- Incoming and outgoing timers at both ends of the logical link, to ensure the links are not timing out prematurely because the timers are set too low.
- The accounting log (using the VMS Accounting Utility), to determine whether the correct process was created or whether a correct process exited prematurely.
- The load on the local and remote nodes, to determine whether the load is preventing the link from being created.
- The path over the network to the remote node. If the circuit is an Ethernet circuit, check the line buffer size parameter to ensure the proper setting.
- The Netserver timeouts, by asking someone to examine the NETSERVER.LOG file at the remote node.
- The proxy settings for your node and for the objects being accessed. (To determine the default proxy access setting for your executor node, specify the NCP command SHOW EXECUTOR CHARACTERISTICS. To examine the proxy access setting for network objects, use the NCP command SHOW KNOWN OBJECTS CHARACTERISTICS.)
- The disk quota, to ensure it is sufficient to create the NETSERVER.LOG file.
- The SYS\$LOGIN file, to determine whether the file protection is set to WORLD:READ.

If a logical link connection is unsuccessful, the link might have timed out for one of the following reasons:

- A heavily loaded node can cause creation of a logical link to take a long time.
- Incoming and outgoing timers might be set too low.

To prevent link timeouts, you can reset the executor parameters INCOMING TIMER and OUTGOING TIMER to higher values at both nodes.

A logical link problem can cause the message "network partner exited" to be displayed. This message indicates that the remote node exited before the logical link was established. Check the following:

- The networking load on the nodes at each end of the logical connection
- The accounting log on the remote node
- Netserver timeouts on the remote node

If you receive a message indicating an invalid account, check that you have the proper authorization to make the logical link connection. However, an invalid account condition can also be reported by the message “network partner exited.” Consequently you should try to have someone check the NETSERVER.LOG file on the remote node.

If performance and response time over the logical link become degraded, the cause might be too much traffic on a path to the target node. If you encounter this problem, consult with the network manager.

**Configuration problems.** The main reason for network errors might be improper configuration of the system. Check your DECnet-VAX configuration, and check the communications cables and connections.

### 6.3.2.3 Asynchronous Connection Problems

Attempts to establish asynchronous DECnet connections with other nodes can fail for a variety of reasons. This section describes some reasons why you might fail to make a static or dynamic connection.

A static asynchronous connection has failed if the static asynchronous DECnet line is started but remains in the ON-STARTING state. To isolate the cause of the problem, check whether the following conditions exist:

- Are the line speeds at both ends of the connection set to the same value?
- If you are using a dialup line, is the modem characteristic set on the terminal? (This must be done before the line is set to asynchronous DDCMP use.)
- Are the two nodes being connected located in the same area in the network (that is, do their node addresses have the same area number) or are both nodes area routers?
- Is the parity on the asynchronous DECnet line set to NONE? If your system is not a VMS system, is the terminal line set to the correct parity?
- Is the terminal line set up to use 8-bit characters?
- If the node already has an active circuit, is the node a routing node?
- If verification is enabled for the circuit, do the passwords set at the two nodes match?

If you are unsuccessful in setting up your terminal line as a static asynchronous DDCMP line, check the following:

- Is the /NOTYPE\_AHEAD qualifier specified for your terminal before you attempt to set up the static asynchronous line? If a type-ahead buffer is associated with your terminal, you might not be able to bring up your terminal line as an asynchronous DECnet line until you stop any process started at the remote node that might own your terminal line.

If dynamic switching is being performed and the asynchronous DECnet connection is not made, first check the following:

- Is DECnet started on both nodes?
- Is the asynchronous DDCMP class driver (NODRIVER) loaded by means of SYS\$SYSTEM:SYSGEN at each node?
- Is the dynamic switch image (DYN SWITCH) installed by means of SYS\$SYSTEM:INSTALL at each node?
- Are virtual terminals enabled on the remote node and, in particular, for the terminal over which you are logged in to the remote node?

If the dynamic asynchronous lines are started but are left in the "ON-STARTING" state, make the following checks:

- Are the two nodes that are being connected located in the same area (that is, do their addresses have the same area number) or are they both area routers?
- Are the routing initialization passwords (transmit and receive passwords) set appropriately at each node?
- Is the INBOUND parameter for the initiating node set correctly in the node database at the node receiving the connection request?
- Is the parity on the asynchronous DECnet line set to NONE? If your system is not a VMS system, is the terminal line set to the correct parity?
- Is the terminal line at the remote node set up to use 8-bit characters?
- If the node already has an active circuit, is the node defined as a routing node?

## 6.4 Summary

### What is a Network, and Which Systems Need One?

A **network** is a means of connecting computers, allowing them to share or transfer information or communications. You need a network if you want to communicate with other computers, if your computer is part of a VAXcluster configuration, or if your computer is a workstation running DECwindows.

With the VMS operating system, the connection to a network is made using DECnet-VAX software.

### **Installing DECnet-VAX**

To install the DECnet-VAX software and connect to a network, you should first be sure that your hardware is in place and connected. If your site has a network or cluster manager, you should obtain information from this individual about the configuration for your system for network operations. In particular, you should know:

- The node name and node address for your system.
- The access method for users on remote systems (for example, the access method through which they can send mail to your system). You can either establish individual default accounts for various network objects (for example, MAIL or FAL), or you can establish a single default DECnet account. When you establish individual default accounts for network objects, you add an increased level of network security to your system.

Install DECnet-VAX as described in Section 6.2.2. First obtain and register the DECnet-VAX PAK using the VMS License Management Utility. Then configure your network either manually or automatically with the NETCONFIG.COM procedure. If you plan to use an asynchronous DECnet connection, you should follow the procedure described in Section 6.2.3.2. Finally, verify that you are properly connected to the network.

### **Installing DECnet-VAX on Certain Workstations**

If you manage a workstation that uses DECwindows and is neither part of a cluster nor connected to a network, then you can save time by installing DECnet-VAX according to the procedure described in Section 6.2.3.4.

### **Starting DECnet-VAX at System Startup**

You must start the DECnet-VAX software each time you boot your system. The easiest way to do this is to execute the procedure SYS\$MANAGER:STARTNET.COM in your site-specific system startup file, SYS\$MANAGER:SYSTARTUP\_V5.COM.

### **NCP Utility**

The Network Control Program (NCP) is the primary system management tool for configuring, controlling, and monitoring the network. In addition to the NCP information discussed in this chapter, the Reference Section includes information about NCP commands and qualifiers.

# Chapter 7

## Setting Up a Local Area VAXcluster Environment

This chapter discusses how to set up a small local area VAXcluster configuration. For the purposes of this manual, a small local area VAXcluster configuration consists of one computer called a **boot server** that serves as the hub of the cluster, and one or more MicroVAX or VAXstation computers that are connected to the boot server. If you want to learn how to set up this type of VAXcluster configuration, then you should read the rest of this chapter. If you manage a VAXcluster environment other than the type described in this chapter, then you should refer to the *VMS VAXcluster Manual*.

This chapter also briefly describes the use of the System Management Utility (SYSMAN) in a VAXcluster environment. With SYSMAN, you can execute commands on all nodes in a cluster from a single node.

### 7.1 What Is a Cluster?

A **cluster** is a group of two or more computers that share some or all of their resources. When a group of VAX computers shares resources in a VAXcluster environment, the storage and computing resources of all of the computers are combined, which can increase the processing capability, communications, and availability of your computing system.

#### 7.1.1 VAXcluster Types

Three types of VAXcluster configurations are possible:

- Local Area VAXcluster configuration
- CI-only VAXcluster configuration
- Mixed-interconnect VAXcluster configuration

### Local Area VAXcluster Configuration

A Local Area VAXcluster configuration is made up of a single VAX computer that serves as the management center of the cluster, plus one or more VAX computers that are connected to this hub. A local area VAXcluster configuration always includes the following parts:

- **Boot server**

A boot server is a VAX or MicroVAX computer, and it serves as the management center of a Local Area VAXcluster environment. The system disk of the boot server contains management files for the entire cluster, including startup files and user authorization information. The boot server must be available and running for the cluster to operate.

Boot servers should be the most powerful machines in the cluster. They should also use the highest bandwidth Ethernet adapters available. You can use any VAX or MicroVAX system supported with VMS Version 5.2 except VAX-11/730 as a boot server.

Refer to the VAXcluster Software Product Description for complete information about supported configurations.

- **Satellite nodes**

A satellite node is a MicroVAX computer that is a member of the cluster. A computer becomes a satellite node when the `CLUSTER_CONFIG.COM` procedure is executed from the boot server to add the computer to the cluster.

### CI-only VAXcluster Configuration

A CI-only VAXcluster configuration is a cluster in which only the computer interconnect is used for communications between the computers in the cluster. In a CI-only VAXcluster configuration, the star coupler is used as the common connection point for all nodes in the cluster, including both VAX computers and Hierarchical Storage Controllers (HSCs).

Nodes in a CI-only VAXcluster configuration can be either of the following:

- VAX computers listed in the VAXcluster SPD
- HSCs

### Mixed-Interconnect VAXcluster Configuration

A **mixed-interconnect** cluster can include both CI-connected VAX computers and MicroVAX systems.

This chapter concentrates on setting up a Local Area VAXcluster configuration with a single boot server. Although some of the management tasks for other VAXcluster types are similar, you should refer to the VAXcluster documentation that is available in the full VMS documentation set for information about managing a CI-only or mixed-interconnect cluster.



## 7.2 Shared Resources

A major benefit of a VAXcluster configuration is the ability to share resources. A **shared resource** is a resource (such as a disk or a queue) that can be accessed and used by any node in a cluster. Data files, application programs, and printers are just a few items that can be accessed by users on a cluster with shared resources, without regard to the particular node on which the files or program or printer might physically reside.

When disks are set up as shared resources in a VAXcluster environment, users have the same environment (password, privileges, access to default login disks, and so on) regardless of the node that is used for logging in. You can realize a more efficient use of mass storage with shared disks, because the information on any device can be used by more than one node—the information does not have to be rewritten in many places.

Print and batch queues can also be set up as shared resources. In a VAXcluster configuration with shared print and batch queues, a single job controller queue file manages the queues for all nodes on the cluster. The job controller file makes the queues available from any node. For example, suppose your VAXcluster configuration has fully shared resources and includes nodes ALBANY, BASEL, and CAIRO. A user logged in to node ALBANY can send a file that physically resides on node BASEL to a printer that is physically connected to node CAIRO, and the user never has to specify (or even know) the nodes for either the file or the printer. For more information about setting up and using print and batch queues in a VAXcluster environment, see Chapter 5.

## 7.3 Preparing a System for a Local Area VAXcluster Environment

In a VAXcluster environment with a single system disk, you need to install the VMS operating system only once, regardless of the number of nodes in the cluster.

To install the operating system, follow the instructions in your computer's installation guide. Before beginning the installation procedure, you must determine the configuration type for your cluster (CI-only, local area, or mixed-interconnect). During the installation of the operating system, you will be asked a series of questions. Table 7-1 lists the questions and answers for Local Area VAXcluster configurations.

**NOTE:** While rebooting at the end of the installation procedure, the system displays messages warning that you must install required licenses. Be sure to install these licenses, as well as the DECnet-VAX license, as soon as the system is available. Procedures for installing the licenses are described in the release notes distributed with the software kit.

## 7-4 Setting Up a Local Area VAXcluster Environment

**Table 7-1: Installation Questions for Local Area VAXcluster Configurations**

Question	Response
Will this node be a cluster member (Y/N)?	Enter Y.
What is the node's DECnet node name?	Enter DECnet node name—For example, ALBANY. The DECnet node name can be from 1 to 6 alphanumeric characters in length and cannot include dollar signs or underscores.
What is the node's DECnet node address?	Enter DECnet node address—For example, 2.2.
Will the Ethernet be used for cluster communications (Y/N)?	Enter Y. The Ethernet is required for cluster (SCS internode) communications in local area configurations.
Enter this cluster's group number:	Enter a number in the range from 1 to 4095 or 61440 to 65535.
Enter this cluster's password:	Enter the cluster password. The password must be from 1 to 31 alphanumeric characters in length and can include dollar signs and underscores.
Re-enter this cluster's password for verification:	Re-enter the password.
Will ALBANY be a disk server (Y/N)?	Enter Y. In local area configurations, the system disk is always served to the cluster.
Will ALBANY serve HSC disks (Y/N)?	Enter N.
Enter a value for ALBANY's ALLOCLASS parameter:	Enter a value of 0 for Local Area VAXcluster configurations covered by this manual.
Does this cluster contain a quorum disk [N]?	Enter N for Local Area VAXcluster configurations covered by this manual.

### 7.3.1 Building a VAXcluster Configuration

Once you have installed the VMS operating system, you can start to build your cluster. This section describes how to build a simple Local Area VAXcluster configuration using the command procedure `SYSS$MANAGER:CLUSTER_CONFIG.COM`. If you find that your cluster configuration is more complex than the type described in this manual, be sure to consult the *VMS VAXcluster Manual* in the Extended VMS Documentation Set.

The command procedure `CLUSTER_CONFIG.COM` is the primary tool that you use for adding a node to your VAXcluster configuration, removing a node from the cluster, or changing the characteristics of a node. This section describes how to use `CLUSTER_CONFIG.COM` to add or remove a satellite node in a Local Area VAXcluster configuration.

## Using CLUSTER\_CONFIG.COM

Before using CLUSTER\_CONFIG.COM, log in to the SYSTEM account on the system that will be your boot server and make sure that DECnet-VAX is up and running. Be sure that your default is set to SYS\$MANAGER; then enter the following command:

```
$ @CLUSTER_CONFIG
```

### 7.3.1.1 Setting Up the Boot Server

The first step in setting up your local area cluster for the first time is to establish the boot server. You must establish the local system as a boot server before you can add any satellites to the cluster.

To establish a node as a boot server, run CLUSTER\_CONFIG.COM and select the CHANGE option from the menu. Then, select the option to enable the local system as a boot server.

### 7.3.1.2 Adding Satellite Nodes

To add satellite nodes to your Local Area VAXcluster configuration, you use the ADD option from the CLUSTER\_CONFIG.COM menu. When you execute CLUSTER\_CONFIG.COM to add a satellite node, you are asked a series of questions for which the command procedure supplies most default values. For local area clusters that are the subject of this manual, the default values are sufficient. (If your cluster has special requirements and you want to learn more about values other than the defaults, you should consult the VAXcluster documentation in the Extended VMS Documentation Set.)

There are some values that you must supply. These include the following:

- DECnet node name and node address for each satellite—The node name has up to 6 alphanumeric characters. The node address should be supplied by your network manager.
- Satellite's Ethernet hardware address—The Ethernet hardware address has the form xx-xx-xx-xx-xx-xx. You must include the hyphens when specifying the hardware address.

To obtain the Ethernet hardware address for MicroVAX II and VAXstation II satellites, enter the following commands at the satellite's console:

```
>>> B/100 XQ
Bootfile: READ_ADDR
```

For MicroVAX 2000 and VAXstation 2000 satellites, enter the following commands at successive console-mode prompts:

```
>>> T 53
2 ?>>> 3
>>> B/100 ES
Bootfile: READ_ADDR
```

## 7-6 Setting Up a Local Area VAXcluster Environment

(In this example, if the second prompt appears as 3 ?>>>, press RETURN.)

For 3xxx series satellites, enter the following command at the satellite's console:

```
>>> SHOW ETHERNET
```

- Workstation windowing system—The windowing system (for example, DECwindows) if your satellite is a workstation.

### 7.4 DECnet-VAX Connections

In any cluster configuration, DECnet-VAX connections are required for all computer nodes. Use of DECnet-VAX facilities ensures that cluster managers can access each node in the cluster from a single terminal, even if terminal-switching facilities are not available.

In local area clusters, DECnet is required both for system management functions and intercomputer communication. For example, DECnet is used for remote booting operations (downline loading of satellite nodes).

In these configurations, DECnet and System Communication Services coexist on the same Ethernet. They share the same data link and physical link protocols, which are implemented by the Ethernet data link drivers, the Ethernet adapters, and the Ethernet itself.

### 7.5 Clusterwide Tasks Using SYSMAN

In a VAXcluster environment, you are sometimes required to perform the same task for each node in the cluster. For example, if you want an executable image that exists on one node to be available as a shared image to users on any node of the cluster, the file would have to be installed as a shared image on each node.

With the System Management Utility (SYSMAN), you can perform system management tasks for all (or some subset of) the nodes on a cluster from one node within the cluster. This section describes some of the system management tasks that you can perform in a Local Area VAXcluster environment with SYSMAN. A description of SYSMAN commands and qualifiers is located in the Reference Section. If you need more information about the complete capabilities of SYSMAN, see the *VMS SYSMAN Utility Manual* in the extended documentation set.

#### 7.5.1 Setting a Clusterwide Environment

SYSMAN is an interactive utility. To use SYSMAN, your process needs OPER privilege on the node where you are logged in, and you must also have authorization for any required privileges on remote nodes where you want to perform tasks using SYSMAN. To run SYSMAN, enter the following command:

```
$ RUN SYS$SYSTEM:SYSMAN  
SYSMAN>
```

To use SYSMAN to apply system management tasks to each node in the cluster, you should first use the SET ENVIRONMENT/CLUSTER command, as follows:

```
SYSMAN> SET ENVIRONMENT /CLUSTER
```

After you enter this command, any subsequent commands that you enter in the SYSMAN Utility will be applied to each node in your cluster. If you want the SYSMAN commands to apply only to certain nodes within the cluster, use the /NODE= qualifier with the SET ENVIRONMENT/CLUSTER command. For example, if you want SYSMAN commands applied only to nodes GREEN and ORANGE in your VAXcluster environment, you would use the following command:

```
SYSMAN> SET ENVIRONMENT /CLUSTER /NODE=(GREEN,ORANGE)
```

Most system management tasks require some privileges, and you must enable the appropriate privileges before you can perform the tasks using SYSMAN. To enable privileges, use the SET PROFILE/PRIVILEGES command in SYSMAN. For example, if you need CMKRNL privilege to perform a task, use the following command sequence:

```
SYSMAN> SET ENVIRONMENT/CLUSTER
SYSMAN> SET PROFILE/PRIVILEGE=CMKRNL
```

You must be authorized to set your privileges on each node that is identified by the SET ENVIRONMENT command.

## 7.5.2 Executing Commands on a Cluster

After setting the environment to the entire cluster, any SYSMAN commands you execute are applied to each node in the VAXcluster configuration. (If you used the /NODE qualifier, then the SYSMAN commands are applied to the nodes that you defined.)

To execute a DCL command with SYSMAN, use the DO command. For example, to show the batch processes on each node in the system (SHOW SYSTEM/BATCH), you would use the following commands:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT /CLUSTER
SYSMAN> DO SHOW SYSTEM /BATCH
```

The SHOW SYSTEM/BATCH command is then executed on each node in the VAXcluster configuration, and the output is displayed on your terminal.

The following example shows how to install a file as shared. In this example, it is assumed that you have previously used the SET ENVIRONMENT /CLUSTER command.

```
SYSMAN> DO INSTALL ADD/OPEN/SHARED WORK_DISK:[ACCOUNTS]SALES
```

## 7-8 Setting Up a Local Area VAXcluster Environment

This command installs the image SALES.EXE on each node in the current SYSMAN environment. As the SYSMAN command is executed on each node, a confirming message is displayed on your terminal, for example:

```
%SYSMAN-I-OUTPUT, Command execution on node RED
%SYSMAN-I-OUTPUT, Command execution on node GREEN
%SYSMAN-I-OUTPUT, Command execution on node BLUE
.
.
.
```

It can also be helpful to use SYSMAN to add or modify the user authorization file for each node. For example, if you want to add user JONES to the system, and you want this user to be able log in to each node in the cluster, you could give the following command in SYSMAN:

```
SYSMAN> DO MCR AUTHORIZE ADD JONES /PASSWORD=FANCIULLA /DEVICE=WORK_DISK -
_SYSMAN> /DIRECTORY=[JONES]
```

## 7.6 Summary

### What is a Cluster?

A cluster is a group of two or more computers that share some or all of their computing and storage resources. VAXcluster configurations can be of the following types:

- Local Area VAXcluster configuration
- CI-only VAXcluster configuration
- Mixed-interconnect VAXcluster configuration

A Local Area VAXcluster configuration has a one VAX computer that serves as the management center of the cluster, plus one or more VAX computers that are connected to the central computer. The central computer in a local area cluster is called the **boot server**, and the other computers in the local area cluster are called **satellite nodes**.

This chapter concentrates only Local Area VAXcluster configurations that use a single boot node.

### Sharing Resources in a VAXcluster Environment

A **shared resource** is a resource (such as a disk or a queue) that can be accessed and used by any node in a cluster. Data files, application programs, and printers are just a few items that can be accessed by users on a cluster with shared resources, without regard to the particular node on which the files or program or printer might physically reside.

### **Creating a Local Area VAXcluster Environment**

To create a Local Area VAXcluster environment, follow these steps:

1. Determine the VAX computer that you want to be your boot server. The boot server should be the most powerful machines in the cluster and should use the highest bandwidth Ethernet adapters available. You can use any VAX or MicroVAX system supported with VMS Version 5.2 except VAX-11/730 as a boot server.
2. Be sure that DECnet-VAX and the appropriate communications hardware are properly installed on each system that is to become a member of the cluster.
3. Install the VMS operating system on the boot server. During the installation process, respond appropriately to the questions that establish the node as a member of a Local Area VAXcluster configuration, as described in Section 7.3.
4. After installing the operating system on the boot server, establish the boot server and add satellite nodes to your Local Area VAXcluster by using the CLUSTER\_CONFIG command procedure, as described in Section 7.3.1.

### **Using SYSMAN in a VAXcluster Environment**

The System Management Utility (SYSMAN) lets you execute commands on all (or some) nodes of a cluster by using just a single command on one node. Use SYSMAN to install software that can be used from any node in the cluster, or to execute any command on all cluster nodes.





# Chapter 8

## Backing Up and Restoring Files

The files on your system contain important information, and it could be damaging to your operation if some or all of the files were lost or otherwise became unusable. If a user inadvertently deletes key files, or if an equipment failure corrupts some files and makes them unusable, the loss of such files could have a severe impact.

As system manager, you can reduce much of the potential impact due to lost or unusable files by saving copies of files on a regular basis and keeping these copies in a safe place. Then, in the case of an unexpected event in which some or all files are lost or unusable, you can restore the files using the copies that you saved.

You use the `BACKUP` command to save copies of files. Saving files with Backup should be part of a system manager's routine tasks; you can use simple command procedures to perform daily backups of all the files on your system.

This chapter describes how to back up your files and how to recover them if needed. Regular backups should be performed on all systems, even when there is only a single user, so all system managers should know about the information described in this chapter.

**NOTE:** With Backup, files can be restored only to their state at the time of the most recent backup. If there are some files for which you might not be able to afford losing even a day's worth of data—for example, critical data files that are updated frequently—then you might also want to use some optional software, such as VAX RMS Journaling or VMS Volume Shadowing, that can be used to avoid the loss of data. To learn more about these products, refer to the documentation sets for them or contact your Digital representative.

## 8.1 Making Backup Copies of Files

Backup copies should exist for every file on your system. By making regular backup copies of these files, you can restore some or all of them if needed. The frequency with which you make backups depends on the conditions and file usage at your site. Backups are usually performed for entire disks (or volumes), although you can back up individual files or directories.

In general, the more frequently files are updated, the more frequently you should make backups. For example, daily backups would be appropriate for disks containing work files that are modified as a regular part of your operation. On the other hand, less frequent backups might be appropriate for disks containing only files that tend to be modified very infrequently.

The media on which copies of files can be saved are magnetic tapes, disks, or diskettes. The media that you use depends upon your hardware configuration and the needs of your site. Magnetic tape is the most commonly used medium for storing files that have been backed up. Magnetic tape is relatively inexpensive, and you can store it easily. Disks have the advantage of being faster for both saving and restoring files. On some systems, diskettes are used for backups because they are the only removable media.

Be sure to keep your backup media in a safe place, away from your system. If an event that corrupts the data on your system's disks also corrupts your backup media, you will be unable to restore the files.

### 8.1.1 Image (Full) and Incremental Backups

There are two general types of backups that save copies of all the files on your system, *image (full) backups* and *incremental backups*:

- **Image Backups**—An image backup (also called a full backup) saves a copy of all the files on a disk (or volume); the image backup is a logical duplicate of the contents of the disk. The first backup that you make on your system must be an image backup.
- **Incremental Backups**—An incremental backup saves only those files that have been created or modified since the most recent backup.

Additionally, you can make file-by-file save operations. In a file-by-file save operation, you back up individual files (or directories).

When choosing between image and incremental backups, you should consider the advantages and disadvantages of each. An incremental backup takes less time and uses less space than an image backup. However, if you need to restore files from the backup copies that you have saved, it is easier and faster to restore copies when your most recent backup was an image save operation. If you choose to use incremental backup, remember that you must also make periodic image backups of your files.

Image backups are most convenient to perform when there are no interactive users on the system. Depending on your system resources, system performance can be affected during the backup process, so it is best to schedule the backup during the least busy times for your system. You can optimize the speed of the backup procedure by ensuring that certain process and system parameters are properly set. See Section 8.2 for information about setting these values.

If all of the files can fit on a single piece of storage media, you do not have to change tapes or disks during the backup. In this case, the backup can be executed by a batch job that is scheduled to run late at night (or at some other time when interactive use of the system is likely to be at a minimum).

When you perform backups, either image or incremental, it is best to notify users that a backup is about to take place. Notify users with the `REPLY/ALL` command, as follows:

```
$ REPLY/ALL "System Backup About To Begin -- Open Files Will Not Be Backed Up
```

When you give this command, the following is displayed on the terminal of every interactive session on the system:

```
Reply received on MYNODE from user SYSTEM at VTA28: 23:35:11
System Backup About to Begin---Open Files Will Not Be Backed Up
```

**OPEN FILES DURING A BACKUP:** When a file is open for writing during a backup, the file is not copied and saved during the backup procedure. However, you can instruct the backup procedure to save these open files by using the `/IGNORE=INTERLOCK` qualifier on the `BACKUP` command. When you use the `/IGNORE=INTERLOCK` qualifier, the contents of the file at the moment of the backup are saved.

The `/IGNORE=INTERLOCK` qualifier can be most useful for files that are constantly open (and would therefore not otherwise be saved). Be careful, though, to recognize that you might be saving inconsistent data, depending on the applications that are writing to the open files. In general, it is best to back up your system when a minimum number of files are open.

### 8.1.2 Save Sets

A **save set** is a file created and used by Backup when you use the `BACKUP` command to save files. The save set, which is written in a format that only Backup can interpret, includes the files that you save with Backup and other information that is used by Backup.

### 8.1.3 Using the BACKUP Command to Save Files

You can use the BACKUP command at DCL level, and you can also use it in a command procedure. Additional information about the BACKUP command is found in the Reference Section of this manual.

The BACKUP command line has three key parts:

BACKUP/qualifiers input\_specifier/qualifiers output\_specifier/qualifiers

1. The BACKUP command itself and its qualifiers—The first part of the BACKUP command line is the word *BACKUP* and those qualifiers that apply to the BACKUP command.
2. The input specifier and its qualifiers—The second part of the BACKUP command line identifies and gives information about the disk and files that currently exist. When you use the BACKUP command to save files, this part of the command line gives information about the files that you want to copy.
3. The output specifier and its qualifiers—The third and final part of the BACKUP command line identifies and gives information about the destination of the files that are to be backed up. When you use the BACKUP command to save files, this part of the command line identifies the tape drive (or disk or diskette) to which the files are to be copied.

### 8.1.4 Making Image Backups of a Disk

To make an image backup that copies all the files on a disk to a magnetic tape, do the following:

- Use the BACKUP command with the /IMAGE and /RECORD qualifiers as first part of the command line. The /IMAGE qualifier identifies the backup operation as an image backup. The /RECORD qualifier provides information for Backup to use when subsequent incremental backups are taken. (If you take an incremental backup, Backup saves only those files that have been created or modified since the most recent BACKUP/RECORD command.)
- Give either the input device name followed by a colon or the volume name as the second part of the command line.
- As the third part of the command line, give the name of the output tape device, followed by a colon and the name of the save set that you want to use. The save set is actually a file, and it must have a file name. When the save set is on magnetic tape, the file name can have no more than 17 characters (including the period delimiter and file type); when the save set is on disk or diskette, it follows the standard VMS file naming rules. Use the /REWIND qualifier to initialize and rewind the tape before the backup procedure begins.

When you **initialize** a tape, the tape is given a **volume label**. A **volume label** is an identifier of one to six characters and is included as part of the header information on the tape. You can use the `/LABEL=` qualifier to choose a volume label of up to six characters; alternatively, the first six characters of the save set name are used to form the volume label if the `/LABEL` qualifier is not used. (For example, if your save set is named `19JUNE1990.SAVE`, then the volume label is `19JUNE` if you do not use the `/LABEL` qualifier.) Backup uses the volume label to ensure that you do not create a save set on the wrong magnetic tape, thus unintentionally overwriting existing data.

For example, suppose you want to save all the files on a disk named `DRA1:` to a magnetic tape on the device named `MTA0:`. The following command line creates a save set named `19JUNE1990.SAV` on the tape that is in `MTA0:`, and that save set will contain all the files on `DRA1:`. Before the files are copied, the tape in `MTA0:` is rewound.

```
$ BACKUP/IMAGE/RECORD DRA1: MTA0:19JUNE1990.SAV/REWIND
```

With this command line, you initialize the tape (with the `/REWIND` command), and the tape has a volume label of `19JUNE` (the first six characters of the save set name).

### 8.1.5 Making Incremental Backups of a Disk

Incremental backups save only those files that have been created or modified since the last image or incremental backup in which the `/RECORD` qualifier was used. Incremental backups can be done more quickly than image backups, because fewer files are saved. Remember, however, that it takes longer to restore an entire disk when one or more incremental backups have been made since the most recent image backup.

To make an incremental backup, use the `/SINCE=BACKUP` qualifier with the `BACKUP` command. The syntax for incremental backups is the same as for image backups; however, you cannot use the `IMAGE` qualifier (because that qualifier specifies an image backup). Also, when you make an incremental backup, you can specify that only certain files be saved, and you can also use input-specifier qualifiers. (See the Reference Section for information about all the qualifiers that you can use with input-specifiers.)

For example, suppose that you had used the command line shown in Section 8.1.4 for an image backup, and you now wanted to make an incremental backup. The following command line makes an incremental backup. It saves a copy of all files on `DRA1:` that were modified since the previous `BACKUP/RECORD` command, storing them in a save set named `20JUNE1990.SAV`.

```
$ BACKUP/RECORD/SINCE=BACKUP DRA1:[*...] MTA0:20JUNE1990.SAV/LABEL=20JUNE
```

In this example, the magnetic tape used for the previous, image backup is used, and it is necessary to identify the volume label using the /LABEL qualifier with the output specifier. Also, because Backup was performing an incremental rather than an image backup, it is necessary to explicitly use the notation DRA1:[\* . . . ] to specify all the files on DRA1.

## 8.1.6 Using Command Procedures for Backups

By using some simple command procedures, you can be sure that your system backups take place when you want them to. The following sections give examples of command procedures for some specific situations.

If you are not familiar with using command procedures, you should read the chapter about command procedures in the *VMS User's Manual*.

### 8.1.6.1 Command Procedure for Nightly Image Backup

The following command procedure performs nightly image backups, backing up all the files on disk DUA0:. The files are copied to a save set on magnetic tape, and the save set is called FULL\_BACKUP.SAV. This procedure is particularly useful for backing up files on a MicroVAX or workstation.

In addition to backing up the files, the command procedure automatically resubmits itself at 2:00 the following morning. Therefore, you need to submit the command procedure only once, and it will execute daily at 2:00 a.m. However, you must physically load a tape each day or else the backup procedure will fail. Even if the backup procedure fails, however, the command procedure will continue to resubmit itself. Also, you must of course have a batch queue available on your system. (See Chapter 5 for information about setting up a batch queue on your system.)

To use the following command procedures, you should follow this process:

1. Create the command procedure as shown in the SYS\$MANAGER directory ([SYSTEM]), and call it SYSTEM\_BACKUP.COM. Edit the command procedure to reflect the name of the disk or disks you want to back up, the name of the tape drive you will use, and the name of the save set you want to assign. The example uses a save set named FULL\_BACKUP.SAV.
2. Write down the name of the save set that you assigned.
3. Submit the command procedure using the following command line. (If you gave your procedure a file name other than SYS\$MANAGER:SYSTEM\_BACKUP.COM, substitute the appropriate file name.)
 

```
$ SUBMIT /AFTER="TOMORROW+2:0" SYS$MANAGER:SYSTEM_BACKUP
```
4. Be sure that a tape is physically loaded on the device that you specified. When the backup is complete, keep the tape in a safe place and do not use the tape again until you next make another image backup of your system.

```

$!
$! Resubmit this procedure --
$ SUBMIT/AFTER="TOMORROW+2:0" SYS$MANAGER:SYSTEM_BACKUP
$!
$ SET NOON
$ ON ERROR THEN GOTO DONE
$ ON CONTROL_Y THEN GOTO DONE
$ SET PROC/PRIV=ALL
$!
$ REPLY/ALL -
  "Full System Backup About to Begin. Open Files Will Not Be Saved"
$!
$ BACKUP /IMAGE DUA0: MUA0:FULL_BACKUP.SAV /REWIND
$!
$ WRITE SYS$OUTPUT "---> Completed backup of DUA0 save set"
$ WRITE SYS$OUTPUT ""
$!
$DONE:
$ DISMOUNT MUA0:
$ EXIT

```

If you want to use a similar procedure to back up files for a system that has more than one disk, you can list each of the devices in the **BACKUP** command line. For example, if you want to back up all files on disks with logical names **WORK\_DISK**, **SYSTEM\_DISK**, and **PAYROLL\_DISK** to a tape drive **MTA0**, substitute the following lines for the **BACKUP** command line in the preceding example:

```

.
.
.
$!
$ BACKUP /IMAGE -
  WORK_DISK, SYSTEM_DISK, PAYROLL_DISK -
  MTA0:SYSTEM_BACKUP.SAV /REWIND
$!
.
.
.

```

If you plan to subsequently perform any incremental backups, you should include the **/RECORD** qualifier in your command line. The **/RECORD** qualifier records the date and time of the backup; Backup uses this information when subsequent incremental backups are taken. When a subsequent incremental backup is made, Backup saves only those files that have been created or modified since the most recent **BACKUP/RECORD** command.

In the event that you want to discontinue using this procedure after you submit it, use the **DELETE/ENTRY** command. To find the entry number, use the **SHOW QUEUE SYS\$BATCH** command. For example, the following sequence shows how to determine the entry number and then delete the batch job:

## 8-8 Backing Up and Restoring Files

```
$ SHOW QUEUE SYS$BATCH
```

```
Batch queue SYS$BATCH
```

Jobname	Username	Entry	Status
-----	-----	-----	-----
SYSTEM_BACKUP	SYSTEM	583	Holding until 19-APR-1990 02:00

```
$ DELETE /entry=583  
$
```

### 8.1.6.2 Command Procedure for Nightly Incremental Backup

You can use a similar command procedure to perform nightly incremental backups of your system. It might be more convenient to perform nightly incremental backups and weekly image backups if either of the following conditions apply:

- Interactive users are on your system at all times of the day and night, and system performance is noticeably affected by backups.
- A full system backup does not fit on a single magnetic tape, but an incremental does fit on a single tape. In this case, an image backup requires the presence of an operator (to change the tape), whereas an incremental backup can be run as a batch job.

Suppose that you take your image system backup on Friday night, and that you want to take incremental backups on every other night at 11:00 p.m. The following command procedure executes an incremental backup on three disks and automatically resubmits itself to run again the following night, except for Friday night.

To use this procedure, follow these steps:

1. Create the command procedure as shown, and call it `INCREMENTAL_BACKUP.COM`. Edit the procedure to reflect the names of the disk or disks that you want to back up, the name of the tape drive that you will use, the volume label of the tape, the name that you want to assign to the save set, and the day of the week (if any) to be omitted in the incremental backup). (In this example, the disks to be backed up are `DRA0`, `DRA1`, and `DRA2`; the tape drive is `MTA0`, the volume label is `INCREM`, the save set is named `INCREMENT.SAV`, and the incremental backup will not be performed on Friday. This example assumes that an image (full) backup is performed on Friday nights.)
2. Be sure that an image backup of the system has been made, and also be sure that you continue to make regular image backups of the system. When you make your image backups, be sure to use the `/RECORD` qualifier (as well as the `/IMAGE` qualifier) in your `BACKUP` command line.
3. Submit the command procedure using the following command line. (If you gave your procedure a file name other than `SYS$MANAGER:INCREMENTAL_BACKUP.COM`, substitute the appropriate file name.)

```
$ SUBMIT /AFTER=23 SYS$MANAGER:INCREMENTAL_BACKUP
```



4. Be sure that a tape is physically loaded on the device that you specified. When the incremental backup is complete, keep the tape in a safe place and do not use the tape again until you make another image backup of your system.

```

$!
$! Resubmit this procedure --
$ SUBMIT/AFTER="TOMORROW+23:0" SYS$MANAGER:INCREMENTAL_BACKUP
$!
$ TODAY = f$cvtime("today",,"weekday")
$ IF TODAY .EQS. "Friday" THEN GOTO DONE
$!
$ SET NOON
$ ON ERROR THEN GOTO DONE
$ ON CONTROL_Y THEN GOTO DONE
$ SET PROC/PRIV=(OPER,BYPASS)
$!
$ REPLY/ALL -
    "Incremental Backup About to Begin.  Open Files Will Not Be Saved"
$!
$ BACKUP/RECORD/SINCE=BACKUP DRA0:,DRA1:,DRA2: -
    MTA0:INCREMENT.SAV /LABEL=INCREM
$!
$ WRITE SYS$OUTPUT "---> Completed backup of save set"
$ WRITE SYS$OUTPUT ""
$!
$DONE:
$ DISMOUNT MTA0:
$ EXIT

```

### 8.1.6.3 Interactive Command Procedure for Backups

The following command procedure can be used interactively to back up a disk to a magnetic tape. After the specified tape drive is allocated, Backup searches the tape's volume header record for a volume label and compares the label you specified with the volume label. If the volume header record contains no volume label, Backup writes the label and expiration date you specified to the volume header record and initializes the tape. Otherwise, Backup compares the tape's volume label with the label you specified and ensures that the tape is expired. If the tape is not expired or the label does not match, the command procedure exits. If the tape is expired and the label matches, Backup writes the expiration date you specified to the volume header record and initializes the tape. After initializing the tape, Backup saves all files in the current default directory tree that have been created or modified since the last save operation to a save set with the name you specified.

## 8-10 Backing Up and Restoring Files

```
$ ! Command procedure DAILYBACK.COM
$ !
$ ! Execute this command procedure interactively,
$ ! by entering the command @[directory]DAILYBACK
$ ! at the DCL prompt.
$ !
$ ! The BACKUP command in this procedure contains the
$ ! output save-set qualifier /REWIND. Therefore, this
$ ! command procedure always initializes the output tape.
$ !
$ ON CONTROL_Y THEN GOTO EXIT
$ ON ERROR THEN GOTO EXIT
$ INQUIRE DRIVE "Enter the drive name (without a colon)"
$ ALLOCATE 'DRIVE'
$ INQUIRE SAVESET_SPEC "Enter the save-set specification"
$ INQUIRE LBL "Enter the tape label"
$ INQUIRE EXP "Enter the tape expiration date"
$ BACKUP/NOASSIST/RECORD/IGNORE=INTERLOCK/SINCE=BACKUP -
[...]'DRIVE':'SAVESET_SPEC'/REWIND/LABEL='LBL'/TAPE_EXPIRATION='EXP'
$ EXIT:
$ DEALLOCATE 'DRIVE'
$ EXIT
```

Digital provides two template command procedures in the SYS\$EXAMPLES directory to assist system managers in designing Backup command procedures. These command procedures are called BACKUSER.COM and RESTUSER.COM.

## 8.2 Preparing Your System for Efficient Backups

You can optimize the efficiency of backups on your system by ensuring that certain parameters are set properly. These parameters include quotas for the process from which backups will be made (that is, the process that enters the BACKUP command, or the process that submits a command procedure), and certain SYSGEN parameters.

Using the Authorize Utility, determine the following quotas for the process that will be used for backups. (Information about the Authorize Utility is located in Chapter 4 and in the Reference Section of this manual.)

```
WSQUOTA
WSEXTENT
PGFLQUO
FILLM
DIOLM
ASTLM
BIOLM
BYTLM
```

The following sequence shows the commands that you would use to run the Authorize Utility and determine process quotas for user SYSTEM. (If you plan to run backups from a different account, determine the process quotas for that account.)

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> SHOW SYSTEM
```

```
Username: SYSTEM                      Owner: SYSTEM MANAGER
Account:  SYSTEM                      UIC:      [1,4] ([SYSTEM])
CLI:     DCL                          Tables: DCLTABLES
Default: SYS$SYSROOT:[SYSMGR]
```

```
Maxjobs:      0  Fillm:      40  Byt1m:      32768
Maxacctjobs:  0  Shrfillm:   0  Pbyt1m:      0
Maxdetach:    0  BIOLm:     18  JTquota:    1024
Prclm:        10 DIOLm:     18  WSdef:      256
Prio:         4  AST1m:     24  WSquo:      512
Queprio:      0  TQELm:    20  WSextent:  2048
CPU:          (none) Enqlm:    200 Pgflquo:  20480
```

```
UAF> EXIT
%UAF-I-NOMODS, no modifications made to system authorization file
%UAF-I-NAFNOMODS, no modifications made to network authorization file
%UAF-I-RDBNOMODS, no modifications made to rights database
$
```

In this example, SYSTEM had the following quotas:

```
WSQUOTA      512
WSEXTENT     2048
PGFLQUO      20480
FILLM        40
DIOLM        18
ASTLM        24
BIOLM        18
BYTLM        32768
```

Now, using the System Generation Utility (SYSGEN), you must also determine the value of the SYSGEN parameters WSMAX and CHANNELCNT, as follows:

```
$ R SYS$SYSTEM:SYSGEN
```

```
SYSGEN> SHOW WSMAX
```

Parameter Name	Current	Default	Minimum	Maximum	Unit	Dynamic
WSMAX	2600	1024	60	100000	Pages	

```
SYSGEN> SHOW CHANNELCNT
```

Parameter Name	Current	Default	Minimum	Maximum	Unit	Dynamic
CHANNELCNT	127	127	21	2047	Channels	

```
SYSGEN> EXIT
$
```

## 8-12 Backing Up and Restoring Files

In this case, the values for WSMAX and CHANNELCNT, as shown in the column marked *Current*, are 2600 and 127, respectively.

Now, compare the values for the process quotas and SYSGEN to the recommended values for the most efficient backup, as shown in Table 8-1:

**Table 8-1: Recommended Process Quotas for Efficient Backups**

Parameter	Facility	Recommended Setting
WSQUOTA	Authorize	Equal to SYSGEN parameter WSMAX
WSEXTENT	Authorize	Equal to WSQUOTA
PGFLQUO	Authorize	At least as large as WSEXTENT
FILLM	Authorize	Equal to SYSGEN parameter CHANNELCNT
DIOLM	Authorize	Either 4096 or three times the value of FILLM, whichever is <i>greater</i>
ASTLM	Authorize	Either 4096 or three times the value of FILLM, whichever is <i>greater</i>
BIOLM	Authorize	Less than or equal to FILLM
BYTLM	Authorize	Greater than or equal to the following value: (256 * FILLM) + (6 * DIOLM)
ENQLM	Authorize	Greater than FILLM

Table 8-2 lists a set of UAF parameter values that are appropriate for many configurations. If your disks are highly fragmented or if your backups will be performed during periods of heavy system use, you should reduce the values shown for WSQUOTA and FILLM.

**Table 8-2: Sample Process Quotas for Efficient Backups**

Parameter	Suggested Value
WSQUOTA	16384
WSEXTENT	16384
PGFLQUO	32768
FILLM	128
DIOLM	4096
ASTLM	4096

(continued on next page)

**Table 8-2 (Cont.): Sample Process Quotas for Efficient Backups**

Parameter	Suggested Value
BIOLM	128
BYTLM	65536
ENQLM	256

If you change any parameters using `AUTHORIZE`, you must log out and log in again before the changes take effect. If you change any of the specified `SYSGEN` parameters, you must shut down your system and reboot before the changes take effect.

### 8.3 Restoring Files from Backup Copies

From time to time, you might need to restore files from your backup copies. To restore files, you use the `BACKUP` command to retrieve the files from a previously created save set.

You can use the `BACKUP` command to restore the following:

- All the files on a disk (or volume or volume set)
- All the files in a specific directory tree (for example, all the files in a user's main directory and subdirectories)
- One or more specific files
- Files from an incremental backup

The procedure for restoring files depends upon the type of restore operation (all files on a disk, directory trees, or specific files) and whether your most recent backup was an image or incremental backup.

In general, the `BACKUP` command line that you use to restore files is as follows:

```
BACKUP save_set_specification output_specifier
```

When you are restoring files from a save set stored on a disk, then you must use the `/SAVE_SET` qualifier with the save set specification in the second part of the `BACKUP` command line.

The sections that follow describe the process you follow and the command syntax you use for various restore operations.

### 8.3.1 Restoring All of the Files on a Disk

If a disk becomes corrupted or if the files on a disk become unusable for any other reason, you would want to restore the contents of the entire disk (or volume or volume set). The procedure you use for this operation depends on whether the most recent backup was an image or incremental backup. Section 8.3.1.1 describes the process for restoring a disk when the most recent backup was an image backup; Section 8.3.1.2 describes the process for restoring a disk when one or more incremental backups have been taken since the most recent image backup.

#### 8.3.1.1 Restoring Files from an Image Backup

To restore the entire contents of a disk when your most recent backup was an image backup (using the `/IMAGE` qualifier, as described in Section 8.1.4), use the following procedure:

1. Logically mount the disk *to* which you will copy the files, using the `/FOREIGN` qualifier. (In order to *logically mount* a device, you use the `MOUNT` command at DCL level. See the *VMS User's Manual* for information about the `MOUNT` command.)
2. Load the tape, disk, or diskette that contains the saved backup copy of your disk.
3. Give the `BACKUP` command with the `/IMAGE` qualifier, using the following syntax:

```
BACKUP /IMAGE device:save_set_specification output_specifier
```

If your backup copy is on a disk or diskette, then you must also use the `/SAVE_SET` qualifier immediately after the save set specifier (*device:save\_set\_specification*).

If you do not know the name of the save set, do one of the following:

- If the save set is on a disk, use the `DIRECTORY` command to determine the name of the save set, for example:

```
$ DIRECTORY BACKUP_DISK:[BACKUPS]
Directory SYSSYSDEVICE:[BACKUPS]
19APRIL1990.SAV;1
Total of 1 file.
```

The save set is named 19APRIL1990.SAV.

- If the save set is on magnetic tape, physically load the tape and then give the following command, substituting the name of the tape drive you use for `MTA0`:

```
$ BACKUP/LIST/REWIND MTA0:
```

```
Listing of save set(s)
```

```
Save set:          19APRIL1990.SAV
Written by:        SYSTEM
UIC:               [000001,000004]
Date:              19-APR-1990 22:03:03.63
.
.
.
```

The save set is named 19APRIL1990.SAV.

4. If your backup copy is on more than one tape or diskette, repeat step 2 for each tape or diskette.
5. Dismount the disk onto which you just restored the files, using the /NOUNLOAD qualifier.

The following example shows this process, using the following elements and assumptions:

- The save set for the saved files is named FULL\_BACKUP.SAV. This save set was created when an image backup was made, using the BACKUP/IMAGE command.
- The tape containing the saved copy of the disk contents is loaded on MTA1:.
- The disk to which the files will be restored is named DRA2:.

```
$ MOUNT/FOREIGN DRA1: ①
$ BACKUP/IMAGE MTA1:FULL_BACKUP.SAV/REWIND DRA2: ②
$ DISMOUNT/NOUNLOAD DRA2: ③
```

In this sequence, the individual command lines do the following:

- ① Logically mounts the disk DRA2. The files will be restored to this disk. The disk, of course, must be physically loaded before this step.
- ② Restores the directory structure and all the files from the save set FULL\_BACKUP.SAV to the disk DRA2.

The /IMAGE qualifier restores a logical duplicate of the original disk, so that the entire directory structure is restored and the files are placed in the proper directories. When you use the /IMAGE qualifier in a restore operation, the disk to which you are restoring the files is initialized. That is, you will be unable to access any information that was previously stored on the disk. If you do not want to use the /IMAGE qualifier (or if the /IMAGE qualifier was not used during the save operation), then you can still restore the directory structure and all of the files by specifying DRA2:[\* . . . ] as your output specifier.

- ③ Logically dismounts the disk.

### 8.3.1.2 Restoring Files from an Incremental Backup

To restore files after you have taken one or more incremental backups since your most recent image backup, you must first restore the most recent image backup and then restore each of the subsequent incremental backups. Use the following procedure to restore files after one or more incremental backups. (Note that the first few steps are similar to the procedure for restoring files from an image backup.)

1. Logically mount the disk *to* which you will copy the files, using the /FOREIGN qualifier. (In order to *logically mount* a device, you use the MOUNT command at DCL level. See the *VMS User's Manual* for information about the MOUNT command.)
2. Load the tape, disk, or diskette that contains the most recent **image backup** of the disk (or volume or volume set).
3. Give the BACKUP command with the /IMAGE qualifier, using the following syntax:

```
BACKUP /IMAGE device:save_set_specification output_specifier
```

If your backup copy is on a disk or diskette, then you must also use the /SAVE\_SET qualifier immediately after the save set specifier (*device:save\_set\_specification*).

4. If your backup copy is on more than one tape or diskette, repeat step 2 for each tape or diskette.
5. Dismount the disk onto which you have just restored the files from the image backup, using the /NOUNLOAD qualifier.
6. Mount the disk that you are restoring as a file-structured volume, using the following syntax:

```
MOUNT device_name: PUBLIC
```

7. Dismount the media that contained the image backup, and mount the tape, disk, or diskette that contains the most recent **incremental backup** of the disk (or volume or volume set).
8. Restore your incremental save sets, beginning with the most recent backup. Use the following syntax to restore an incremental backup:

```
BACKUP/INCREMENTAL save_set_specifier device_specifier
```

Remember that you must use the /SAVE\_SET qualifier after the save set specifier if your backup copies are on a disk or diskette.



Continue restoring the incremental backups in reverse chronological order, until you have processed all of the incremental backups since the most recent image backup. If the incremental backups are on more than one tape, diskette, or disk, then you must mount each of these successively.

When you have processed the most recent incremental backup, the restore operation is complete.

The following example shows the process of restoring an entire disk after a series of incremental backups, using these elements and assumptions:

- The save set for the image backup is named `WORK_DISK_BACKUP.SAV`. This save set was created using the `BACKUP/IMAGE` command.
- The save sets for the incremental backups are named as follows:

```
WORK_DISK_16_JAN.SAV
WORK_DISK_17_JAN.SAV
WORK_DISK_18_JAN.SAV
```

- Both the image and the incremental backup copies are on the disk named `DBA3`, which is already mounted.
- The disk to which the files will be restored is named `DRA2`.

```
$ MOUNT/FOREIGN DRA2: ❶
$ BACKUP/IMAGE DBA3:WORK_DISK_BACKUP.SAV/SAVE_SET DRA2: ❷
$ DISMOUNT/NOUNLOAD DRA2: ❸
$ MOUNT DRA2: PUBLIC ❹
$ BACKUP/INCREMENTAL DBA3:WORK_DISK_18_JAN.SAV/SAVE_SET DRA2: ❺
$ BACKUP/INCREMENTAL DBA3:WORK_DISK_17_JAN.SAV/SAVE_SET DRA2: ❻
$ BACKUP/INCREMENTAL DBA3:WORK_DISK_16_JAN.SAV/SAVE_SET DRA2: ❼
```

In this sequence, the individual command lines do the following:

- ❶ Logically mounts the disk `DRA2`. The files will be restored to this disk.
- ❷ Restores the directory structure and all the files from the save set `WORK_DISK_BACKUP.SAV` to the disk `DRA2`. This was an image backup, which must be the first save set you restore when you want to restore incremental backup copies.
- ❸ Logically dismounts the disk `DRA2`.
- ❹ Remounts the disk `DRA2`, this time as a public, files-structured volume.
- ❺ Restores the most recent incremental backup.
- ❻ Restores the next incremental backup.
- ❼ Restores the last incremental backup.

Restoring the incremental backups in reverse chronological order is the most efficient way to restore files. When you have restored the last incremental backup, the restoration process is complete.

### 8.3.2 Restoring an Individual Directory Structure

If the contents of an entire directory structure are lost, it is simple to restore the directory structure with the `BACKUP` command and the saved backup copies that you have made. The `BACKUP` command automatically creates any subdirectories, and all files are placed in their proper directories or subdirectories. (You can also use the `BACKUP` command in the same way to copy a directory tree from one disk to another disk, or even from one system to another system.)

To restore a directory tree structure, use the following syntax:

```
BACKUP save_set_specifier/SELECT=[dir...] output_specifier:[dir..]
```

Note that if you are restoring files from a disk, then you must also use the `/SAVE_SET` qualifier with the save set specifier.

For example, suppose that all of the files in directory `WORK_DISK:[FINANCE]` were inadvertently deleted. The save set (on a tape, from an image backup taken the previous night) is named `FULL_BACKUP.SAV`, and it will be mounted on drive `MTA0`. To restore the directory tree, follow this process:

```
$ BACKUP MTA1:FULL_BACKUP.SAV/SELECT=[FINANCE...] -  
_ $ WORK_DISK:[FINANCE...]
```

This command restores the directory tree `[FINANCE ... ]` to the device `WORK_DISK`.

Note that you do not use the `MOUNT/FOREIGN` command for the disk to which you are restoring files when you restore individual directories or files (rather than an entire disk or volume).

When restoring files from incremental backups, remember the following:

1. First restore the files from the most recent image backup, using the `/RECORD` qualifier with the `BACKUP` command.
2. Restore the files from the incremental backups, beginning with the most recent incremental backup and then continuing in reverse chronological order. Use the `/INCREMENTAL` qualifier with the `BACKUP` command for each incremental backup that you process.

### 8.3.3 Restoring an Individual File

The `BACKUP` command lets you restore individual files from a saved backup copy to a disk. To restore an individual file that has been lost or corrupted, first make the backup copy available by making sure that the tape, diskette, or disk is mounted. Then enter the `BACKUP` command, using the `/SELECT=` qualifier with the input specifier to identify the file or files that you want to restore.

For example, suppose you are asked to restore the file `DUA0:[SALES]JANUARY_SALES.DAT`. Your most recent backup is an image backup, on a magnetic tape, with a save set named `FULL_BACKUP.SAV`, and tape will be mounted in `MUA0`. You could use the following sequence to restore the file:

```
$ BACKUP MUA0:FULL_BACKUP.SAV/REWIND/SELECT=[SALES]JANUARY_SALES.DAT -
_ $ DUA0:[SALES]*.*
```

The `/REWIND` qualifier to the input specifier ensures that the tape is read from the beginning of the tape. Note that when the `/REWIND` qualifier is used with the input specifier, it ensures only that the tape is rewound. When the `/REWIND` qualifier is used with the output specifier (in saving files), the tape is automatically initialized.

Note that you do not use the device name (`MUA0`) when specifying the file with the `/SELECT` qualifier; however, you do use the device name with the output specifier.

### 8.3.4 Listing the Contents of a Save Set

To list the contents of a save set, you must use the `BACKUP` command. There are two ways to list the contents of a save set:

- Use the `/JOURNAL` qualifier each time you back up your files. Then, use the `BACKUP/JOURNAL/LIST` command to list the contents of the save set. You use neither an input specifier nor an output specifier with this command.
- Make available the save set containing your backup copies by ensuring that the backup media is mounted, and then use the `BACKUP/LIST` command. With this command, you use the backup media and the save set as your input specifier, and you do not use an output specifier.

For example, suppose you had backed up a disk with this command:

```
$ BACKUP /IMAGE /JOURNAL=SYS$MANAGER:FULL_BACKUP.BJL -
_ $ WORK_DISK: MTA1:FULL_BACKUP.SAV
```

You can list the contents of the save set on your terminal by using the following command:

```
$ BACKUP /LIST /JOURNAL=SYS$MANAGER:FULL_BACKUP.BJL
```

You can also direct the contents of the save set to be listed in a file that you can read or edit with a text editor by supplying a file specification with the `/LIST` qualifier.

## 8-20 Backing Up and Restoring Files

For example, suppose that the save set containing your backup was on a tape that had been mounted on MTA0. To write the contents of the save set to an ASCII file, enter the following command:

```
$ BACKUP /LIST=SYS$MANAGER:BACKUP_FILES.DAT -  
_ $ /JOURNAL=SYS$MANAGER:FULL_BACKUP.BJL MTA0:FULL_BACKUP.SAV
```

This can be useful when you want to select individual files from the save set and restore them to a disk. You can first list all the files in the save set in a file (using the */JOURNAL=file-spec* and */LIST=file-spec* qualifiers). Edit the resulting output file, creating a command procedure that restores only those files that you want to restore. In your command procedure, use the same syntax for the BACKUP command as when you restore individual files at DCL level.

### 8.4 Standalone Backup

Standalone Backup is bootstrapped into main memory instead of running under the control of the VMS operating system. You can bootstrap standalone Backup from console media or from a Files-11 disk. The installation and operations guide for your VAX processor describes how to build a standalone Backup kit and bootstrap standalone Backup from the console medium. This section describes how to build a standalone Backup kit on a Files-11 disk and contains an example of a standalone Backup operation.

Digital recommends that you use standalone Backup to save and restore the system disk. This is because files open for system write access will not be saved completely if you use online Backup to save and restore the system disk.

Standalone Backup employs a subset of Backup qualifiers and only performs image and physical operations. Before using standalone Backup, read the descriptions of the command qualifiers */IMAGE* and */PHYSICAL*.

In a standalone Backup image operation, as in an online Backup image operation, if the output volume is a disk, all files on the output volume are stored contiguously. Contiguous storage of files eliminates disk fragmentation and creates contiguous free blocks of disk space.

Before you bootstrap standalone Backup from a Files-11 disk, you must have executed the *SYS\$UPDATE:STABACKIT.COM* command procedure to create a standalone Backup kit on the disk. You must have the user privileges *BYPASS*, *CMKRNL*, *CMEEXEC*, *LOG\_IO*, *SYSNAM*, *VOLPRO*, and *OPER* (or the user privilege *SETPRV*) to execute *STABACKIT.COM*.

*STABACKIT.COM* prompts you for a target device and places the files in directories *[SYSn.SYSEXE]* and *[SYSn.SYS\$LDR]* on the target device. The target device can be either a system disk or a user disk. If the directories *[SYSn.SYSEXE]* and *[SYSn.SYS\$LDR]* do not already exist on the target device, *STABACKIT.COM* creates the directories. *STABACKIT.COM* lists the standalone Backup files at your terminal as they are copied.

The following example creates a standalone Backup kit on the system disk of node MYNODE (the logical name for the system disk is SYS\$SYSDEVICE):

```
$ SET DEFAULT SYS$UPDATE
$ @STABACKIT

Enter the name of the device on which to build the kit: SYS$SYSDEVICE

Sysgen parameters for standalone VMS have been placed in file
      SYS$SYSROOT:[SYSUPD]VAXVMSSYS.PAR-TEMP-00000033;1
%CREATE-I-EXISTS, STA$TARGET already exists
%COPY-S-COPIED, SYS$SYSROOT:[SYSEXE]STASYSGEN.EXE;1 copied to
      _MYNODE$DUA0:[SYSE.SYSEXE]SYSINIT.EXE;1 (79 blocks)
%COPY-S-COPIED, SYS$SYSROOT:[SYSEXE]SYSBOOT.EXE;1 copied to
      _MYNODE$DUA0:[SYSE.SYSEXE]SYSBOOT.EXE;1 (91 blocks)
%COPY-S-COPIED, SYS$SYSROOT:[SYSUPD]VAXVMSSYS.PAR-TEMP-00000033;1 copied
      to _MYNODE$DUA0:[SYSE.SYSEXE]VAXVMSSYS.PAR;1 (15 blocks)
%SET-I-ENTERED, DUA0:[000000]SYS0.DIR;1 entered as
      _MYNODE$DUA0:[SYSE]SYSCOMMON.DIR;1
%COPY-S-COPIED, SYS$SYSROOT:[SYSEXE]STABACKUP.EXE;1 copied to
      _MYNODE$DUA0:[SYSE.SYSEXE]STANDALON.EXE;1 (409 blocks)
%DELETE-I-FILDEL, SYS$SYSROOT:[SYSUPD]VAXVMSSYS.PAR-TEMP-00000033;1
      deleted (16 blocks)

Ending time   19-APR-1990 16:22:00.85
Starting time 19-APR-1990 16:21:13.36

The kit is complete.
```

The installation and operations guide for your processor contains instructions for bootstrapping standalone backup.

If your system has at least two megabytes of memory, you can perform more than one standalone Backup operation without rebooting standalone Backup between operations. The following example bootstraps standalone Backup on a MicroVAX II system from a disk named DUA0, saves the contents of DUA0 to a magnetic tape in drive MUA0, and restores the image save set from the MUA0 to DUA0:

```
>>> B/E0000000 DUA0:
%BACKUP-I-IDENT, Stand-alone BACKUP V5.2; the date is 19-APR-1990 15:22:35.53
$ BACKUP/IMAGE/VERIFY DUA0: MUA0:FULLBACK.BCK/REWIND/LABEL=DEC29
%BACKUP-I-NOBACKUP, DUA0:[SYS0.SYSEXE]PAGEFILE.SYS;1 data not copied, file
marked NOBACKUP
%BACKUP-I-NOBACKUP, DUA0:[SYS0.SYSEXE]SWAPFILE.SYS;1 data not copied, file
marked NOBACKUP
%BACKUP-I-STARTVERIFY, starting verification pass
%BACKUP-I-PROCDONE, Operation completed. Processing finished at 19-APR-1990
16:34:45.20
```

If you do not want to perform another standalone BACKUP operation, use the console to halt the system.

```
If you do want to perform another standalone BACKUP operation,
ensure the standalone application volume is online and ready.
Enter "YES" to continue: YES
%BACKUP-I-IDENT, Stand-alone BACKUP V5.2; the date is 19-APR-1990 15:22:35.53
$ BACKUP/IMAGE/VERIFY MUA0:FULLBACK.BCK/REWIND/LABEL=AUG29 DUA0:
```

## 8.5 Backup and Magnetic Tape

Throughout this chapter, the use of magnetic tape in backup procedures has been demonstrated in various examples. This section describes some of the considerations you should keep in mind when using magnetic tape.

### 8.5.1 Automatic Tape Unloading

When a Backup save operation requires the use of more than one magnetic tape, Backup automatically unloads magnetic tapes when they are full.

When you attempt a Backup save operation to a write-protected magnetic tape, Backup automatically unloads the tape and displays a `WRITENABLE` message. After the tape unloads, you can remove it from the drive, insert a write ring, and replace the tape in the drive. Backup displays the `WRITENABLE` message on your terminal if you specified the command qualifier `/NOASSIST` or on the operator terminal if you did not specify `/NOASSIST`. From the operator terminal, enter the `REPLY/TO` command to restart the save operation. From your terminal, enter `YES` to restart the save operation.

### 8.5.2 Tape Label Processing

By default, Backup processes information stored in the volume header record of the tape before writing to or reading from a magnetic tape. The volume header record contains volume protection information, an expiration date, and a volume label. By processing the volume protection information, Backup ensures that you have the right to access the volume in the manner you requested. By processing the tape expiration date, Backup prevents you from initializing a magnetic tape that has not yet expired. By comparing the volume label specified in the Backup command line to the volume label of the tape, Backup prevents you from creating a save set on the wrong magnetic tape or from reading the wrong tape.

You can prevent Backup from processing the tape expiration date and the volume label by specifying the command qualifier `/IGNORE=LABEL_PROCESSING`.

### 8.5.3 Assigning Volume Labels to Magnetic Tapes

Magnetic tape volume labels can contain a maximum of six characters. You can use any ANSI “a” character in a magnetic tape volume label. The ANSI “a” characters include numbers, uppercase letters, and any one of the following nonalphanumeric characters:

```
! " % ' ( ) * + , _ . / : ; < = > ?
```

If you use any of the preceding nonalphanumeric characters, you must enclose the volume label with quotation marks.

Label your magnetic tapes according to the data contained on the tapes. The following table presents some suggestions for labeling tapes:

Label	Type of Backup	Expiration Date
DLY101	Daily, group 1, volume number 1	Expires in 7 days
DLY102	Daily, group 1, volume number 2	Expires in 7 days
WKY101	Weekly, group 1, volume number 1	Expires in 4 weeks
WKY201	Weekly, group 2, volume number 1	Expires in 4 weeks
MTH101	Monthly, group 1, volume number 1	Expires in 12 months
YRY101	Yearly, group 1, volume number 1	Expires in 5 years

## 8.6 Summary

### Making Regular Backups

You should make backup copies of the files on your system on a regular and scheduled basis. It is easier to back up entire disks rather than individual files.

### Image (Full) and Incremental Backups

You can make either image backups (also called full backups) or incremental backups. Image backups save a copy of all files, while incremental backups save a copy of those files that have been created or modified since the most recent backup. Incremental backups are useful *only* when an image backup, using the /RECORD qualifier, has previously been taken.

Incremental backups can be done more quickly than image backups and they require less storage space. However, files that have been backed up with a combination of image and incremental backups are more complicated to restore than files that have been backed up in an image backup. It might be reasonable to make an image backup weekly and to make daily incremental backups for disks where files are updated frequently. For small systems where a single magnetic tape cassette can store all the files on the system (for example, many workstation configurations), it can be useful to use a batch job that backs up the entire system daily, resubmitting itself automatically.

### Save Sets

When you back up files in a save operation, the files are stored in units called save sets. A save set is a file created and used by Backup when you use the BACKUP command to save files. Save sets are written in a format that only Backup can interpret, and they include both the files that you save with Backup and other information that is used by Backup.

### Media Used for Backups

You can save files on magnetic tape, diskette, or disk.

### The BACKUP Command Line

The BACKUP command line has three parts: the BACKUP command, the input specifier, and the output specifier.

The first part of the command line is the BACKUP command and any qualifiers to the BACKUP command itself (for example, /RECORD or /INCREMENTAL).

The input specifier identifies the source data for the Backup operation. When you are using Backup to save current copies of files, the input specifier identifies the disk on which the files reside and, if you are not backing up the entire disk, the files or directory structures that you are saving. The input specifier also includes any qualifiers that apply to the input specifier—for example, the /SAVE\_SET qualifier when you are restoring files from a disk.

The output specifier identifies the destination for the Backup operation and any qualifiers that apply to it.

### Qualifiers to BACKUP

When saving image backups of a disk, use the /IMAGE qualifier to the BACKUP command. You can also use the /RECORD qualifier if you expect to use incremental backups for the files, and you can use the /JOURNAL qualifier if you might later want to obtain a list of files in the save set.

When saving incremental backups of a disk or a subset of its files, use the /SINCE=BACKUP and /RECORD qualifiers.

### Command Procedures

You can use command procedures to reduce the work needed to back up files. If no operator intervention is needed for your backups, then you can submit the command procedures to run as batch jobs, preferably when the system is least busy. If operator intervention is necessary—for example, to change tapes during a backup—then you can still use command procedures to reduce your workload and ensure accuracy in your command lines.

In addition to the sample command procedures shown in this manual, the directory SYS\$EXAMPLES includes the command procedures BACKUSER.COM (for saving backup copies) and RESTUSER.COM (for restoring saved copies). You can copy these procedures and edit them to conform to your needs.

### Set Proper Parameters for Backup

You can maximize the efficiency of your backup by using the Authorize Utility to set certain parameters in the User Authorization File (SYSUAF.DAT). These parameters should be set only for the account from which backups will be taken.



## Restoring Files

To restore files to a disk from a backup copy, you also use the `BACKUP` command. When restoring files, the `save` set containing your saved copy (on either tape, diskette, or disk) is the input specifier for the `BACKUP` command. The disk and, if your restore operation is not for the entire disk, the target directory or directory structure is the output specifier.

## Restoring Files from Incremental Backups

If you are restoring files and your most recent backup was an incremental backup, then you must first restore the most recent image backup. After restoring the image backup, restore each of the incremental backups, using the `/INCREMENTAL` qualifier in each `BACKUP` command line. When restoring the incremental backups, begin with the most recent incremental backup and continue in reverse chronological order.

## Listing the Contents of a Save Set

Use the `/JOURNAL` qualifier to the `BACKUP` command when saving files if you want to generate a journal file that lists the contents of the save set you create. This journal file can be read only by using the `BACKUP/LIST` command.

To determine the files that are contained in a save set, use the `BACKUP/LIST` command. If you used the `/JOURNAL` file when saving copies of your files, then you can use the `BACKUP/LIST` command to read the journal file. When no journal file exists, the save set itself must be on line (that is, on a tape, disk, or diskette that is mounted and available) in order for you to use the `BACKUP/LIST` command.



# Chapter 9

## Maintaining Acceptable System Performance

Performance management of a VMS system means optimizing your hardware and software resources for the current work load. This task entails several related activities:

- Acquiring a thorough familiarity with your work load and an understanding of how that work load uses the system's resources. This knowledge, combined with an appreciation of the VMS resource management mechanisms, will enable you to establish realistic standards for system performance in areas such as the following:
  - Interactive and batch throughput
  - Interactive response time
  - Batch job turnaround time
- Routinely monitoring system operating conditions to determine if, when, and why a given resource is approaching capacity.
- Investigating reports from users of degraded performance.
- Planning for changes in the system work load or hardware configuration and being prepared to make any necessary adjustments to system values.
- Performing, after installation, certain optional system management operations.

This chapter introduces the basic concepts of performance management. It is not meant to be used as a tutorial for tuning your system.

### 9.1 Knowing Your Work Load

One of the most important assets that a system manager brings to any performance evaluation is an understanding of the normal work load and operating conditions of the system. Each system manager must assume the responsibility for understanding the system's work load sufficiently to be able to recognize normal and abnormal operating conditions; to predict the effects of changes in applications, operations, or usage; and to recognize typical throughput rates. The system manager should be able to answer questions such as the following:

- What is the typical number of users on the system at each time of day?
- What is the typical response time for various tasks for this number of users, at each hour of operation?
- What are the peak hours of operation?
- Which jobs typically run at which time of day?
- Which commonly run jobs are intensive consumers of the CPU, memory, and disk space?
- Which applications involve the most image activations?
- Which parts of the system software, if any, have been modified or user-written, such as device drivers?
- Are there any known system bottlenecks? Are there any anticipated ones?

If you are new to VMS system management, you should observe system operation using the following tools:

- Monitor Utility
- Accounting Utility
- SHOW commands (available through DCL)

Over time you will learn about metrics such as the typical page fault rate for your system, the typical CPU usage, the normal memory usage, and typical modes of operation, and you will begin to see how certain activities affect system performance. As you continue to monitor your system, you will come to know what range of values is acceptable, and you will be better prepared to detect unusual conditions.

Routine evaluation of the system is critical for effective performance management. The best way to avoid problems is to anticipate them; you should not wait for problems to develop before you learn how the system performs. You can learn more about your system's operation if you use the Monitor and Accounting utilities regularly to capture and analyze certain key data items. By observing and collecting this data, you will also be able to see usage trends and predict when your system will approach its capacity.

When you use the tools that measure and report system operations, keep in mind that the tools themselves use some system resources. Be careful, therefore, in selecting the items you want to measure and the frequency with which you collect the data. If you use the tools excessively, the consumption of system resources to collect, store, and analyze the data can distort your picture of the system's work load and capacity. The best approach is to have a plan for collecting and analyzing the data.

### 9.1.1 Using the Monitor Utility (MONITOR)

You can develop a database of performance information for your system by running MONITOR continuously as a background process. The directory with the logical name SYS\$EXAMPLES includes three command procedures that you can use to establish the database. Instructions for installing and running the procedures are contained in the comments at the beginning of each one. Following is a brief summary of these procedures:

- SUBMON.COM—Starts MONITOR.COM as a detached process. You should invoke SUBMON.COM from the DCL procedure SYS\$MANAGER:SYSTARTUP\_V5.COM.
- MONITOR.COM—Creates a summary file from the recording file of the previous boot, then begins recording for this boot. The recording interval is 10 minutes.
- MONSUM.COM—Generates two clusterwide multifile summary reports; one for the previous 24 hours, and one for the previous day's prime-time period (9 A.M. to 6 P.M.). These are mailed to the system manager, and then the procedure resubmits itself to run each day at midnight.

While MONITOR data is recorded continuously, a summary report can cover any contiguous time segment. The command file MONSUM.COM, which is executed every midnight, generates and mails the two multifile summary reports described previously. These reports are not saved as files, so if you want to keep them, you must either extract them from your mail file or alter the MONSUM.COM command procedure to save them.

### 9.1.2 Using the Accounting Utility (ACCOUNTING)

The Accounting Utility can be used to generate reports that indicate how well the system is performing. Of particular interest to performance management is image-level accounting, which records information about the system resources consumed by the execution of specific images. By being aware of the images that use the most resources at your site, you can better direct your efforts toward controlling them and the resources they consume.

## 9-4 Maintaining Acceptable System Performance

Images used frequently are typically good candidates for code sharing, whereas images that consume large quantities of various resources might be forced to run in a batch queue. In batch queues, the number of simultaneous processes can be controlled. Using a series of commands like those in the following example, you can produce a report containing the resource usage information necessary to manage images.

**NOTE:** It is assumed in the following example that image-level accounting records have been collected previously. (You enable image-level record collection by entering the DCL command SET ACCOUNTING/ENABLE=IMAGE.)

```
$ ACCOUNTING /TYPE=IMAGE /OUTPUT=BYNAM.LIS -
_$ /SUMMARY=IMAGE -
_$ /REPORT=(PROCESSOR,ELAPSED,DIRECT_IO,FAULTS,RECORDS)
$ SORT BYNAM.LIS BYNAM.ORD /KEY=(POS=16,SIZ=13,DESCEND)
```

```
.
.
(Edit BYNAM.ORD to relocate heading lines)
.
.
$ TYPE BYNAM.ORD
```

You should be careful when using image-level accounting on your system. As a rule, you should enable image-level accounting only when you plan to invoke ACCOUNTING to process the information provided in the file SYS\$MANAGER:ACCOUNTNG.DAT. Once you have collected enough data for your purposes, disable image-level accounting by entering the DCL command SET ACCOUNTING /DISABLE=Image. While image activation data can be very helpful in performance analysis, it can be a waste of processing time and disk storage if the data is collected but never used.

### 9.1.3 Managing Work Load

System performance is directly proportional to the efficiency of work load management. Each site must develop its own strategy for this key task. Before adjusting any system values, answer the following questions:

- Is there a time of day when the work load “peaks,” that is, when it is noticeably heavier than at other times?
- Is there any way to balance the work load better? Perhaps some voluntary measures can be adopted by users, after appropriate discussion.
- Could any jobs run better as batch jobs, preferably during nonpeak hours?
- Have primary and secondary hours of operation been employed with users? If not, could system performance benefit by adopting this practice? If the primary and secondary hours are in use, are the choices of hours the most appropriate for all users? (Plan to review this issue every time you either

add or remove users or applications, to ensure that the desired balance is maintained.)

- Can future applications be designed to work around any known or expected system bottlenecks? Can present applications be redesigned somewhat, for the same purpose?
- Are you using to the utmost the code-sharing ability that the VMS system offers you? If not, you will find that code sharing provides an excellent means to conserve memory, thereby improving performance over the life of the system.

Do not adjust any system values until you are satisfied that all these issues are resolved and that your work load management strategy is appropriate.

### 9.1.4 Distributing Work Load

You should distribute the work load as evenly as possible over the time your system is running. Although the work schedule for your site can make it difficult to schedule interactive users at optimum times, the following techniques might be helpful:

- Run large jobs as batch jobs—Establish a site policy that encourages the submission of large jobs on a batch basis. Regulate the number of batch streams so that batch usage is high when interactive usage is low. You might also want to use DCL command qualifiers to run batch jobs at lower priority, adjust the working set sizes, and control the number of concurrent jobs.
- Restrict system use—Do not permit more users to log in at one time than the system can support with an adequate response time. You can restrict the number of interactive users with the DCL command SET LOGINS/INTERACTIVE. You can also control the number of concurrent processes with the MAXPROCESSCNT system parameter, and the number of remote terminals allowed to access the system at one time with the RJOBLIM system parameter.

You might also restrict use of the system by groups of users to certain days and hours of the day. You can use the Authorize Utility to define the permitted login hours for each user. In particular, refer to the AUTHORIZE qualifiers /PRIMEDAYS, /P\_RESTRICT, /PFLAGS, /SFLAGS, and /S\_RESTRICT. Remember you can use the DCL command SET DAY to override the conventional day of the week associations for primary and secondary days. For example, you might need to specify a primary day of the week as a secondary day when it is a holiday.

- Design applications to reduce demand on binding resources—If you know where your system bottlenecks are or where they will likely occur in the near future, you can distribute the work load more evenly by planning usage that minimizes demand on the bottleneck points.

### 9.1.5 Installing Known Images

If you have programs that are frequently used on your system, you should consider installing them as known images. In general, a program should be installed as a known image if it has one or more of the following attributes:

- It is frequently run
- It is usually run concurrently by several processes
- It requires special privileges

Chapter 2 describes how to install programs as known images.

By specifying appropriate qualifiers to `INSTALL` commands, you can assign any of the following attributes to known images:

- **Permanently open**—Directory information on the image file remains permanently resident, eliminating the usual directory search required to locate a file. The cost of keeping an image file permanently open is approximately one page of nonpaged dynamic memory per file.
- **Header resident**—The header of the image file (native images only) remains permanently resident, saving one disk I/O operation per file access. For images with single-block file headers, the cost is less than one page of paged dynamic memory per file; for images with multiblock headers, the cost varies according to the header block count. The images must also be declared as permanently open.
- **Privileged**—Amplified privileges are temporarily assigned to any process running the image (executable images only), permitting the process to exceed its user authorization file (UAF) privilege restrictions during execution of the image. In this way, users with normal privileges can run programs that require higher than normal privileges.
- **Protected**—A shareable image contains protected code, that is, code that runs in Kernel or Executive mode but that can be called by a user-level image. Protected images must be declared shared.
- **Shared**—More than one user can access the read-only and noncopy-on-reference read/write sections of the image concurrently, so that only one copy of those sections ever need be in physical memory. (Copy-on-reference sections always require a separate copy for each process.) The image is implicitly declared as permanently open.
- **Writable**—When a shared noncopy-on-reference writable section is removed from physical memory (for paging reasons or because no processes are referencing it), it is written back to the image file. Any updates made by processes mapped to the section, therefore, are preserved (while the initial values are lost). The image must also be declared as shared.



### 9.1.6 Tuning a System

Tuning is the process of altering various system values to obtain the optimum *overall* performance possible from any given configuration and work load. However, the process does not include the acquisition and installation of additional memory or devices, although in many cases such additions (when made at the appropriate time) can vastly improve system operation and performance.

Always aim for best overall performance, that is, performance viewed over time. The work load is constantly changing on most systems. System parameters that produce optimal performance at one time might not produce optimal performance a short time later as the work load changes. Your goal is to establish values that, on the average, produce the best overall performance.

Before you take any action, you must recognize that the following sources of performance problems cannot be cured by adjusting system values:

- Improper operation
- Unreasonable performance expectations
- Insufficient memory for the applications attempted
- Inadequate hardware configuration for the work load, such as too slow a processor, too few buses for the devices, too few disks, and so forth
- Improper device choices for the work load, such as using disks with insufficient speed or capacity
- Hardware malfunctions
- Human errors, such as poor application design or allowing one process to consume all available resources

When you make adjustments, you normally select a very small number of values for change, based on a careful analysis of the behavior being observed. These values are usually either system parameters or entries in the User Authorization File (UAF) that affect particular users.

Normally, system parameters are modified automatically by the system using AUTOGEN; AUTOGEN uses system configuration data to automatically set system parameters. You can also use SYSGEN to manually alter system parameters.

One of AUTOGEN's special features is that it makes automatic adjustments for you in associated parameters. To control the values in the UAF, you use the Authorize Utility.

### 9.1.7 Predicting When Tuning Is Required

Under most conditions, tuning is rarely required for VMS systems. The AUTOGEN.COM command procedure, which is included in the operating system, establishes initial values for all the configuration-dependent system parameters so that they match your particular configuration. Additionally, the system includes features that in a limited way permit it to adjust itself dynamically during operation. That is, the system detects the need for adjustment in certain areas, such as the nonpaged dynamic pool, working set size, and the number of pages on the free and modified page lists. The system makes rough adjustments in these areas automatically. As a result, these areas can grow dynamically, as appropriate, during normal operation.

A frequent reason for disappointment in system performance is ultimately due to insufficient hardware capacity. Once the demand on a system exceeds its capacity, adjusting system values will not result in any significant improvements, simply because such adjustments are a means of trading off or juggling existing resources.

Although tuning is rarely required, you should recognize that system tuning might be needed under the following conditions:

1. If you have adjusted your system for optimal performance with current resources and then acquire new capacity, you must plan to compensate for the new configuration. In this situation, the first and most important action is to execute the AUTOGEN command procedure.
2. If you anticipate a dramatic change in your work load, you should expect to compensate for the new work load.

### 9.1.8 Evaluating Tuning Success

Whenever you adjust your system, you should monitor its behavior afterward, to be sure that you have obtained the desired results. To observe results, use MONITOR and the various forms of the DCL SHOW command.

For example, you might consider running some programs whose results you believe are fixed and reproducible, at the same time that you run your normal work load. If you run the programs and measure their running times under nearly identical work load conditions both before and after your adjustments, you can obtain a basis for comparison.

However, when applying this technique, remember to take the measurements under very similar work load conditions. Also, remember that this test alone does not provide conclusive proof of success. There is always the possibility that your adjustments might have favored the performance of the image you are measuring—to the detriment of other images. Therefore, in all cases, continue to observe system behavior closely for a time after you make any changes.

### 9.1.9 Performance Options

Following is a list of optional system management operations, normally performed after installation, that often result in improved overall performance. Note, however, that not all options are appropriate at every site.

- **Decompress system libraries**—Most of the libraries shipped with Version 4 and later versions of the VMS operating system are in a compressed format in order to conserve disk space. The system dynamically decompresses them whenever they are accessed, and the resulting performance slowdown is especially noticeable during link operations and when requesting online help. If you have sufficient disk space, decompressing the libraries improves both CPU and elapsed time performance. To do this, invoke the command procedure `SYS$UPDATE:LIBDECOMP.COM`. The decompressed object libraries take up about 25 percent more disk space than when compressed; the decompressed help libraries take up about 50 percent more disk space.
- **Disable file system high-water marking**—This security feature guarantees that users cannot read data they have not written. It is implemented by erasing the previous contents of the disk blocks allocated every time a file is created or extended. High-water marking is set by default whenever a volume is initialized.

Disabling this feature improves system performance by a variable amount, depending on the frequency of new file creation, the frequency of extending existing files, and the fragmentation of the volume. To disable high-water marking, you can specify the `/NOHIGHWATER` qualifier when initializing the volume, or you can enter the `SET VOLUME` command using the following syntax at any time:

```
SET VOLUME/NOHIGHWATER_MARKING device-spec[:]
```

Then dismount and remount the volume. However, you should consider the security implications of disabling this feature.

- **Set RMS file extend parameters**—Because files extend in increments of twice the multiblock count (default 16 blocks), system defaults provide file extension of only 32 blocks. Thus, when files are created or extended, increased I/O can slow performance. The problem can be corrected by specifying larger values for `SYSGEN` file extend parameters or by setting the system parameter `RMS_EXTEND_SIZE=80`.
- **Relink images**—Beginning with VMS Version 4.0, the VMS Run-Time Library (VMS RTL) was separated into several smaller libraries. Running images linked under previous versions of the VMS operating system will therefore incur the image activation costs of mapping all of the libraries, even if only one is needed. You can improve performance by relinking pre-Version 4.0 images that reference run-time library routines, so that only the required libraries are mapped and activated.

## 9-10 Maintaining Acceptable System Performance

- Install frequently used images—When an image is accessed concurrently by more than one process on a routine basis, install the image with the Install Utility, specifying the /OPEN, /SHARED, and /HEADER\_RESIDENT qualifiers. You will thereby ensure that all processes use the same physical copy of the image, and that the image will be activated in the most efficient way.

Generally, an image takes about two additional physical pages when installed /OPEN/HEADER\_RESIDENT/SHARED. The utility's LIST/FULL command shows the highest number of concurrent accesses to an image installed with the /SHARED qualifier. This information can help you decide whether installing an image is worth the space. For more information on the Install Utility, refer to the Reference Section.

- Reduce system disk I/O—You can move frequently accessed files off the system disk and use logical names or, where necessary, other pointers to access them. For example:
  - SYSUAF.DAT (SYSUAF is the logical name)
  - RIGHTSLIST.DAT (RIGHTSLIST is the logical name)
  - VMSMAIL.DAT (VMSMAIL is the logical name)
  - NETPROXY.DAT (NETPROXY is the logical name)
  - JBCSYSQUE.DAT (File specification parameter for the START/QUEUE/MANAGER command)
  - ERRFMT log files (SYS\$ERRORLOG is the logical name)
  - MONITOR log files (SYS\$MONITOR is the logical name)
  - Default DECnet account (DECNET record in SYSUAF file)

You can also consider moving paging and swapping activity off the system disk by creating large secondary page and swap files on a less heavily used disk.

However, be sure to understand the nature of system values before adjusting them. Without the proper level of understanding, you might degrade, rather than improve, overall performance.

While investigating the cause of an apparent performance problem, it is wise to keep in mind that tuning is a last resort solution.

## 9.2 Summary

### Performance Assessment Is Site Dependent

For any given hardware configuration, the system performance is likely to vary according to the work characteristics of your site. Some of the primary factors affecting performance include the number of interactive users on the system, the types of jobs and images that are run, and the balancing of the use of computer resources.

If you can identify specific bottlenecks (for example, certain images that seem to use a disproportional amount of CPU or memory), then you can begin to reduce the negative impacts of the bottlenecks (for example, by restricting the use of such images during peak hours).

### Performance Monitoring Tools

MONITOR can show you where system resources are being used. You can use MONITOR interactively (to show current conditions), and you can also use it to provide a report of resource usage at regular intervals.

The Accounting Utility can be used to record the amount of system resources used by specific images. You can use the Accounting Utility to identify images that might be downgrading performance; such images might then be run only at off-peak hours or in batch mode.

The DCL SHOW command has many features for system management. Use this command to identify the current status of various system conditions. (See the *VMS User's Manual* for more information about the SHOW command.)

### Work Load

The more that you can evenly distribute your work load, the more predictable will be your system performance. When possible, schedule large jobs to be run in batch during off-peak hours. If necessary, you can help balance the work load by restricting the number of interactive users allowed on your system at any one time, by reducing the priority of batch queues, and by installing programs as known images.

### System Tuning

Although improved performance can often be achieved by acquiring additional memory or devices, such a solution is not always available, especially in the short term. In the VMS operating system, configuration parameters are automatically set by the AUTOGEN procedure whenever you boot your system. You can also manually change parameters with SYSGEN. Remember, though, that tuning should be the last resort to solving a performance problem, not a first resort.



# Chapter 10

## Operator Tasks

Certain system management operations require your attention on a regular basis in order to maintain the system properly. Following are some of the activities you may perform in your role as system operator:

- Performing regular backups of user data
- Saving crash dumps following a system failure
- Printing and resetting the operator log file
- Printing and resetting the error log file
- Collecting information in the accounting log file

This chapter describes how to perform each of these tasks.

### 10.1 Performing Backups

One of the most commonly performed system operations is backing up files on public volumes. Backing up a volume means copying the contents of the volume to another volume or set of volumes. It is a precautionary measure to allow you to recover from the loss or destruction of valuable information. Most sites establish a policy and a schedule for regularly backing up files on public volumes.

See Chapter 8 for information about performing system backups, establishing a periodic backup schedule, and using the Backup Utility.

### 10.2 Maintaining System Log Files

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:

## 10-2 Operator Tasks

- **System dump file**

The system dump file assists you in analyzing the cause of the system failure. In the event of a severe system failure, the VMS operating system 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 10.2.1).

- **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 10.2.2).

- **Operator log file**

The Operator Communication Process (OPCOM) records system events in the operator log file (see Section 10.2.3).

- **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 10.2.4).

### 10.2.1 The System Dump File

The following requirements must be met before the VMS operating system can write a complete dump file:

- The system must not be halted 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 halting 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.SYS` 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 buffers with the default set to 4.

- The system parameter DUMPBUG must be set to 1 (the default is 1).

## 10.2.2 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:

1. A set of executive routines that detect errors and events and write relevant information into error log buffers in memory
2. 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.)
3. 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

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 terminal and stops 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
```

### Using Error Reports

The error reports generated by the Error Log Utility are useful in two ways:

1. They aid preventive maintenance by identifying areas within the system that show potential for failure.
2. 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 Digital 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 Digital Field Service. 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 Field Service representative 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 Field Service 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 Field Service representative can proceed to correct the device.

Error reports allow you to anticipate potential failures. In turn, Field Service 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.

### Maintaining the Error Log Files

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 morning at 9:00. 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.

### Printing Error Log Files

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
```

The following example demonstrates these steps:

```
$ 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.

### 10.2.3 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 process (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's log file:

- Initialization of the operator's 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 SYSGEN Utility
- Security alarm messages
- DECnet-VAX status messages

Example 10-1 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.

#### 10.2.3.1 Types of OPCOM Messages

This section describes some of the messages you might find in the operator's log file.

##### Initialization Messages

When you enter the REPLY/LOG command, the current operator's 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
```

**Example 10-1: Sample Operator Log File (SYS\$MANAGER:OPERATOR.LOG)**

```

##### OPCOM, 19-APR-1990 22:33:54.07 #####
Operator '_ZEUS$VT333:' has been disabled, user JONES
##### OPCOM, 19-APR-1990 22:34:15.47 #####
Operator '_ZEUS$VT333:' has been enabled, user SMITH
##### OPCOM, 19-APR-1990 22:34:15.57 #####
operator status for '_ZEUS$VT333:'
PRINTER, TAPES, DISKS, DEVICES
##### OPCOM, 19-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, 19-APR-1990 22:39:54.37 #####
request 1 was satisfied.
##### OPCOM, 19-APR-1990 22:40:23.54 #####
message from user SYSTEM
Volume "KLATU" mounted, on physical device MTA0:
##### OPCOM, 19-APR-1990 22:40:38.02 #####
request 2, from user PUBLIC
MOUNT new relative volume 2 () on MTA0:
##### OPCOM, 19-APR-1990 22:41:07.54 #####
message from user SYSTEM
Volume "KLATU" dismantled, on physical device MTA0:
15-APR-1986 22:42:14.81, request 2 completed by operator OPA0
##### OPCOM, 19-APR-1990 22:46:47.96 #####
request 4, from user PUBLIC
_TTB5:, This is a sample user request with reply expected.
##### OPCOM, 19-APR-1990 22:47:38.50 #####
request 4 was canceled
##### OPCOM, 19-APR-1990 22:48:21.15 #####
message from user PUBLIC
_TTB5:, This is a sample user request without a reply expected.
##### OPCOM, 19-APR-1990 22:49:07.90 #####
Device DMA0: is offline.
Mount verification in progress.
##### OPCOM, 19-APR-1990 22:49:20.22 #####
Mount verification completed for device DMA0:
##### OPCOM, 19-APR-1990 22:49:37.64 #####
Device DMA0: has been write locked.
Mount verification in progress.
##### OPCOM, 19-APR-1990 23:33:54.07 #####
message from user NETACP
DECnet shutting down

```

**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's log file at periodic intervals until the device is explicitly returned to online status.

The device status message appears in the operator's log file in the following format:

```
%OPCOM, dd-mmm-yyyy hh:mm:ss.cc, device device-name is offline
```

## 10-8 Operator Tasks

The devices for which this message can appear are card readers, line printers, and magnetic tapes.

### Terminal Enable and Disable Messages

You designate a terminal as an operator's terminal by entering the **REPLY/ENABLE** command from the desired terminal. OPCOM confirms the request by displaying the following message at the operator's terminal and in the operator's log file:

```
%OPCOM, dd-mmm-yyyy hh:mm:ss.cc, operator enabled, operator terminal-name
```

This message tells you which terminal has been established as an operator's terminal and when it was established.

If a terminal has been designated as an operator's terminal for a particular function, OPCOM displays the name of that function or operator class. For example, if you enter the command **REPLY/ENABLE=TAPES**, OPCOM displays the following message:

```
%OPCOM, 19-APR-1990 10:25:35.74, operator enabled, operator TTE1
```

```
%OPCOM, 19-APR-1990 10:25:38.82, operator status for operator TTE1  
TAPES
```

OPCOM confirms that the terminal is established as an operator's terminal and indicates that the terminal can only receive and respond to requests concerning magnetic tape-oriented events, such as the mounting and dismounting of tapes.

A terminal that has been designated as an operator's terminal is automatically returned to nonoperator status when the operator logs out. To return the terminal to normal (nonoperator) status without logging off, enter the **REPLY/DISABLE** command from the terminal. OPCOM confirms that the terminal is no longer an operator's terminal by displaying a message in the following format both at the operator's terminal and in the operator's 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's 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's terminal and in the operator's 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's 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, 19-APR-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.

### Volume Mount and Dismount Messages

Perhaps the widest range of operator messages occurs with volume mounts and dismounts. See Example 10-1 for examples of messages relating to mount verification and operator-assisted mounts.

### 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's 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.

If the user enters a REQUEST/TO command, the request is recorded in the operator's log file in the format described above, but without a request identification number, as follows:

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

When you respond to a user's request and specify the /TO qualifier, the response is recorded in the operator's log file in the following format:

```
response message
%OPCOM, dd-mmm-yyyy 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's log file in the following format:

```
%OPCOM, dd-mmm-yyyy 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's log file because the request has not yet been completed (that is, the request has not been fulfilled or aborted).

## 10-10 Operator Tasks

When a user enters a **REQUEST/REPLY** command and you have disabled all terminals as operator's terminals, OPCOM records all subsequent user's requests in the log file in the format shown above, 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's log file.

### **SYSGEN Messages**

Users with **CMKRNL** privilege can use the System Generation Utility (**SYSGEN**) to change system parameters in the running (active) system. Users with the **SYSRV** 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-mmm-yyyy hh:mm:ss.cc, message from user user-name
%SYSGEN-I-WRITExxx, system-mode system parameters modified by process ID n
into file y
```

### **Security Alarm Messages**

Security alarm messages are included in the operator log file if you enable a security operator terminal and specific alarm events with the **SET AUDIT/ENABLE** command. Alarm messages are sent to the security operator terminal when the selected events occur. The following is an example of a security alarm OPCOM message:

```
%OPCOM, 19-APR-1990 12:27:52.26, security alarm on node HERA/
System UAF record modification
```

You can use the command procedure **SYS\$MANAGER:SECAUDIT.COM** to selectively extract information from the operator's log file. Output from **SECAUDIT** is displayed on **SYS\$OUTPUT**. If you want to write the records to a file, you include the file specification with the **/OUTPUT** qualifier. The following command writes the records to the file **BREAKINS.DAT** in your default directory:

```
$ @SYS$MANAGER:SECAUDIT/OUTPUT=BREAKINS.DAT
```

#### **10.2.3.2 Maintaining the Operator Log File**

The operator log file, **OPERATOR.LOG**, normally resides on the system disk in the **[SYSMGR]** directory. A new version of **OPERATOR.LOG** is created each time the system is rebooted. You can also use the **DCL** command **REPLY/LOG** to create a new version of the file at any time. Note that there is one operator log file per node; it is not a shared file.

You should create new versions of the operator log file regularly and store these copies for reference. The file is in ASCII format and can be printed. The current version of the operator log file can be accessed only by creating a new log file. Section 10.2.3.3 describes how to print copies of 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. 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, use the following method 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
```

### 10.2.3.3 Printing the Operator Log File

Perform the following operation to produce a printed copy of the most recent version of the operator log file. (You must have OPER privilege.)

1. Use the following command to enable the terminal as an operator terminal:

```
$ REPLY/ENABLE
```

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

```
$ REPLY/LOG
```

3. Set the default to SYS\$MANAGER and enter the following command to list all versions of the file:

```
$ DIRECTORY OPERATOR.LOG
```

4. 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.

5. Print the operator log file by entering the following command:

```
$ 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.

## 10-12 Operator Tasks

```
$ REPLY/ENABLE
$ REPLY/LOG

%%%%%%%%%% OPCOM, 19-APR-1990 12:29:24.52 %%%%%%%%%%%
logfile initialized by operator MARS$VTA2:
logfile is SYS$MANAGER:OPERATOR.LOG

$ SET DEFAULT SYS$MANAGER
$ 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
```

### 10.2.3.4 Restarting OPCOM

You can restart OPCOM if for some reason it is deleted or suspended. Simply invoke the **STARTUP** command procedure in the [SYSEXE] directory and specify one parameter, as in the following example:

```
$ @SYS$SYSTEM:STARTUP OPCOM
```

## 10.2.4 The Accounting Log File

The Accounting facility 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, 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 access is sufficient 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**.

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 listed above.) 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)
- **Directory location:** `SYS$MANAGER`
- **File organization:** sequential
- **Record length:** variable
- **Record types:** eight

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.

## 10-14 Operator Tasks

### 10.2.4.1 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. 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.

### 10.2.4.2 Accounting Report Formats

The Accounting Utility 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 indicate 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.

By default, the output is directed to SYS\$OUTPUT. However, you can specify an output file with the /OUTPUT qualifier. The three output formats used for displaying data are brief (the default), full, and summary listings. The following example illustrates a summary output:

```
$ ACCOUNTING/SUMMARY=(ACCOUNT, USER) /REPORT=(RECORDS, ELAPSED, PROCESSOR)
```

From:	5-APR-1988 16:33	To:	19-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

## 10.3 Summary

### Operator Tasks

The tasks for an operator vary according to the job requirements at each site. Some common operator tasks include backing up files on the system, saving crash dumps following a system failure, printing and resetting the operator and error log files, and collecting information in the accounting log file.

## Backups

To perform a backup, the backup media must be available, and the appropriate BACKUP commands must be entered. If files are backed up to magnetic tape or diskette, the operator ensures that the tape or diskette is mounted on the appropriate device. If a single tape or diskette is not large enough to accommodate the backup, the operator should be ready to change the tape or diskette during the backup procedure. If files are backed up to disk, the operator should make sure that the proper disk is in place.

BACKUP procedures are described in Chapter 8.

## Log files

Operators are often responsible for maintaining the log files generated by the VMS operating system.

The *system dump file* is generated when the system shuts down due to a severe system failure. The system dump file produces a crash dump of the state of the system at the time that the error was detected. Using the DCL command ANALYZE/CRASH\_DUMP, you can analyze the dump and help determine the cause of the system failure.

The *error log file* contains device and CPU error messages. You can examine reports from the error log file using the ANALYZE/ERROR\_LOG command.

The *operator log file* is a text file that records system events and user requests sent to the operator terminal by the operator communication process (OPCOM). 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 *accounting log file* records the use of system resources and is the source of the accounting reports. You can examine the contents of the accounting log file with the ACCOUNTING command.



# Chapter 11

## System Security Issues

As the person responsible for the day-to-day system management, you play an important role in ensuring the security of your system. Therefore, you should familiarize yourself with the security features available with the VMS operating system and implement the features needed to protect systems, users, and files from damage caused by tampering. This chapter outlines the security features available with the VMS operating system and suggests procedures to reduce the threat of a break-in on your system or cluster.

Effective operating system security measures help prevent unauthorized access and theft of proprietary software, software plans, and computer time. These measures can also protect equipment, software, and files from damage caused by tampering.

This chapter provides system managers with an overview of security measures available with the VMS operating system. In this chapter, the expression *security manager* refers to a system manager who is also responsible for system security.

### 11.1 Types of Computer Security Problems

The source of a security breach on a computer system can usually be traced to one of three categories: user irresponsibility, user probing, or user penetration.

#### 11.1.1 User Irresponsibility

**User irresponsibility** refers to situations where the user purposely or accidentally causes some noticeable damage. An example would be a user who is authorized to access certain files making a copy of a key file to sell.

There is little that an operating system can do to protect sites from this source of security failures. The problem frequently lies in application design deficiencies or inconsistent use of available controls by users and the security manager. Sometimes the failure to enforce adequate environmental security unwittingly encourages this type of security problem.

## 11-2 System Security Issues

Even the best security system will fail if implemented inconsistently. This, along with the failure to motivate your users to observe good security practices, will make your system vulnerable to security failures caused by user irresponsibility.

### 11.1.2 User Probing

**User probing** refers to situations where a user exploits insufficiently protected parts of the system. Some users consider gaining access to a forbidden system area as an intellectual challenge, playing a game of user-versus-system. Although intentions might be harmless, theft of services is a crime. Users with more serious intent might seek confidential information, attempt embezzlement, or even destroy data by probing. Always treat user probing seriously.

VMS provides many security features to combat user probing. Based on security needs, the security manager implements features either on a temporary or permanent basis. These features are discussed in later sections.

### 11.1.3 User Penetration

**Penetration** refers to situations where the user breaks through security controls to gain access to the system. While VMS has security features making penetration extremely difficult, it is impossible to make any operating system completely impenetrable.

A user who succeeds in penetrating a system is both skilled and malicious. Thus, penetration is the most serious and potentially dangerous type of security breach. With proper implementation of VMS security features, however, it is also the rarest security breach, requiring the offender to possess unusual skills and perseverance.

## 11.2 Levels of Security Requirements

Each site has unique security requirements. Some sites might need limited measures because they are able to tolerate some forms of unauthorized access with little adverse effect. At the other extreme are those sites that cannot tolerate even the slightest probing, such as strategic military defense centers. In between are many commercial sites, such as banks.

To ascertain your security requirements, answer the questions in Table 11-1. Your answers can help determine your security needs to be low, medium, or high.



**Table 11-1: Event Tolerance as a Measure of Security Requirements**

Question: Could You Tolerate the Following Event?	Level of Security Requirements Based on Toleration Responses		
	Low	Medium	High
A user knowing the images being executed on your system	Y	Y	N
A user knowing the names of another user's files	Y	Y	N
A user accessing the file of another user in the group	Y	Y	N
An outsider knowing the name of the system just dialed into	Y	Y	N
A user copying files of other users	Y	N	N
A user reading another user's electronic mail	Y	N	N
A user writing data into another user's file	Y	N	N
A user deleting another user's file	Y	N	N
A user being able to read sections of a disk that might contain various old files	Y	N	N
A user consuming machine time and resources to perform unrelated or unauthorized work, possibly even playing games	Y	N	N

If you can tolerate most of the events listed, your security requirements are quite low. If your answers are equally mixed between yes and no, your requirements are in the medium to high range. Generally, those sites that are most intolerant to the events have very high levels of security requirements.

When reviewing security needs, do not confuse a weakness in site operations or recovery procedures as a security problem. Ensure that your operations policies are effective and consistent before evaluating your system security requirements.

## 11.3 The Secure System Environment

There are two sources of security problems outside the operating system domain: employee carelessness and facility vulnerability. If you have a careless or malicious employee or your facility is insecure, none of the security measures discussed in this guide will protect you from security breaches.

Most system penetration occurs through these environmental weaknesses. It is much easier to physically remove a small reel of tape than it is to break access protection codes or change file protection.

Digital strongly encourages you to stress environmental considerations as well as operating system protections when reviewing site security.

The following sections discuss VMS operating system security measures. When deciding on which of these measures to implement, it is important for you to assess site security needs realistically. While instituting adequate security for your site is essential, instituting more security than is actually necessary is costly and time-consuming.

When deciding which security measures to apply to your system, remember the following:

- The most secure system is also the most difficult to use.
- Increasing security can increase costs in terms of slower access to data, slower machine operations, and slower system performance.
- More security measures require more personnel time. (Increased security requires increased employee hours.)

The VMS operating system provides the basic mechanisms to control access to the system and its data. It also provides monitoring tools to ensure that access is restricted to authorized users. However, many computer crimes are committed by authorized users with no violation of the operating system's security controls.

Therefore, the security of your operation depends on how you apply these security features and how you control your employees and your site. By first building appropriate supervisory controls into your application and designing your application with the goal of minimizing opportunities for abuse, you can then implement VMS operating system security features and produce a less vulnerable environment.

Topics discussed in this chapter include the following:

- Setting up a site security policy
- Managing passwords
- Controlling break-in detection
- Displaying the break-in database

- Protecting files and directories with ACLs
- Creating a project account
- Security auditing

## 11.4 Managing Passwords

A site needing average security protection always requires use of passwords. Sites with more security needs frequently impose a double password scheme (see Section 11.4.3) requiring primary and secondary passwords, and possibly system passwords as well.

### 11.4.1 Initial Passwords

When you open an account for a new user with the Authorize Utility, you must give the user a user name and an initial password. When you assign temporary initial passwords, observe all guidelines recommended in Section 11.4.6. You should consider using the automatic password generator. Avoid any obvious pattern when assigning passwords.

To use the automatic password generator while using the Authorize Utility to open an account, add the `/GENERATE_PASSWORD` qualifier to either the `ADD` or the `COPY` command. The system responds by offering you a list of automatically generated password choices. Select one of these passwords, and continue setting up the account.

When you add a new user to the UAF, you might want to define that user's password as having expired previously using the `AUTHORIZE` qualifier `/PWDEXPIRED`. This forces the user to change the initial password when first logging in. The system behaves just as if the password had reached its expiration date, as described in Section 11.4.4.

Pre-expired passwords are conspicuous in the UAF record listing. The entry for the date of the last password change carries the following notation:

<pre-expired>

By default, the VMS operating system forces new users to change their passwords the first time they log in. Encourage your site to use a training program for its users that includes information about changing passwords frequently and other techniques that promote system security.

## 11.4.2 System Passwords

System passwords are used to control access to terminals that might be targets for unauthorized use, as follows:

- All terminals using dialup lines or public data networks for access
- Terminals on lines that are publicly accessible and not tightly secured, such as those at computer laboratories at universities
- Terminals not frequently inspected
- Terminals intended for use only as spare devices
- Terminals the security manager wants to reserve for security operations

Implementing system passwords is a two-stage operation involving the DCL commands `SET TERMINAL` and `SET PASSWORD`. First, you must decide which terminals require system passwords. Then, for each terminal, you enter the DCL command `SET TERMINAL/SYSPWD/PERMANENT`. When you are satisfied that you have selected the right terminals, incorporate these commands into the site-specific startup command procedure so that the terminal setup work is done automatically at system startup time. You can remove the restriction on a terminal at any time by invoking the DCL command `SET TERMINAL/NOSYSPWD/PERMANENT` for that terminal.

Then choose a system password and implement it with the DCL command `SET PASSWORD/SYSTEM`, which requires the `SECURITY` privilege. This command prompts you for the password and then asks you to re-enter it for verification, just as is done for user passwords. To request automatic password generation, include the `/GENERATE` qualifier.

To enable the use of the system password for the remote class of logins (those accomplished through the DCL command `SET HOST`), set the appropriate bit in the default terminal characteristics parameter using `SYSGEN`. This is bit 19 (hexadecimal value 80000) in the parameter `TTY_DEFCHAR2`. Note that if you set this bit, you must invoke the DCL command `SET TERMINAL/NOSYSPWD/PERMANENT` to disable system passwords for each terminal where you do not want the feature. (As before, consider placing the `SET TERMINAL` commands you have tested in the site-specific startup command procedure.) Follow the steps in the preceding paragraph to set the system password.

When choosing a system password, select a non-English string of characters and digits, with a minimum length of 6. The system password is not subject to expiration. Change the password frequently. Always change the system password as soon as a person who knows the password leaves. Share the system password only with those who need to know.

The system password is stored in a separate UAF record and cannot be displayed. The DCL command SET PASSWORD/SYSTEM (the normal means of setting and changing the system password) requires that you enter the old system password prior to changing it. Use the AUTHORIZE command MODIFY/SYSTEM\_PASSWORD to change the system password without having to specify the old password, as shown in the following command:

```
UAF> MODIFY/SYSTEM_PASSWORD=ABRACADABRA
```

The primary function of the system password is to form a first line of defense for publicly accessible ports and to prevent potential intruders from learning the identity of the system. However, requiring system passwords can appear unfriendly when authorized users are unaware that they are required on certain terminals. To avoid false reports of defective terminals or systems, inform your users which terminals allocated for their use require system passwords.

Where system passwords are not applied to either control access through dialup lines or on publicly accessed lines, few people might know the system password. There is the possibility of encumbered operations if the personnel who know the password are unavailable, incapacitated, or forget the password. Solve this problem by invoking AUTHORIZE and entering the MODIFY/SYSTEM\_PASSWORD command. The SYSPRV privilege is required.

### 11.4.3 Primary and Secondary Passwords

The use of dual passwords is cumbersome and mainly needed at sites with high-level security concerns. Dual passwords offer three advantages: when used on a widespread basis, they facilitate the verification of the physical identity of each user at login time through visual contact; when used in limited cases, they single out accounts that can be logged in to only when two persons are present; they also prevent accounts from being accessed through DECnet using simple access control.

Sites with medium security requirements might want to use dual passwords as a tool when there are unexplained break-ins after the password has been changed and the use of the password generator has been enforced. Select problem accounts, and make them a temporary target of this restriction. If the problem goes away when you institute personal verification through the secondary password, you know you have a personnel problem. Most likely, the authorized user is revealing the password for the account to one or more other users who are abusing the account.

Implement dual passwords with the AUTHORIZE qualifier /PASSWORD. To impose dual passwords on a new account, invoke AUTHORIZE and use the following syntax:

```
UAF> ADD username /PASSWORD="( "", secondarypwd)
```

## 11-8 System Security Issues

To impose a secondary password on an existing account, use the same syntax, substituting the `MODIFY` command for `ADD`. For example, to modify the account for user `JONES`, you could use the following command:

```
UAF> MODIFY JONES /PASSWORD=(olmifraj,bogfrapto)
```

This command does not affect the primary password that already exists for the account, but it adds the requirement that a secondary password be provided at each subsequent login. The secondary password acquires the same password lifetime and minimum length values in effect for the primary password. If the `/FLAGS=GENPWD` qualifier has been specified for this account, the secondary password can be changed only under the control of the automatic password generator.

**NOTE:** While secondary passwords can be specified for accounts requiring remote access using the `DCL` command `SET HOST`, they cannot be specified for accounts requiring network file access using access control strings. Do not specify secondary passwords on accounts that require network access, or request remote security managers to set up proxy accounts for those users requiring file access to other nodes in the network.

### 11.4.4 Enforcing Minimum Password Standards

Security managers can use `AUTHORIZE` to impose minimum password standards for individual users. Specifically, qualifiers and login flags provided by `AUTHORIZE` control the minimum password length, how soon passwords expire, and whether the user is forced to change passwords at expiration.

#### Password Expiration

With the `AUTHORIZE` qualifier `/PWDLIFETIME`, you can establish the maximum length of time that can elapse between password changes before the user will be forced to change the password or lose access to the account. By default, the value of `/PWDLIFETIME` is 180 days. You can change the frequency requirements for user password changes by specifying a different delta time value for the qualifier. For example, to require a user to change the password every 60 days, you would specify the qualifier as `/PWDLIFETIME=60-0`.

The `/PWDLIFETIME` qualifier applies to both primary and secondary user passwords, but not to the system password. Each primary and secondary password for a user is subject to the same maximum lifetime. However, the passwords can change at separate times. As soon as the user completes a password change, that individual password's clock is reset; the new password value can exist unchanged for the length of time dictated by `/PWDLIFETIME`.

The use of a password lifetime forces the user to change the password regularly. The lifetime can be different for different users. Users who have access to critical files generally should have the shortest password lifetimes.

System passwords have an unlimited lifetime. Therefore, change the system password regularly.

### Forcing Expired Password Changes

By default, users are forced to change expired passwords when logging in. Users whose passwords have expired are prompted for new passwords at login. This password feature is valid only when a password expiration date is specified with the `/PWDLIFETIME` qualifier.

To disable forced password changes, specify the following qualifier to the `ADD` or `MODIFY` command:

```
/FLAGS=DISFORCE_PWD_CHANGE
```

Once disabled, the forced password feature can be re-enabled by clearing the login flag, using the following syntax:

```
/FLAGS=NODISFORCE_PWD_CHANGE
```

Users who log in and are prompted to change expired passwords can abort the login by pressing `CTRL/Y`.

**NOTE:** If secondary passwords are in effect and both primary and secondary passwords have expired, the user is forced to change both passwords. If the user changes the primary password and presses `CTRL/Y` before changing the secondary password, the user is logged out, and no password change is recorded.

### Minimum Password Length

With the `AUTHORIZE` qualifier `/PWDMINIMUM`, you can direct that all password choices must be a minimum number of characters in length. Users can still specify passwords up to the maximum length of 31 characters.

The minimum length that you specify applies both to primary and secondary passwords and is required only when users change passwords with the `DCL` command `SET PASSWORD`. As system manager, you can specify initial passwords (with `AUTHORIZE`) that are shorter than the minimum. However, doing so could confuse your users unnecessarily. Furthermore, initial passwords inherently introduce security weaknesses. By selecting short initial passwords, you compound the problem. Generally, it is good practice to observe the same rules you expect your users to follow.

A minimum password length is always in effect for each user. The default minimum password length for each user is the value for `PWDMINIMUM` for the `DEFAULT` account. You can determine this value with the following sequence:

```
$ SET DEFAULT SYSS$SYSTEM
$ RUN AUTHORIZE
UAF> (SHOW DEFAULT)
```

## 11-10 System Security Issues

Use the `/PWDMINIMUM` qualifier to specify a different minimum password length for any individual user or to change the default minimum password length in the `DEFAULT` account.

If a user enters the DCL command `SET PASSWORD/GENERATE=n` to generate new password choices automatically,  $n$  must be a value at least as great as the minimum value in effect. If  $n$  is less than the current minimum enforced in the UAF, it is disregarded; no message appears. The five password choices that the VMS operating system generates for the user comply with the current minimum password length.

The password generator creates passwords that range in length between  $n$  and  $n+2$ , where  $n$  is the specified or minimum password length. In addition, the maximum values for  $n$  and  $n+2$  that the password generator can accommodate are 10 and 12, respectively. Longer passwords require an inordinate amount of CPU time to generate.

The system password is not subject to a minimum length. Guidelines that apply to user passwords are equally applicable to system passwords. Choose system passwords that are 6 to 10 characters long.

### 11.4.5 Requiring the Password Generator

The `/FLAGS=GENPWD` qualifier in `AUTHORIZE` allows you to force the use of the automatic password generator when a user changes a password. At some sites, all accounts are created with this qualifier. At other sites, the security manager can be more selective.

Criteria for requiring use of the password generator should be whether or not the user might have access to sensitive data that must not be compromised by a break-in.

If your policy is to request voluntary use of the password generator, and users are not cooperating, you can force users to use the password generator by adding the `/FLAGS=GENPWD` qualifier to most or all user accounts. You can also add the `AUTHORIZE` qualifier `/FLAGS=LOCKPWD` to user accounts to prevent users from changing passwords. Only you as system manager will be authorized to change passwords.

### 11.4.6 Protecting Passwords

Observe the following guidelines to protect passwords:

- Make certain the password for the `SYSTEM` account, which is a standard account on all VMS systems, is secure and is changed regularly.
- Disable any accounts that are not used regularly with the `AUTHORIZE` qualifier `/FLAGS=DISUSER`.



- Do not permit an outside or an in-house service organization to dictate the password for an account they use to service your system. Such service groups tend to use the same password on all systems, and their accounts are usually privileged. On seldom-used accounts, set the AUTHORIZE flag DISUSER, and enable the account only when it is needed. You can also change the password immediately after each use and notify the service group of the new password.
- Delete accounts no longer in use.
- If you have an account on a system that stores passwords in plaintext (unencrypted), choose a different password on all of your other accounts.
- Do not leave listings where they could be read or stolen.
- Maintain adequate protection of authorization files. Note that the system user authorization file (SYSUAF.DAT) and network proxy authorization file (NETPROXY.DAT) are owned by the system account ([SYSTEM]). There should be no other users in this group. Accordingly, the categories SYSTEM, OWNER, and GROUP are synonymous. Normally the default UIC-based file protection for these authorization files is adequate.

The following actions are not strictly for password protection, but they reduce the potential of password detection or limit the extent of the damage if passwords are discovered or bypassed:

- Avoid giving multiple users access to the same account.
- Protect telephone numbers for dialup lines connected to your system.
- Make all accounts that do not require a password captive accounts.
- Extend privileges to users carefully.
- Ensure that the files containing components of the operating system are adequately protected.

## 11.5 Controlling Break-In Detection

This section describes how to set up break-in detection and evasion and how to display the break-in database.

### 11.5.1 Controlling the Number of Retries on Dialups

You can control the number of login attempts the user is allowed through a dialup line. If the user makes a typing mistake after obtaining the connection, the user does not automatically lose the connection. This option is useful for authorized users, while still restricting the number of unauthorized attempts.

## 11-12 System Security Issues

To implement control of retries, use the following two **LGI parameters** provided with **SYSGEN**: **LGI\_RETRY\_TMO** and **LGI\_RETRY\_LIM**. If you do not change the parameters, the default values allow the users three retries with a 20-second interval between each. This means that users will lose the connection only if they fail to specify a valid password in three tries, or if they spend more than 20 seconds between two of their tries.

Note that these values apply to every user on the system who is permitted to access the system through a dialup line.

The following example illustrates setting the total number of retry attempts to six, allowing a half-minute interval between tries. Because these **LGI** parameters are dynamic, you could change them and test them before performing the **SYSGEN** command **WRITE CURRENT** and rebooting the system.

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> SET LGI_RETRY_LIM 6
SYSGEN> SET LGI_RETRY_TMO 30
SYSGEN> WRITE ACTIVE

{OPCOM messages show modification has been made}

SYSGEN> EXIT
$
```

### 11.5.2 Controlling Break-In Detection and Evasion

Section 11.5.1 shows how to control the number of login retries for users dialing in. By limiting the number of retries to a reasonable number on each dialup login, you make the job of dialing up and trying every password combination more difficult for outsiders.

You should keep in mind that controlling dialup retries is only a part of an overall security program and is not, in itself, sufficient to avoid break-ins. An obstacle like redialing is not going to prove an effective deterrent to a persistent intruder; moreover, this technique applies only to dialups.

The VMS operating system offers additional methods of discouraging break-in attempts. These methods also use **SYSGEN** parameters in the **LGI** category. One of the parameters (**LGI\_BRK\_LIM**) defines a threshold count for login failures. When the count of login failures exceeds the **LGI\_BRK\_LIM** value within a reasonable time interval, the system assumes that a break-in is in progress. Only login failures caused by specifying invalid passwords are counted, and they must be from a specific source. That source can be any of the following combinations:

- A specific terminal and a specific valid user name. As described in a following section, you can override this default to count failures by user name only. Attempted logins using invalid user names never trigger break-in detection; however, they are counted together as a single class per terminal and are used to trigger security alarms. (See Section 11.8 for information about security alarms.)

- A specific remote node and a specific remote user name.
- The user name of the creator of a detached process.

By default, LGI\_BRK\_LIM permits five failed login attempts from one of these sources. (Security managers can adjust the value of LGI\_BRK\_LIM with SYSGEN.)

The SYSGEN parameter LGI\_BRK\_TERM controls the association of terminals and user names for counting failures. By default, the VMS operating system sets this parameter to 1 so that terminals and login failures are tracked together. If you set this parameter to zero (0), the terminal is not included in the association; the failures are counted based on user name only.

Another key parameter, LGI\_BRK\_TMO, controls the time period in which login failures are detected and recorded. The initial failure on each source is given an expiration time that represents the current time plus the delta time given by LGI\_BRK\_TMO. Each additional failure on that source adds another delta of LGI\_BRK\_TMO to that entry, thus extending the length of time that breakin detection is in effect. The cumulative effect is that the more failures made by a source, the greater the window of time in which additional failures will count toward the critical number defined by LGI\_BRK\_LIM. If no more failures occur by the time the expiration point is reached, the number of accumulated failures for that source is reset to zero. Note, however, that the failure count is not reset by a successful login.

For example, assume the default values are in effect. LGI\_BRK\_LIM specifies no more than five login failures from one source. LGI\_BRK\_TMO is set for 5 minutes. Assume that an outsider starts sending user names and passwords to the system. When the first password fails, the clock starts to run and the user has four more tries in the next 5 minutes. When the second attempt fails about 30 seconds later, the user has three tries left that will be counted over the next 9.5 minutes. When the third attempt fails 30 seconds later, the login failure observation time extends to 14 minutes. The fourth failure occurs about one minute later; the fifth failure occurs within another 30 seconds. By this time, the observation time has reached 22.5 minutes. As a result, the next login failure from that source within 22.5 minutes will trigger evasive action.

The system tolerates an average rate of login failures that is the reciprocal of the parameter LGI\_BRK\_TMO. For example, if the default value of LGI\_BRK\_TMO (300 seconds or 5 minutes) is in effect, the average rate of tolerable login failures is one every 5 minutes. When the rate of login failures exceeds the tolerable rate, and the critical number of five failures is reached (the default value of LGI\_BRK\_LIM), the system concludes a break-in is in progress and initiates evasive action.

The system stops accepting logins from the offending source for a period of time. When the source is a terminal (when LGI\_BRK\_TERM equals 1), for a period of time no one can log in from that terminal with the user name that is under suspicion. (However, other users can log in from that terminal.) A remote user triggering break-in evasion is prohibited from logging in from that node for a

## 11-14 System Security Issues

period of time. Consequently, login attempts that provide valid user name and password combinations that should otherwise succeed are rejected during this interval, but only from the presumed intruder at that source. Once the interval elapses, operations return to normal. As a result of this form of evasive action, outsiders are less likely to learn the correct password by using repetitive login attempts.

The duration of the evasive action is controlled by the LGI\_HID\_TIM parameter. The length of time depends on an additional random number (in the range of 1 to 1.5) used as a multiplier. The product of LGI\_HID\_TIM and the random number yields the actual duration of evasive action. The formula could be represented as follows:

$$\text{Evasion time} = \text{LGI\_HID\_TIM} * (\text{random number})$$

The inclusion of a random amount of time helps obscure the true evasion time. An outsider who learned the value of LGI\_HID\_TIM could not be assured that the evasive action would persist for exactly that length of time.

The parameters described in the previous sections affect all terminals, users, and nodes that access the system. Because these parameters are dynamic, you can reset them without rebooting the system.

If the values of LGI\_BRK\_LIM and LGI\_BRK\_TMO can be learned or guessed, the outsider can attempt a system break-in over sufficiently long intervals that suspicion is not triggered. The outsider can also change terminals, nodes, and user names frequently enough to avoid detection. Do not rely on these break-in techniques as the sole means of security on your system.

The technique of counting failures per terminal and user name raises the potential for break-in because the password guess rate for a particular user name is multiplied by the number of available terminals. Each terminal is counted as a separate source for break-in detection. The benefit of this approach, however, is that it sharply reduces the denial of service problem that could result from simply counting failures per terminal or per user name. (A malicious user could disable an entire terminal room or user's account for a period of time if failures are counted for each user name alone.)

By setting LGI\_BRK\_TERM to zero, you can detect attempts more quickly, at the expense of increasing the risk of denial of service to legitimate users.

The SYSGEN parameter LGI\_BRK\_DISUSER makes the effects of break-in detection more severe. If you set this parameter to 1, the VMS operating system sets the DISUSER flag in the UAF record for the account where the break-in was attempted. Thus, that user name is disabled until you manually intervene. However, the service denial effects of this option can be very severe. A malicious user can put all known accounts, including yours, out of service in a short time. To recover, you must log in on the system console where the SYSTEM account is always allowed to log in. The VMS operating system stores information in the break-in database about login failures that originate from a specific source.

### 11.5.3 Displaying the Break-In Database

Use the DCL command `SHOW INTRUSION` to display the contents of the break-in database and the `DELETE/INTRUSION_RECORD` command to remove entries from the break-in database. See the Reference Section of the *VMS User's Manual* for additional information about these commands. Entries in the break-in database have the following format:

```
Intrusion      Type      Count      Expiration      Source
```

The information provided in the fields in each entry is as follows:

Intrusion	Class of intrusion.
Type	Severity of intrusion as defined by the threshold count for login failures. The <code>SYSGEN</code> parameter, <code>LGI_BRK_LIM</code> , defines the threshold count.
Count	Number of login failures associated with a particular source.
Expiration	Absolute time when the VMS operating system stops keeping track of login failure. The <code>SYSGEN</code> parameter, <code>LGI_BRK_TMO</code> , controls this time.
Source	Origin of the login failure.

The information in the break-in database is controlled by the `SYSGEN` parameters in the `LGI` category.

## 11.6 Protecting Files and Directories with ACLs

The VMS operating system offers two primary protection mechanisms. The first, **standard UIC-based protection**, is based on the user identification code (UIC) and is applied to all user files. It controls access to files according to the user categories `SYSTEM`, `OWNER`, `GROUP`, and `WORLD`.

The second file protection mechanism uses **access control lists (ACLs)**, which employ a more refined level of protection on files than that available with UIC-based protection. ACLs can be used to grant or deny file access to individual users or groups of users, independent of the UIC.

It is assumed that you are familiar with the default file protection available with the UIC-based protection scheme. For more information about file protection based on UICs, see the *VMS User's Manual*.

ACLs are important file protection tools available to all VMS users and are generally used at sites with medium to high security requirements. ACLs are also prevalent in environments with complex patterns of file sharing. As security requirements increase, so does the use of ACLs.

ACLs consist of access control list entries (ACEs) that grant or deny access to system objects, such as files and devices. Each ACE specifies a user or group of users and the type of access permitted. ACLs define access more precisely than the default UIC-based protection scheme by allowing you to create groups of users independent of the users' UICs.

The VMS operating system provides a file called a **rights database** that contains a list of special names called **identifiers** as well as a list of the users specified as **holders** of identifiers. The security manager uses the AUTHORIZE to maintain the rights database, adding and removing identifiers and holders of identifiers as necessary. By allowing groups of users to hold identifiers, the manager has created a group designation that differs from the one used with the user's UIC. This alternative method of grouping is more finely tailored to the uses the holders of the identifier are expected to make of the objects. This method also permits each user to be a member of multiple overlapping groups.

Each time you log in, the system creates a **process rights list** for you containing a list of the identifiers in the rights database associated with your process. When you attempt to access objects protected with ACLs, the system searches the object's ACL for an identifier granting access that matches one of the identifiers in your process rights list.

The following sections describe the relationship between ACLs and identifiers in more detail.

### 11.6.1 Creating and Maintaining ACLs

Use the VMS ACL Editor to create and edit an ACL on a specific object. You can also use the DCL command SET ACL to manipulate (add, delete, or copy) entire ACLs or individual ACEs on more than one object at a time.

The following DCL commands can be used to display ACLs:

- SHOW ACL
- DIRECTORY/ACL
- DIRECTORY/SECURITY
- DIRECTORY/FULL

In general, you will find the DCL commands SET ACL and SHOW ACL sufficient for creating and displaying most ACLs, although the ACL editor is an important utility for more extensive ACL work.

### 11.6.2 Identifiers

Identifiers in an ACL specify the users who are allowed or denied access to an object. Following are the three types of identifiers:

- **UIC identifiers**—Depend on the user identification codes (UICs) that uniquely identify each user on the system. Typically the UIC identifiers are presented in numeric or abbreviated alphanumeric format. For example, a UIC identifier might adopt the numeric format of the UIC, such as [306,210], or just the member name from the alphanumeric format UIC, such as JONES, where the full alphanumeric UIC is [GROUP1,JONES].

- **General identifiers**—Defined by the security manager in the system rights database to identify groups of users on the system. For example, TERM3BIO, WARD5WORKERS, DATAENTRY, and RESERVDESK would identify the third term biology students, the campaign workers for Ward 5, the data entry personnel, or the people who handle the reservations desk, respectively.
- **System-defined identifiers**—Describe certain types of users based on their use of the system. For example, BATCH, NETWORK, DIALUP, INTERACTIVE, LOCAL, and REMOTE correspond directly to the type of login the user executed.

When you log in, the identifiers you hold in the rights database (including your UIC and your system-defined identifiers) are copied into a rights list that is part of your current process. The rights list is the structure that the VMS operating system uses to perform all protection checks. Additional identifiers might appear in your rights list; they were put there either by VMS Login software or by software specific to your installation. These identifiers represent qualifications about your login and the state of the system.

### UIC Identifiers

While the most common types of UIC identifiers are either numeric format UICs or user names, full alphanumeric UICs or UICs in hexadecimal format are accepted as UIC identifiers. Thus, you might see the following UIC identifiers:

```
[PROGRAMMERS,J_JONES]      {alphanumeric format UIC}
J_JONES                    {username from alphanumeric format UIC}
[341,311]                  {numeric format UIC}
%X08001006                 {hexadecimal format UIC}
```

Each of these formats uniquely identifies a user.

The system automatically adds a UIC identifier to the system rights database when each new account is created.

### General Identifiers

A general identifier, defined in the system rights database, is an alphanumeric string of 1 to 31 characters that must contain at least one alphabetic character. It can include the characters A through Z, dollar signs (\$), underscores (\_), and the numbers 0 through 9.

Use the Authorize Utility (AUTHORIZE) to create general identifiers in the system rights database and to assign them to system users, as follows:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> ADD/IDENTIFIER PROJECTX
UAF> GRANT/IDENTIFIER PROJECTX WILLIAMS
```

See the Reference Section for descriptions and examples of all AUTHORIZE commands.

### System-Defined Identifiers

System-defined identifiers are automatically defined by the system when the rights database is created at system installation time. The following identifiers, which correspond directly to the possible types of login classes, are system defined:

BATCH	All access attempts made by batch jobs.
NETWORK	All access attempts made by DECnet tasks.
INTERACTIVE	All access attempts made by interactive processes.
LOCAL	All access attempts made by users logged in at local terminals.
DIALUP	All access attempts made by users logged in at dialup terminals.
REMOTE	All access attempts made by users logged in through a network.

In addition, a system node identifier of the form `SYS$NODE_node_name` is created by the site-independent startup procedure (`STARTUP.COM` in `SYS$SYSTEM`).

A user automatically becomes a holder of one or more of these identifiers during login. The VMS Login software adds the appropriate identifiers to the process rights list.

### 11.6.3 Access Control List Entries

As shown in the previous section, you create identifiers in the rights database and assign users as holders of the identifiers. You then define which access to grant or deny holders of the identifier for each object (such as a file) requiring this level of protection. Because several identifiers might be required to represent access needs for each object, it is typical to create a list of multiple entries. Each entry defines the access rights to be granted or denied the holders of the identifier named in that entry. This list is the Access Control List, or ACL. Each entry in this list is called an access control list entry (ACE).

Like the defaults for UIC-based protection, you can set up default ACLs. As a result, some users might be unaware that their files have ACLs and might never change ACLs themselves. Other users are actively involved in creating and maintaining their own ACLs.

To summarize, ACLs can be created by the system by default, by you for specific objects, and by users to protect their own files. Users can create ACLs only for objects they own (typically files residing in their directories) or to which they have the same access as the object owner.

An ACL consists of ACEs that grant or deny access to a particular system object, such as a file, a directory, or a device. Because ACLs can define access more selectively than UIC-based protection, ACLs allow users to fine tune the action taken when access is requested for an object. Typically, you use ACLs to provide users from several UIC groups access to a system object without having to grant



WORLD access to the object. ACLs can perform other functions, such as directing security alarms to be set off when access to an object succeeds or fails.

When the system receives a request for access to an object that has an ACL, the system searches each entry in the ACL sequentially for the first match. It stops searching at the first match. If another match exists further down in the ACL, it has no effect. Thus, ACEs that identify specific users should appear in the ACL before ACEs that identify broader classes of users, as follows:

```
(IDENTIFIER=WILLIAMS,ACCESS=READ+EXECUTE)
(IDENTIFIER=CS101,ACCESS=NONE)
```

Assume that user WILLIAMS holds the CS101 identifier. In the previous example, WILLIAMS is granted READ and EXECUTE access to the object. If the ACEs were switched, user WILLIAMS might be denied access to the object.

The use of ACLs is optional. Although the use of ACLs can enhance the security of system objects in any installation through a more detailed definition of who is allowed what kind of access, user time must be spent in creating and maintaining the ACLs, and processor time is required to perform the functions that ACLs mandate.

Each ACL consists of one or more ACEs. There is no limit to the number of ACEs that an ACL can contain or to the number of characters in an ACE; however, very long ACLs increase the amount of time necessary to gain access to an object.

The most common type of ACE is the Identifier ACE, which controls the type of access allowed to a particular user or group of users.

In general, the format of an ACE is as follows:

```
(type[,options][,access_to_grant])
```

### Identifier ACE

An identifier ACE controls the types of access allowed to specific users based on user identification. Following is the format for an identifier ACE:

```
(IDENTIFIER=identifier[,options][,access])
```

**Specifying Identifiers in Identifier ACEs:** The first field in the identifier ACE is the keyword IDENTIFIER followed by one or more identifiers,

The system takes the access action included in the ACE only for the user who matches all the identifiers. For example, if you wanted to grant read access to user [301,25] running a batch job, you would specify the identifier ACE as follows:

```
(IDENTIFIER=[301,25]+BATCH,ACCESS=READ)
```

Although it is unusual for a number of users to share the same UIC, it is likely that a number of users will share the same general identifier. Users with the same general identifier do not need to be in the same UIC-based group. Furthermore, a single user can be associated with a number of different general identifiers as defined in the rights database. The creator of an ACL has considerable flexibility in selecting sets of users and defining access capabilities for them.

For example, the user identified by the UIC [301,25] is a member of the UIC-based group 301. That user might be the only member of group 301 who is also associated with the general identifier PERSONNEL. An ACE defining a particular type of access for the users associated with the general identifier PERSONNEL grants that type of access to that user, but not to the other members of group 301.

**Specifying Options in Identifier ACEs:** The options field in an identifier ACE controls whether an ACE is copied to new versions of the file, can be displayed, or can be deleted. This field in an identifier ACE begins with the keyword OPTIONS and takes one or more of the following keywords:

DEFAULT	Indicates that an ACE is to be included in the ACL of any files created within a directory. When the ACE is propagated, the DEFAULT indicator is removed from the ACL of the created file. This option is valid only for directory files. A default ACE does not grant or deny access; it just affects the ACL of new files.
HIDDEN	Indicates that this ACE should be changed only by the application that added it. The ACL editor does not permit modification or deletion. Thus, the ACL editor displays the ACE only to show its relative position within the ACL, not to facilitate editing of the ACE. The DCL DIRECTORY and SHOW ACL commands do not display hidden ACEs.
PROTECTED	Indicates that an ACE will be preserved even when an attempt is made to delete the entire ACL. A protected ACE must be deleted specifically with the ACL editor or by specifying the ACE on the command line of the DCL command SET ACL.
NOPROPAGATE	Indicates that, when copying an ACL from one version of a file to a later version of the same file, the ACE is not copied to the newer version.
NONE	Indicates that no options apply to an ACE. Although you can enter OPTIONS=NONE when you create the ACE, OPTIONS=NONE is not displayed when the ACE is displayed.

Connect multiple options with plus signs (+). If you specify any other options with the NONE option, the other options take precedence.

**Identifier ACE for a Directory:** The OPTIONS=DEFAULT option of an identifier ACE allows users to define one or more default ACEs for inclusion in the ACLs for files created in a particular directory. A default ACE is supplied for all new files created in that directory; any existing files are not supplied with the default ACE. Thus, if you want all files in the directory [MALCOLM] to have an ACE that permit read and write access to users with the PERSONNEL identifier, you could include the following ACE in the ACL for the file MALCOLM.DIR:

(IDENTIFIER=PERSONNEL,OPTIONS=DEFAULT,ACCESS=READ+WRITE)

As a result of this ACE, any file created in the [MALCOLM] directory has the following ACE:

```
(IDENTIFIER=PERSONNEL,ACCESS=READ+WRITE)
```

Notice that the DEFAULT option does not appear in the file's ACE. However, any subdirectory created in the MALCOLM directory has the DEFAULT option as part of its ACE so that the default ACE will be propagated throughout the entire directory tree.

**Specifying Access in Identifier ACEs:** The third field in an identifier ACE specifies what type of access you are allowing the users identified in the first field of the ACE. This field begins with the keyword ACCESS followed by a string of access actions connected by plus signs. The following types of access are allowed in an identifier ACE:

READ	Accessor can read a file, read from a disk, or allocate a device.
WRITE	Accessor can read or write a file.
EXECUTE	Accessor can execute an image file or look up entries in a directory by explicitly specifying file names.
DELETE	Accessor can delete a file.
CONTROL	Accessor has all the privileges of the object's owner.
NONE	Accessor has no access to the object.

**Sample Identifier ACEs:** The most common type of ACL is one that defines the access to a file for a group of users. In the following ACL example, access to a file is based on the identity of a user. PERSONNEL, SECURITY, and SECRETARIES are general identifiers assigned to appropriate sets of users by the system manager using AUTHORIZE. NETWORK is a system-defined identifier, while [20,\*] and [SALES,JONES] are examples of UIC identifiers.

```
(IDENTIFIER=SECURITY,OPTIONS=PROTECTED,ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL)
(IDENTIFIER=PERSONNEL,ACCESS=READ+WRITE+EXECUTE+DELETE)
(IDENTIFIER=SECRETARIES,ACCESS=READ+WRITE)
(IDENTIFIER=[20,*],ACCESS=READ)
(IDENTIFIER=NETWORK,ACCESS=NONE)
(IDENTIFIER=[SALES,JONES],ACCESS=NONE)
```

In the preceding example, the ACE providing the greatest amount of file access is listed at the top of the ACL. Any users holding both the SECURITY and PERSONNEL identifiers obtain maximum access rights through the first match, which is the SECURITY identifier. In this example, the user with UIC [SALES,JONES] is prohibited from any access to the file unless that user also happens to have one of the general identifiers (which is an oversight on the part of the creator of the ACL). If the ACL creator wants to be absolutely certain that the user with UIC [SALES,JONES] could not possibly gain access to the file, the ACE at the bottom of the ACL should be moved to the top.

The order of the ACEs in the example permits a number of users to gain types of file access over the DECnet-VAX network. The users with the identifiers of SECURITY, PERSONNEL, SECRETARIES, and UIC [20,\*] can all gain some access over the network, although only those with the identifier SECURITY can gain full access. The fifth ACE prevents all other users from network access. While this might be the intent of the ACL creator, it would be an unfortunate oversight if it were not. Remember that the system searches the ACL sequentially and grants the user only the access specified in the first matching ACE. All subsequent ACEs are ignored.

The first ACE is the only ACE containing an option field (the PROTECTED option). Using this option prevents the first ACE from being deleted unless you have explicitly deleted the ACE with the ACL editor, or unless you have specified the ACE with the SET ACL/DELETE command.

### 11.6.4 Summary of ACLs

The following recommendations will help you manage ACLs:

- Do not assume that specifying ACCESS=NONE for an identifier will absolutely prohibit the holders of the identifier from accessing the object. Frequently, users in either the SYSTEM or OWNER category are still entitled to whatever access the UIC-based protection affords that category. If the users hold special privileges, they might be granted the access requested through the privilege.
- Watch out for errors in the order that ACEs appear in the ACL. Place the ACEs that deny access to specific users at the top of your ACLs, so that the user will not obtain access by holding another identifier. Sometimes you use wildcards in the UIC-format identifiers to deny access to large groups of users. Such an ACE properly belongs at the bottom of the ACL, not at the top. Place the ACEs that grant the widest access rights immediately before the most restrictive ACEs. This technique ensures that users who hold multiple identifiers do not obtain restricted access rights on the first match when another identifier they hold could grant more generous rights. Remember that a user can receive only the access rights granted through the first matching identifier.
- Do not place ACLs on all objects. This is usually unnecessary even at medium-level security sites. Too many ACLs can cause performance penalties to appear on the system. Instead of using ACLs, group files so that only a few directories need default ACEs that propagate to many or all files.
- Use general identifiers to create practical groups of users to avoid unnecessarily long ACLs.
- Update ACLs when you delete user accounts. Always maintain the shortest and most current ACLs. Again, using general identifiers instead of individual users helps alleviate this maintenance problem.

## 11.7 Creating a Project Account

To allow for more flexible management and accounting of disk space, identifiers can carry the optional resource attribute. This attribute, when present on an identifier, allows file space to be owned by and charged to that identifier. Thus, when a project or department-specific identifier is the owner of a directory, the space used by files created in the directory can be charged to the appropriate department or project rather than to the individual who created them. When users work on multiple projects, they can charge their disk space requirements to the related project rather than to their personal accounts.

Another important advantage of setting up a project account is to give control of the protection of the account and its files to you (the system manager) rather than to the users of the account. This helps to assure that all files created within the project account will be adequately and uniformly protected.

To set up a project identifier and directory, perform the following steps:

1. Using **AUTHORIZE**, create the project identifier with the resource attribute in the rights database. The following example creates the identifier **PROJECTX**:

```
$ RUN SYS$SYSTEM:AUTHORIZE
UAF> ADD/IDENTIFIER PROJECTX /ATTRIBUTES=RESOURCE
```

2. Grant the identifier to the appropriate individuals with the resource attribute.

```
UAF> GRANT/IDENTIFIER PROJECTX user1 /ATTRIBUTES=RESOURCE
UAF> GRANT/IDENTIFIER PROJECTX user2 /ATTRIBUTES=RESOURCE
```

```

.
.
.
```

3. Create the disk quota authorization for the project identifier. For example, the following command invokes the VMS System Management (**SYSMAN**) Utility and assigns the identifier **PROJECTX** 2000 blocks of disk quota with 200 blocks of overdraft:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> DISKQUOTA ADD PROJECTX /PERMQUOTA=2000 /OVERDRAFT=200
```

4. Create the project directory. For example, the following **DCL** command creates the project directory [**PROJECTX**] and establishes the identifier **PROJECTX** as the owner:

```
$ CREATE/DIRECTORY [PROJECTX] /OWNER=[PROJECTX]
```

5. Set up the necessary **ACL** on the project directory to allow holders of the **PROJECTX** identifier access to the directory. For example, the following **DCL** command places an **ACL** on the directory [**PROJECTX**] that permits any holder of the identifier **PROJECTX** to gain **READ**, **WRITE**, or **EXECUTE** access to the directory. The second **ACE** specifies that files created in the directory will receive the same **ACE** as a default.

## 11-24 System Security Issues

```
$ SET DIRECTORY [PROJECTX] /ACL= (-  
_ $ (IDENTIFIER=PROJECTX, ACCESS=READ+EXECUTE) , -  
_ $ (IDENTIFIER=PROJECTX, OPTIONS=DEFAULT, ACCESS=READ+EXECUTE) )
```

Access must be granted through ACL entries, since the owner identifier of the directory and the files does not match the UIC of any of the project members; thus, only SYSTEM and WORLD access are available through the UIC-based protection mask. The first ACE of the specified ACL gives all project members READ and EXECUTE access to the directory; the second ACE gives the same access for all files created in the directory. (The DEFAULT option in the second ACE specifies that the ACE is to be copied to each file created in the directory.)

Note that project members are not allowed to delete (or control) files created by others. However, the members each have complete access to files they have created in the directory, because the file system supplies an additional ACE that grants the file creator CONTROL access plus the access specified in the OWNER field of the UIC-based protection mask. This ACE only appears when the owner of the created file does not match the UIC of the creator, as is the case for files created in an account owned by a project identifier.

Thus, when project member CRANDALL creates files in the [PROJECTX] directory, the files receive the following access control list:

```
(IDENTIFIER=CRANDALL, OPTIONS=NOPROPAGATE, ACCESS=READ+WRITE+EXECUTE+DEFAULT+CONTROL)  
(IDENTIFIER=PROJECTX, ACCESS=READ+EXECUTE)
```

This example assumes that the OWNER field grants full (RWED) access. Because this is not always true (the systemwide default set by the SYSGEN parameter RMS\_FILEPROT might have been changed, or a user might have specified a process-specific default protection mask with the DCL command SET PROTECTION/DEFAULT), you might want to ensure consistency by providing a default protection ACE in the project directory ACL, as follows:

```
$ SET DIRECTORY [PROJECTX] /ACL= (-  
_ $ (DEFAULT_PROTECTION, S:RWED, O:RWED, G, W) , -  
_ $ (IDENTIFIER=PROJECTX, ACCESS=READ+EXECUTE) , -  
_ $ (IDENTIFIER=PROJECTX, OPTIONS=DEFAULT, ACCESS=READ+EXECUTE) )
```

The UIC protection specified in the default protection ACE is applied to all files created in the project directory.

## 11.8 Security Auditing

Security alarms are messages sent to the operator's terminal indicating specific events. Alarms can help you detect outsiders' attempts to break into the system and can be used to monitor undesirable activity at your site. For example, you might enable an alarm that sends a message to the operator's terminal whenever a UAF record changes.

When dealing with security alarms, carefully select and enable the events to be audited, enable an operator terminal, and monitor and make use of the alarm information.

## 11.8.1 Enabling Security Alarms

Before enabling security auditing on your Local Area VAXcluster, ensure that the Operator Communications (OPCOM) process has been started in your site-specific startup command procedure. The `AUDIT_SERVER` process is the mechanism used to write all security alarms to the system security audit log file.

To enable security auditing, specify the DCL command `SET AUDIT` in the following format:

```
SET AUDIT /ALARM /ENABLE=keyword[,...]
```

By default, the `AUTHORIZATION` and `BREAKIN` keywords are enabled by default when you use the `SET AUDIT /ALARM` command. The `AUTHORIZATION` keyword monitors modifications to the system UAF file, network proxy authorization file, rights database, or changes to system and user passwords; the `BREAKIN` keyword monitors successful break-in attempts.

In addition to `AUTHORIZATION` and `BREAKIN`, you can enable other classes of events by listing one or more of the following keywords to the `/ENABLE` qualifier:

- `ACL`—Event requested by an ACL on a file or global section
- `ALL`—All possible events
- `FILE_ACCESS`—Selected types of access (privileged and nonprivileged) to files and global sections
- `INSTALL`—Installation of images
- `LOGFAILURE`—Failed login attempt
- `LOGIN`—Successful login attempt
- `LOGOUT`—Logout
- `MOUNT`—Volume mounts and dismounts

See the Reference Section of the *VMS User's Manual* for more information about the `SET AUDIT` command.

Because security auditing affects system performance, enable security alarms only for the most important events. The following security alarm features are presented in order of decreasing priority and increasing system cost:

1. Enable security auditing for `LOGFAIL` and `BREAKIN`. This is the best way to detect probing by outsiders (and insiders looking for accounts). All sites needing security should enable alarms for these events.
2. Enable security auditing for `LOGIN`. Auditing successful logins from the more suspicious sources like `REMOTE` and `DIALUP` provides the best way to track which accounts are being used. An audit record is written before a user's identity can be disguised.

3. Enable the FILE=FAILURE security audit. This technique audits all file protection violations and is an excellent method of catching probers.
4. Apply ACL-based file access auditing to detect write access to critical system files. The most important files to audit are shown in Table 11-2. You might want to audit only successful access to these files to detect penetrations, and you might want to audit access failures to detect probing as well.

Note that some files in Table 11-2 are written during normal system operation. For example, SYSUAF.DAT is written during each login, and SYSMGR.DIR is written when the system boots.

5. Audit use of privilege to access files (either write access or all forms of access). Implement the security audit with FILE=(SYSPRV,BYPASS,READALL,GRPPRV). Note that this class of auditing can produce a large volume of output because privileges are often used in normal system operation for such tasks as mail delivery and operator backups.

**Table 11-2: System Files Benefiting from ACL-Based File Access Auditing**

Device and Directory	File Name
SYS\$SYSTEM	AUTHORIZE.EXE
	F11BXQP.EXE
	LOGINOUT.EXE
	DCL.EXE
	JOBCTL.EXE
	JBCSYSQUE.DAT
	SYSUAF.DAT
	NETPROXY.DAT
	RIGHTSLIST.DAT
	STARTUP.COM
SYS\$LIBRARY	SECURESHR.EXE
SYS\$MANAGER	SYSTARTUP_V5.COM
	VMSIMAGES.DAT
SYS\$SYSROOT	[000000]SYSEXE.DIR
	[000000]SYSLIB.DIR
	[000000]SYS\$LDR.DIR
	[000000]SYSMGR.DIR



## 11.8.2 Enabling an Operator Terminal

Before you enable alarms for particular events, enable an operator's terminal. Choose a terminal that provides hardcopy output and is in a secure location. The following DCL command enables the terminal from which the command is entered:

```
$ REPLY/ENABLE=SECURITY
```

Any number of terminals can be enabled as operators. If you designate one terminal as the operator's terminal, add the following lines to the site-specific startup command procedure (usually `SYS$MANAGER:SYSTARTUP_V5.COM`) to send alarms to the terminal and disable them on the system console:

```
$ DEFINE/USER SYS$COMMAND OPA0:
$ REPLY/DISABLE=SECURITY
$ DEFINE/USER SYS$COMMAND TTA3:
$ REPLY/ENABLE=SECURITY
```

Security audits are always written to the system security audit log file even if no operator terminal is enabled.

## 11.8.3 Enabling Alarm Messages

After you enable an operator terminal, enable specific alarm events with the `SET AUDIT/ENABLE` command. Alarm messages are then sent to the operator terminal when the selected events occur. Security alarms appear as follows:

```
***** OPCOM 19-APR-1990 12:27:52.26 ***** ①
Security alarm on LASSIE / System UAF record modification ②
      Time:          19-APR-1990 12:27:52.25 ③
      PID:           23C00155 ④
      User Name:     MENACE ④
      Rec Mod:       GOWER
      Fields Mod:    PRIVILEGES
```

The information included in the message depends on the type of event; in all cases, the alarm message contains the following four elements:

- ① OPCOM heading, which includes the date and time the alarm was sent
- ② Type of alarm event
- ③ Date and time the alarm event occurred
- ④ The user who caused the event, as identified by the user name and process identification (PID)

Other information contained in alarm messages is specific to the type of event that the alarm signaled.

## 11.9 The Audit Analysis Utility—A Security Auditing Tool

The Audit Analysis Utility (ANALYZE/AUDIT) enables system managers and site security administrators to selectively extract and display information from security audit or security archive log files. Using ANALYZE/AUDIT qualifiers, you can choose among a variety of audit analysis report formats and select specific audit events to be included in the report.

**NOTE:** ANALYZE/AUDIT replaces the SECAUDIT.COM command procedure that was included in previous versions of the VMS operating system.

You can use ANALYZE/AUDIT to produce audit analysis reports in one of the following forms:

- A brief (one-line) listing (/BRIEF)
- A full listing (/FULL)
- A summary of all security events processed (/SUMMARY)
- A binary output file (/BINARY)

You can use the brief output format from ANALYZE/AUDIT to perform a daily inspection of the security audit log file. If any of the selected events arouses your suspicion, you can produce a full-format listing of selected events and perform a more complete inspection of the data, as described in Section 11.12.

Listings generated by ANALYZE/AUDIT can be based on a number of selection criteria: use the /BEFORE and /SINCE qualifiers to extract security events logged during a specific period of time; use the /EVENT\_TYPE qualifier to list all security event messages of a specific event class; use the /SELECT and /IGNORE qualifiers to produce a listing that includes or excludes security event messages based upon the fields contained in the messages. (For example, /SELECT=USERNAME=JSNOOP lists only security event messages generated by user JSNOOP.)

Use the SET AUDIT command to enable and disable the recording of specific audit events in the system security audit log file and to modify characteristics of the audit server; use the SHOW AUDIT command to display the list of system events for which auditing is enabled. See the *VMS User's Manual* for descriptions of the DCL commands SET and SHOW AUDIT.

This section describes how to use ANALYZE/AUDIT to inspect the volume of security events logged to the system security audit log file.

## 11.10 ANALYZE/AUDIT Command Line Format

Use the DCL command ANALYZE/AUDIT to perform audit analysis operations on security audit or archive files. An ANALYZE/AUDIT command can specify the name of one or more audit log files, as follows:

```
ANALYZE/AUDIT [file-spec,...]
```

The default system security audit log file is SYS\$MANAGER:SECURITY\_AUDIT.AUDIT\$JOURNAL. You can omit the **file-spec** parameter if you use this file as your system security audit log file and your current default is set to SYS\$MANAGER.

In addition to the system security audit log file, you can use the ANALYZE/AUDIT command to extract security event messages from binary files created with previous ANALYZE/AUDIT commands, as well as from security archive files.

## 11.11 ANALYZE/AUDIT Output

You can specify a number of different forms of output from ANALYZE/AUDIT; brief listing (the default), full listing, summary report, 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 audit analysis file is produced that can later be processed using other audit analysis commands. If you accept the default brief output format (/BRIEF) or request a full format (/FULL) or summary (/SUMMARY) display, an ASCII file is produced.

### Brief Listing Format

The brief listing format provides one line for each record in the system security audit log file being processed. As shown in Example 11-1, output always includes date and time, type of record, subtype, node name, user name, process ID, and terminal.

**Example 11-1: Sample Brief Listing**

---

Date / Time	Type	Subtype	Node	Username	ID	Term
19-APR-1990 16:00:03.37	ACCESS	FILE_ACCESS	HERE	SYSTEM	5B600AE4	
19-APR-1990 16:00:59.66	LOGIN	SUBPROCESS	GONE	PIPERSKI	3BA011D4	
19-APR-1990 16:02:37.31	LOGIN	SUBPROCESS	GONE	MILANT	000000D5	
19-APR-1990 16:06:36.40	LOGFAIL	LOCAL	SUPER	MBILLS	000000E5	_LTA1:
.						
:						
.						

---

**Full Listing Format**

The full listing format provides all the data for each record in the system security audit log file being processed. There are small variations in record formats, based on the presence or absence of data in the record. Example 11-2 displays a single audit analysis record in the full format.

**Example 11-2: Sample Full Listing**

---

```
Security alarm (SECURITY) and security audit (SECURITY) on K9, system id: 19681
 / Local interactive login failure
Event time:          19-APR-1990 16:06:36.40
PID:                 000000D6
Username:            PGEORGE
Terminal name:      _LTA2:
Status:              %LOGIN-F-INVPWD, invalid password
```

---

**Summary Listing Format**

A summary report lists the total number of audit messages generated throughout the ANALYZE/AUDIT period for each class of security event. The summary report provides you with a method of quickly spotting potential security problems.

Example 11-3 illustrates a summary output for the following command:

```
$ ANALYZE/AUDIT/SUMMARY
```

**Example 11-3: Sample Summary Output**


---

Total records read:	6315	Records selected:	6315
Record buffer size:	512	Format buffer size:	128
Server messages:	0	Customer messages:	0
Digital CSS messages:	0	Layered prod messages:	0
Audit changes:	40	Installed db changes:	54
Login failures:	22	Breakin attempts:	1
Successful logins:	560	Successful logouts:	347
System UAF changes:	5	Network UAF changes:	0
Rights db changes:	0	Object accesses:	5111
Volume (dis)mounts:	176		

---

You can produce a summary report by itself, as shown in the previous example, or in combination with a brief or full format audit analysis report, as shown in the following example:

```
$ ANALYZE/AUDIT/BRIEF/SINCE=TODAY/SUMMARY/OUTPUT=TODAY.LIS
```

The command in this example creates a file named TODAY.LIS containing a brief format listing of all security audit messages generated since midnight of the current day followed by a summary of all the records generated.

**Binary Output:** A binary output file is an encoded data file that is created when an ANALYZE/AUDIT request includes the /BINARY qualifier. This file contains a set of the audit records from the input (source) system security audit log files. Your selection criteria determines the set of records included in the binary file. You can use binary output files as source files for future audit analysis requests, to format and display the data, to create a summary file, or to record a new binary file with different selection criteria.

With the /BINARY qualifier, all records that match the selection criteria are written to the binary output file.

## 11.12 Using ANALYZE/AUDIT

This section describes how to perform a successful audit analysis of your system. Although the way you use ANALYZE/AUDIT will depend upon the security needs at your site, there are a number of common steps which you should follow, regardless of the extent to which you use ANALYZE/AUDIT:

- Before you begin producing audit analysis reports, familiarize yourself with the normal operation of your system. Understand what types of security events are generated as part of normal system operation. This knowledge enables you to determine when to disregard system security events as uninteresting or irrelevant and when to suspect a security problem.
- Develop a procedure for generating and reviewing audit analysis reports on a periodic basis in order to determine which security events logged to the system security audit log file warrant a more thorough investigation.

## 11-32 System Security Issues

- Perform a detailed investigation of selected security events when your regular audit analysis leads you to suspect a security problem.

### 11.12.1 Recognizing Common System Events

Before using ANALYZE/AUDIT, you should familiarize yourself with the patterns of typical system use which constitute normal system operation. By becoming as familiar as possible with the ways in which your system is normally used, you can more easily distinguish between a security event message that can be ignored and one that requires further investigation.

As a system manager, you should be able to answer the following questions before performing an audit analysis:

- What are the typical hours of operation for most users of the system?
- Which images generate system security events as part of other applications?
- Are there specific users who normally operate with advanced privileges?
- Are there any regular batch or network jobs that run at specific times of the day?

By knowing the answers to these questions, you can eliminate false alarms which otherwise might have caused you to wrongly suspect a security problem.

At this point, you are ready to generate routine audit analysis reports to monitor security activity on your system.

### 11.12.2 Performing a Periodic Audit Analysis

While it is possible to generate a wide variety of audit analysis reports, the most common type of report you will probably generate is the daily listing. Typically, you might create a command procedure that runs in a batch job every evening before midnight to generate a report of the day's security event messages. The following example shows the ANALYZE/AUDIT command you would use to generate this report:

```
$ ANALYZE/AUDIT/SINCE=TODAY/OUTPUT=19APR1990.AUDIT
$ MAIL/SUBJECT="Security Events" 19APR1990.AUDIT SYSTEM
```

The first command in this example produces a file named 19APR1990.AUDIT which contains all the security event messages generated during the current day. The second command sends the file to the system manager for examination. By default, the report is produced using the brief (one line) format.

An alternate method of producing a daily audit analysis report is to generate a full format listing of selected security audit records, as shown in the following example:

```
$ ANALYZE/AUDIT/FULL/SINCE=TODAY/OUTPUT=19APR1990.AUDIT -
_ $ /EVENT_TYPE=(BREAKIN,NETUAF,RIGHTSDB,SYUAF)
$ MAIL/SUBJECT="Security Events" 19APR1990.AUDIT SYSTEM
```

It is important that you review audit analysis reports as soon as possible. The sooner you inspect the reports, the sooner you become aware of any possible breach of security on the system and determine the extent of the problem. You can make the inspection of the previous day's audit analysis report a regular part of your morning routine, or you can create a program that reviews the report and notifies you through MAIL when suspicious events have been found.

### 11.12.3 Performing a Detailed Audit Analysis

When a routine audit analysis leads you to suspect that the security of your system has been compromised—through an actual or attempted breakin, repeated login failures, or any other suspicious security events—you can investigate the source of the security event through a more detailed inspection of the system security audit log file.

For example, suppose that you see the security events shown in Example 11-4 during a routine inspection of the previous day's audit analysis report:

#### Example 11-4: Spotting Suspicious Activity in the Audit Analysis Report

Date / Time	Type	Subtype	Node	Username	ID	Term
.	.	.	.	.	.	.
19-APR-1990 16:06:09.17	LOGFAIL	REMOTE	BOSTON	JGARNER	5BC002EA	_RTA14:
19-APR-1990 16:06:22.01	LOGFAIL	REMOTE	BOSTON	JGARNER	5BC002EA	_RTA14:
19-APR-1990 16:06:34.17	LOGFAIL	REMOTE	BOSTON	JGARNER	5BC002EA	_RTA14:
19-APR-1990 16:06:45.50	LOGFAIL	REMOTE	BOSTON	JGARNER	5BC002EA	_RTA14:
19-APR-1990 16:07:12.39	LOGIN	REMOTE	BOSTON	JGARNER	5BC002EA	_RTA14:
19-APR-1990 16:23:42.45	SYSUAF	SYSUAF_ADD	BOSTON	JGARNER	5BC002EA	_RTA14:
.	.	.	.	.	.	.

The security events displayed in this report indicate that user JGARNER logged in to the system following four unsuccessful login attempts. Shortly after logging in, user JGARNER created a new account in the system user authorization file (SYSUAF).

At this point, you must determine whether this behavior is normal or abnormal. Is user JGARNER authorized to add new user accounts to the system? If you believe that the security of your system has been compromised, examine in greater detail the system security audit log file to determine the extent of the damage to your system, as shown in the following example:

```
$ ANALYZE/AUDIT/FULL/SINCE=19-APR-1990:16:06
```

The command in this example generates a full format listing of all security audit events recorded in the system security audit log file since user JGARNER first attempted to log in to the system. In full format, all the data available for each record in the system security audit log file is displayed. Using the previous example, you can find out the name of the remote user who logged in under the

## 11-34 System Security Issues

local JGARNER account and the node from which the login was made, as shown in Example 11-5.

### Example 11-5: An Example of a Full Format Audit Analysis Report

---

```
.  
.  
Security alarm (SECURITY) and security audit (SECURITY) on BOSTON,  
  system id: 19941 / Remote interactive login failure  
Event time:      19-APR-1990 16:06:09.17  
PID:            5BC002EA  
Username:       JGARNER  
Terminal name:  _RTA14:  
Remote nodename: NASHWA           Remote node id:      7300  
Remote username: FOLEY  
Status:        %LOGIN-F-INVPWD, invalid password  
.  
.  
.
```

---

The information displayed in this example indicates that the login was made by user FOLEY from remote node NASHWA. Your next step is to determine whether the security event was generated by user FOLEY or by someone who has broken in to remote node NASHWA through the FOLEY account.

### 11.12.4 Using Interactive Mode Commands

ANALYZE/AUDIT offers an alternative method of analyzing system security events logged to the system security audit log file: interactive command mode. At any time during a full or brief audit analysis listing, you can interrupt the report being displayed and enter interactive command mode by using the CTRL/C key combination, as shown in Example 11-6:



**Example 11-6: Entering Interactive Command Mode**


---

Date / Time	Type	Subtype	Node	Username	ID	Term
.	.	.	.	.	.	.
19-APR-1990 10:25:09.17	LOGFAIL	REMOTE	THERCK MINTICK		4AD003AB	_RTA99:
19-APR-1990 10:25:22.01	LOGFAIL	REMOTE	THERCK MINTICK		4AD003AB	_RTA99:
19-APR-1990 10:25:34.17	LOGFAIL	REMOTE	THERCK MINTICK		4AD003AB	_RTA99:
19-APR-1990 10:25:45.50	LOGFAIL	REMOTE	THERCK MINTICK		4AD003AB	_RTA99:
26-APR-1990 09:01:52.11	LOGFAIL	REMOTE	ALGONE MORRIS		2640020A	_RTA11:
.	.	.	.	.	.	.
.	.	.	.	.	.	.
<b>CTRL/C</b>						
COMMAND>						

---

At the *COMMAND>* prompt, you can enter interactive mode commands to generate a new audit analysis report using a different set of audit analysis criteria or reposition yourself within the system security audit log file. Use the *CONTINUE* command to return to the original full or brief audit analysis listing, or use the *EXIT* command to terminate the report and return to the *DCL* command level. See the Reference Section for a list of all interactive mode commands.

**Online Help**

The Audit Analysis Utility provides a *HELP* facility that contains information about all interactive mode commands. Enter the *HELP* command at the *COMMAND>* prompt for descriptions of each interactive mode command.

**11.13 Summary****Types of Security Problems**

Security issues on most systems are generally caused by irresponsibility, probing, or penetration. The tolerance that your site might have to a breach of security depends on the type of work that takes place at your site and on your system.

**Environmental Considerations**

A secure system environment is a key to system security. Digital strongly encourages you to stress environmental considerations as well as operating system protections when reviewing site security.

## Operating System Protections

The following measures can increase your system's security:

- **Managing passwords**—You can use a system password, set requirements for minimum password length, require users to use only system generated passwords, and you can require users to change their passwords frequently.
- **Controlling break-in detection**—You can control the number of retries allowed on dialup logins, and you can periodically review the break-in database.
- **File and directory protection**—In addition to the standard VMS file protection system, you can use Access Control Lists (ACLs) to limit and control access to files and directories.
- **Security auditing**—You can use the operator's terminal and log file to monitor security alarm messages.
- **The Audit Analysis Utility**—You can use ANALYZE/AUDIT to selectively extract and display information from security audit or security archive log files.

## **Reference Section**



---

## Accounting Utility

The Accounting Utility (ACCOUNTING) processes system accounting files to produce reports and summaries of system usage.

### format

**ACCOUNTING** [*file-spec*[,...]]

### parameter

*file-spec*[,...]

Specifies one or more accounting files as input to be processed by ACCOUNTING. If you specify more than one file name, separate them with commas. If you omit the file-spec parameter, data is processed from the current accounting file, SYS\$MANAGER:ACCOUNTNG.DAT.

Wildcard characters are allowed in the file specification.

### usage summary

The following DCL command invokes ACCOUNTING:

```
$ ACCOUNTING [file-spec[,...]]
```

Each ACCOUNTING request runs until it completes. To terminate an ACCOUNTING request before completion, press CTRL/Y.

You can direct ACCOUNTING output to any supported terminal device or to a disk or tape file by specifying the /OUTPUT qualifier.

**Use of ACCOUNTING requires read access to the input accounting file.**

## ACC-2 ACCOUNTING /ACCOUNT

### ACCOUNTING Qualifiers

This section explains ACCOUNTING qualifiers and provides examples of their use. The qualifiers follow the standard rules of DCL grammar.

---

#### /ACCOUNT

Controls whether only those records matching the specified account name are selected. If you omit the qualifier or specify /NOACCOUNT, the account name is not used to select records.

#### format

```
/ACCOUNT=(["-"],account-name[...])  
/NOACCOUNT
```

#### keywords

“\_”

Specifies that all records are selected except those matching any specified account name.

***account name*[...]**

Specifies the account name used to select records. The account name matches the account name specified in the user authorization file.

When you specify the /ACCOUNT qualifier, specify at least one account name. If you specify more than one account name, separate them with commas, and enclose the list in parentheses.

#### example

```
$ ACCOUNTING /ACCOUNT=(MISHA,MARCO)
```

The command in this example selects records matching the accounts MISHA and MARCO.

---

#### /ADDRESS

Controls whether only those records matching the specified remote node-address are selected. If you omit the qualifier or specify /NOADDRESS, the node-address is not used to select records.

#### format

```
/ADDRESS=(["-"],node-address[...])  
/NOADDRESS
```

## keywords

**“\_”**

Specifies that all records are selected except those matching any specified node address.

***node address[,...]***

Specifies the node address used to select records.

The node address is a unique numeric identifier for DECnet nodes. Use the following formula to calculate the node address:

$$\text{node address} = (\text{area-number} * 1024) + \text{node-number}$$

When you specify the /ADDRESS qualifier, specify at least one node address. If you specify more than one node address, separate them with commas, and enclose the list in parentheses.

## example

```
$ ACCOUNTING /ADDRESS=19656
```

The command in this example selects records that have remote node address fields that are equivalent to the DECnet address 19656 or DECnet node address 19.200.

---

## /BEFORE

Controls whether only those records dated earlier than the specified time are selected. If you specify /NOBEFORE or omit the qualifier, time is not used to select records.

## format

***/BEFORE[=time]***

***/NOBEFORE***

## keyword

***time***

Specifies the time used to select records. Records dated earlier than the specified time are selected. You can specify an absolute time, delta time, or a combination of the two.

## example

```
$ ACCOUNTING /BEFORE=31-DEC-1990
```

The command in this example selects all records dated earlier than December 31, 1990.

**ACC-4 ACCOUNTING**  
**/BINARY**

---

**/BINARY**

Controls whether output is a binary accounting file.

**format**

**/BINARY**  
**/NOBINARY**

**description**

When **/BINARY** is specified, the output file, specified using the **/OUTPUT** qualifier, contains image copies of the selected input records. If you specify **/NOBINARY** or omit the qualifier, the output file contains formatted ASCII records.

The **/BINARY**, **/BRIEF**, **/FULL**, and **/SUMMARY** qualifiers cannot be used in combination with each other.

**example**

```
$ ACCOUNTING /BINARY /OUTPUT=MYACC.DAT
```

The command in this example writes accounting data in binary format to the file **MYACC.DAT**.

---

**/BRIEF**

Controls whether a brief format is used in ASCII displays.

**format**

**/BRIEF**  
**/NOBRIEF**

**description**

By default, records are displayed in the brief format. You must specify **/FULL** to have the full contents of each selected record displayed.

The **/BINARY**, **/BRIEF**, **/FULL**, and **/SUMMARY** qualifiers cannot be used in combination with each other.

**example**

```
$ ACCOUNTING /OUTPUT=MYACC.DAT
```

The command in this example produces an ASCII file in brief format. The file is written to **MYACC.DAT**.



---

**/ENTRY**

Controls whether only those records matching the specified queue entry are selected. If you specify **/NOENTRY** or omit the qualifier, the queue entry is not used to select records.

**format**

**/ENTRY**=(["-"],*queue-entry*[,...])

**/NOENTRY**

**keywords**

**"\_"**

Specifies that all records are selected except those matching any specified queue entry.

***queue-entry*[,...]**

Specifies the queue entry identifier used to select records. The queue entry is a unique numeric identifier assigned to entries in device and batch queues.

When you specify the **/ENTRY** qualifier, specify at least one queue entry. If you specify more than one queue entry, separate them with commas, and enclose the list in parentheses.

**example**

```
$ ACCOUNTING /ENTRY=("-" ,25)
```

The command in this example selects records for all queue entries except number 25.

---

**/FULL**

Controls whether a full format is used in ASCII displays. If you specify **/NOFULL** or omit the qualifier, records are displayed in the brief format.

**format**

**/FULL**

**/NOFULL**

**description**

By default, records are displayed in the brief format. You must specify **/FULL** to have the full contents of each selected record displayed.

The **/BINARY** **/BRIEF**, **/FULL**, and **/SUMMARY** qualifiers cannot be used in combination with each other.

## ACC-6 ACCOUNTING /FULL

### example

```
$ ACCOUNTING /FULL
```

The command in this example displays the full contents of each selected record.

---

### /IDENT

Controls whether only those records matching the specified process ID are selected. If you specify /NOIDENT or omit the qualifier, the process ID is not used to select records.

### format

```
/IDENT=(["-"],process-id[,...])
```

```
/NOIDENT
```

### keywords

“-”

Specifies that all records are selected except those matching the specified process ID.

*process-id[,...]*

Specifies the process ID used to select records. When you specify /IDENT, specify at least one process ID. If you specify more than one process ID, separate them with commas, and enclose the list in parentheses.

### example

```
$ ACCOUNTING /IDENT=(25634,045A6B)
```

The command in this example selects records matching the process IDs 25634 and 045A6B.

---

### /IMAGE

Controls whether only those records matching the specified image name are selected. If you specify /NOIMAGE or omit the qualifier, the image name is not used to select records.

### format

```
/IMAGE=(["-"],image-name[,...])
```

```
/NOIMAGE
```

## keywords

**"\_"**

Specifies that all records are selected except those that match the specified image name.

***image-name[,...]***

Specifies the image name used to select records. Specify only the file name portion of the image file specification, such as EDT.

When you specify /IMAGE, specify at least one image name. If you specify more than one image name, separate them with commas, and enclose the list in parentheses.

## example

```
$ ACCOUNTING /IMAGE=("_",SYSGEN)
```

The command in this example selects records for all images except SYSGEN.

---

## /JOB

Controls whether only those records matching the specified job name are selected. A job name is assigned to an entry in a device or batch queue. If you specify /NOJOB or omit the qualifier, the job name is not used to select records.

## format

***/JOB=(["\_",]job-name[,...])***

***/NOJOB***

## keywords

**"\_"**

Specifies that all records are selected except those matching any specified job name.

***job-name[,...]***

Specifies the job name used to select records. When you specify /JOB, specify at least one job name. If you specify more than one job name, separate them with commas, and enclose the list in parentheses.

## example

```
$ ACCOUNTING /JOB=(MYJOB1,MYJOB2)
```

The command in this example selects all records that match the job names MYJOB1 and MYJOB2.

**ACC-8 ACCOUNTING**  
**/LOG**

---

**/LOG**

Controls whether informational messages (input file names, selected record counts, rejected record counts) are displayed to SYS\$OUTPUT.

**format**

**/LOG**  
**/NOLOG**

**description**

By default, messages are not displayed. If more than one input file is specified in an ACCOUNTING command with the /LOG qualifier, there is one logging message for each file, and a total is provided.

**example**

\$ ACCOUNTING /LOG

Date / Time	Type	Subtype	Username	ID	Source	Status
31-DEC-1990 13:42:44	FILE			00000000		00000000
31-DEC-1990 13:53:29	PROCESS	BATCH	SYSTEM	20800116		10030001
31-DEC-1990 13:53:38	SYSINIT		SYSTEM	20800104		107781AB
31-DEC-1990 13:58:04	PROCESS	INTERACTIVE	MATTHEWS	20800128	TTF5:	00000001
31-DEC-1990 14:10:29	PROCESS	NETWORK	ROBIN_NET	20800132	AXEL	10000000
31-DEC-1990 14:28:56	PROCESS	SUBPROCESS	SMITH	2080013E		10000001
31-DEC-1990 14:33:31	PRINT		JONES	21400117		00040001

.  
.  
.

%ACC-I-INPUT, SYS\$SYSROOT:[SYSMGR]ACCOUNTNG.DAT;1, 33 selected, 0 rejected

The command in this example displays accounting records and informational messages such as selected and rejected record counts.

---

**/NODE**

Controls whether only those records matching the specified remote DECnet node name are selected. If you specify /NONODE or omit the qualifier, the node name is not used to select records.

**format**

**/NODE=(["-",]node-name[,...])**  
**/NONODE**

## keywords

**"\_"**

Specifies that all records are selected except those matching any specified remote node name.

***node-name[,...]***

Specifies the remote node name used to select records. Colons (:) are not allowed in the node name specification.

When you specify /NODE, you must specify at least one node name. If you specify more than one node name, separate them with commas, and enclose the list in parentheses.

## example

```
$ ACCOUNTING /NODE= ("_", NOROT, ROBERT, SEESHA)
```

The command in this example selects records for all remote node names except those named in the list.

---

## /OUTPUT

Specifies where to direct accounting output. If you omit the qualifier, selected records are output to SYS\$OUTPUT.

## format

***/OUTPUT[=file-spec]***

***/NOOUTPUT***

## keyword

***file-spec[,...]***

Specifies the name of the file that is to contain the selected records.

If you omit the device or directory specification, the current device and default directory are used. If you omit the file name, the file name of the input file is used. If you omit the file type and the output is ASCII, the default file type is LIS. If you omit the file type and the output is binary (/BINARY), the default file type is DAT.

## example

```
$ ACCOUNTING /BINARY /OUTPUT=STAT.DAT
```

The command in this example selects accounting records and outputs them in binary to the file STAT.DAT.

## **/OWNER**

Controls whether only those records matching the specified owner process ID are selected. If you specify /NOOWNER or omit the qualifier, the owner process ID is not used to select records.

### **format**

**/OWNER=**(["-"],*owner-process-id*[,...])  
**/NOOWNER**

### **keywords**

**"\_"**

Specifies that all records are selected except those matching any specified owner process ID.

***owner-process-id*[,...]**

Specifies the owner process identification number used to select records. Owner process IDs are present only in subprocesses to specify the process id of their owner process.

When you specify /OWNER, specify at least one owner process ID. If you specify more than one, separate them with commas, and enclose the list in parentheses.

---

## **/PRIORITY**

Controls whether only those records matching the specified base process priority are selected. If you specify /NOPRIORITY or omit the qualifier, the priority is not used to select records.

### **format**

**/PRIORITY=**(["-"],*priority*[,...])  
**/NOPRIORITY**

### **keywords**

**"\_"**

Specifies that all records are selected except those matching any specified base process priority.

***priority*[,...]**

Specifies the base process priority used to select records.

When you specify /PRIORITY, specify at least one priority. If you specify more than one priority, separate them with commas, and enclose the list in parentheses.

**example**

```
$ ACCOUNTING /PRIORITY=3
```

The command in this example selects records that match a base process priority of 3.

---

**/PROCESS**

Controls whether only those process-termination records matching the specified process type are selected. If you specify `/NOPROCESS` or omit the qualifier, the process type is not used to select records.

To produce records for interactive processes, you must enable both `PROCESS` and `INTERACTIVE` logging using the `SET ACCOUNTING` command.

**format**

```
/PROCESS=(["-"]process-type[,...])  
/NOPROCESS
```

**keywords**

**"-"**

Specifies that all records are selected except those matching any specified process type.

***process-type*[,...]**

Specifies the process type used to select records.

When you specify `/PROCESS`, specify at least one process type. If you specify more than one process type, separate them with commas, and enclose the list in parentheses.

You can specify any of the following process types: `BATCH`, `DETACHED`, `INTERACTIVE`, `NETWORK`, and `SUBPROCESS`.

**example**

```
$ ACCOUNTING /PROCESS=("-", INTERACTIVE, DETACHED)
```

The command in this example selects all records except those that match the process types `INTERACTIVE` or `DETACHED`.

## /QUEUE

Controls whether only those records matching the specified queue name are selected. If you specify /NOQUEUE or omit the qualifier, the queue name is not used to select records.

### format

**/QUEUE=**(["\_",]*queue-name*[,...])  
**/NOQUEUE**

### keywords

“\_”

Specifies that all records are selected except those matching any specified queue name.

***queue-name***[,...]

Specifies the queue name used to select records. A queue name is a unique identifier for a device or batch queue.

When you specify /QUEUE, specify at least one queue name. If you specify more than one queue name, separate them with commas, and enclose the list in parentheses.

---

## /REJECTED

Controls whether records that do not match the selection criteria are output to a specified file. Unselected records are always in binary format. If you specify /NOREJECTED or omit the qualifier, unselected records are not output.

### format

**/REJECTED**[=*file-spec*]  
**/NOREJECTED**

### keyword

***file-spec***

Specifies the name of the file to contain unselected records. If you omit the device or directory specification, the current device and default directory are used. If you omit the file name, the file name of the input file is used. If you omit the file type, REJ is used.



## example

```
$ ACCOUNTING /REJECTED=ACCOUNTING
```

The command in this example outputs all unselected records to the file ACCOUNTING.REJ.

---

## /REMOTE\_ID

Controls whether only those records matching the specified remote ID are selected. The remote ID identifies the process or user on a remote node. If you specify /REMOTE\_ID or omit the qualifier, the remote ID is not used to select records.

## format

```
/REMOTE_ID=(["-"],remote-id[,...])  
/NOREMOTE_ID
```

## keywords

“-”

Specifies that all records are selected except those matching any specified remote ID.

### *remote-id*

Specifies the remote process identification code used to select records. The exact format of a remote ID varies with the context and DECnet implementation. For VMS systems, the remote ID is always the user name.

When you specify /REMOTE\_ID, specify at least one remote ID. If you specify more than one remote ID, separate them with commas, and enclose the list in parentheses.

## example

```
$ ACCOUNTING /REMOTE_ID=ROBIN
```

The command in this example requests accounting information for the remote user ROBIN.

---

**/REPORT**

Controls whether a specified item is included in a summary report. One column is generated on the summarization report for each item specified. Items are summarized either as totals or maximum values. The /REPORT qualifier requires the /SUMMARY qualifier.

**format**

**/REPORT**[(*report-item*[,...])] **/NOREPORT**

**keyword**

***report-item***[,...]

Specifies the report item used to select records.

You can specify any of the following items:

---

<b>Keyword</b>	<b>Meaning</b>	<b>How Summarized</b>
BUFFERED_IO	Buffered IOs	Total
DIRECT_IO	Direct IOs	Total
ELAPSED	Elapsed time	Total
EXECUTION	Image execution count	Total
FAULTS	Page faults	Total
GETS	VMS RMS gets issued by symbiont	Total
PAGE_FILE	Page file usage	Maximum
PAGE_READS	Page read IOs	Total
PAGES	Pages printed	Total
PROCESSOR	Processor time consumed	Total
QIOS	Printer QIOS issued by symbiont	Total
RECORDS	Records in file (default)	Total
VOLUMES	Volumes mounted	Total
WORKING_SET	Working set size	Maximum

---

If you specify more than one report item, separate them with commas, and enclose the list in parentheses.

## description

If you specify /REPORT without a keyword (or if you specify /SUMMARY and do not specify /REPORT), /REPORT=RECORDS is assumed.

To obtain a summary by image (when image accounting is enabled) showing the number of times individual images were executed, specify /SUMMARY=IMAGE/REPORT=RECORDS. These qualifiers will display the total number of termination records for each image.

Many report items are present in only a few types of accounting records. If records are selected that do not contain a report value that has been requested, a default value, usually 0, is used.

## example

```
$ ACCOUNTING /SUMMARY /REPORT=(DIRECT_IO,BUFFERED_IO)
```

The command in this example produces a summary report of direct I/O and buffered I/O records.

---

## /SINCE

Controls whether only those records dated the same or later than a specified time are selected. If you specify /NOSINCE or omit the qualifier, time is not used to select records.

## format

*/SINCE[=*time*]*

*/NOSINCE*

## keyword

### *time*

Specifies the time used to select records. Records dated the same or later than the specified time are selected. You can specify an absolute time, delta time, or a combination of the two.

If you specify /SINCE without the time, midnight of the current day is used.

## example

```
$ ACCOUNTING /SINCE=31-DEC-1990
```

The command in this example selects records dated later than December 31, 1990.

## **/SORT**

Specifies the sequence of records in the brief or full listing. The /SORT qualifier can be used with the /BINARY, /BRIEF, and /FULL qualifiers but not with /SUMMARY.

### **format**

**/SORT**=[(-]*sort-item*[,...]]

**/NOSORT**

### **keywords**

-

Specifies that the sort field is used as a descending key. By default, keys are assumed to be ascending.

***sort-item***[,...]

Specifies the sort item used to select records.

When you specify /SORT, specify at least one sort item. If you specify more than one sort item, separate the items with commas, and enclose the list in parentheses.

You can specify any of the following sort items:

<b>Keyword</b>	<b>Meaning</b>
ACCOUNT	User's account name
ADDRESS	Remote node address
BUFFERED_IO	Buffered IO count
DIRECT_IO	Direct IO count
ELAPSED	Elapsed time
ENTRY	Number of batch or print job queue entry
EXECUTION	Image execution count
FAULTS	Page faults
FINISHED	Termination time or time record was written
GETS	Number of gets from the file to be printed
IDENT	Process identification
IMAGE	Image name
JOB	Name of batch or print job
NODE	Remote node name
OWNER	Owner process identification
PAGES	Number of pages printed
PAGE_FILE	Peak page file usage
PAGE_READS	Page read IOs
PRIORITY	Process base priority
PROCESS	Process type
PROCESSOR	Processor time
QIOS	Number of QIOs to the printer
QUEUE	Name of queue
QUEUED	Time batch or print job was queued
STARTED	Start time
STATUS	Exit status
TERMINAL	Terminal name
TYPE	Record type
UIC	User identification code
USER	User's name
VOLUMES	Number of volumes mounted
WORKING_SET	Peak working set size

## ACC-18 ACCOUNTING

### /SORT

#### description

If a sort item is preceded by a minus sign (-), that field is used as a descending key. By default, keys are assumed to be ascending.

The selected records are sorted according to the sequence specified by the sort items given with the /SORT qualifier prior to writing the records to the designated output file. Unselected records are not sorted. The ordering of sort items determines the relative ranking of the keys.

If a sort item specifies a field that is not present in a record, that record becomes unselected and will be reflected as such in the counts of selected and rejected records. For example, /SORT=IMAGE would cause nonimage-termination records to be excluded, since image-termination records are the only record types that contain image names. Similarly, /SORT=PAGES would exclude nonprint-termination records.

#### example

```
$ ACCOUNTING /SORT=(PROCESS,FAULTS,IMAGE)
```

The command in this example sorts the selected records in the sequence specified by the /SORT qualifier.

---

#### /STATUS

Controls whether only those records matching the specified exit status are selected. The exit status refers to the final completion status of the process or image. If you specify /NOSTATUS or omit the qualifier, the exit status is not used to select records.

#### format

**/STATUS=**(["-",]*exit-status*[,...])

**/NOSTATUS**

#### keywords

**"\_"**

Specifies that all records are selected except those matching any specified exit status.

***exit-status*[,...]**

Specifies the exit status used to select records.

When you specify /STATUS, specify at least one exit status. If you specify more than one exit status, separate them with commas, and enclose the list in parentheses. Specify each status as a character string of hexadecimal numerals.

**example**

\$ ACCOUNTING /STATUS=10D38064

The command in this example selects all records that have a status field value of 10D38064 in hexadecimal.

**/SUMMARY**

Specifies that a summary of the selected records, grouped by the list of summary keys, be produced. Use the /REPORT qualifier to control what information is summarized. If you omit the /REPORT qualifier, /REPORT=RECORDS is assumed. The /SUMMARY qualifier is required with the /REPORT qualifier.

If you specify /NOSUMMARY or omit the qualifier, no summarization occurs.

**format**

**/SUMMARY**[(*summary-item*[...])]  
**/NOSUMMARY**

**keyword**

***summary-item*[...]**

Specifies the summary item used to select records. You can specify any of the following summary items:

Summary Item	Outputs
ACCOUNT	Account name from the UAF
DATE	YYYY MMM DD
DAY	Day of month (1-31)
HOUR	Hour of day (0-23)
IMAGE	Image name
JOB	Name of batch job or print job
MONTH	Month of year (1-12)
NODE	Remote node name
PROCESS	Process type
QUEUE	Batch or device queue name
TERMINAL	Terminal name
TYPE	Type of record (logout, batch)

# ACC-20 ACCOUNTING /SUMMARY

---

Summary Item	Outputs
UIC	User identification code
USER	User name from UAF
WEEKDAY	Day of week (0=Sunday, 1=Monday, and so on)
YEAR	Year

---

If you specify /SUMMARY without a value, /SUMMARY=USER is assumed.

If you specify more than one summary item, separate them with commas, and enclose the list in parentheses.

## description

The summarized items are sorted in ascending order and listed in the same left-to-right sequence given in the list of summary items. The output is sent to SYS\$OUTPUT unless specifically directed elsewhere by the /OUTPUT qualifier.

The /BINARY, /BRIEF, /FULL, and /SUMMARY qualifiers cannot be used in combination with each other.

**NOTE:** Report item totals on summary reports can be misleading if you do not know the number of records that were added together to produce the totals. Use the /REPORT=RECORDS qualifier to show the number of records that were added to produce each total.

## example

```
$ ACCOUNTING /SUMMARY=IMAGE
```

The command in this example generates a summary report of all image records.

---

## /TERMINAL

Controls whether only those records matching the specified terminal names are selected. Terminal names are associated with interactive processes. If you specify /NOTERMINAL or omit the qualifier, the terminal name is not used to select records.

## format

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



## keywords

**"\_"**

Specifies that all records are selected except those matching any specified terminal name.

***terminal-name[,...]***

Specifies the terminal name used to select records.

When you specify /TERMINAL, specify at least one terminal name.

Specify terminal names as standard device names and include the colon (: ) (for example, TTA6:).

If you specify more than one terminal name, separate them with commas, and enclose the list in parentheses.

## example

```
$ ACCOUNTING /TERMINAL=TTB3:
```

The command in this example selects records that match the terminal name TTB3.

---

## /TITLE

Specifies the title to be printed in the center of the first line of summary reports. The title line includes the beginning and ending times for the data summary at the left and right margins, respectively.

## format

***/TITLE=title***

**/NOTITLE**

## keyword

***title***

Specifies the title to be printed on the summary report. If the title includes spaces or special characters, you must enclose it in quotation marks ("").

## example

```
$ ACCOUNTING /SUMMARY=IMAGE /TITLE="JUNE ACCOUNTING REPORT"
```

The command in this example selects image records for a summary report and writes the title "JUNE ACCOUNTING REPORT" at the top of the report.

---

## /TYPE

Controls whether only those records matching the specified record type are selected. If you specify /NOTYPE or omit the qualifier, the record type is not used to select records.

### format

**/TYPE**=(["\_",]*record-type*[,...])  
**/NOTYPE**

### keywords

"\_"

Specifies that all records are selected except those matching any specified record type.

*record-type*[,...]

Specifies the record type used to select records. You can specify any of the following record types:

---

Record Type	Meaning
FILE	Accounting file forward and backward pointers
IMAGE	Termination of image
LOGFAIL	Unsuccessful conclusion of a login attempt
PRINT	Termination of print job
PROCESS	Termination of process
SYSINIT	System initialization
UNKNOWN	Any record not recognized as one of the other specified record types
USER	Arbitrary user messages

---

When you specify /TYPE, specify at least one record type. If you specify more than one record type, separate them with commas, and enclose the list in parentheses.

### example

```
§ ACCOUNTING /TYPE=PRINT
```

The command in this example selects records that match the record type PRINT.

---

**/UIC**

Controls whether only those records matching the specified user identification code (UIC) are selected. If you specify /NOUIC or omit the qualifier, the UIC is not used to select records.

**format**

**/UIC=**(["-",]uic[,...])  
**/NOUIC**

**keywords**

**"\_"**

Specifies that all records are selected except those matching any specified UIC.

**uic[,...]**

Specifies the user identification code (UIC) used to select records.

When you specify /UIC, specify at least one UIC. If you specify more than one UIC, separate them with commas, and enclose the list in parentheses. You may specify the UIC in numeric or alphanumeric format. You may use the asterisk (\*) as a wildcard character.

**example**

```
§ ACCOUNTING /UIC=[360,*]
```

The command in this example selects records that match UICs having a group number of 360.

---

**/USER**

Controls whether only those records matching the specified user name are selected. The user name matches the user name in the user authorization file. If you specify /NOUSER or omit the qualifier, the user name is not used to select records.

**format**

**/USER=**(["-",]username[,...])  
**/NOUSER**

**keywords**

**"\_"**

Specifies that all records are selected except those matching any specified user name.

**ACC-24 ACCOUNTING  
/USER**

***username[,...]***

Specifies the user name used to select records.

When you specify /USER, specify at least one user name. If you specify more than one user name, separate them with commas, and enclose the list in parentheses.

**example**

```
$ ACCOUNTING /USER=("-", SASHA)
```

The command in this example selects all records except those that match the user name SASHA.

---

## Audit Analysis Utility

The Audit Analysis Utility (ANALYZE/AUDIT) processes security audit messages to produce reports and summaries of security events on the system.

### format

**ANALYZE/AUDIT** [*file-spec*[,...]]

### parameter

***file-spec*[,...]**

Specifies one or more security audit log files as input to be processed by the Audit Analysis Utility. If you specify more than one file name, separate them with commas. If your current directory is the system manager directory and you omit the **file-spec** parameter, data is processed from the default security audit log file, SYS\$MANAGER:SECURITY\_AUDIT.AUDIT\$JOURNAL.

Wildcard characters are allowed in the file specification.

### usage summary

The following DCL command invokes the Audit Analysis Utility:

**ANALYZE/AUDIT** [*file-spec*[,...]]

Each ANALYZE/AUDIT request runs until it completes. To terminate an ANALYZE/AUDIT request before completion, press CTRL/Z.

You can direct ANALYZE/AUDIT output to any supported terminal device or to a disk or tape file by specifying the /OUTPUT qualifier.

**NOTE:** Use of the Audit Analysis Utility requires no special privileges other than access to the files specified on the command line.

## AUD-2 ANALYZE/AUDIT /BEFORE

### ANALYZE/AUDIT Qualifiers

This section describes qualifiers for the ANALYZE/AUDIT command and provides examples of their use. The qualifiers follow the standard rules of DCL grammar, as described in the *VMS DCL Concepts Manual*.

---

#### **/BEFORE**

Controls whether records dated earlier than the specified time are selected.

#### **format**

**/BEFORE[=*time*]**  
**/NOBEFORE**

#### **keyword**

##### *time*

Specifies the time used to select records. Records dated earlier than the specified time are selected. You can specify an absolute time, delta time, or a combination of the two. Observe the syntax rules for date and time described in the *VMS DCL Concepts Manual*.

#### **description**

By default, all records in the security audit log file may be examined. You must specify **/BEFORE** to discard records created after a specific point in time.

#### **example**

```
$ ANALYZE/AUDIT /BEFORE=25-NOV-1989 -  
_ $ SYS$MANAGER:SECURITY_AUDIT.AUDIT$JOURNAL
```

The command in this example selects all records dated earlier than November 25, 1989.

---

#### **/BINARY**

Controls whether output is a binary file.

#### **format**

**/BINARY**  
**/NOBINARY**

## keywords

None.

## description

When /BINARY is specified, the output file, specified using the /OUTPUT qualifier, contains image copies of the selected input records. If you specify /NOBINARY or omit the qualifier, the output file contains ASCII records.

By default, if you specify /BINARY and do not include the /OUTPUT qualifier, an output file named AUDIT.AUDIT\$JOURNAL is created.

The /BINARY, /BRIEF, and /FULL qualifiers cannot be used in combination.

## example

```
$ ANALYZE/AUDIT /BINARY/EVENT_TYPE=LOGFAIL -  
_ $ SYSS$MANAGER:SECURITY_AUDIT.AUDIT$JOURNAL
```

The command in this example writes all login failure audit messages from the current security audit log file to the binary file AUDIT.AUDIT\$JOURNAL.

---

## /BRIEF

Controls whether a brief (one line per record) format is used in ASCII displays.

## format

*/BRIEF (default)*  
**/NOBRIEF**

## keywords

None.

## description

By default, records are displayed in the brief format. You must specify /FULL to have the full contents of each selected audit event record displayed.

The /BINARY, /BRIEF, and /FULL qualifiers cannot be used in combination.

## AUD-4 ANALYZE/AUDIT /BRIEF

### example

```
$ ANALYZE/AUDIT /OUTPUT=AUDIT.LIS -  
_ $ SYS$MANAGER:SECURITY_AUDIT.AUDIT$JOURNAL
```

The command in this example produces an ASCII file in brief format by default. The file is written to AUDIT.LIS.

---

### /EVENT\_TYPE

Selects the classes of events to be extracted from the security log file. If you omit the qualifier or specify the ALL keyword, the event type is not used to select records.

### format

*/EVENT\_TYPE=event-type[,...] )*

### keyword

*event type[,...]*

Specifies the classes of events used to select records. You can specify any of the following event types:

---

Event Type	Meaning
[NO]ACCESS	Object access
[NO]ALL	All event types
[NO]AUDIT	Use of SET AUDIT command
[NO]BREAKIN	Breakin detection
[NO]INSTALL	Install operation
[NO]LOGFAIL	Unsuccessful login attempt
[NO]LOGIN	Successful login
[NO]LOGOUT	Successful logout
[NO]MOUNT	Execution of MOUNT or DISMOUNT command
[NO]NETUAF	Modification of the network proxy authorization file
[NO]RIGHTSDB	Modification of the rights database
[NO]SYSUAF	Modification of the system user authorization file

---

Specifying the negated form of an event class (for example, NOLOGFAIL) excludes the specified event class from the audit analysis report.



### example

```
$ ANALYZE/AUDIT /EVENT_TYPE=LOGFAIL -  
_ $ SYSS$MANAGER:SECURITY_AUDIT.AUDIT$JOURNAL
```

The command in this example selects records that match the event type LOGFAIL.

---

### /FULL

Controls whether a full format is used in ASCII displays. If you specify /NOFULL or omit the qualifier, records are displayed in the brief format.

### format

```
/FULL  
/NOFULL (default)
```

### keywords

None.

### description

By default, records are displayed in the brief format. You must specify /FULL to have the full contents of each selected record displayed.

The /BINARY, /BRIEF, and /FULL qualifiers cannot be used in combination.

### example

```
$ ANALYZE/AUDIT /FULL -  
_ $ SYSS$MANAGER:SECURITY_AUDIT.AUDIT$JOURNAL
```

The command in this example displays the full contents of each selected record.

---

### /IGNORE

Controls whether records matching the specified criteria are excluded.

### format

```
/IGNORE=criteria[,...]
```

## AUD-6 ANALYZE/AUDIT /IGNORE

### keyword

#### *criteria[,...]*

Specifies that all records are selected except those matching any of the specified exclusion criteria. See the /SELECT qualifier description for a list of the possible criteria to use with the /IGNORE qualifier.

### description

Use the /IGNORE qualifier to exclude specific groups of audit records from the audit analysis report. When more than one keyword from the list of possible exclusion criteria are specified, records that meet any of the criteria are excluded.

### example

```
$ ANALYZE/AUDIT /IGNORE=(SYSTEM=NAME=WIPER, USERNAME=MILANT) -  
_ $ SYS$MANAGER:SECURITY_AUDIT.AUDIT$JOURNAL
```

The command in this example excludes from the audit analysis report all records in the audit log file generated from node WIPER or from user MILANT (on any node).

---

## /INTERACTIVE

Controls whether interactive command mode is enabled when the Audit Analysis Utility is invoked.

### format

```
/INTERACTIVE (default)  
/NOINTERACTIVE
```

### keywords

None.

### description

Interactive command mode, enabled by default, allows you to interrupt the audit analysis report being displayed and issue commands that modify the criteria used to select or exclude records for the report.

To interrupt a full or brief audit analysis report and enter interactive mode commands, press CTRL/C. Enter commands at the *COMMAND>* prompt. Enter the CONTINUE command to leave interactive command mode and continue the audit analysis report or EXIT to terminate the session. See the command section for a complete description of each interactive mode command.

Specify `/NOINTERACTIVE` to disable interactive command mode.

**NOTE:** Upon entering command mode, the current record is displayed in full format. The record may not match the selection or exclusion criteria specified in the previous ANALYZE/AUDIT command.

### example

```
$ ANALYZE/AUDIT /FULL -  
_ $ SYS$MANAGER:SECURITY_AUDIT.AUDIT$JOURNAL
```

The command in this example produces a full format display of the selected records. New records are displayed every three seconds. (See the `/PAUSE` qualifier description to find how to modify the duration of each record displayed.) Use the CTRL/C key combination to interrupt the display and enter interactive mode commands.

---

### /OUTPUT

Specifies where to direct output from the Audit Analysis Utility. If you omit the qualifier, selected ASCII records are output to `SYS$OUTPUT`.

### format

```
/OUTPUT[=file-spec]  
/NOOUTPUT
```

### keyword

*file-spec[,...]*

Specifies the name of the file that is to contain the selected records. If you omit the device and directory specification, the current device and directory specification are used. If you omit the file name and type, the default file name `AUDIT.LIS` is used. If the output is binary (`/BINARY`) and you omit the `/OUTPUT` qualifier, the binary information is output to the file `AUDIT.AUDIT$JOURNAL`.

### example

```
$ ANALYZE/AUDIT /BINARY/OUTPUT=BIN122588.DAT -  
_ $ SYS$MANAGER:SECURITY_AUDIT.AUDIT$JOURNAL
```

The command in this example selects audit records and outputs them in binary format to the file `BIN122588.DAT`.

## AUD-8 ANALYZE/AUDIT /PAUSE

---

### **/PAUSE**

Specifies the length of time each record is displayed for full-format (/FULL) displays.

#### **format**

**/PAUSE=seconds**

#### **keyword**

##### **seconds**

Specifies the duration (in seconds) of the full screen display. A value of 0 specifies that the system should not pause before displaying the next selected record. The default is 3 seconds.

#### **description**

The /PAUSE qualifier can only be used with full-format (/FULL) displays to specify the length of time each record is displayed. By default, each record is displayed for a period of 3 seconds. A value of 0 results in a continuous display of audit records.

#### **example**

```
$ ANALYZE/AUDIT /FULL/PAUSE=1 -  
_ $ SYS$MANAGER:SECURITY_AUDIT.AUDIT$JOURNAL
```

The command in this example displays a selected record in full format every second. You can interrupt the display and enter interactive mode commands at any time by pressing CTRL/C. (See the Interactive Mode Command section for more information.)

---

### **/SELECT**

Controls whether records matching the specified criteria are selected.

#### **format**

**/SELECT=(criteria[,...])**

**/NOSELECT**

#### **keyword**

##### **criteria[,...]**

Specifies the criteria to be used to select records. If you omit the /SELECT qualifier, all event records are selected.

The possible criteria that can be specified are as follows:

***ACCESS=(type,...)***

Specifies the type of object access upon which the selection is based. It may be any of the following: READ, WRITE, EXECUTE, DELETE, or CONTROL.

***ACCOUNT=(name,...)***

Specifies the account name upon which selection is based. Full wildcarding of the account name is allowed.

***DEVICE\_NAME=(name,...)***

Specifies the name of the device to be used in the selection of event records. Full wildcarding of the device name is allowed.

***DISMOUNT\_FLAGS=(name,...)***

Specifies the names of the volume dismounting flags upon which selection is based. The available names are ABORT, CLUSTER, NOUNLOAD, and UNIT.

***HOLDER=(list,...)***

Specifies the characteristics of the identifier holder to be used in selecting event records.

<b>Keyword</b>	<b>Description</b>
NAME=name	Specifies the name of the holder. Full wildcarding of the name is allowed.
OWNER=value	Specifies the owner (UIC) of the holder.

***IDENTIFIER=(attr,...)***

Specifies that some attribute of an identifier should be used in selecting event records.

<b>Keyword</b>	<b>Description</b>
ATTRIBUTE=name	Specifies the name of the particular attribute. The available names are RESOURCE and DYNAMIC.
NAME=name	Specifies the original name of the identifier. Full wildcarding of the name is allowed.
NEW_NAME=name	Specifies the new name of the identifier. Full wildcarding is allowed.
VALUE=value	Specifies the original value of the identifier.
NEW_VALUE=value	Specifies the new value of the identifier.

***IMAGE\_NAME=(name,...)***

Specifies the name of the image to be used when selecting event records. Full wildcarding of the image name is allowed.

**AUD-10    ANALYZE/AUDIT  
/SELECT**

***INSTALL=(type,...)***

Specifies the type of installation event to be considered when selecting event records.

<b>Keyword</b>	<b>Description</b>
FILE=name	Specifies the name of the installed file. Full wildcarding is allowed.
FLAGS=name	Specifies the names of the flags that correspond to the INSTALL qualifiers. (For example, OPEN corresponds to /OPEN.)
PRIVILEGES=name	Specifies the names of the privileges with which the file was installed.

***LOCAL=(list,...)***

Specifies the characteristics of the local (proxy) account to be used when selecting event records.

<b>Keyword</b>	<b>Description</b>
USERNAME=name	Specifies the name of the local account used. Full wildcarding of the name is allowed.

***LOGICAL\_NAME=(name,...)***

Specifies the logical name of the volume mounted (or dismounted) upon which selection is based. Full wildcarding of the logical name is allowed.

***MOUNT\_FLAGS=(name,...)***

Specifies the names of the volume mounting flags upon which selection is based. The available names are

CACHE=(NONE,WRITETHROUGH)  
CLUSTER  
DATACHECK=(READ,WRITE)  
FOREIGN  
GROUP  
INITIALIZATION=(ALLOCATE,CONTINUATION)  
INTERCHANGE  
MESSAGE  
NOASSIST  
NOAUTO  
NODISKQ  
NOHDR3  
NOLABEL  
NOWRITE

OVERRIDE=(options[,...]) {  
                                   ACCESSIBILITY  
                                   EXPIRATION  
                                   IDENTIFICATION  
                                   SETID  
                                   LOCK  
                                   OWNER\_IDENTIFIER }  
  
 SHARE  
 SYSTEM

**OBJECT=(list,...)**

Specifies which characteristics of an object should be used in selecting event records.

Keyword	Description
IDENTIFICATION=value	Specifies a unique object identification for the object (currently this is only the file identification (file ID) for a file).
NAME=name	Specifies the name of the object. Full wildcarding of the name is allowed.
OWNER=value	Specifies the owner (identifier value) of the object.
TYPE=name	Specifies the general object type. The available types are as follows:  <div style="margin-left: 40px;">           FILE            SYSTEM_GLOBAL_SECTION            GROUP_GLOBAL_SECTION         </div>

**PARENT=(list,...)**

Specifies which characteristics of the parent process (when a subprocess causes an event record to be generated) are used in selecting event records.

Keyword	Description
IDENTIFICATION=value	Specifies the process identification (PID) of the parent process.
NAME=name	Specifies the name of the parent process. Full wildcarding of the name is allowed.
OWNER=value	Specifies the owner (identifier value) of the parent process.
USERNAME=name	Specifies the user name of the parent process. Full wildcarding of the name is allowed.

**AUD-12 ANALYZE/AUDIT  
/SELECT**

***PRIVILEGES\_USED=(privs,...)***

Specifies the privileges of the process to be used when selecting event records. Specify any of the following privileges: SYSPRV, BYPASS, GRPPRV, and READALL.

***PROCESS=(list,...)***

Specifies the characteristics of the process to be used when selecting event records.

<b>Keyword</b>	<b>Description</b>
IDENTIFICATION=value	Specifies the PID of the process.
NAME=name	Specifies the name of the process. Full wildcarding of the process name is allowed.

***REMOTE=(list,...)***

Specifies that some characteristic of the network request is to be used in selecting event records.

<b>Keyword</b>	<b>Description</b>
IDENTIFICATION=value	Specifies the DECnet address.
NODENAME=name	Specifies the DECnet node name. Full wildcarding of the node name is allowed.
USERNAME=name	Specifies the remote user name. Full wildcarding of the remote user name is allowed.

***STATUS=type***

Specifies the type of success status to be used in selecting event records.

<b>Keyword</b>	<b>Description</b>
SUCCESSFUL	Specifies a generic success class.
FAILURE	Specifies a generic failure class.
CODE=value	Specifies a specific completion status.

***SYSTEM=(list,...)***

Specifies the characteristics of the system to be used in selecting event records.



---

<b>Keyword</b>	<b>Description</b>
IDENTIFICATION=value	Specifies the numeric identification of the system.
NAME=name	Specifies the name of the system.

---

***TERMINAL=(name,...)***

Specifies the name of the terminal to be used when selecting event records. Full wildcarding of the terminal name is allowed.

***USERNAME=(name,...)***

Specifies the user name to be used when selecting event records. Full wildcarding of the user name is allowed.

***VOLUME\_NAME=(name,...)***

Specifies that the name of the mounted (or dismounted) volume is to be used in selecting event records. Full wildcarding of the volume name is allowed.

**example**

```
$ ANALYZE/AUDIT /FULL/SELECT=USERNAME=JOHNSON -  
_ $ SYSS$MANAGER:SECURITY_AUDIT.AUDIT$JOURNAL
```

The command in this example selects all records written to the security audit log file that were generated by user JOHNSON.

---

**/SINCE**

Controls whether records dated the same or later than the specified time are selected.

**format**

***/SINCE[=time]***

***/NOSINCE***

**keyword*****time***

Specifies the time used to select records. Records dated the same or later than the specified time are selected. You can specify an absolute time, delta time, or a combination of the two. Observe the syntax rules for date and time described in the *VMS DCL Concepts Manual*.

If you specify /SINCE without the time, midnight of the current day is used.

# AUD-14 ANALYZE/AUDIT /SINCE

## example

```
$ ANALYZE/AUDIT /SINCE=25-JUL-1989 -  
_ $ SYSS$MANAGER:SECURITY_AUDIT.AUDIT$JOURNAL
```

The command in this example selects records dated later than July 25, 1989.

---

## /SUMMARY

Specifies that a summary of the selected records be produced after all records are processed.

You can use the /SUMMARY qualifier alone or in combination with the /BRIEF, /BINARY, or /FULL qualifier.

## format

```
/SUMMARY  
/NOSUMMARY
```

## keywords

None.

## example

```
$ ANALYZE/AUDIT /SUMMARY
```

The command in this example generates a summary report of all records processed, as shown in the following display:

Total records read:	10831	Records selected:	10831
Record buffer size:	512	Format buffer size:	128
Server messages:	0	Customer messages:	0
Digital CSS messages:	0	Layered prod messages:	0
Audit changes:	169	Installed db changes:	322
Login failures:	246	Breakin attempts:	0
Successful logins:	1719	Successful logouts:	951
System UAF changes:	33	Network UAF changes:	0
Rights db changes:	3	Object accesses:	6976
Volume (dis)mounts:	412		

## ANALYZE/AUDIT Interactive Mode Commands

This section describes the interactive mode commands available with the Audit Analysis Utility. The qualifiers for this section follow the standard rules of DCL grammar.

To enter interactive mode commands, press CTRL/C at any time during the processing of a full or brief interactive display. At the *COMMAND>* prompt, you can enter additional interactive mode commands, the *CONTINUE* command to resume processing of the event records, or *EXIT* to terminate the session.

---

### CONTINUE

Resumes processing of event records.

#### format

**CONTINUE**

#### parameters

None.

#### qualifiers

None.

#### example

```
COMMAND> DISPLAY/SINCE=25-JUN-1989/SELECT=USERNAME=POST  
COMMAND> CONTINUE
```

The first command in this example selects only event records generated by user JOHNSON after June 25, 1989. The second command in the example displays a report based on the new selection criteria.

---

### DISPLAY

Changes the criteria used to select event records. For a more complete description of any one of the following qualifiers, refer to the description of the same qualifier and its keywords in the preceding ANALYZE/AUDIT qualifier section.

#### format

**DISPLAY**

# AUD-16 ANALYZE/AUDIT DISPLAY

## parameters

None.

## qualifiers

### ***/BEFORE=time***

Controls whether only those records dated earlier than the specified time are selected.

### ***/BRIEF***

Controls whether a brief (one line per record) format is used in ASCII displays.

### ***/EVENT\_TYPE=event-type[,...]***

Controls whether only those records matching the specified event type are selected.

### ***/FULL***

Controls whether a full format for each record is used in ASCII displays.

### ***/IGNORE=criteria[,...]***

Controls whether records matching the specified criteria are excluded. If you specify */IGNORE* two or more times, the criteria is combined. To specify a new set of exclusion criteria, include the */REMOVE* qualifier with the */IGNORE* qualifier.

### ***/PAUSE=seconds***

For full-format displays (*/FULL*), specifies the length of time each record is displayed.

### ***/REMOVE***

Controls whether the criteria specified by the */IGNORE* and */SELECT* qualifiers are no longer to be used to select event records to be displayed.

### ***/SELECT=criteria[,...]***

Controls whether only those records matching the specified criteria are selected. If you specify */SELECT* two or more times, the criteria is combined. To specify a new set of selection criteria, include the */REMOVE* qualifier with the */SELECT* qualifier.

### ***/SINCE[=time]***

Controls whether only those records dated the same or later than the specified time are selected.

## example

```
COMMAND> DISPLAY/EVENT_TYPE=SYSUAF  
COMMAND> CONTINUE
```

The first command in this example selects records that were generated as a result of a modification to the system user authorization file (SYSUAF). The second command displays the selected records.

```
COMMAND> DISPLAY/SELECT=USERNAME=CRICK  
COMMAND> CONTINUE
```

```
.  
.
```

```
CTRL/C
```

```
COMMAND> DISPLAY/SELECT=USERNAME=WATSON  
COMMAND> CONTINUE
```

The first DISPLAY command in this example selects records that were generated by user CRICK. The second command displays the selected records. The next DISPLAY command selects records that were generated by user WATSON. The last command in the example displays all records generated by users CRICK and WATSON.

---

## EXIT

Terminates the session.

## format

EXIT

## parameters

None.

## qualifiers

None.

---

## HELP

Provides online help information for using ANALYZE/AUDIT interactive mode commands.

## format

HELP *[topic]*

## AUD-18 ANALYZE/AUDIT HELP

### parameter

#### *topic*

Specifies the command for which help information is to be displayed. If you omit the keyword, HELP displays a list of available help topics, and prompts you for a particular keyword.

### qualifiers

None.

### example

```
COMMAND> HELP DISPLAY
```

The command in this example displays help information about the interactive mode command DISPLAY.

---

## LIST

Changes the criteria used to select event records. The LIST command is a synonym for DISPLAY. For a more complete description of any one of the following qualifiers, refer to the description of the same qualifier and its keywords in the preceding ANALYZE/AUDIT qualifier section.

### format

LIST

### parameters

None.

### qualifiers

#### */BEFORE=time*

Controls whether only those records dated earlier than the specified time are selected.

#### */BRIEF*

Controls whether a brief (one line per record) format is used in ASCII displays.

#### */EVENT\_TYPE=event-type[,...]*

Controls whether only those records matching the specified record type are selected.

#### */FULL*

Controls whether a full format for each record is used in ASCII displays.

***/IGNORE=criteria[,...]***

Controls whether records matching the specified criteria are excluded. If you specify **/IGNORE** two or more times, the criteria is combined. To specify a new set of exclusion criteria, include the **/REMOVE** qualifier with the **/IGNORE** qualifier.

***/PAUSE=seconds***

For full-format displays (**/FULL**), specifies the duration for each record displayed.

***/REMOVE***

Controls whether the criteria specified by the **/IGNORE** and **/SELECT** qualifiers are no longer to be used to select event records to be displayed.

***/SELECT=criteria[,...]***

Controls whether only those records matching the specified criteria are selected. If you specify **/SELECT** two or more times, the criteria is combined. To specify a new set of exclusion criteria, include the **/REMOVE** qualifier with the **/SELECT** qualifier.

***/SINCE[=time]***

Controls whether only those records dated the same or later than the specified time are selected.

**example**

```
COMMAND> LIST/EVENT_TYPE=SYSUAF  
COMMAND> CONTINUE
```

The first command in this example selects records that were generated as a result of a modification to the system user authorization file (SYSUAF). The second command displays the selected records.

---

**NEXT FILE**

Controls whether the current security audit log file is closed and the next log file opened. If there are no other audit log files to open, the audit analysis session is terminated and control returns to DCL.

**format**

**NEXT FILE**

**parameters**

None.

**AUD-20    ANALYZE/AUDIT  
NEXT FILE**

**qualifiers**

None.

---

**NEXT RECORD**

Controls whether the next audit record is displayed. The **NEXT RECORD** command is synonymous with the command **POSITION**.

**format**

**NEXT RECORD**

**parameters**

None.

**qualifiers**

None.

---

**POSITION**

Moves the full-format display forward or backward the specified number of event records.

**format**

**POSITION** *number*

**parameter**

*number*

For positive numbers, displays the record that is the specified number of records after the current record. For negative numbers, displays the record that is the specified number of records before the current record.

**qualifiers**

None.

**example**

COMMAND> POSITION 100

The command in this example moves the display forward 100 event records.

COMMAND> POSITION -100

The command in this example moves the display back 100 event records.



---

**SHOW**

Displays information about selection or exclusion criteria currently being used to select event records.

**format**

**SHOW** *option[,...]*

**parameter**

*option[,...]*

Displays information about selection or exclusion criteria currently being used to select records. Specify one or more of the following options:

---

<b>Option</b>	<b>Meaning</b>
ALL	Displays all criteria being used to select event records.
EXCLUSION_CRITERIA	Displays the criteria being used to exclude event records.
SELECTION_CRITERIA	Displays the criteria being used to select event records.

---

**qualifiers**

None.

**example**

COMMAND> SHOW SELECTION\_CRITERIA

The command in this example displays the selection criteria currently in use to select records.



---

## Analyze/Disk\_Structure Utility

The Analyze/Disk\_Structure Utility checks the readability and validity of Files-11 Structure Level 1 and Structure Level 2 disk volumes, and reports errors and inconsistencies.

You can detect most classes of errors by invoking the utility once and using its defaults.

### format

**ANALYZE/DISK\_STRUCTURE** *device-name:[/qualifier]*

### parameter

#### ***device-name***

Specifies the disk volume or volume set to be verified. If you specify a volume set, all volumes of the volume set must be mounted as Files-11 volumes. (For information on the Mount Utility, refer to the *VMS Mount Utility Manual*.)

### usage summary

Use the following command to invoke the utility:

```
$ ANALYZE/DISK_STRUCTURE device-name: /qualifiers
```

You can terminate an ANALYZE/DISK\_STRUCTURE session by entering CTRL/C or CTRL/Y while the utility executes. You cannot resume operation by using the DCL command CONTINUE.

By default, ANALYZE/DISK\_STRUCTURE directs all output to your terminal. Use the /USAGE or /LIST qualifiers to direct output to a file.

**To repair a disk effectively, you must have read, write, and delete access to all files on the disk. To effectively scan a disk (/NOREPAIR), you must have read access to all files on the disk.**

**For a complete explanation of file access, see the *Guide to VMS System Security*.**

**You can safely use ANALYZE/DISK\_STRUCTURE on a disk that is concurrently being used for other file operations. If you specify /REPAIR, the utility locks the volume before performing any operations; this blocks volume modification. Because other users cannot create, delete, extend, or truncate files, repair operations are unimpeded and the volume is left in a consistent state.**

## ADSK-2 Analyze/Disk\_Structure Utility

**If you specify /NOREPAIR, the volume is not locked; the utility does not attempt to write to the disk. However, if users perform file operations while you run the utility, you may receive error messages that incorrectly indicate file damage. To avoid this problem, Digital recommends you run ANALYZE/DISK\_STRUCTURE when the disk is in a quiescent state.**

## ANALYZE/DISK\_STRUCTURE Qualifiers

---

### /[NO]CONFIRM

Determines whether the Analyze/Disk\_Structure Utility prompts you to confirm each repair. If you respond with Y or YES, the utility performs the repair. Otherwise, the repair is not performed.

#### format

/[NO]CONFIRM

#### description

You can only use the /CONFIRM qualifier with the /REPAIR qualifier. The default is /NOCONFIRM.

#### example

```
$ ANALYZE/DISK_STRUCTURE DBAO:/REPAIR/CONFIRM
%VERIFY-I-BACKLINK, incorrect directory back link [SYS0]SYSMAINT.DIR;1
Repair this error? (Y or N): Y
%VERIFY-I-BACKLINK, incorrect directory back link [SYSEXE]SYSBOOT.EXE;1
Repair this error? (Y or N): N
```

The command in this example causes the Analyze/Disk\_Structure Utility to prompt you for confirmation before performing the indicated repair operation.

---

### /[NO]LIST[=*filespec*]

Determines whether the Analyze/Disk\_Structure Utility produces a listing of the index file.

#### format

/LIST[=*filespec*]  
/NOLIST

#### description

If you specify /LIST, the utility produces a file that contains a listing of all FIDs, file names, and file owners. If you omit the file specification, the default is SYS\$OUTPUT. If you include a file specification without a file type, the default type is LIS. You cannot use wildcard characters in the file specification.

The default is /NOLIST.

## ADSK-4 ANALYZE/DISK\_STRUCTURE /[NO]LIST[=filespec]

### example

```
$ ANALYZE/DISK_STRUCTURE DLA2:/LIST=INDEX
$ TYPE INDEX
Listing of index file on DLA2:
31-DEC-1988 20:54:42.22

(00000001,00001,001)  INDEXF.SYS;1
                        [1,1]
(00000002,00002,001)  BITMAP.SYS;1
                        [1,1]
(00000003,00003,001)  BADBLK.SYS;1
                        [1,1]
(00000004,00004,001)  000000.DIR;1
                        [1,1]
(00000005,00005,001)  CORIMG.SYS;1
                        [1,1]
.
.
.
$
```

In this example, ANALYZE/DISK\_STRUCTURE did not find errors on the device DLA2. Since the file INDEX was specified without a file type, the system assumes a default file type of LIS. The subsequent TYPE command displays the contents of the file INDEX.LIS.

---

## /[NO]READ\_CHECK

Determines whether the Analyze/Disk\_Structure Utility performs a read check of all allocated blocks on the specified disk. When the Analyze/Disk\_Structure Utility performs a read check, it reads the disk twice; this ensures that it reads the disk correctly. The default is /NOREAD\_CHECK.

### format

/[NO]READ\_CHECK

### example

```
$ ANALYZE/DISK_STRUCTURE DMA1:/READ_CHECK
```

The command in this example directs ANALYZE/DISK\_STRUCTURE to perform a read check on all allocated blocks on the device DMA1.

## **/[NO]REPAIR**

Determines whether the Analyze/Disk\_Structure Utility repairs errors that are detected in the file structure of the specified device.

### **format**

**/[NO]REPAIR**

### **description**

The Analyze/Disk\_Structure Utility does not perform any repair operation unless you specify the /REPAIR qualifier. The file structure is software write-locked during a repair operation. The default is /NOREPAIR.

### **example**

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

The command in this example causes ANALYZE/DISK\_STRUCTURE to perform a repair on all errors found in the file structure of device DBA1.

---

## **/USAGE[=filespec]**

Specifies that a disk usage accounting file should be produced, in addition to the other specified functions of the Analyze/Disk\_Structure Utility.

### **format**

**/USAGE[=filespec]**

### **description**

If all or part of the file specification is omitted, ANALYZE/DISK\_STRUCTURE assumes a default file specification of USAGE.DAT. The file is placed in the current default directory.

### **example**

```
$ ANALYZE/DISK_STRUCTURE DBA1:/USAGE  
$ DIRECTORY USAGE
```

```
Directory DISK$DEFAULT:[ACCOUNT]
```

```
USAGE.DAT;3
```

```
Total of 1 file.
```

The first command in this example causes ANALYZE/DISK\_STRUCTURE to produce a disk usage accounting file. Since a file specification was not provided in the command line, ANALYZE/DISK\_STRUCTURE uses both the default file name and directory [ACCOUNT]USAGE.DAT. The DIRECTORY command instructs the system to display all default information.





---

## Authorize Utility

The Authorize Utility (AUTHORIZE) is a system management tool that allows you to control access to the system and to allocate user resources.

### format

**RUN AUTHORIZE**

### usage summary

To invoke AUTHORIZE, set your process default device and directory to SYS\$SYSTEM, and type RUN AUTHORIZE. To terminate AUTHORIZE, enter the EXIT command at the UAF> prompt, or press CTRL/Z.

To create a listing file of reports for selected UAF records, enter the LIST command at the UAF> prompt. For more information on listing reports, see the description of the LIST command.

**NOTE:** Use of the Authorize Utility requires write access to SYSUAF.DAT, NETPROXY.DAT, or RIGHTSLIST.DAT in the SYS\$SYSTEM directory. Write access to these files is normally restricted to users with the system UIC or the SYSPRV or BYPASS privilege.

## AUTH-2 AUTHORIZE AUTHORIZE Qualifiers

### AUTHORIZE Qualifiers

Table AUTH-1 describes the qualifiers which are common to the ADD, COPY, DEFAULT, and MODIFY qualifiers.

**Table AUTH-1: Summary of Qualifiers for the ADD, COPY, DEFAULT, and MODIFY Commands**

Qualifier	Function
/ACCESS [=(range[,...])]	<p>Specifies hours of access for all modes of access. Syntax for range specification is:</p> <p>/[NO]ACCESS=( [PRIMARY], [n-m], [n], [...], [SECONDARY], [n-m], [n], [...])</p> <p>Specify hours as integers from 0 to 23, inclusive. Hours may be specified as single hours (n), or as ranges of hours (n-m). If the ending hour of a range is earlier than the starting hour, the range extends from the starting hour through midnight to the ending hour. The first set of hours after the keyword PRIMARY specifies hours on primary days; the second set of hours after the keyword SECONDARY specifies hours on secondary days. Note that hours are <i>inclusive</i>; that is, if you grant access during a given hour, access extends to the end of that hour.</p>
/ACCOUNT=account-name	<p>Specifies a 1 through 8 alphanumeric character string that is the default name for the account (for example, a billing name or number). By default, a blank account name is assigned.</p>
/ADD_IDENTIFIER /NOADD_IDENTIFIER	<p>Adds identifiers for the user name and account name to the rights database.</p>
/ASTLM=value	<p>An integer with a minimum value of 2 specifying the number of ASTs the user can have queued at one time.</p>
/BATCH[=(range[,...])]	<p>Specifies hours of access permitted for batch jobs. For a description of the range specification, see the /ACCESS qualifier.</p>
/BIOLM=value	<p>Specifies a buffered I/O count limit for the BIOLM field of the UAF record. The buffered I/O count limit is the maximum number of buffered I/O operations, such as terminal I/O, that can be outstanding at one time.</p>
/BYTLM=value	<p>Specifies the buffered I/O byte limit for the BYTLM field of the UAF record. The buffered I/O byte limit is the maximum number of bytes of nonpaged system dynamic memory that a user's job may consume at one time. Nonpaged dynamic memory is used for operations such as I/O buffering, mailboxes, file-access windows.</p>

(continued on next page)

**Table AUTH-1 (Cont.): Summary of Qualifiers for the ADD, COPY, DEFAULT, and MODIFY Commands**

<b>Qualifier</b>	<b>Function</b>
/CLI=cli-name	Specifies the name of the default command language interpreter (CLI) for the CLI field of the UAF record. The cli-name is 1 through 12 alphanumeric characters and should be either DCL or MCR. By default, the DCL CLI is used.
/CLITABLES	Specifies user-defined CLI tables for the account, from 1 to 31 characters. If none is specified, LOGINOUT uses the default CLI.
/CPU TIME=time	Specifies the maximum process CPU time for the CPU field of the UAF record. The maximum process CPU time is the maximum CPU time a user's process can take per session. You must specify a delta-time value. The default of 0 means infinite time.
/DEFPRIVILEGES =( <b>[NO]</b> privname[,...])	Specifies default privileges for the user; that is, those enabled at login time. A NO prefix removes a privilege from the user. The keyword <b>[NO]ALL</b> specified with the /DEFPRIVILEGES qualifier disables or enables all user privileges.
/DEVICE=device-name	Specifies the name of the user's default device at login. The device-name is a 1 through 31 alphanumeric character string. If you omit the colon from the device-name value, a colon is appended. The default blank value is interpreted as SYS\$SYSDISK.
/DIALUP [=(range[,...])]	Specifies hours of access permitted for dial-up logins. For a description of the range specification, see the /ACCESS qualifier.
/DIOLM=value	Specifies the direct I/O count limit for the DIOLM field of the UAF record. The direct I/O count limit is the maximum number of direct I/O operations (usually disk) that can be outstanding at one time. The value is an integer of at least 2 and has a default of 18.
/DIRECTORY =directory-name	Specifies the default directory-name for the DIRECTORY field of the UAF record. The directory-name is 1 through 63 alphanumeric characters. Brackets are added to the directory name if omitted. By default, the directory-name <b>[USER]</b> is assigned.
/ENQLM=value	Specifies the lock queue limit for the ENQLM field of the UAF record. The lock queue limit is the maximum number of locks that can be queued at one time.
/EXPIRATION=time	Expiration date and time of the account. Default is 180 days for nonprivileged users.

(continued on next page)

**AUTH-4 AUTHORIZE**  
**AUTHORIZE Qualifiers**

**Table AUTH-1 (Cont.): Summary of Qualifiers for the ADD, COPY, DEFAULT, and MODIFY Commands**

<b>Qualifier</b>	<b>Function</b>
/FILLM=value	Specifies the open file limit for the FILLM field of the UAF record. The open file limit is the maximum number of files that can be open at one time, including active network logical links.
/FLAGS =([NO]option[,...])	Specifies login flags for the user. A NO in front of the flag clears the flag. The following are valid options:
AUDIT	Audits all security-related actions.
AUTOLOGIN	Restricts the account to the autologin mechanism.
CAPTIVE	Places user under the control of the login command procedure; denies the user access to the DCL command level.
DEFCLI	Restricts the user to using the default command language interpreter and CLI tables.
DISCTLY	Disables the CTRL/Y function.
DISFORCE_PWD_CHANGE	Removes the requirement that the user must change expired passwords at login.
DISIMAGE	Prevents the user from executing the RUN or MCR commands or from using the foreign command mechanism in DCL.
DISMAIL	Prevents mail delivery to the user.
DISNEWMAIL	Suppresses announcements of new mail at login time.
DISRECONNECT	Disables automated reconnection to an existing process when a terminal connection has been interrupted.
DISREPORT	Suppresses time of last login and other security reports.
DISUSER	Prevents the user from logging in.

(continued on next page)

**Table AUTH-1 (Cont.): Summary of Qualifiers for the ADD, COPY, DEFAULT, and MODIFY Commands**

Qualifier	Function
	DISWELCOME      Suppresses the system login message.
	GENPWD            Requires the user to use generated passwords.
	LOCKPWD          Prevents the user from changing the password for the account.
	PWD_EXPIRED      Marks password as expired.
	PWD2_EXPIRED     Marks second password as expired.
	RESTRICTED        Prevents the user from accessing the DCL command level (disables CTRL/Y interrupts) until the system and user login command procedure are executed.
/GENERATE_PASSWORD [=keyword]	Invokes the password generator to generate user passwords. Specify one of the following keywords:
	ALL                Generate primary and secondary passwords
	BOTH              Generate primary and secondary passwords
	CURRENT          Generate primary, secondary, or both passwords as specified for the DEFAULT account
	PRIMARY          Generate primary password only
	SECONDARY        Generate secondary password only
	Note that the /GENERATE_PASSWORD and /PASSWORD qualifiers are mutually exclusive.
/INTERACTIVE [=(range[,...])]	Specifies hours of access for interactive logins. For a description of the range specification, see the /ACCESS qualifier.
/JTQUOTA=value	Specifies the initial byte quota with which the job-wide logical name table is to be created.
/LGICMD=file-spec	Specifies the name of the default login command file. Defaults to the device specified for /DEVICE, the directory specified for /DIRECTORY, a file name of LOGIN, and a file type of COM.
/LOCAL[=(range[,...])]	Specifies hours of access for interactive logins via local terminals. For a description of the range specification, see the /ACCESS qualifier.
/MAXACCTJOBS=value	Specifies the maximum number of batch, interactive, and detached processes which may be active at one time for all users of the same account. The default value of 0 represents an unlimited number.

(continued on next page)

**AUTH-6 AUTHORIZE**  
**AUTHORIZE Qualifiers**

**Table AUTH-1 (Cont.): Summary of Qualifiers for the ADD, COPY, DEFAULT, and MODIFY Commands**

<b>Qualifier</b>	<b>Function</b>
<code>/MAXDETACH=value</code>	Specifies the active process limit for the MAXDETACH field of the UAF record. The active process limit is the total number of detached processes permitted at one time. The keyword NONE indicates that the user cannot create detached processes. The default value of 0 represents an unlimited number.
<code>/MAXJOBS=value</code>	Specifies the maximum number of processes (interactive, batch, detached, and network) which may be active at one time for the specified user. The first four network jobs are not counted. The default value of 0 represents an unlimited number.
<code>/MODIFY_IDENTIFIER</code>	Specifies whether the identifier associated with a user record is to be modified in the rights database. The qualifier only applies if the UIC or user name qualifier field in the UAF is modified. The default is /MODIFY_IDENTIFIER.
<code>/NETWORK [=(range[,...])]</code>	Specifies hours of access for network batch jobs. For a description of the range specification, see the /ACCESS qualifier.
<code>/OWNER=owner-name</code>	The owner-name specifies the name of the owner of the account. This name can be used, for example, for billing purposes. The owner-name is 1 through 31 characters and has a blank name for its default.
<code>/PASSWORD=(password1 [,password2])</code>	Specifies up to two passwords for login. Passwords can be from 0 to 31 characters in length, and can include alphanumeric characters, dollar signs, and underscores. If omitted, password defaults to USER. To set only the first password, specify /PASSWORD=password1; to set both the first and second password, specify /PASSWORD=(password1,password2). To change the first password without affecting the second, specify /PASSWORD=(password,""). To change the second password without affecting the first, specify /PASSWORD="",password. To set both passwords to null, specify /NOPASSWORD.
<code>/PGFLQUOTA=value</code>	Specifies the paging file limit for the PGFLQUOTA field of the UAF record. The paging file limit is the maximum number of pages that the user's process can use in the system paging file. The minimum value is 2048 pages for typical interactive processes.
<code>/PRCLM=value</code>	Specifies the subprocess creation limit for the PRCLM field of the UAF record. The subprocess creation limit is the maximum number of subprocesses that can exist at one time for the user's process.

(continued on next page)

AUTHORIZE AUTH-7  
AUTHORIZE Qualifiers

**Table AUTH-1 (Cont.): Summary of Qualifiers for the ADD, COPY, DEFAULT, and MODIFY Commands**

Qualifier	Function
<code>/PRIMEDAYS =([NO]day[,...])</code>	Defines the primary and secondary days of the week for logging in. Specify primary days as MON, TUE, WED, THU, FRI, SAT, and SUN. Specify secondary days as NOMON, NOTUE, NOWED, NOTHU, NOFRI, NOSAT, and NOSUN. Defaults to MON, TUE, WED, THU, FRI, NOSAT, NOSUN. Any days omitted from the list take their default value.
<code>/PRIORITY=value</code>	Specifies the default base priority for the PRIO field of the UAF record. The value is an integer in the range of 0 through 31 with a default value of 4 for timesharing users.
<code>/PRIVILEGES =([NO]privname[,...])</code>	Specifies a list of privileges that the user is granted at login. NO in front of a privilege removes the privilege. A specification of ALL means all privileges. Defaults to NETMBX and TMPMBX.
<code>/[NO]PWDEXPIRED</code>	Specifies whether a password is valid only for the first login. In order to log in to the account after the first session, the user must specify a new password during this session with the DCL command SET PASSWORD. The /PWDEXPIRED qualifier only affects accounts having a nonzero password lifetime.
<code>/[NO]PWDLIFETIME=time</code>	Specifies or negates the length of time a password is valid. You must specify a delta-time value. If a period longer than the specified time has elapsed when the user logs in, a warning message is displayed, and the password is marked as expired. The default is 90 00:00.
<code>/PWDMINIMUM=value</code>	Specifies minimum password length in characters (default is 6). Note that this value is enforced only by the DCL command SET PASSWORD. Passwords in violation of this value may be specified to AUTHORIZE.
<code>/REMOTE [=(range[,...])]</code>	Specifies hours of access permitted for interactive login via network remote terminals (that is, SET HOST). For a description of the range specification, see the /ACCESS qualifier.
<code>/REMOVE_IDENTIFIER</code>	Specifies whether the user name and account name identifiers should be removed from the rights database when a UAF record is removed from SYSUAF.DAT. This qualifier is used only with the REMOVE command. The account name identifier is removed only if there are no remaining UAF records with the same group as the deleted record. If identifiers should not be removed, specify /NOREMOVE_IDENTIFIER. The default is /REMOVE_IDENTIFIER.

(continued on next page)

**AUTH-8 AUTHORIZE**  
**AUTHORIZE Qualifiers**

**Table AUTH-1 (Cont.): Summary of Qualifiers for the ADD, COPY, DEFAULT, and MODIFY Commands**

<b>Qualifier</b>	<b>Function</b>
<code>/SHRFILLM=value</code>	Specifies the maximum number of shared files the user may have open at one time. The default value of 0 represents an infinite number.
<code>/TQELM</code>	Specifies the total number of entries in the timer queue, plus the number of temporary common event flag clusters that the user can have at one time.
<code>/UIC=uic</code>	Specifies the user identification code (UIC) for the UIC field of the UAF record. The UIC value, specified in octal, is a group and member number separated by a comma and enclosed in brackets. The group number must be in the range 1-37776 (octal), the member number in the range 0-177776 (octal). The default UIC value is [200,200].
<code>/WSDEFAULT=value</code>	Specifies the size in pages of the user's default working set. The minimum size is 50 pages.
<code>/WSEXTENT=value</code>	Specifies the size in pages of the user's working set extent. The minimum size is 50 pages.
<code>/WSQUOTA=value</code>	Specifies the size in pages of the user's working set quota. The minimum size is 50 pages.



## AUTHORIZE Commands

This section describes the AUTHORIZE commands and provides examples of their use. You can abbreviate any command, keyword, or qualifier as long as the abbreviation is not ambiguous. The asterisk and the percent sign can be used as wildcard characters in the specification of user names, node names, and UICs.

---

### ADD

Adds a user record to the system UAF and corresponding identifiers to the rights database.

#### format

**ADD** *newusername*

#### parameter

##### *newusername*

Specifies the name of the user record to be included in the system UAF. The **newusername** parameter is a string of 1 through 12 alphanumeric characters and may contain underscores. Although dollar signs are permitted, they are usually reserved for system names.

While fully numeric **newusernames** are permitted, fully numeric identifiers are not. Numeric **newusernames** do not receive corresponding identifiers and should be avoided.

#### qualifiers

##### **See Table AUTH-1.**

Qualifiers not specified take their values from the DEFAULT record, except that the default password is always USER. Typically, you take defaults on the limits, priority, privileges, command interpreter, and sometimes device; as a result, you type only the password, UIC, directory, owner, account, and sometimes device.

**NOTE:** When you add a new record to the UAF and a rights database exists, an identifier with the user name is added to the rights database (unless you specify the /NOADD\_IDENTIFIER qualifier). If the record is the first member of a new UIC group, and you specify an account name with the record, a group identifier corresponding to the account name is also added to the rights database.

# AUTH-10 AUTHORIZE ADD

## example

```
UAF> ADD ROBIN /PASSWORD=SP0152/UIC=[014,006] -  
_ /DEVICE=SYSS$USER/DIRECTORY=[ROBIN]/CLITABLES=DCLTABLES -  
_ /OWNER="JOSEPH ROBIN" /ACCOUNT=INV  
%UAF-I-ADDMSG, user record successfully added  
%UAF-I-RDBADDMSGU, identifier ROBIN value: [000014,000006] added to  
RIGHTSLIST.DAT  
%UAF-I-RDBADDMSGU, identifier INV value: [000014,177777] added to RIGHTSLIST.DAT
```

This example illustrates the typical ADD command and qualifiers. The record that results from this command appears in the description of the SHOW command.

---

## ADD/IDENTIFIER

Adds an identifier to the rights database.

### format

**ADD/IDENTIFIER** [*id-name*]

### parameter

#### *id-name*

Specifies the name of the identifier to be added to the rights database. If you omit the name, you must specify the /USER qualifier. The identifier name is a string of 1 through 31 alphanumeric characters that may contain underscores and dollar signs. The name must contain at least one nonnumeric character.

### qualifiers

**/ATTRIBUTES=(keyword[,...])**

Specifies attributes to be associated with the new identifier. The following are valid keywords:

- |              |  |
|--------------|--|
| [NO]RESOURCE | Determines whether holders of the identifier may charge resources to it. The default is NORESOURCE.                                    |
| [NO]DYNAMIC  | Determines whether unprivileged holders of the identifier may add or remove it from the process rights list. The default is NODYNAMIC. |

**/USER=user-spec**

Scans the UAF record for the specified user and creates the corresponding identifier. Specify **user-spec** by user name or UIC. You can use the asterisk wildcard to specify multiple user names or UICs. Full use of the asterisk and percent wildcards is permitted for user names; UICs must be in the form [\*,\*], [n,\*], [\*,n], or [n,n]. A wildcard user name specification (\*) creates identifiers alphabetically by user name; a wildcard UIC specification ([\*,\*]) creates them in numerical order by UIC.

***/VALUE=value-specifier***

Specifies the value to be attached to the identifier. The following are valid formats for the value-specifier:

**IDENTIFIER:integer** An integer value in the range of 65,536 to 268,435,455. You may also specify the value in hexadecimal (precede the value with %X) or octal (precede the value with %O).

Note that %X80000000 is added to the value you specify in order to differentiate general identifiers from UIC identifiers.

**UIC:uic** A UIC value in the standard UIC format

If the **/VALUE** qualifier is not specified, **AUTHORIZE** assigns an unused identifier value.

**example**

```
UAF> ADD/IDENTIFIER/VALUE=UIC:[300,011] INVENTORY
%UAF-I-RDBADMSGU, identifier INVENTORY value: [000300,000011] added to
RIGHTSLIST.DAT
```

The command in this example adds an identifier named **INVENTORY** to the rights database. By default, the identifier is not marked as a resource.

---

**ADD/PROXY**

Adds user entries to the network proxy authorization file.

**format**

**ADD/PROXY** *node::remote-user local-user[,...]*

**parameters**

***node***

Specifies a node name (1 through 6 alphanumeric characters). If you specify an asterisk, the specified remote user on all nodes is served by the account specified as **local-user**.

***remote-user***

Specifies the user name or UIC of a user at a remote node. If you specify an asterisk, all users at the specified node are served by the local user. You can also specify a wildcard asterisk in the group and member fields of the UIC.

***local-user***

Specifies the user names of from 1 to 16 users on the local node. If you specify an asterisk, a local-user name equal to remote-user name will be used.

## AUTH-12 AUTHORIZE ADD/PROXY

### positional qualifier

#### */DEFAULT*

Establishes the specified user name as the default proxy account. The remote user can request proxy access to an authorized account other than the default proxy account by specifying the name of the proxy account in the access control string of the network operation.

### example

```
UAF> ADD/PROXY MISHA:* MARCO/DEFAULT, OSCAR  
%UAF-I-NAFADDMSG, record successfully added to NETPROXY.DAT
```

The command in this example specifies that any user on the remote node MISHA can, by default, use the MARCO account on the local node for DECnet tasks such as remote file access. Remote users can also access the OSCAR proxy account by specifying the user name OSCAR in the access control string when remote node access is attempted.

---

### COPY

Creates a new system UAF record that duplicates an existing UAF record.

### format

**COPY** *oldusername newusername*

### parameters

#### *oldusername*

Old user name for an existing user record.

#### *newusername*

New user name for a new user record. The user name is a string of 1 through 12 alphanumeric characters.

### qualifiers

#### *See Table AUTH-1.*

Qualifiers not specified in the command remain unchanged. However, since password verification includes the user name as well as the password, it will generally fail when you attempt to use a new user name with an old password. (Only null passwords can be effectively transferred from one user record to another by the COPY command.) Include the password whenever you use the COPY command.

## example

```
UAF> COPY ROBIN SPARROW /PASSWORD=SP0152
%UAF-I-COPMSG, user record copied
%UAF-E-RDBADDERRU, unable to add SPARROW value: [000014,00006] to RIGHTSLIST.DAT
-SYSTEM-F-DUPIDENT, duplicate identifier
```

The command in this example adds a record for Thomas Sparrow that is identical, except for the password, to that of Joseph Robin. Note that since there is no change in the UIC value, no identifier is added to RIGHTSLIST.DAT. AUTHORIZE issues a “duplicate identifier” error message.

---

## CREATE/PROXY

Creates and initializes the network proxy authorization file, NETPROXY.DAT.

### format

**CREATE/PROXY**

---

## CREATE/RIGHTS

Creates and initializes the rights database, RIGHTSLIST.DAT.

### format

**CREATE/RIGHTS**

## example

```
UAF> CREATE/RIGHTS
%UAF-E-RDBCREERR, unable to create RIGHTSLIST.DAT
-RMS-E-FEX, file already exists, not superseded
```

You can use the command in this example to create and initialize a new rights database. Note, however, that RIGHTSLIST.DAT is created automatically during the installation process. Thus you must delete or rename the existing file before creating a new one.

## AUTH-14 AUTHORIZE DEFAULT

---

### DEFAULT

Modifies the system UAF's DEFAULT record.

### format

DEFAULT

### qualifiers

*See Table AUTH-1.*

Qualifiers not specified in the command remain unchanged.

### example

```
UAF> DEFAULT /DEVICE=SYS$USER/LGICMD=SYS$MANAGER:SECURELGN -  
_ /PRIVILEGES=(TMPMBX, GRPNAM, GROUP)  
%-UAF-MDFYMSG, user record(s) updated
```

The command in this example modifies the DEFAULT record, changing the default device, default login command file, and default privileges.

---

### EXIT

Enables you to exit from AUTHORIZE and return to DCL command level. You can also return to command level by pressing CTRL/Z.

### format

EXIT

---

### GRANT/IDENTIFIER

Grants the specified identifier to the user.

### format

GRANT/IDENTIFIER *id-name user-spec*

### parameters

#### *id-name*

Specifies the identifier name. Specify the name in identifier ID format (see the ADD/IDENTIFIER command).

#### *user-spec*

Specifies the UIC identifier corresponding to the user (see the ADD/IDENTIFIER command).

## qualifier

**/ATTRIBUTES=(keyword[,...])**

Specifies attributes to be associated with the identifier. The following are valid keywords:

- |              |  |
|--------------|--|
| [NO]RESOURCE | Determines whether holders of the identifier may charge resources to it. The default is NORESOURCE.                                    |
| [NO]DYNAMIC  | Determines whether unprivileged holders of the identifier can add or remove it from the process rights list. The default is NODYNAMIC. |

## example

```
UAF> GRANT/IDENTIFIER INVENTORY [300,015]
%UAF-I-GRANTMSG, identifier INVENTORY granted to CRAMER
```

The command in this example grants the identifier `INVENTORY` to a user with the UIC [300,015]. The user Cramer becomes the holder of the identifier and any resources associated with it. The following command produces the same result:

```
UAF> GRANT/IDENTIFIER INVENTORY CRAMER
```

---

## HELP

Lists and explains `AUTHORIZE` commands and qualifiers.

## format

**HELP** *[command-name]*

## parameter

***command-name***

Specifies the name of an `AUTHORIZE` command.

## example

```
UAF> HELP MODIFY/WSDEFAULT
```

The command in this example displays information about the `/WSDEFAULT` qualifier:

```
MODIFY
```

```
/WSDEFAULT=n
```

```
Initial limit of a working set for the user process.
```

# AUTH-16 AUTHORIZE LIST

---

## LIST

Writes reports for selected UAF records to a listing file, SYSUAF.LIS.

### format

**LIST** [*user-spec*]

### parameter

#### ***user-spec***

Specifies the user name or UIC of the desired UAF record. If you omit the user-spec parameter, the user records of all users are listed. The asterisk and percent sign wildcards are permitted in the user name.

### qualifiers

#### ***/BRIEF***

Specifies that a brief report be written to SYSUAF.LIS. */BRIEF* is the default qualifier.

#### ***/FULL***

Specifies that a full report be written to SYSUAF.LIS, including identifiers held by the user.

### example

```
UAF> LIST ROBIN/FULL
%UAF-I-LSTMSG1, writing listing file
%UAF-I-LSTMSG2, listing file SYSUAF.LIS complete
```

This command lists a full report for the user record ROBIN.

---

## LIST/IDENTIFIER

Creates a listing file (RIGHTSLIST.LIS) to which identifier information is written.

### format

**LIST/IDENTIFIER** [*id-name*]

### parameter

#### ***id-name***

Specifies an identifier name. You can specify the asterisk wildcard character (\*) to list all identifiers. If you omit the identifier name, you must specify */USER* or */VALUE*.



## qualifiers

### ***/BRIEF***

Specifies a brief listing in which only the identifier name, value and attributes appear.

### ***/FULL***

Specifies a full listing, in which the names of the identifier's holders are displayed along with the identifier's name, value, and attributes. */FULL* is the default listing format.

### ***/USER=user-spec***

Specifies one or more users whose identifiers are to be listed. **User-spec** may be a user name or UIC. You can use the asterisk wildcard to specify multiple user names or UICs. Full use of the asterisk and percent wildcards is permitted for user names; UICs must be in the form [\*,\*], [n,\*], [\*,n], or [n,n]. A wildcard user name specification (\*) lists identifiers alphabetically by user name; a wildcard UIC specification ([\*,\*]) lists them numerically by UIC.

### ***/VALUE=value-specifier***

Specifies the value of the identifier to be listed. The following are valid formats for the value-specifier:

IDENTIFIER:integer	An integer value in the range of 65,536 to 268,435,455. You may also specify the value in hexadecimal (precede the value with %X) or octal (precede the value with %O).  Note that %X80000000 is added to the value you specify in order to differentiate general identifiers from UIC identifiers.
UIC:uic	A UIC value in the standard UIC format.

## example

```
UAF> LIST/IDENTIFIER INVENTORY
%UAF-I-LSTMSG1, writing listing file
%UAF-I-RLSTMSG, listing file RIGHTSLIST.LIS complete
```

The command in this example generates a full listing for the identifier **INVENTORY**, including its value (in hexadecimal), holders, and attributes.

**AUTH-18    AUTHORIZE  
             LIST/PROXY**

---

**LIST/PROXY**

Creates a listing file of the network proxy database entries.

**format**

**LIST/PROXY**

---

**LIST/RIGHTS**

Lists identifiers held by the specified identifier or, if /USER is specified, all identifiers held by the specified users.

**format**

**LIST/RIGHTS** *[id-name]*

**parameter**

*[id-name]*

Specifies the name of the identifier associated with the user. Specify the identifier in UIC format. If you omit the identifier name, you must specify the /USER qualifier.

**qualifier**

**/USER=user-spec**

Specifies a user whose identifiers are to be listed. **User-spec** may be a user name or UIC. You can use the asterisk wildcard to specify multiple user names or UICs. Full use of the asterisk and percent wildcards is permitted for user names; UICs must be in the form *[\*,\*]*, *[n,\*]*, *[\*,n]*, or *[n,n]*. A wildcard user name specification (*\**) or wildcard UIC specification (*[\*,\*]*) lists all identifiers held by users. The wildcard user name specification lists holders' user names alphabetically; the wildcard UIC specification lists them in the numerical order of their UICs.

---

**MODIFY**

Changes values in a system UAF user record.

**format**

**MODIFY** *username /qualifier[,...]*

## parameter

### *username*

Specifies the name of a user in the system UAF. The asterisk and percent sign wild card characters are permitted in the user name. When you specify a single asterisk for the user name, you modify the records of all users.

## qualifiers

*See Table AUTH-1.*

Qualifiers not specified in the command remain unchanged.

## example

```
UAF> MODIFY ROBIN /PASSWORD=SP0172
%UAF-I-MDFYMSG, user record(s) updated
```

The command in this example changes the password for user ROBIN without altering any other values in the record.

---

## MODIFY/IDENTIFIER

Modifies an identifier in the rights database.

## format

**MODIFY/IDENTIFIER** *id-name*

## parameter

### *id-name*

Specifies the name of an identifier to be modified.

## qualifiers

**/ATTRIBUTES=(keyword[,...])**

Specifies attributes to be associated with the modified identifier. The following are valid keywords:

- |              |   |
|--------------|---|
| [NO]RESOURCE | Determines whether holders of the identifier can charge resources to it.<br><br>If you specify RESOURCE, a holder named with the /HOLDER qualifier gains the right to charge resources to the identifier. If you specify NORESOURCE, the holder loses the right to charge resources. If you specify NORESOURCE and do not name any holder (if /HOLDER is not specified), all holders lose the right to charge resources. The default is NORESOURCE. |
|--------------|---|

## AUTH-20 AUTHORIZE MODIFY/IDENTIFIER

[NO]DYNAMIC      Determines whether unprivileged holders of the identifier can add or remove it from the process rights list. The default is NODYNAMIC.

### ***/HOLDER=username***

Specifies the holder of an identifier whose attributes are to be modified. The /HOLDER qualifier is used only in conjunction with the /ATTRIBUTES qualifier. If you specify /HOLDER, the /NAME and /VALUE qualifiers are ignored.

### ***/NAME=id-name***

Specifies a new identifier name to be associated with the identifier.

### ***/VALUE=value-specifier***

Specifies a new identifier value. Note that an identifier value cannot be modified from a UIC to a non-UIC format or vice versa. The following are valid formats for the value-specifier:

IDENTIFIER:integer      An integer value in the range of 65,536 to 268,435,455. You can also specify the value in hexadecimal (precede the value with %X) or octal (precede the value with %O).

Note that %X80000000 is added to the value you specify in order to differentiate general identifiers from UIC identifiers.

UIC:uic      A UIC value in the standard UIC format.

## example

```
UAF> MODIFY/IDENTIFIER OLD_ID /NAME=NEW_ID
%UAF-I-RDBMDFYMSG, identifier OLD_ID modified
```

The command in this example changes the name of the OLD\_ID identifier to NEW\_ID.

---

## MODIFY/PROXY

Modifies an entry in the network proxy authorization file (NETPROXY.DAT).

### format

**MODIFY/PROXY** *node::remote-user*

### parameters

#### ***node***

Specifies a node name (1 through 6 alphanumeric characters). If you specify an asterisk, the specified remote user on all nodes is served by the local user.

***remote-user***

Specifies the user name of a user at a remote node. If you specify an asterisk, all users at the specified node are served by the local-user.

For non-VMS systems which implement DECnet Phase IV+, specifies the UIC of a user at a remote node. You can specify a wildcard asterisk in the group and member fields of the UIC.

**qualifier**

***/DEFAULT[=local-user]***

***/NODEFAULT***

Designates the default user name on the local node through which proxy access from the remote user is directed. If */NODEFAULT* is specified, removes the default designation.

**example**

```
UAF> MODIFY/PROXY MISHA::MARCO /DEFAULT=JOHNSON
%UAF-I-NAFADDMMSG, record successfully modified in NETPROXY.DAT
```

The command in this example changes the default proxy account for user MARCO on the remote node MISHA to the JOHNSON account.

---

**MODIFY/SYSTEM\_PASSWORD**

Changes the system password.

**format**

**MODIFY/SYSTEM\_PASSWORD=*system-password***

**parameter**

***system-password***

Specifies the new system password.

**example**

```
UAF> MODIFY/SYSTEM_PASSWORD=ABRACADABRA
UAF>
```

This command changes the system password to ABRACADABRA.

## AUTH-22 AUTHORIZE REMOVE

---

### REMOVE

Deletes a system UAF user record and corresponding identifiers in the rights database. The DEFAULT and SYSTEM records cannot be deleted.

#### format

**REMOVE** *username*

#### parameter

*username*

Specifies the name of a user in the system UAF.

#### qualifier

**/[NO]REMOVE\_IDENTIFIER**

Specifies whether the user name and account name identifiers should be removed from the rights database when a record is removed from the UAF. If there are two UAF records with the same UIC, the user name identifier is removed only when the second record is deleted. Similarly, the account name identifier is removed only if there are no remaining UAF records with the same group as the deleted record.

#### example

```
UAF> REMOVE ROBIN
%UAF-I-REMMMSG, record removed from SYSUAF.DAT
%UAF-I-RDBREMMMSGU, identifier ROBIN value: [000014,000006] removed from
RIGHTSLIST.DAT
```

The command in this example deletes the record for user ROBIN from the system UAF and ROBIN's UIC identifier from RIGHTSLIST.DAT.

---

### REMOVE/IDENTIFIER

Removes an identifier from the rights database.

#### format

**REMOVE/IDENTIFIER** *id-name*

#### parameter

*id-name*

Specifies the name of an identifier in the rights database.

## example

```
UAF> REMOVE/IDENTIFIER Q1SALES
%UAF-I-RDBREMSGU, identifier Q1SALES value %X80010024 removed from
RIGHTSLIST.DAT
```

The command in this example removes the identifier Q1SALES from the rights database. All of its holder records are removed with it.

---

## REMOVE/PROXY

Deletes network proxy access for the specified remote user. The /PROXY qualifier is required.

### format

```
REMOVE/PROXY node::remote-user [local-user,...]
```

### parameters

#### *node*

Specifies the name of a network node in the network UAF.

#### *remote-user*

Specifies the user name or UIC of a user on a remote node. The asterisk wildcard character is permitted in the remote-user specification.

#### *local-user*

Specifies the user name of from 1 to 16 users on the local node. If no local user is specified, proxy access to all local accounts is removed.

## example

```
UAF> REMOVE/PROXY MISHA::MARCO
%UAF-I-NAFDONEMSG, record removed from NETPROXY.DAT
```

The command in this example deletes the record for MISHA::MARCO from the network proxy authorization file, removing all proxy access to the local node for user MARCO on node MISHA.

---

## RENAME

Renames a system UAF record.

### format

```
RENAME oldusername newusername
```

## AUTH-24 AUTHORIZE RENAME

### parameters

***oldusername***

Specifies the name of a user currently in the system UAF.

***newusername***

Specifies the new user name.

### qualifiers

***/[NO]MODIFY\_IDENTIFIER***

Specifies whether the corresponding identifier is renamed.

***/[NO]PASSWORD[=(password[,password2])]***

See Table AUTH-1.

Because password verification includes the user name as well as the password, it will generally fail when you attempt to use a new user name with an old password. You must include a new password whenever you use the RENAME command unless you specify a null password with /NOPASSWORD.

***/GENERATE\_PASSWORD***

See Table AUTH-1.

### example

```
UAF> RENAME HAWKES KRAMERDOVE/PASSWORD=MARANNKRA
%UAF-I-ZZPRACREN, proxies to HAWKES renamed
%UAF-I-RENMSG, user record renamed
%UAF-I-RDBMDFYMSG, identifier HAWKES modified
```

The command in this example changes the name of the account Hawkes to Kramerdove, modifies the user name identifier for the account, and renames all proxies to the account.

---

## RENAME/IDENTIFIER

Renames an identifier in the rights database.

### format

**RENAME/IDENTIFIER** *old-id-name new-id-name*

### parameters

***old-id-name***

Specifies the name of an identifier to be renamed.

***new-id-name***

Specifies the new identifier name.



## example

```
UAF> RENAME/IDENTIFIER Q1SALES Q2SALES  
%UAF-I-RDBMDFYMSG, identifier Q1SALES modified
```

The command in this example renames the identifier Q1SALES to Q2SALES.

---

## REVOKE/IDENTIFIER

Revokes an identifier held by a user.

### format

```
REVOKE/IDENTIFIER id-name user-spec
```

### parameters

#### *id-name*

The identifier name. Specify the name in identifier ID format (see the ADD/IDENTIFIER command).

#### *user-spec*

An identifier (UIC or non-UIC format) that specifies the user (see the ADD/IDENTIFIER command).

## example

```
UAF> REVOKE/IDENTIFIER INVENTORY CRAMER  
%UAF-I-REVOKEMSG, identifier INVENTORY revoked from CRAMER
```

The command in this example revokes the identifier INVENTORY from the user Cramer. Cramer loses the identifier and any resources associated with it.

Note that, since rights identifiers are stored in numeric format, it is not necessary to change records for users holding a renamed identifier.

---

## SHOW

Displays reports for selected UAF records on the current SYS\$OUTPUT device.

### format

```
SHOW user-spec
```

# AUTH-26 AUTHORIZE SHOW

## parameter

### *user-spec*

Specifies the user name or UIC of the desired UAF record. If you omit the user-spec parameter, the UAF records of all users are listed. The asterisk and percent sign wildcard characters are permitted in the user name.

## qualifiers

### */BRIEF*

Specifies that a brief report be displayed. If you omit the /BRIEF qualifier, a full report is displayed.

### */FULL*

Specifies that a full report be displayed, including identifiers held by the user.

## example

UAF> SHOW ROBIN

The command in this example displays a full report for the user ROBIN. The display corresponds to the first example in the description of the ADD command. Note that most defaults are in effect.

```
Username: ROBIN                               Owner: JOSEPH ROBIN
Account: VMS                                  UIC: [14,6] ([INV,ROBIN])
CLI: DCL                                       Tables: DCLTABLES
Default: SYS$USER:[ROBIN]
LGICMD:
Login Flags:
Primary days: Mon Tue Wed Thu Fri
Secondary days:                               Sat Sun
No access restrictions
Expiration: (none) Pwdminimum: 6 Login Fails: 0
Pwdlifetime: (none) Pwdchange: 15-APR-1989 14:08
Last Login: (none) (interactive), (none) (non-interactive)
Maxjobs: 0 Fillm: 20 Byt1m: 12480
Maxacctjobs: 0 Shrfillm: 0 Pbyt1m: 0
Maxdetach: 0 BI01m: 6 JTquota: 1024
Prclm: 2 DI01m: 6 WSdef: 300
Prio: 4 AST1m: 10 WSquo: 350
Queprio: 0 TQE1m: 10 WSextent: 700
CPU: (none) Enqlm: 30 Pgflquo: 12480
Authorized Privileges:
TMPMBX NETMBX
Default Privileges:
TMPMBX NETMBX
Identifier Value Attributes
CLASS_CA101 %X80010032 NORESOURCE NODYNAMIC
CLASS_PY102 %X80010049 NORESOURCE NODYNAMIC
```

**NOTE:** The quotas Pbyt1m and Queprio are not implemented for Version 5.0 and thus are not documented in this manual.

## SHOW/IDENTIFIER

Displays information about the identifier on the current SYS\$OUTPUT device.

### format

**SHOW/IDENTIFIER** [*id-name*]

### parameter

#### *id-name*

Specifies an identifier name. If you omit the identifier name, you must specify /USER or /VALUE.

### qualifiers

#### **/BRIEF**

Specifies a brief listing, in which only the identifier name, value, and attributes are displayed. /BRIEF is the default format for the SHOW/IDENTIFIER command.

#### **/FULL**

Specifies a full listing in which the names of the identifier's holders are displayed along with the identifier's name, value, and attributes.

#### **/USER=user-spec**

Specifies one or more users whose identifiers are to be displayed. **User-spec** can be a user name or UIC. You can use the asterisk wildcard to specify multiple user names or UICs. Full use of the asterisk and percent wildcards is permitted for user names; UICs must be in the form [\*,\*], [n,\*], [\*,n], or [n,n]. A wildcard user name specification (\*) displays identifiers alphabetically by user name; a wildcard UIC specification ([\*,\*]) displays them numerically by UIC.

#### **/VALUE=value-specifier**

Specifies a value in any valid format (see the LIST/IDENTIFIER command).

### example

UAF> SHOW/IDENTIFIER/FULL INVENTORY

The command in this example would produce output similar to the following:

Name	Value	Attributes
INVENTORY	%X80010006	NORESOURCE NODYNAMIC
Holder	Attributes	
ANDERSON	NORESOURCE	NODYNAMIC
BROWN	NORESOURCE	NODYNAMIC
CRAMER	NORESOURCE	NODYNAMIC

## SHOW/PROXY

Displays all authorized proxy access for the specified remote user. The /PROXY qualifier is required.

### format

**SHOW/PROXY** *node::remote-user*

### parameters

#### *node*

Specifies the name of a network node in the network UAF. The asterisk wildcard is permitted in the node specification.

#### *remote-user*

Specifies the user name or UIC of a user on a remote node. The asterisk wildcard is permitted in the remote-user specification.

### example

```
UAF> SHOW/PROXY SAMPLE::[200,100]
```

Default proxies are flagged with an \*

```
SAMPLE::[200,100]
```

```
MARCO *  
PROXY3
```

```
PROXY2
```

The command in this example displays all authorized proxy access for the user on node SAMPLE with a UIC of [200,100]. The default proxy account can be changed from MARCO to PROXY2 or PROXY3 with the MODIFY/PROXY command.

---

## SHOW/RIGHTS

Displays the identifiers held by the specified identifiers or, if /USER is specified, all identifiers held by the specified users.

### format

**SHOW/RIGHTS** *[user-spec]*

### parameter

#### *[user-spec]*

The name of the identifier associated with the user. Specify the identifier in UIC format. If you omit the identifier name, you must specify the /USER qualifier.

**qualifier**

***/USER=user-spec***

Specifies one or more users whose identifiers are to be listed. **User-spec** can be a user name or UIC. You can use the asterisk wildcard to specify multiple user names or UICs. Full use of the asterisk and percent wildcards is permitted for user names; UICs must be in the form [\*,\*], [n,\*], [\*,n], or [n,n]. A wildcard user name specification (\*) or wildcard UIC specification ([\*,\*]) displays all identifiers held by users. The wildcard user name specification displays holders' user names alphabetically; the wildcard UIC specification displays them in the numerical order of their UICs.

**example**

UAF> SHOW/RIGHTS ANDERSON

This command displays all identifiers held by the user ANDERSON. For example:

Name	Value	Attributes
INVENTORY	%X80010006	NORESOURCE NODYNAMIC
PAYROLL	%X80010022	NORESOURCE NODYNAMIC

Note that the following formats of the command produce the same result:

SHOW/RIGHTS/USER=ANDERSON  
 SHOW/RIGHTS/USER=[300,015]



---

## Backup Utility

By duplicating files or volumes of files, the Backup Utility (BACKUP) protects data from loss or corruption.

BACKUP is intended for use primarily by system managers and operators to protect public media. However, anyone can use BACKUP to make personal BACKUP copies and to transport files between VMS systems.

Standalone BACKUP is a version of the Backup Utility that is bootstrapped into main memory instead of running under the control of the VMS operating system. Standalone BACKUP uses a subset of BACKUP qualifiers to perform image and physical BACKUP operations.

### format

**BACKUP** *input-specifier output-specifier*

### parameter

#### *input specifier*

Specifies the input for the BACKUP operation. The input specifier can be a standard VMS file specification, a BACKUP save-set specification, or a device name. If the input specifier is a save-set specification on disk, it must include the input save-set qualifier /SAVE\_SET.

DECnet-VAX node names are allowed only in save-set specifications.

Wildcards are permitted in standard VMS file specifications and in save-set specifications if they are on magnetic tape.

#### *output specifier*

Specifies the output for the BACKUP operation. The output specifier, like the input specifier, can be either a standard VMS file specification, a BACKUP save-set specification, or a device name. If the output specifier is a save set on disk, it must include the output save-set qualifier /SAVE\_SET.

DECnet-VAX node names are allowed only in save-set specifications.

You can use wildcard characters if the output specifier is a Files-11 volume. You cannot use wildcard characters if the output specifier is a BACKUP save set or a volume created by a BACKUP/PHYSICAL or BACKUP/IMAGE operation.

## **BCK-2 Backup Utility**

### **usage summary**

To invoke online BACKUP, enter an appropriate BACKUP command at the DCL prompt.

When you enter a BACKUP command, BACKUP evaluates the input and output specifier and qualifiers to determine the type of operation to perform. BACKUP uses the input specifier to locate the input to the utility and directs output to the output specifier, which can be a file or a save set on disk or a save set on magnetic tape.

After executing the command, BACKUP returns to DCL command level. If you want to halt the execution of a BACKUP command prematurely, press CTRL/Y. If BACKUP is creating a file when you press CTRL/Y, the file is closed immediately and only partially created.

You need the user privilege TMPMBX to send messages to operator terminals when using BACKUP in batch mode. If you are performing a save operation to a volume set of sequential disks, you must have the user privilege PHY\_IO or LOG\_IO to write to a continuation volume. The use of several BACKUP qualifiers also requires privileges; these are noted in the appropriate qualifier descriptions.



## BACKUP Qualifiers

This section provides detailed descriptions of each BACKUP qualifier and includes examples. Each qualifier description identifies the qualifier type.

BACKUP has five types of qualifiers: command qualifiers, input file-selection qualifiers, input save-set qualifiers, output file qualifiers, and output save-set qualifiers, as follows:

- **Command qualifiers** allow you to modify the default action of a BACKUP command. You can place command qualifiers anywhere in the command line. Command qualifiers act upon every file in the input or output specifier.
- **Input file-selection qualifiers** select files from the input specifier. Place them immediately after the input specifier.
- **Input save-set qualifiers** affect the way BACKUP handles an input save set during a restore operation. Place them immediately after the input specifier.
- **Output file qualifiers** change the way output files are restored. Place them immediately after the output specifier.
- **Output save-set qualifiers** affect the way BACKUP processes an output save set during a save operation. Place them immediately after the output specifier.

**NOTE:** You cannot use input and output qualifiers in image operations.

---

## /ASSIST

### Command Qualifier

Allows operator or user intervention if a request to mount a magnetic tape fails during a BACKUP operation.

### format

*/[NO]ASSIST input-specifier output-specifier*

### description

The /ASSIST qualifier causes BACKUP to send messages to operator terminals when a failure occurs during a BACKUP mount request for a magnetic tape. BACKUP sends messages to operator terminals enabled to receive TAPES and CENTRAL messages. (See the description of the REPLY command for information about enabling and disabling operator terminals.) If a failure occurs, the operator can either abort the operation or correct the error condition and allow the operation to continue.

## BCK-4 BACKUP /ASSIST

If no operator terminal is enabled to receive TAPES and CENTRAL messages and to respond to a mount assist request, a message is displayed informing the user of the situation. If a volume is placed in the requested drive, no additional operator response is necessary. Any operator reply to a mount request is written to SYS\$OUTPUT. When BACKUP is run interactively, SYS\$OUTPUT is the user's terminal. When BACKUP is run in batch mode, SYS\$OUTPUT is the batch job log file.

If you specify /NOASSIST, mount messages appear on your terminal and are not sent to the operator.

The default is /ASSIST. Specifying /NOASSIST when BACKUP is run in batch mode has no effect.

### example

```
$ BACKUP/NOASSIST [PAYROLL]*.*;* MTA1:PAYROLL.BCK/LABEL=WKY101
```

This command mounts the volume labeled WKY101 on the MTA1 tape drive and copies all files in the [PAYROLL] directory to a save set named PAYROLL.BCK. The /NOASSIST qualifier directs BACKUP to send mount messages to your terminal rather than to the operator terminal. The WKY101 label indicates that WKY101 is a weekly BACKUP tape in group 1, volume number 01. (If the volume label of the tape is not WKY101, you can direct BACKUP to write the save set to the tape by choosing the OVERWRITE option at the BACKUP> prompt.)

---

## /BACKUP

### Input File-Selection Qualifier

Selects files according to the BACKUP date written in the file header record by the BACKUP/RECORD command.

### format

*input-specifier*/BEFORE=*time*/BACKUP *output-specifier*  
*input-specifier*/SINCE=*time*/BACKUP *output-specifier*

### description

The /BACKUP qualifier is valid with Files-11 Structure Level 2 volumes only and must be used with either the /BEFORE or /SINCE qualifier. You cannot use /BACKUP with the /CREATED, /MODIFIED, or /EXPIRED qualifiers, in an image operation or in a physical operation.

/BACKUP selects files by comparing the date and time recorded in the BACKUP field of the file header record with the date and time specified with the /BEFORE or /SINCE qualifier. The date and time recorded in the file header record is the date and time the file was last saved or copied using the /RECORD command qualifier.

When you use /BACKUP with /BEFORE, files with a BACKUP date prior to the specified date or time are selected. Files with no BACKUP date (/RECORD was not specified when the file was saved or copied) are also selected.

When you use /BACKUP with /SINCE, files with a BACKUP date equal to or later than the specified date or time are selected. Files with no BACKUP date (/RECORD was not specified when the file was saved or copied) are also selected.

### example

```
$ BACKUP/RECORD  
_FROM: [PAYROLL]*.*;*/BEFORE=01-SEP-1990/BACKUP  
_TO: MTA1:SEP01.BCK
```

In this command, the /BACKUP qualifier combined with the /BEFORE qualifier saves all versions of all files in the directory [PAYROLL] that have a BACKUP date written before September 1, 1990. The command qualifier /RECORD writes the date and time of the save operation to the file header record of each saved file.

---

## /BEFORE

### Input File-Selection Qualifier

Selects files dated earlier than the date and time you specify.

### format

*input-specifier*/BEFORE=*time* *output-specifier*

### description

The /BEFORE qualifier selects files by comparing the date and time in the specified field of each file header record with the date and time you specify in the command line. The following list shows the other input file-selection qualifiers you can use with /BEFORE and their functions. Use only one of these other qualifiers at a time in your command line.

/BACKUP	Selects files last saved or copied by BACKUP/RECORD before the date specified. Also selects files with no BACKUP date.
/CREATED	Selects files created before the date specified.
/EXPIRED	Selects files that have expired as of the date specified.
/MODIFIED	Selects files last modified before the date specified. If you specify /BEFORE without another qualifier, /MODIFIED is used by default.

## BCK-6 BACKUP /BEFORE

Specify the date and time as a delta time or as an absolute time using the format [dd-mmm-yyyy[:]][hh:mm:ss.cc]. You can also use one of the following reserved words to specify the date and time:

BACKUP	The BACKUP date of the file written by a previous BACKUP/RECORD operation (available only on Files-11 Structure Level 2 volumes)
TODAY	The current day, month, and year at 00:00:00.0 o'clock
TOMORROW	24 hours after midnight last night
YESTERDAY	24 hours before midnight last night

The /BEFORE qualifier is not valid in incremental restore operations.

### example

```
$ BACKUP [POLICIES]*.*;*/BEFORE=TODAY/EXPIRED DMA1:OLDPOL.BCK/SAVE_SET
```

This command saves all files in the directory [POLICIES] that have expiration dates preceding today's date.

---

## /BLOCK\_SIZE

### Output Save-Set Qualifier

Specifies the output block size in bytes for data records in a BACKUP save set.

### format

*input-specifier output-save-set-speed*/BLOCK\_SIZE=*n*

### description

The minimum block size is 2048 bytes; the maximum block size is 65,024 bytes. The actual block size written is adjusted using the constraints of the BACKUP format. The block size cannot be rounded up over the maximum block size.

If you specify /BLOCK\_SIZE in a magnetic tape save operation, BACKUP ignores any block size defined by the /BLOCK\_SIZE qualifier to the DCL command MOUNT.

If the block size is set to a large value for a save set on magnetic tape, it is possible for the magnetic tape to run off its reel or for a large number of write errors to be logged. If this occurs, avoid using large block sizes. If the problem recurs with the same magnetic tape, avoid using that tape for future BACKUP operations.

The default block size for magnetic tape is 8192 bytes; the default for disk is 32,256 bytes.

## example

```
$ BACKUP/RECORD DRA2:[LEE...]/SINCE=BACKUP MTA0:SAVEWORK.BCK/BLOCK_SIZE=10000
```

This command saves a directory tree on DRA2 to a magnetic tape mounted on drive MTA0. The input file-selection qualifier /SINCE=BACKUP instructs BACKUP to process only those files in the specified directory tree that have been modified since the last BACKUP/RECORD operation. The output save-set qualifier /BLOCK\_SIZE directs BACKUP to assign a block size of 10,240 (BACKUP rounds the specified block size of 10,000 up to the next multiple of 512).

## /BRIEF

### Command Qualifier

Lists the file specification, size, and creation date for each file in the save set. (The size listed is the actual size of the file saved, rather than the number of blocks allocated to the file.) The /BRIEF qualifier is valid only with the /LIST qualifier and is the default format for BACKUP listings. Specify the /FULL qualifier to list information provided by the DCL command DIRECTORY/FULL.

## format

*/LIST/BRIEF save-set-spec*

## example

```
$ BACKUP/LIST/BRIEF DBA2:[SAVE]23MAR90.BCK/SAVE_SET
```

Listing of save set(s)

```
Save set:          23MAR90.BCK
Written by:       MOROCI
UIC:             [000200,000200]
Date:            23-MAR-1990 14:18:16.96
Command:         BACKUP [SAVE] DBA2:[SAVE]23MAR90.BCK/SAVE_SET
Operating system: VAX/VMS version 5.2
BACKUP version:  V5.2
CPU ID register: 08000000
Node name:       _SUZI:
Written on:      _DBA2:
Block size:     32,256
Group size:     10
Buffer count:    3
```

```
[SAVE]LAST.DAT;1      1  18-AUG-1989 14:11
[SAVE]INFO.TXT;4     5   4-FEB-1990 13:12
[SAVE]WORK.DAT;3    33  1-DEC-1989 10:02
```

```
Total of 3 files, 39 blocks
End of save set
```

## BCK-8 BACKUP /BRIEF

This command lists the BACKUP summary information and the file name, size, and creation date for each file in the save set. Note that the input save-set qualifier /SAVE\_SET is required to identify the input specifier as a save set on a Files-11 medium.

---

### /BUFFER\_COUNT

#### Command Qualifier

This qualifier is obsolete. You can still specify the /BUFFER\_COUNT qualifier, although it has no effect. (This ensures that command procedures containing this qualifier will still operate correctly.) Digital recommends that you remove the /BUFFER\_COUNT qualifier from command procedures.

---

### /BY\_OWNER

#### Input File-Selection Qualifier

Selects files for processing according to the user identification code (UIC).

#### format

*input-specifier*/BY\_OWNER[=*uic*] *output-specifier*

#### description

If you specify /BY\_OWNER without a UIC, BACKUP selects all files whose UIC matches that of the current process.

Specify either a numeric UIC as octal numbers or an alphanumeric UIC in the form [g,m]. Wildcards are permitted. Note that the brackets are required.

[g,m]

- g An octal number in the range 0 through 37776 representing the group number or an alphanumeric group name
- m An octal number in the range 0 through 177776 representing the member number or an alphanumeric member name

If you do not specify /BY\_OWNER, BACKUP processes all files specified by the input specifier.

## example

```
$ BACKUP [SNOW...]/BY_OWNER MT$DRIVE:SNOW.BCK/LABEL=TAPE01
```

In this example, BACKUP mounts the tape with the label TAPE01 on drive MT\$DRIVE and saves all files in the directory and subdirectories of [SNOW] with the UIC of the current default process to the save set SNOW.BCK.

## /BY\_OWNER

### Output File Qualifier

Redefines the owner user identification code (UIC) for restored files.

### format

*input-specifier output-specifier/BY\_OWNER[=option]*

### description

The following are available options:

default	Sets the owner UIC to the user's current default UIC. This option is the default if the /BY_OWNER qualifier is not specified, except in image and incremental restore operations, when ORIGINAL is the default option.
ORIGINAL	Retains the owner UIC of the file being restored. This option is the default if the /BY_OWNER qualifier is specified, but no option is selected. This option is also the default for incremental restore operations. To use this option, the UIC must be yours, or you must have the SYSPRV user privilege or be the owner of the output volume.
PARENT	Sets the owner UIC to the owner UIC of the directory to which the file is being restored or copied. To use this option, the parent UIC must be yours, or you must have the SYSPRV user privilege or be the owner of the output volume.
[uic]	Sets the owner UIC to the UIC specified. Use the [g,m] format (as described in the input file-selection qualifier /BY_OWNER). To use this option, the UIC must be yours, or you must have the SYSPRV user privilege or be the owner of the output volume.

In restore operations where the command qualifier /IMAGE or /INCREMENTAL is specified, the default is /BY\_OWNER=ORIGINAL.

## example

```
$ BACKUP DBA2:ACCOUNTS.BCK/SAVE_SET [CLEAVER...]/BY_OWNER=PARENT
```

In this example, the sequential-disk save set ACCOUNTS.BCK is restored to the directory tree [CLEAVER...], assigning each restored file the owner UIC of the [CLEAVER] directory.

## BCK-10 BACKUP /BY\_OWNER

---

### /BY\_OWNER

#### Output Save-Set Qualifier

Specifies the owner user identification code (UIC) of the save set.

#### format

*input-specifier output-save-set-spec*/BY\_OWNER=*uic*

#### description

If the /BY\_OWNER qualifier is omitted, the UIC of the current process is used. To use this qualifier on Files-11 save sets, you need the user privilege SYSPRV, or the UIC must be your own.

Specify either a numeric UIC as octal numbers or an alphanumeric UIC in the form [g,m]. Wildcards are permitted. Note that the brackets are required.

[g,m]

- g** An octal number in the range 0 through 37776 representing the group number or alphanumeric group name
- m** An octal number in the range 0 through 177776 representing the member number or alphanumeric member name

#### example

```
$ BACKUP [CLEAVER...] MFA2:ACCOUNTS.BCK/BY_OWNER=[301,310]/LABEL=TAPE01
```

In this example, BACKUP mounts the tape with the label TAPE01 on drive MFA2. Next, BACKUP saves the directory tree [CLEAVER...] to a save set named ACCOUNTS.BCK. The output save-set qualifier /BY\_OWNER assigns an owner UIC of [301,310] to the save set.

---

### /COMMENT

#### Output Save-Set Qualifier

Places a comment in an output save set. If the comment string is longer than one word or if it contains nonalphanumeric characters, it must be enclosed in quotation marks ( " "). A comment can contain up to 1024 characters.

#### format

*input-specifier output-save-set-spec* /COMMENT=*string*



## example

```
$ BACKUP [REMARKS] DMA1:20JULREM.BCK/SAVE_SET -
_$ /COMMENT="Remote operations for July 20, 1990"
$ BACKUP/LIST DMA1:20JULREM.BCK/SAVE_SET
Listing of save set

Save set:          20JULREM.BCK
Written by:       WALRUS
UIC:              [360,054]
Date:             20-JUL-1990 14:22:06.62
Command:          BACKUP [REMARKS] DMA1:20JULREM.BCK/SAVE_SET/COMMENT=Remote
operations for July 20, 1990
Operating system: VMS Version V5.2
BACKUP version:   V5.2
CPU ID register:  0138084C
Node name:        _ABBEY::
Written on:       _ABBEY$DMA1:
Block size:       32256
Group size:       10
Buffer count:     3

[REMARKS]BAC.RES;1          2  30-JUL-1990 14:13
[REMARKS]COM.LIS;1         1  30-JUL-1990 14:04
[REMARKS]DTOP.DIR;1       1  30-JUL-1990 14:18
.
.
.
Total of 40 files, 535 blocks
End of save set
```

The first BACKUP command saves the directory [REMARKS] to a sequential-disk save set and records a comment. The BACKUP/LIST command displays the contents of the newly created save set. Note that the /SAVE\_SET qualifier is required when creating a save set on disk.

---

## /COMPARE

### Command Qualifier

Compares the save set, device, file, or files specified by the first parameter with the contents of the Files-11 device, file, or files specified by the second parameter and displays an error message if it finds a difference.

### format

```
/COMPARE file-spec file-spec
/COMPARE save-set-spec file-spec
/IMAGE/COMPARE device-spec device-spec
/PHYSICAL/COMPARE device-spec device-spec
```

## BCK-12 BACKUP /COMPARE

### description

In a BACKUP compare operation, the first parameter can be a Files-11 file or a wildcard character representing a set of files, a BACKUP save set on disk or magnetic tape, a tape device, or a disk device. The second parameter must be a Files-11 disk file, a wildcard character representing a set of files or a Files-11 disk device, unless you specify the command qualifier /PHYSICAL. When you specify /PHYSICAL, and the first parameter specifies a disk device, both disks in the compare operation must be mounted with the /FOREIGN qualifier.

BACKUP displays the following error message if it encounters a difference between files it compares:

```
%BACKUP-E-VERIFYERR, verification error for . . .
```

Use the /COMPARE qualifier to compare a save set with original files or to compare files or volumes copied using BACKUP with original files. Because BACKUP processes files by blocks, comparing files not produced by BACKUP is likely to cause mismatch errors in files that are apparently identical.

If you do not specify a version number with the file specification, the default is ;\* (the asterisk wildcard character), which processes all versions of the file.

Both parameters in a compare operation are input specifiers.

If you are comparing two entire Files-11 volumes, use an image compare operation, as follows:

```
$ BACKUP/IMAGE/COMPARE DBA1: DBA2:
```

You cannot use the command qualifier /DELETE or /RECORD in compare operations.

Do not perform compare operations on files that were restored or copied using the output file qualifier /NEW\_VERSION because this qualifier causes version numbers to change.

### example

```
$ BACKUP/COMPARE JAZZ.DAT BLUES.DAT
```

This example compares two Files-11 files. Since no version number is specified, BACKUP compares all versions of each file.

---

**/CONFIRM**

**Input File-Selection Qualifier**

Displays prompts on your terminal for confirmation before processing each file. If you want the file to be processed, enter Y or YES and press RETURN.

**format**

*input-specifier/CONFIRM output-specifier*

**example**

```
$ BACKUP *.LIS/CONFIRM/LOG DLA2:LIST.BCK/SAVE_SET
DISK$DEFAULT:[WONDER]CRE.LIS;1, copy? (Y or N): Y
%BACKUP-S-COPIED, copied DISK$DEFAULT:[WONDER]CRE.LIS;1
DISK$DEFAULT:[WONDER]CRETIME.LIS;1, copy? (Y or N): Y
%BACKUP-S-COPIED, copied DISK$DEFAULT:[WONDER]CRETIME.LIS;1
DISK$DEFAULT:[WONDER]EXC.LIS;1, copy? (Y or N): Y
%BACKUP-S-COPIED, copied DISK$DEFAULT:[WONDER]EXC.LIS;1
DISK$DEFAULT:[WONDER]REB.LIS;1, copy? (Y or N): N
DISK$DEFAULT:[WONDER]SETREB.LIS;1, copy? (Y or N): Y
%BACKUP-S-COPIED, copied DISK$DEFAULT:[WONDER]SETREB.LIS;1
DISK$DEFAULT:[WONDER]VERS.LIS;1, copy? (Y or N): N
.
.
.
$
```

This command locates all files with a file type of LIS and prompts for confirmation before saving each file to LIST.BCK on DLA2. The command qualifier /LOG displays information about each file as it is processed. Note that the output save-set qualifier SAVE\_SET qualifier is required when creating a save set on disk.

---

**/CRC**

**Input Save-Set Qualifier**

Specifies that the software Cyclic Redundancy Check (CRC) is to be performed.

**format**

*input-save-set-spec/[NO]CRC output-specifier*

## BCK-14 BACKUP /CRC

### description

The default is /CRC. To disable CRC checking, specify /NOCRC; note that use of /NOCRC reduces processing time but increases the risk of data loss.

The method for performing CRC checking emulation is approximately 40% faster than the method used before VMS Version 5.2. This improves BACKUP performance on the following processors, which emulate CRC in software:

- MicroVAX II/VAXstation II
- MicroVAX 2000/VAXstation 2000
- MicroVAX 3200/VAXstation 3200
- MicroVAX 3500/VAXstation 3500
- MicroVAX 3600
- VAX 6200-series

### example

```
$ BACKUP MTA2:928SAVE.BCK/NOCRC []
```

This command restores the save set 928SAVE.BCK to the current default directory, indicated by ([]); the input save-set qualifier /NOCRC disables Cyclic Redundancy Checking.

---

## /CRC

### Output Save-Set Qualifier

Specifies whether the software Cyclic Redundancy Check (CRC) is to be computed and stored in the data blocks of the output save set.

### format

*input-specifier output-save-set-spec*[NO]CRC

### description

The default is /CRC. To disable checking, use /NOCRC; note that use of /NOCRC reduces processing time but increases the risk of data loss.

The method for performing CRC checking emulation approximately 40% faster than the method used before VMS Version 5.2. This improves BACKUP performance on the following processors, which emulate CRC in software:

- MicroVAX II/VAXstation II
- MicroVAX 2000/VAXstation 2000
- MicroVAX 3200/VAXstation 3200
- MicroVAX 3500/VAXstation 3500
- MicroVAX 3600
- VAX 6200-series

### example

```
$ BACKUP/RECORD []/SINCE=BACKUP MTA2:928SAVE.BCK/NOCRC
```

This command saves all files in the current default directory that have been created or modified since the last BACKUP/RECORD operation to the save set 928SAVE.BCK; the output save-set qualifier /NOCRC disables Cyclic Redundancy Checking.

## /CREATED

### Input File-Selection Qualifier

Selects files according to the value of the creation date field in each file header record.

### format

*input-specifier*/BEFORE=*time*/CREATED *output-specifier*

*input-specifier*/SINCE=*time*/CREATED *output-specifier*

### description

You must use either the /BEFORE qualifier or the /SINCE qualifier with /CREATED. The date and time you specify to /BEFORE or /SINCE determine which files should be processed.

You cannot use /CREATED with the /BACKUP, /MODIFIED, or /EXPIRED qualifiers.

### example

```
$ BACKUP *.GNC/SINCE=YESTERDAY/CREATED DLA2:[SAVEDIR]/SAVE_SET
```

The command in this example saves all files with a file type of GNC created since yesterday (24 hours before midnight last night).

## BCK-16 BACKUP /DELETE

---

### /DELETE

#### Command Qualifier

Specifies that a BACKUP save or copy operation is to delete the selected input files from the input volume after all files have been processed.

#### format

*/DELETE file-spec save-set-spec*

#### description

The /DELETE qualifier is valid only when used in a BACKUP save or copy operation. You must have sufficient privilege to delete files; if you do not, files protected against deletion are not deleted. If you use the command qualifier /VERIFY with /DELETE, files that fail verification are not deleted.

You cannot use /DELETE with the /RECORD or /COMPARE command qualifiers.

#### example

```
$ BACKUP/DELETE BOP.DAT MTA0:BOP.BCK/LABEL=DANCE
```

In this example, the file BOP.DAT will be deleted after the save set BOP.BCK is successfully created on MTA0.

---

### /DENSITY

#### Output Save-Set Qualifier

Specifies the recording density of the output magnetic tape in bits per inch (bpi). The output save-set qualifier /REWIND is required if you specify /DENSITY.

#### format

*input-specifier output-save-set-spec/DENSITY=*n**

#### description

The value you specify must be supported by your magnetic tape hardware. If you omit this qualifier, the default density is the current density on the output tape drive.

The /DENSITY qualifier is incompatible with the output save-set qualifier /NOREWIND. You must specify the output save-set qualifier /REWIND to initialize the magnetic tape when using the /DENSITY qualifier. When you specify /DENSITY/REWIND, BACKUP rewinds the tape to

the beginning-of-tape. Then BACKUP initializes the tape with the new density, removing access to all data that previously resided on the tape.

### example

```
$ BACKUP *.PAS MTA2:SAVEPAS.BCK/DENSITY=1600/REWIND/LABEL=PASCAL
```

The magnetic tape on drive MTA2 is initialized. All files with a file type of PAS in the current default directory are saved to the save set SAVEPAS.BCK. The /DENSITY qualifier sets the recording density to 1600 bpi.

---

## /EXCLUDE

### Input File-Selection Qualifier

Excludes files that otherwise meet the selection criteria for a save operation. The excluded files are not processed.

### format

*input-specifier/EXCLUDE=(file-spec[...]) output-specifier*

### description

If you specify more than one file, separate the file specifications with commas and enclose the list in parentheses. Do not use a device specification when you define the files to be selected. You can use most standard wildcard characters, but you cannot use wildcard characters denoting latest versions of files (;) or relative versions of files (;n).

Note that BACKUP does not apply temporary file specification defaults within the list. Each file specification independently takes its defaults from the file specification [000000 ... ]\*.\*;\*

If you specify directory files (files with the file type DIR), your command is processed but the directory files are not excluded (they are processed). BACKUP uses directory files to facilitate incremental restore operations.

### example

```
$ BACKUP  
_FROM: DRA2:[CONTRACTS]/BEFORE=TODAY/EXCLUDE=(*.OBJ,*.MAI)  
_TO: MFA0:CONTRACT.BCK/LABEL=DLY102
```

All files in the directory [CONTRACTS] that have a modification date prior to today (the current day, month, and year at 00:00:00.0 o'clock) are saved to the save set CONTRACT.BCK on drive MFA0, except for those with a file type of OBJ or MAI.

## /EXPIRED

### Input File-Selection Qualifier

Selects files according to the value of the expiration date field in each file header record.

### format

*input-specifier*/BEFORE=*time* /EXPIRED *output-specifier*

*input-specifier*/SINCE=*time* /EXPIRED *output-specifier*

### description

You must use the input file-selection qualifier /BEFORE or /SINCE with /EXPIRED. The date and time you specify to /BEFORE or /SINCE determines which files are processed.

You cannot use /EXPIRED with the input file-selection qualifiers /BACKUP, /MODIFIED, or /CREATED.

### example

```
$ BACKUP [CONTRACTS]/BEFORE=TOMORROW/EXPIRED MTA1:30DEC.BCK/LABEL=WK04
```

This command saves all files in the directory [CONTRACTS] that have an expiration date prior to tomorrow (24 hours after midnight last night) to a save set named 30DEC.BCK.

---

## /FAST

### Command Qualifier

Processes the input specifier using a fast file scan to reduce processing time. The input specifier must be a Files-11 disk.

### format

*/FAST input-specifier output-specifier*

### description

The fast file scan reads the index file on the Files-11 disk specified by the input specifier and creates a table of files that match the qualifiers you specified.

To perform a fast file scan, you need write access to the INDEXF.SYS file on the input medium, or the input medium must be write-locked. This requirement is necessary because BACKUP opens the index file to synchronize with the file system, whether or not any update is made.



A fast file scan is most useful when the input specifier includes most of the files on the volume, and file-selection qualifiers (such as those that pertain to date or owner) specify a relatively small set of the files named. Because image operations implicitly use the fast file scan, the /FAST qualifier is ignored if used with the command qualifier /IMAGE.

You cannot use /FAST in restore operations.

### example

```
$ BACKUP/FAST
  _FROM: DBA1:[*...]/MODIFIED/SINCE=TODAY
  _TO: MTA0:13NOVBAK.BCK,MTA1:/LABEL=WK201
```

In this example, all files on the disk DBA1 that have been modified today are saved to a multireel tape save set named 13NOVBAK.BCK. The /FAST qualifier is used to reduce processing time.

## /FULL

### Command Qualifier

Lists the file information produced by the command qualifier /LIST in the format provided by the DCL command DIRECTORY/FULL.

### format

*/LIST/FULL input-specifier [output-specifier]*

### description

The /FULL qualifier is valid only with the command qualifier /LIST.

If you do not specify /FULL with /LIST, the /LIST qualifier uses the default command qualifier /BRIEF and lists only the file specification, size, and creation date of each file. When you specify /FULL, the list includes more information from the file header records, such as the BACKUP date, date of last modification, number of blocks allocated to the file, file protection and organization, and record attributes.

### example

```
$ BACKUP/LIST/FULL MTA1:ROCK.BCK
Listing of save set(s)
```

**BCK-20    BACKUP  
/FULL**

Save set:            ROCK.BCK  
Written by:         RINGO  
UIC:                [000200,000300]  
Date:               20-AUG-1990 15:39:38.89  
Command:            BACKUP [.STONES] MTA0:ROCK.BCK/LABEL=BACKUP  
Operating system:   VAX/VMS version 5.2  
BACKUP version:     V5.2  
CPU ID register:    08000000  
Node name:          \_SUZI::  
Written on:         \_MTA0:  
Block size:         8192  
Group size:         10  
Buffer count:       30

[RINGO.STONES]GRAPHITE.DAT;1

Size:               1/1                    Created: 18-AUG-1990 14:10  
Owner: [000200,000200]   Revised: 18-AUG-1990 14:10 (2)  
File ID: (91,7,1)       Expires: [None specified]  
Backup: [No backup done]

File protection:    System:RWED, Owner:RWED, Group:RE, World:  
File organization:   Sequential  
File attributes:    Allocation = 1, Extend = 0  
                      Global Buffer Count = 0  
Record format:      Variable length, maximum 255 bytes  
Record attributes:   Carriage return

[RINGO.STONES]GRANITE.DAT;1

Size:               1/1                    Created: 18-AUG-1990 14:11  
Owner: [000200,000200]   Revised: 18-AUG-1990 14:11 (2)  
File ID: (92,9,1)       Expires: [None specified]  
Backup: [No backup done]

File protection:    System:RWED, Owner:RWED, Group:RE, World:  
File organization:   Sequential  
File attributes:    Allocation = 1, Extend = 0  
                      Global Buffer Count = 0  
Record format:      Variable length, maximum 255 bytes  
Record attributes:   Carriage return

.  
.  
.  
Total of 4 files, 16 blocks

End of save set

**The command in this example lists the files in save set MTA1:ROCK.BCK  
in full format.**

## **/GROUP\_SIZE**

### **Output Save-Set Qualifier**

Defines the number of blocks BACKUP places in each redundancy group.

### **format**

*input-specifier output-save-set-spec* **GROUP\_SIZE=n**

### **description**

BACKUP writes redundant information to output save sets to protect against data loss. Using the redundant information, BACKUP can correct one “uncorrectable” read error in each redundancy group. The /GROUP\_SIZE qualifier specifies the number of output blocks written to each redundancy group. The value of *n* can be from 0 through 100. The default value is 10.

If you define a value of 0 for /GROUP\_SIZE, no redundancy groups are created for the save set.

### **example**

```
$ BACKUP/RECORD DBA1:[*...]/SINCE=BACKUP TAPE:SAVEWORK.BCK/GROUP_SIZE=5
```

This BACKUP command saves all files in the current default directory tree that have been modified since the last BACKUP/RECORD operation; the /GROUP\_SIZE defines the redundancy group size as 5 blocks.

---

## **/IGNORE=option**

### **Command Qualifier**

Specifies that a BACKUP save or copy operation is to override restrictions placed on files or not perform tape label processing checks.

### **format**

*/IGNORE=option input-specifier output-specifier*

### **description**

The /IGNORE=option qualifier has three options:

## BCK-22 BACKUP /IGNORE=option

INTERLOCK	Processes files that otherwise could not be processed because of file access conflicts. This option can be used to save or copy files currently open for writing. Note that no synchronization is made with the process writing the file, so the file data that is copied might be inconsistent with the input file, depending on the circumstances (for example, if another user is editing the file, the contents might change). When a file open for writing is processed, BACKUP issues the message: %BACKUP-W-ACCONFLICT, 'filename' is open for write by another user.  The INTERLOCK option is especially useful if you have files that are open so much of the time that they might not otherwise be saved. The use of this option requires the user privilege SYSPRV, a system UIC, or ownership of the volume.
LABEL_PROCESSING	Saves the contents of files to the specified magnetic tape volume regardless of the information contained in the volume header record. BACKUP will not verify the volume label or expiration date before writing information to the tape volume.
NOBACKUP	Causes BACKUP to save both the file header record and the contents of files marked with the NOBACKUP flag by the /NOBACKUP qualifier of the DCL command SET FILE. If you do not specify this option, BACKUP saves only the file header record of files marked with the NOBACKUP flag.

### example

```
$ BACKUP/IGNORE=INTERLOCK  
_FROM: DUA0:[SUSAN...]  
_TO: MTA0:SONGBIRD.BCK/LABEL=TAPE01
```

This command saves an entire directory tree and the files in all subdirectories, including any files that are open.

```
$ BACKUP/IGNORE=LABEL_PROCESSING *.*;* MFA1:MYFILES.BCK/REWIND
```

This command rewinds the tape in drive MFA1 to the beginning-of-tape marker, initializes the tape, and creates a save set containing all files in the user's current directory. The command qualifier /IGNORE=LABEL\_PROCESSING specifies that no tape label processing checks are done before BACKUP initializes the tape. When the tape is initialized, access to data that previously resided on the tape is lost.

---

## /IMAGE

### Command Qualifier

Directs BACKUP to process an entire volume or volume set.

### format

*/IMAGE input-specifier output-specifier*

### description

To use the /IMAGE qualifier, you need write access to the volume index file (INDEXF.SYS) and the bit map file (BITMAP.SYS), or the input medium must be write-locked. BACKUP opens the index file to synchronize with the file system (no update is made). Finally, you must have read access to all files on the input medium.

**NOTE:** The input and output devices in an image operation must be different except in an image save operation when the output device is a Files-11 disk save set.

If the output volume is a disk, all files on the output volume are stored contiguously. Contiguous storage of files eliminates disk fragmentation and creates contiguous free blocks of disk space.

Because all files on the input volume are processed, you cannot use input file-selection qualifiers in image copy or save operations. You can, however, restore files and directories selectively from an image save set.

When performing image operations on volume sets (more than one volume), the number of volumes specified by the output specifier must be equal to the number of volumes in the input volume set.

In an image save or copy operation, BACKUP attempts to save or copy all files on the input disk volume including files marked for deletion and lost files (files without a directory entry). However, there are two types of files that a BACKUP image operation does not save or copy by default. These are files that are flagged as NOBACKUP by the DCL command SET FILE/NOBACKUP and files that are open for write access by another user at the time of the image save operation. If you want these files to be included, specify the command qualifier /IGNORE in the BACKUP command line. The command qualifier /IGNORE=NOBACKUP directs BACKUP to save or copy files flagged as NOBACKUP. The command qualifier /IGNORE=INTERLOCK directs BACKUP to save or copy files open for write access by another user.

## BCK-24 BACKUP /IMAGE

An image restore or copy operation initializes the output volume or volume set. The initialization data comes from the save-volume summary record of the input volume unless the command qualifier /NOINITIALIZE is specified. Specifying /NOINITIALIZE directs BACKUP to initialize the output volume using volume initialization data that already exists on the output volume.

In image restore and copy operations, every file is restored or copied. The output volume must be mounted using the /FOREIGN qualifier. The new volume is a functionally equivalent copy of the input volume; however, file placement will change. Files are stored contiguously on the output volume.

You cannot change the structure level of the output volume in an image restore or copy operation.

### example

```
$ MOUNT/FOREIGN DMA1:
%MOUNT-I-MOUNTED, mounted on NODE$DMA1:
$ BACKUP/IMAGE/LOG DLA2: DMA1:
%BACKUP-S-CREATED, created DMA1:[000000]000000.DIR;1
%BACKUP-S-CREATED, created DMA1:[000000]BACKUP.SYS;1
%BACKUP-S-CREATED, created DMA1:[000000]CONTIN.SYS;1
%BACKUP-S-CREATED, created DMA1:[000000]CORIMG.SYS;1
%BACKUP-S-CREATED, created DMA1:[000000]ELLA.DIR;1
%BACKUP-S-CREATED, created DMA1:[ELLA]SCAT.DAT;1
%BACKUP-S-CREATED, created DMA1:[000000]JOE.DIR;1
%BACKUP-S-CREATED, created DMA1:[JOE]STRINGS.DAT;1
%BACKUP-S-CREATED, created DMA1:[000000]OSCAR.DIR;1
%BACKUP-S-CREATED, created DMA1:[OSCAR]KEYS.DAT;1
%BACKUP-S-CREATED, created DMA1:[000000]VOLSET.SYS;1
.
.
.
$
```

The MOUNT command prepares the target disk for the image copy operation. The command qualifier /LOG directs BACKUP to display information about each file copied on your terminal. The BACKUP command initializes DMA1 and copies the disk volume DLA2 to DMA1. All files on DMA1 are stored contiguously.

---

## /INCREMENTAL

### Command Qualifier

Allows you to restore an incremental save set.

### format

**/INCREMENTAL** *save-set-spec disk-device-name*

### description

Use **/INCREMENTAL** only in restore operations that restore incremental save sets. When you use **/INCREMENTAL**, the output specifier must specify a device only; file specifications are not allowed. Also, input save-set qualifiers are not allowed in incremental restore operations.

You can create incremental save sets with the command qualifier **/RECORD** and the file-selection qualifier **/SINCE=BACKUP** or **/SINCE=date**. Most sites perform daily incremental save operations to keep copies of files created or modified that day, and periodic full backups to keep a copy of all files on the disk volume. (Digital recommends that you use the command qualifier **/IMAGE** to perform full backups.)

If a disk volume is lost, corrupted, or destroyed, its contents can be recreated by performing the following tasks:

1. Restore the latest full backup using the command qualifiers **/IMAGE** and **/RECORD**.
2. Restore any incremental save sets since the last full backup, in reverse chronological order, using the **/INCREMENTAL** qualifier.

After you restore the save sets in this order, the output disk volume contains the same files it contained when the most recent incremental save operation was performed.

When the **/INCREMENTAL** qualifier is used, the **/BY\_OWNER=ORIGINAL** qualifier is assumed; therefore, you do not need to specify **/BY\_OWNER** unless you want to change the original UICs. The **/INCREMENTAL** qualifier can be used only on Files-11 Structure Level 2 volumes.

### example

If you have been performing a combination of full backups and incremental save operations on a public volume, and the public volume is lost, corrupted, or destroyed, use a procedure like the following to create a new copy of the public volume. First, restore the volume from the latest full backup with an image restore operation. Note that the **/RECORD** qualifier is necessary to perform the operation correctly.

## BCK-26 BACKUP /INCREMENTAL

```
$ MOUNT/FOREIGN DRA0:  
%MOUNT-I-MOUNTED, mounted on _DRA0:  
$ BACKUP/IMAGE/RECORD MTA0:FULLJUN90,MTA1 DRA0:  
%BACKUP-I-RESUME, resuming operation on volume 2  
%BACKUP-I-RESUME, resuming operation on volume 3  
%BACKUP-I-RESUME, resuming operation on volume 4  
.  
.  
.  
$ DISMOUNT/NOUNLOAD DRA0:
```

Next, mount the disk as a file-structured volume and restore the incremental save sets in reverse chronological order. Finally, restore the weekly incremental save sets. The /INCREMENTAL qualifier must be used where shown in the following example to obtain the correct results.

```
$ MOUNT DRA0: PUBLIC  
%MOUNT-I-MOUNTED, PUBLIC mounted on _DRA0:  
$ BACKUP/INCREMENTAL MTA0:INCD17JUN DRA0:  
$ BACKUP/INCREMENTAL MTA0:INCD16JUN DRA0:  
$ BACKUP/INCREMENTAL MTA0:INCD15JUN DRA0:  
$ BACKUP/INCREMENTAL MTA0:INCW14JUN DRA0:  
$ BACKUP/INCREMENTAL MTA0:INCW7JUN DRA0:
```

Note that BACKUP restores the volume correctly regardless of the order in which the incremental save sets are applied; using reverse chronological order is most efficient.

---

## /INITIALIZE

### Command Qualifier

Initializes an output disk volume, making its entire previous contents unavailable.

### format

*/[NO]INITIALIZE input-specifier output-specifier*

### description

The /[NO]INITIALIZE qualifier is valid only when used with the command qualifier /IMAGE during restore or copy operations or when saving files to a sequential-disk save set.

When used with the command qualifier /IMAGE in a restore or copy operation, the /INITIALIZE qualifier directs BACKUP to initialize the output volume using volume initialization data from the save-volume summary record on the input volume.



The /NOINITIALIZE qualifier directs BACKUP to reinitialize the output volume using the existing initialization data on that volume; the output volume must have been previously initialized as a Files-11 volume. When the output volume is initialized, existing data on the volume is lost. The structure level of the output volume must be the same as the structure level of the save set being restored.

For image restore and copy operations on Files-11 volumes, the default is /INITIALIZE.

If you use the /INITIALIZE qualifier when creating sequential-disk save sets, BACKUP initializes the first output volume in the sequential-disk save set, as well as subsequent volumes. By default, BACKUP does not initialize the first volume of a sequential-disk save set but does initialize subsequent volumes of a multivolume sequential-disk save set.

### example

```
$ BACKUP/IMAGE/NOINITIALIZE DBA0: DBA2:
```

This command causes the output volume DBA2 to be reinitialized using the volume initialization data that exists on DBA2. The contents of DBA0 are then copied to DBA2.

---

## /INTERCHANGE

### Command Qualifier

Directs BACKUP to process files in a manner suitable for data interchange (software distribution) by excluding information that would prevent other utilities or sites from reading the BACKUP save set.

### format

```
/INTERCHANGE input-specifier output-specifier
```

### description

The effects of the /INTERCHANGE qualifier are as follows:

- Directories not selected as files are not copied.
- Access control lists are not copied.
- Block size on magnetic tape is limited to 8,192 bytes.
- Normal error recovery is used to write magnetic tapes so that there are no bad records on the resulting magnetic tape.

## BCK-28 BACKUP /INTERCHANGE

### example

```
$ BACKUP/RECORD/INTERCHANGE [ACCOUNTS]/SINCE=BACKUP MFA0:SAVACC.BCK
```

The command in this example saves all files in the directory [ACCOUNTS] that have been modified since the last BACKUP/RECORD operation. The /INTERCHANGE qualifier ensures that the processed files are suitable for data interchange.

---

## /JOURNAL

### Command Qualifier

Specifies that a BACKUP save operation is to create a BACKUP journal file or append information to a BACKUP journal file. Lists the contents of a BACKUP journal file when combined with the command qualifier /LIST.

### format

*/JOURNAL[=file-spec] input-specifier output-specifier*

*/JOURNAL[=file-spec]/LIST[=file-spec]*

### description

A BACKUP journal file contains records of BACKUP save operations and the file specifications of saved files. Use the command qualifier /JOURNAL[=file-spec] in a BACKUP save operation to create a journal file.

If you do not include a file specification with the command qualifier /JOURNAL, the name of the BACKUP journal file defaults to SYS\$DISK:[]BACKUP.BJL. You can specify another file name, however. (The file specification of a journal file cannot include a node name; the default file type for a journal file is BJL.) If the specified journal file does not exist, it is created; if the journal file does exist, the new journal information is appended to the existing journal file.

Start a new version of a journal file by creating a zero-length file using the DCL command CREATE or a text editor.

To list the contents of a BACKUP journal file, use the /JOURNAL=[file-spec] qualifier with the /LIST qualifier, but do not specify an input or output specifier. By default, the list is displayed on SYS\$OUTPUT, but it is written to an output file if you specify a file with /LIST.

When listing a journal file, you can use the file-selection qualifiers /BEFORE, /SINCE, and /EXCLUDE to search for specific files. (In this context, the /BEFORE and /SINCE qualifiers refer to the time when the save set was created, not the time when the files in the save set were created.) Also, by specifying a file in a multivolume save set, you can

search the journal file to find which volume the file is in. You can then mount that volume and restore the file.

Journal files are not created for physical save operations (save operations performed with the command qualifier /PHYSICAL).

### example

```
$ BACKUP/JOURNAL=LAR.BJL [LARRY]*.*;* MFA0:YET.BCK
```

This command saves all versions of all files in the directory [LARRY] to the save set YET.BCK on MFA0. The /JOURNAL qualifier creates a record of the saved files in a journal file named LAR.BJL in the current default directory.

---

## /LABEL

### Output Save-Set Qualifier

Specifies the volume labels for the magnetic tapes to which the save set is written.

### format

*input-specifier output-save-set-spec*/LABEL=(string[,...])

### description

Use the label qualifier to specify the one- to six-character volume labels for the magnetic tapes to which the save set is written.

You can specify either a single label or a list of labels with the /LABEL qualifier. If you do not specify the /LABEL qualifier, BACKUP uses the first six characters of the save-set name as the volume label of the first tape. If you specify a label that is longer than six characters, BACKUP truncates the label to six characters.

If the save set continues to another tape, and you did not specify a volume label for the tape, BACKUP uses the first four characters of the previous tape's volume label followed by the volume number of the tape. For example, if the first tape in a save set is labeled AAAABB, the second tape in a save set is labeled AAAA02, and the third tape is labeled AAAA03.

Before writing a save set to magnetic tape, BACKUP compares the label specified in the command line to the volume label of the tape. (If the tape has no volume label and you specified the output save-set qualifier /REWIND, BACKUP writes the label you specified to the volume header record of the tape.) If the volume label is less than six characters long, BACKUP pads the volume label with the blank character to six characters.

## BCK-30 BACKUP /LABEL

The first four characters of the volume label must either exactly match the first four characters of the label specified in the BACKUP command line, or the first four characters of the volume label must end with one or more underscore characters. If the first four characters of the volume label end with one or more underscore characters, and the label specified in the command line matches the part of the volume label that appears before the underscore characters, BACKUP accepts the match. (For example, the volume label ABN\_ matches the command line label ABN but does not match the command line label ABNE.) If either the fifth or sixth characters of the volume label is a number between zero and nine, BACKUP does not compare these characters with corresponding characters in the label specified in the BACKUP command line. Otherwise, the fifth and sixth characters in the volume label must match the corresponding characters in the label specified in the BACKUP command line exactly.

The following table illustrates volume labels that match labels specified in the BACKUP command line:

Label Specified in Command Line	Matching Volume Labels
MAR	MAR, MAR_, MAR_nn
MAR_	MAR_, MAR_nn
MARK	MARK, MARKnn
MARKER	MARKER, MARKnn

If the label you specify matches the tape's volume label, the BACKUP save operation proceeds. If you specify more than one label with the /LABEL qualifier, the BACKUP save operation succeeds if any of the labels you specify match the tape's volume label. For example, if the tape's volume label is MA1686, the save operation will succeed if you specify the the following list of labels with the /LABEL qualifier:

```
/LABEL=(MA1684,MA1685,MA1686)
```

If the label you specified does not match the tape's volume label, BACKUP displays the following messages and prompt on your terminal if you specified the command qualifier /NOASSIST or on the operator terminal if you did not specify /NOASSIST:

```
%BACKUP-W-MOUNTERR, volume 'number' on 'device' was not mounted because  
its label does not match the one requested  
Specify option (QUIT, NEW tape or OVERWRITE tape)  
BACKUP>
```

Specify QUIT to abort the BACKUP operation and unload the magnetic tape. Specify NEW to direct BACKUP to prompt for a new tape. Specify OVERWRITE to direct BACKUP to ignore the label mismatch, mount the tape, initialize the tape if you specified the output save-set qualifier /REWIND, and write the save set to the tape.

You can specify the command qualifier /IGNORE=LABEL\_PROCESSING to prevent BACKUP from verifying the volume label of the tape.

## example

```
$ BACKUP [PAYROLL] MTA0:30NOV.BCK/LABEL=PAY
```

This command causes BACKUP to check the volume label of the tape mounted on drive MTA0. If the volume label is PAY, BACKUP saves the directory [PAYROLL] to a save set named 30NOV.BCK.

---

## /LIST

### Command Qualifier

Lists information about a BACKUP save set and about the files in a save set. The list can be displayed on your terminal or written to a file.

### format

*/LIST[=*file-spec*] *save-set-spec**

### description

Use the /LIST qualifier by itself or in conjunction with any other operation (save, restore, copy, compare, or journal). If /LIST is specified by itself (not with a save, restore, copy, compare or journal operation), the input specifier must refer to a save set, and the output specifier must be omitted.

Before you can list the contents of a save set, the media containing the save set must be inserted into an appropriate drive. If the save set is stored on a disk, the disk must be mounted as a Files-11 volume or as a foreign volume. BACKUP mounts magnetic tapes automatically as part of the list operation.

By default, the list information is displayed on your terminal; however, you can specify a file to which to write the list information.

When you use the /LIST qualifier with standalone BACKUP and you direct output to a file (/LIST=*file-spec*), the file specification must refer to either a terminal or a printer.

## BCK-32 BACKUP /LIST

You can use either the command qualifier **/BRIEF** or **/FULL** with the **/LIST** qualifier. The **/BRIEF** qualifier directs BACKUP to list each file's size in blocks and its creation date. The **/FULL** qualifier directs BACKUP to list additional information about each file in the same format as the information provided by the DCL command **DIRECTORY/FULL**. The default is **/BRIEF**.

Do not use the command qualifier **/LOG** with **/LIST** when the output for **/LIST** is directed to the terminal; if you do, you will receive confusing output.

### example

```
$ BACKUP/LIST DBA2:[SAVE]23MAR90.BCK/SAVE_SET
```

Listing of save set(s)

```
Save set:          23MAR90.BCK
Written by:       MOROCI
UIC:              [000200,000200]
Date:             23-MAR-1990 14:18:16.96
Command:          BACKUP [SAVE] DBA2:[SAVE]23MAR90.BCK/SAVE_SET
Operating system: VAX/VMS version 5.2
BACKUP version:   V5.2
CPU ID register:  08000000
Node name:        _SUZI::
Written on:       _DBA2:
Block size:       32,256
Group size:       10
Buffer count:     3
```

```
[SAVE]LAST.DAT;1      1  18-AUG-1989 14:11
[SAVE]INFO.TXT;4     5   4-FEB-1990 13:12
[SAVE]WORK.DAT;3    33  1-DEC-1989 10:02
```

```
Total of 3 files, 39 blocks
End of save set
```

This command lists the BACKUP summary information and the file name, size, and creation date for each file in the save set. Note that the **/SAVE\_SET** qualifier is required to identify the input specifier as a save set on a Files-11 disk.

---

## /LOG

### Command Qualifier

Determines whether the file specification of each file processed is displayed on **SYSD\$OUTPUT** during the operation. The default is **/NOLOG**.

## format

**/[NO]LOG** *input-specifier output-specifier*

## example

```
$ BACKUP/LOG [SAVE]23MAR90.BCK/SAVE SET DBA2:[PLI.WORK]
%BACKUP-S-CREATED, created DBA2:[PLI.WORK]ANOTHER.DAT;1
%BACKUP-S-CREATED, created DBA2:[PLI.WORK]LAST.DAT;1
%BACKUP-S-CREATED, created DBA2:[PLI.WORK]THAT.DAT;1
%BACKUP-S-CREATED, created DBA2:[PLI.WORK]THIS.DAT;2
.
.
.
```

In this example, the file specifications of the files restored to the directory named [PLI.WORK] on DBA2 are logged to SYS\$OUTPUT.

## /MODIFIED

### Input File-Selection Qualifier

Selects files according to the value of the modified date field (the date the file was last modified) in each file header record.

## format

*input-specifier***/BEFORE=***time* **/MODIFIED** *output-specifier*  
*input-specifier***/SINCE=***time* **/MODIFIED** *output-specifier*

## description

You must use the /MODIFIED qualifier with either the input file-selection qualifier /BEFORE or /SINCE. The date and time you specify with /BEFORE or /SINCE determines which files are processed.

You cannot use /MODIFIED with the input file-selection qualifiers /BACKUP, /CREATED, or /EXPIRED.

## example

```
$ BACKUP [SUNDANCE...] /BEFORE=TODAY /MODIFIED MFA1:MOD.BCK
```

This command saves all files in the directory tree [SUNDANCE] whose modification dates precede today (00:00:00.0 o'clock of the current day, month, and year).

## **/NEW\_VERSION**

### **Output File Qualifier**

Creates a new version of a file if a file with an identical specification already exists at the location to which the file is being restored or copied.

### **format**

*input-specifier output-specifier***/NEW\_VERSION**

### **description**

If BACKUP attempts to copy or restore a file when a file with an identical directory name, file name, type, and equal or higher version number already exists, a new file is created with the same name and type and a version number one higher than the highest existing version.

If you do not use **/NEW\_VERSION**, **/REPLACE**, or **/OVERLAY**, and the version number of the file being restored is identical or less than the version number of the existing file, BACKUP reports an error in copying or restoring the file.

Note that when copying or restoring files using the **/NEW\_VERSION** qualifier, files are processed in decreasing version number order and are created in ascending order. The result is that the version numbers are inverted.

Because this qualifier causes version numbers to change, using it with the **/COMPARE** or **/VERIFY** qualifiers will cause unpredictable results. Digital recommends that you do not use the **/NEW\_VERSION** qualifier with the **/COMPARE** or **/VERIFY** qualifiers.

### **example**

```
$ BACKUP MTA1:NOV30REC.BCK/SELECT=*.DAT [RECORDS...]/NEW_VERSION
```

This example restores all files with the file type of DAT from the magnetic-tape save set NOV30REC.BCK to the directory [RECORDS]. The **/NEW\_VERSION** qualifier instructs BACKUP to restore each file with the file type DAT regardless of whether a file with the same file specification already exists.



---

## /OVERLAY

### Output File Qualifier

Writes the input file over a file with an identical specification at the output location.

### format

*input-specifier output-specifier***OVERLAY**

### description

If BACKUP attempts to copy or restore a file when a file with an identical directory name, file name, type, and version number already exists, the new version of the file is written over the existing version. The file identification of the new version is the same as the file identification of the file that is overwritten.

The physical location of the file on disk does not change. If /OVERLAY is specified, and the new file is larger than the one already present, BACKUP allocates more blocks on the disk and extends the file.

When you do not use /OVERLAY, /REPLACE, or /NEW\_VERSION, and the version number of the file being restored is identical to the version number of the existing file, BACKUP reports an error in copying or restoring the file.

### example

```
$ BACKUP DRA1:MAR30SAV.BCK/SAVE_SET [RECORDS...]/OVERLAY
```

The sequential-disk save set MAR30SAV.BCK is restored to the directory tree [RECORDS...]. If a file from the save set has a specification that is identical to a file that already exists in [RECORDS...], the /OVERLAY qualifier directs BACKUP to write over the existing version.

---

## /OWNER\_UIC

### Input File-Selection Qualifier

Selects files for processing according to the specified user identification code (UIC).

The /OWNER\_UIC qualifier has been superseded by /BY\_OWNER. Digital recommends that you substitute /BY\_OWNER for /OWNER\_UIC in command procedures and operator instructions. See the description of /BY\_OWNER for more information.

## **/OWNER\_UIC**

### **Output File Qualifier**

Redefines the owner user identification code (UIC) for restored files.

The **/OWNER\_UIC** qualifier has been superseded by **/BY\_OWNER**. Digital recommends that you substitute **/BY\_OWNER** for **/OWNER\_UIC** in command procedures and operator instructions. See the description of **/BY\_OWNER** for more information.

---

## **/OWNER\_UIC**

### **Output Save-Set Qualifier**

Specifies the owner user identification code (UIC) of the save set. The **/OWNER\_UIC** qualifier has been superseded by **/BY\_OWNER**. Digital recommends that you substitute **/BY\_OWNER** for **/OWNER\_UIC** in command procedures and operator instructions. See the description of **/BY\_OWNER** for more information.

---

## **/PHYSICAL**

### **Command Qualifier**

Specifies that a **BACKUP** operation is to ignore any file structure on the input volume and to process the volume in terms of logical blocks.

### **format**

**/PHYSICAL** *input-specifier output-specifier*

### **description**

In a physical operation, **BACKUP** saves, restores, copies, or compares the entire volume in terms of logical blocks.

The input and output specifiers for physical volumes must be device names, and they cannot be the same device. Also, the following qualifiers are ignored if specified with **/PHYSICAL: /DELETE, /IMAGE, /INCREMENTAL, /JOURNAL, and /RECORD**.

For physical copy operations between disks, the output disk must be the same type of device as the input disk; for example, a **BACKUP/PHYSICAL** operation cannot be performed between an **RP05** input disk and an **RP06** output disk. The output disk must not have a bad block in any location that corresponds to a good block on the input disk. (This restriction does not apply to **RA-series** disks.)

For physical save operations between disks, the output disk must be the same type of disk as the input disk or a larger-capacity disk. The output disk must not have a bad block in any location that corresponds to a good block on the input disk. (This restriction does not apply to RA-series disks.)

For physical restore operations between disks, the output disk must be the same type of device as the disk from which the save set was created. The output disk must not have a bad block in any location that corresponds to a good block on the disk from which the save set was created. (This restriction does not apply to RA-series disks.)

An output disk of a physical operation must be mounted using the DCL command MOUNT/FOREIGN. An input disk of a physical operation must either be mounted using the DCL command MOUNT/FOREIGN, or the user must have the user privilege LOG\_IO or PHY\_IO.

You can perform physical save and restore operations using magnetic tapes. BACKUP mounts magnetic tapes automatically as foreign devices.

A save set written using the /PHYSICAL qualifier can only be read as a physical save set; conversely, a file-structured save set can only be read with file-structured restore or compare operations.

**NOTE:** BACKUP/PHYSICAL does not copy the first track (track 0) of RX01 and RX02 diskettes; Digital does not support track 0.

## example

```
$ MOUNT/FOREIGN DYAO:  
$ MOUNT/FOREIGN DYAI:  
$ BACKUP/PHYSICAL DYAO: DYAI:
```

This example mounts RX02 diskettes in DYAO and DYAI as foreign devices and copies the contents of the diskette mounted in DYAO to the diskette mounted in DYAI.

---

## /PROTECTION

### Output Save-Set Qualifier

When you create a save set on disk, this qualifier defines the protection to be applied to an output save set. When you create a save set on magnetic tape, this qualifier defines the protection to be applied to the magnetic tape volume. (All save sets created subsequently on the tape will receive this same protection until the tape is initialized.)

## BCK-38 BACKUP /PROTECTION

### format

*input-specifier output-save-set-spec***PROTECTION**[=(code)]

### description

Because the file system treats a BACKUP save set as a single file, it is crucial that you protect save sets adequately. If you do not specify adequate protection, anyone who has access to a save set can access any file in the save set.

The protection code indicates the type of access (read, write, execute, and delete) available to the four categories of users (system, owner, group, and world).

If the save set is written to either a Files-11 disk or a sequential disk and /PROTECTION is not specified, BACKUP applies the process default protection to the save set. If /PROTECTION is specified, any protection categories not specified default to your default process protection.

Protection information is written to the volume header record of a magnetic tape, and applies to all save sets stored on the tape. Therefore, you must specify the output save-set qualifier /REWIND in order to specify the /PROTECTION qualifier for a magnetic tape. (If you do not specify /REWIND with /PROTECTION, the protection information, if any, in the volume header record is not changed.) If the save set is written to magnetic tape and /PROTECTION is not specified, BACKUP applies **no** protection to the tape. If you specify /PROTECTION, any protection categories that you do not specify default to your default process protection.

In order to initialize a magnetic tape volume that was previously initialized with the /PROTECTION qualifier, you must own the volume (your UIC matches the UIC of the volume) or have the VOLPRO privilege.

### example

```
$ BACKUP
  FROM: [CLEAVER...]
  TO: MFA2:ACCOUNTS.BCK/BY_OWNER=[301,310]/REWIND/LABEL=BANK01-
  $/PROTECTION=(S:RWE,O:RWED,G:RE,W)
```

This command saves the directory tree [CLEAVER...] to a save set named ACCOUNTS.BCK on the magnetic tape labeled BANK01. The output save-set qualifier /REWIND directs BACKUP to rewind the tape and initialize it before performing the save operation. The output save-set qualifier /BY\_OWNER assigns an owner UIC of [301,310] to the magnetic tape. The /PROTECTION qualifier assigns the owner of the magnetic tape read, write, execute, and delete access. SYSTEM users are assigned read, write, and execute access; GROUP users are assigned read and execute access; WORLD users are assigned no access.

## **/RECORD**

### **Command Qualifier**

Records the current date and time in the BACKUP date field of each file header record once a file is successfully saved or copied.

### **format**

**/RECORD** *input-specifier output-specifier*

### **description**

The /RECORD qualifier can be used only on Files-11 Structure Level 2 volumes. The user privilege SYSPRV is required to use the /RECORD qualifier on files other than those owned by your UIC.

When you use /RECORD in a copy or save operation, BACKUP writes the date and time that the copy or save set was created in the BACKUP date field of each file header record.

When you use /RECORD to perform incremental save operations on a disk volume, do not allow other users to use /RECORD in their BACKUP operations on the same disk volume. If other users specify /RECORD, the dates in the BACKUP date fields of file header records will change. This will make it impossible for you to save all files created or modified since you last performed a save operation.

If you use the command qualifier /VERIFY with /RECORD, files that fail verification are not recorded.

If /RECORD is not specified, the BACKUP date field of each processed file is not changed.

You cannot use the /RECORD qualifier with the command qualifier /DELETE or /COMPARE.

### **example**

```
$ BACKUP/RECORD DBA1:[*...]/SINCE=BACKUP MTA0:13MAY.BCK
```

This command saves all files on DBA1 that have been created or modified since the last save operation and records the current date and time in each file header record.

## **/REPLACE**

### **Output File Qualifier**

Replaces a file on the output specifier with an identically-named file from the input specifier.

### **format**

*input-specifier output-specifier/REPLACE*

### **description**

When you use /REPLACE in a copy or restore operation, and an identically-named file exists in both the input and output specifier, BACKUP does the following:

- Copies or restores a new version of the file with the same directory specification, file name, type, and version number
- Deletes the copy of the file that previously existed on the output disk

In this way, the previous copy of the file is replaced with the restored version. Note that the version number is not incremented because the old copy of the file is deleted. If you want to keep the versions from both the input and the output specifiers, use the output file qualifier /NEW\_VERSION.

If you do not use /REPLACE, /OVERLAY, or /NEW\_VERSION, and the version number of the file being restored is identical to the version number of the existing file, BACKUP reports an error and does not restore the file.

### **example**

```
$ BACKUP MUA0:SAVEWORK.BCK/SELECT=[LEE...] DUA0:[LEE...]/REPLACE
```

The command in this example restores the directory tree [LEE...] (and all files in the directory tree) from a magnetic-tape save set to disk. The input save-set qualifier /SELECT specifies the directory tree to be selected from the save set, and the output file qualifier /REPLACE instructs BACKUP to first create a new version of an input file if there is a file on the output medium with the same file specification and then to delete the file that originally existed on the output medium.

---

## /REWIND

### Input Save-Set Qualifier

Rewinds the input tape reel to the beginning-of-tape marker before reading the input volume.

### format

*input-save-set-spec*[NO]REWIND *output-specifier*

### description

The /[NO]REWIND qualifier is for magnetic tape volumes only.

/REWIND directs BACKUP to rewind the input magnetic tape to the beginning-of-tape marker before reading the input volume. Then BACKUP locates the input save set. In this way, BACKUP can find the input save set if it is located before the current tape position.

/NOREWIND indicates that BACKUP should not rewind the input volume before processing the command. Instead, BACKUP proceeds toward the logical end-of-tape (the end of the last save set stored on the tape). Therefore, if the specified save set is located before the current position of the tape, BACKUP is unable to find it.

The default is /NOREWIND. You must specify /REWIND to rewind the tape.

### example

```
$ BACKUP MFA1:CONTRACTS.BCK/REWIND DBA2:[*...]/BY_OWNER=ORIGINAL
```

In this example, the save set CONTRACTS.BCK is restored to the disk volume mounted on DBA2. The /REWIND qualifier rewinds the magnetic tape to the beginning-of-tape marker before reading the input volume to search for CONTRACTS.BCK. The output file qualifier /BY\_OWNER restores the original owner UICs.

---

## /REWIND

### Output Save-Set Qualifier

Rewinds the output tape to the beginning-of-tape marker and initializes the output tape. The /NOREWIND qualifier causes the tape to wind forward to the logical end-of-tape (the end of the last save set stored on the tape) and to begin writing the save set there.

## BCK-42    **BACKUP** **/REWIND**

### **format**

*input-specifier output-save-set-spec***[NO]REWIND**

### **description**

The **/[NO]REWIND** qualifier is for magnetic tape volumes only.

If you specify **/REWIND**, **BACKUP** rewinds to the beginning of the magnetic tape and searches the volume header record for a volume label. If the volume header record contains no volume label, **BACKUP** writes the label specified in the **BACKUP** command to the volume header record, initializes the tape, and creates the save set on the tape.

If no label is specified explicitly in the command line, **BACKUP** uses the first six characters of the save-set name as the volume label of the first tape in a multivolume save set and the first four characters of the save-set name followed by the volume number of the tape as the volume label of subsequent tapes. You can also specify a label or list of labels explicitly with the **/LABEL** qualifier. If you do not specify enough labels with the **/LABEL** qualifier, **BACKUP** uses the first four characters of the final label in the list followed by the volume number of the tape as the volume label of subsequent tapes.

If **BACKUP** finds a volume label on the tape, it compares the volume label with the label you specified in the **BACKUP** command line (either explicitly with the **/LABEL** qualifier or implicitly through the save set name) and ensures that the tape is expired.

If the volume label is less than six characters long, **BACKUP** pads the volume label with the blank character to six characters. The first four characters of the volume label must either match the first four characters of the label specified in the **BACKUP** command line exactly, or the first four characters of the volume label must end with one or more underscore characters. If the first four characters of the volume label end with one or more underscore characters, and the label specified in the command line matches the part of the volume label that appears before the underscore characters, **BACKUP** accepts the match. (For example, the volume label **ABN\_** matches the command line label **ABN** but does not match the command line label **ABNE**.) If either the fifth or sixth character of the volume label is a number between zero and nine, **BACKUP** does not compare these characters with corresponding characters in the label specified in the **BACKUP** command line. Otherwise, the fifth and sixth characters in the volume label must match the corresponding characters in the label specified in the **BACKUP** command line exactly. The following table illustrates volume labels that match labels specified in the **BACKUP** command line:



Label Specified in the Command Line	Matching Volume Labels
MAR	MAR, MAR_, MAR_nn
MAR_	MAR_, MAR_nn
MARK	MARK, MARKnn
MARKER	MARKER, MARKnn

You can specify more than one label with the /LABEL qualifier. If any label specified in the BACKUP command line matches the volume label of the tape and the tape is expired, BACKUP overwrites the volume label of the tape with the same volume label.

By overwriting the tape's volume label, BACKUP initializes the tape, removing access to any data that previously resided on the tape and preparing the tape to receive new data. During the initialization process, BACKUP writes the values specified with the output save-set qualifiers /TAPE\_EXPIRATION, /PROTECTION, and /BY\_OWNER to the volume header record. (If these qualifiers are not specified, the default tape expiration date is today, the default protection is none, and the owner UIC of the tape is the UIC of the current process.) After initializing the tape, BACKUP writes the save set to the tape.

If the label in the BACKUP command line did not match the volume label of the tape, BACKUP displays the following message and prompt on your terminal if you specified the command qualifier /NOASSIST or on the operator terminal if you did not specify /NOASSIST:

```
%BACKUP-W-MOUNTERR, volume 'number' on 'device' was not mounted because
  its label does not match the one requested
Specify option (QUIT, NEW tape or OVERWRITE tape)
BACKUP>
```

If you enter QUIT at the BACKUP> prompt, BACKUP aborts, unloads the magnetic tape, and issues the following message:

```
%BACKUP-F-ABORT, operator requested abort on fatal error
```

If you enter NEW at the BACKUP> prompt, BACKUP unloads the magnetic tape and issues the following prompt for a new tape:

```
%BACKUP-I-READYWRITE, mount volume 'volume-number' on '_device-name': for writing
Enter "YES" when ready:
```

If you enter OVERWRITE at the BACKUP> prompt, BACKUP overwrites the old volume label with the new volume label. (OVERWRITE instructs BACKUP to ignore the fact that either the tape has not expired or that the labels do not match.) By overwriting the tape's volume label, BACKUP initializes the tape, removing access to any data that previously resided on the tape and preparing the tape to receive new data.

## BCK-44    **BACKUP** **/REWIND**

During the initialization process, **BACKUP** writes the values specified with the output save-set qualifiers **/TAPE\_EXPIRATION**, **/PROTECTION** and **/BY\_OWNER** to the volume header record. After initializing the tape, **BACKUP** writes the save set to the tape.

If the tape is not expired, **BACKUP** displays the following message and prompt on your terminal if you specified the command qualifier **/NOASSIST** or on the operator terminal if you did not specify **/NOASSIST**:

```
%BACKUP-W-MOUNTERR, volume 'number' on 'device' was not mounted because  
  its expiration date is in the future  
Specify option (QUIT, NEW tape or OVERWRITE tape)  
BACKUP>
```

Always specify **/REWIND** when the output tape has a non-ANSI or non-ISO label or when the output tape has never been initialized

The **/NOREWIND** qualifier directs **BACKUP** to compare the volume label of the tape with the label you specified in the **BACKUP** command before performing the save operation. You can specify a label explicitly with the **/LABEL** qualifier; otherwise, **BACKUP** uses the first six characters of the save-set name as the volume label. If the volume label does not match the label you specified, **BACKUP** displays the following message and prompt on your terminal if you specified the command qualifier **/NOASSIST** or on the operator terminal if you did not specify **/NOASSIST**:

```
%BACKUP-W-MOUNTERR, volume 'number' on 'device' was not mounted because  
  its label does not match the one requested  
Specify option (QUIT, NEW tape or OVERWRITE tape)  
BACKUP>
```

If you choose the **OVERWRITE** option, **BACKUP** ignores the fact that the volume labels do not match. If the labels match, or if you choose the **OVERWRITE** option, **BACKUP** winds the tape forward to the logical end-of-tape (the end of the last save set stored on the tape) and writes the save set to the tape. If the logical end-of-tape is also the physical end of the tape, **BACKUP** requests a new tape. Because **BACKUP** searches for the end of data on the tape, you cannot write a new save set to a tape if it ends with a save set that is continued onto another tape.

Although the **/NOREWIND** qualifier does not initialize the first tape in a multivolume save set, **BACKUP** initializes subsequent tapes in a multivolume save set. **BACKUP** ensures that the tape is expired and that the tape labels match before initializing subsequent volumes in a multivolume save set.

The default is **/NOREWIND**. You must specify **/REWIND** to rewind and initialize a magnetic tape volume.

## example

```
$ BACKUP
  FROM: *.RNO
  TO: MTA0:DSRSAVE.BCK/REWIND/LABEL=DSR01/TAPE_EXPIRATION=29-DEC-1989
```

The command in this example initializes a new magnetic tape and writes the volume label DSR01 and a tape expiration date of December 29, 1989, to the tape's volume header record. Then this command saves all files in the current default directory with a file type of RNO to the magnetic tape save set named DSRSAVE.BCK.

## /SAVE\_SET

### Input Save-Set Qualifier

Directs BACKUP to treat the input file as a BACKUP save set. You must specify /SAVE\_SET when the input specifier refers to a BACKUP save set on disk.

### format

*input-save-set-spec*/SAVE\_SET *output-specifier*

### description

The /SAVE\_SET qualifier allows you to refer to a BACKUP save set on a local Files-11 disk, a remote Files-11 disk, or a sequential disk. If you do not specify /SAVE\_SET, an input specifier that refers to a disk is treated as a Files-11 file. An input specifier that refers to tape is always treated as a BACKUP save set.

## example

```
$ BACKUP DBA2:[BACKUP]1212MAR3.BCK/SAVE_SET DBA1:[*...]
```

This command restores a save set named 1212MAR3.BCK from DBA2 to DBA1.

## /SAVE\_SET

### Output Save-Set Qualifier

Directs BACKUP to treat the output file as a BACKUP save set. You must specify the /SAVE\_SET qualifier when the output specifier refers to a BACKUP save set on disk.

## BCK-46 BACKUP /SAVE\_SET

### format

*input-specifier output-save-set-spec*/SAVE\_SET

### description

The /SAVE\_SET qualifier allows you to create a BACKUP save set on a local Files-11 disk, a remote Files-11 disk, or a sequential disk. If you do not specify /SAVE\_SET, an output specifier that refers to disk is treated as a Files-11 file. An output specifier that refers to tape is always treated as a BACKUP save set.

### example

```
$ BACKUP [HILL] DBA1:[BACKUP]SEP28.BCK/SAVE_SET
```

This command saves the directory [HILL] to a save set named SEP28.BCK on a Files-11 disk.

---

## /SELECT

### Input Save-Set Qualifier

Selects the specified files for processing.

### format

*input-save-set-spec*/SELECT=(*file-spec*[,...]) *output-specifier*

### description

If you specify more than one file, separate the file specifications with commas and enclose the list in parentheses. Do not use a device specification when you define the files to be selected. You can use most standard wildcard characters, but you cannot use wildcard characters denoting latest version of files (;) and relative versions of files (-n).

Note that BACKUP does not apply temporary file specification defaults within the list. Each file specification independently takes its defaults from the file specification [000000 . . . ]\*.\*.\*.

You cannot use /SELECT in image restore operations.

### example

```
$ BACKUP DBA1:JUL20.BCK/SAVE_SET/SELECT=[SNOW]BALL.PAS [WINTER.GAME]BALL.PAS
```

This command selects a file named [SNOW]BALL.PAS from a sequential-disk save set and restores it to the directory [WINTER.GAME] on the current default device.

---

## /SINCE

### Input File-Selection Qualifier

Selects files dated equal to or later than the specified date and time.

### format

*input-specifier/SINCE=time output-specifier*

### description

The /SINCE qualifier selects files by comparing the date and time in the specified field of each file header record with the date and time you specify in the command line. The following table shows the input file-selection qualifiers you can use with /SINCE and their functions. Use only one of these qualifiers at a time in your command line.

/BACKUP	Selects files last saved or copied by BACKUP/RECORD since the date specified. Also selects files with no BACKUP date.
/CREATED	Selects files created since the date specified.
/EXPIRED	Selects files that have expired since the date specified.
/MODIFIED	Selects files last modified since the date specified. If you specify /SINCE without another qualifier, /MODIFIED is used by default.

Specify the date and time as a delta time or as an absolute time using the format [dd-mmm-yyyy[:]][hh:mm:ss.cc]. You can also use one of the following reserved words to specify the date and time:

BACKUP	The BACKUP/RECORD operation (available only on Files-11 Structure Level 2 volumes)
TODAY	The current day, month, and year at 00:00:00.0 o'clock
TOMORROW	24 hours after midnight last night
YESTERDAY	24 hours before midnight last night

### example

```
$ BACKUP [PLI.WORK]/SINCE=YESTERDAY/MODIFIED [PLI.SAV]
```

This command copies selected files in the directory [PLI.WORK] to the directory [PLI.SAV]. Only those files that have been modified since 24 hours preceding midnight last night are processed. Note that the /MODIFIED qualifier is not required because its action is the default when the /SINCE qualifier is specified.

## **/TAPE\_EXPIRATION**

### **Output Save-Set Qualifier**

Writes the date on which the tape will expire to the volume header record. The output save-set qualifier /REWIND must be specified with the /TAPE\_EXPIRATION qualifier.

### **format**

*input-specifier output-save-set-spec/TAPE\_EXPIRATION[=date]*

### **description**

When you specify the output save-set qualifier /REWIND during a save operation to magnetic tape, BACKUP checks that the tape has expired before initializing the tape. Initializing the tape removes access to data previously stored on the tape.

Digital recommends that you specify an expiration date whenever you create a BACKUP save set on magnetic tape using /REWIND. Daily BACKUP tapes should expire in seven days, weekly BACKUP tapes should expire in one month, and monthly BACKUP tapes should expire in one year.

Specify the date in the following format:

dd:mmm:yyyy

where:

dd                            is the date.

mmm                         is a three-letter abbreviation of the month.

yyyy                        is the year.

If you do not specify an expiration date, today's date is written to the volume header record when you perform a save operation using /REWIND.

### **example**

```
$ BACKUP DBA1: MTA0:13NOVBAK.BCK/REWIND/TAPE_EXPIRATION=20-NOV-1990/LABEL=NOV02
```

In this example, the entire contents of the disk DBA1 are saved to a save set named 13NOVBAK.BCK. The tape will expire in seven days on November 20, 1990.

## /TRUNCATE

### Command Qualifier

Controls whether a copy or restore operation truncates a sequential output file at the end-of-file (EOF) when restoring it.

### format

**/[NO]TRUNCATE** *input-specifier output-specifier*

### description

By default, a copy or restore operation uses the allocation of the input file to determine the size of the output file. Specify **/TRUNCATE** if you want the output files to be truncated at the end-of-file (EOF).

### example

```
$ DIRECTORY/SIZE [FRANKIE]ORIGINAL.DAT
Directory DMA0:[FRANKIE]
ORIGINAL.DAT          35
Total of 1 file, 35 blocks
$ COPY ORIGINAL.DAT EXTENDED.DAT/ALLOCATION=500
$ BACKUP [FRANKIE]EXTENDED.DAT MFA0:20JUL.BCK/LABEL=WKLY03
$ BACKUP/TRUNCATE MFA0:20JUL.BCK/LABEL=WKLY03 DMA0:[FRANKIE]
```

This sequence of commands does the following:

- Determines that the file ORIGINAL.DAT is 35 blocks long.
- Copies ORIGINAL.DAT to EXTENDED.DAT, allocating 500 blocks for EXTENDED.DAT.
- Saves the file EXTENDED.DAT to a save set named 20JUL.BCK on MFA0. BACKUP writes the file allocation size in the file header record of the saved file but saves only 35 blocks in the save set.
- Restores the save set file on MFA0 to a volume mounted on DMA0 and truncates the output files at the EOF. The restored file is 35 blocks long.

## **/VERIFY**

### **Command Qualifier**

Specifies that the contents of the output specifier be compared with the contents of the input specifier after a save, restore, or copy operation is completed.

### **format**

*/VERIFY input-specifier output-specifier*

### **description**

The **/VERIFY** qualifier is different from the command qualifier **/COMPARE**. Unlike the **/VERIFY** qualifier, the command qualifier **/COMPARE** cannot be used in a save, restore, copy, or list operation. The **/VERIFY** qualifier directs **BACKUP** to perform the copy, save, or restore operation first and then to perform the compare operation.

On file-structured copy operations, each file is compared after it is copied. On physical copy operations, the volume is compared after it is copied. For a save or restore operation, the verification is performed in a separate pass and is preceded by the following informational message:

`%BACKUP-I-STARTVERIFY, starting verification pass`

If a file does not compare successfully, **BACKUP** displays the following error message:

`%BACKUP-E-VERIFYERR, verification error for block 'block-number'  
of 'disk:[directory]file_name.file_type;version_number'`

The **/VERIFY** qualifier does not work on a restore or copy operation when the **/NEW\_VERSION** output file qualifier is also used. Because the **/NEW\_VERSION** qualifier reassigns output file versions, it is not possible to correctly associate the created output files with the input files from which they were copied.



## example

```
$ BACKUP/VERIFY/LOG *.LIS MFA0:LIST.BCK
%BACKUP-S-COPIED, copied DISK$DEFAULT:[WONDER]CRE.LIS;1
%BACKUP-S-COPIED, copied DISK$DEFAULT:[WONDER]CRETIME.LIS;1
%BACKUP-S-COPIED, copied DISK$DEFAULT:[WONDER]EXC.LIS;1
%BACKUP-S-COPIED, copied DISK$DEFAULT:[WONDER]NOREB.LIS;1
%BACKUP-S-COPIED, copied DISK$DEFAULT:[WONDER]REB.LIS;1
%BACKUP-S-COPIED, copied DISK$DEFAULT:[WONDER]SETREB.LIS;1
%BACKUP-S-COPIED, copied DISK$DEFAULT:[WONDER]VERS.LIS;1
%BACKUP-I-STARTVERIFY, starting verification pass
%BACKUP-S-COMPARED, compared DISK$DEFAULT:[WONDER]CRE.LIS;1
%BACKUP-S-COMPARED, compared DISK$DEFAULT:[WONDER]CRETIME.LIS;1
%BACKUP-S-COMPARED, compared DISK$DEFAULT:[WONDER]EXC.LIS;1
%BACKUP-S-COMPARED, compared DISK$DEFAULT:[WONDER]NOREB.LIS;1
%BACKUP-S-COMPARED, compared DISK$DEFAULT:[WONDER]REB.LIS;1
%BACKUP-S-COMPARED, compared DISK$DEFAULT:[WONDER]SETREB.LIS;1
%BACKUP-S-COMPARED, compared DISK$DEFAULT:[WONDER]VERS.LIS;1
```

This example creates a magnetic-tape save set on MFA0 and starts the verification pass after the save operation is completed. The /LOG qualifier displays the file names as they are processed.

---

## /VOLUME

### Command Qualifier

Indicates that a specific disk volume in a disk volume set is to be processed. The /VOLUME qualifier is valid only when used with the /IMAGE qualifier.

### format

*/IMAGE/VOLUME=*n* input-specifier output-specifier*

### description

The /VOLUME qualifier allows you to perform an image save, restore, or copy operation using one more disk drive than the number of disks in the input volume set. When you use /VOLUME, you must write-lock the entire input volume set.

When performing an image copy or save operation with the /VOLUME qualifier, all disks in the input volume set must be mounted. Mount the volumes of the target volume set one at a time. Enter a separate BACKUP command for each disk in the input volume set. A save set created with the /VOLUME qualifier must be restored using the /VOLUME qualifier.

## BCK-52    **BACKUP**           **/VOLUME**

You can restore any image save set with the **/VOLUME** qualifer. All disks in the output volume set must be mounted. Mount the disks in the input volume set one at a time. You cannot use the command qualifier **/NOINITIALIZE** in the restore operation with the command qualifier **/VOLUME**.

In a compare operation that uses the **/VOLUME** qualifier to compare two disk volume sets, all disks in both volume sets must be mounted. In a selected-volume compare operation between a save set on tape and a disk volume set, all disks in the disk volume set must be mounted.

### **example**

```
$ BACKUP/IMAGE/VOLUME=3 DISK$PUBLIC DRA1:
```

This command creates a functionally equivalent copy of the third volume of a volume set named **DISK\$PUBLIC** to **DRA1**. The disk mounted in **DRA1** becomes the third volume of the image-copy volume set.

---

## Bad Block Locator Utility

The Bad Block Locator Utility (BAD) analyzes block-addressable devices and records the locations of blocks that cannot store data reliably.

### format

**ANALYZE/MEDIA** *device*

### parameter

#### *device*

Specifies the device containing the volume that BAD will analyze. The device name has the form ddcu: or logical-name.

### usage summary

To invoke BAD, enter the command ANALYZE/MEDIA at the DCL prompt along with any parameters or qualifiers. Once invoked, BAD runs until completion. When BAD terminates, control is returned to the DCL command level.

To write the contents of the Detected Bad Block File (DBBF) to an output file, specify the /OUTPUT qualifier, as described in the following section.

## BAD-2 BAD BLOCK LOCATOR /BAD\_BLOCKS

### BAD BLOCK LOCATOR Qualifiers

This section presents qualifiers for the ANALYZE/MEDIA command in alphabetical order. The qualifiers follow the standard rules of DCL syntax. Thus, you can abbreviate any qualifier or keyword as long as the abbreviation is not ambiguous. The asterisk and the percent sign can be used as wildcard characters unless otherwise noted.

---

### /BAD\_BLOCKS

Adds the specified bad blocks to the Detected Bad Block File (DBBF). If the /BAD\_BLOCKS qualifier is specified along with the /EXERCISE qualifier, the medium is tested and the bad blocks are added to the DBBF.

#### format

`/BAD_BLOCKS [(=list)]`

#### keyword

##### *list*

Specifies codes for the bad block locations to be added to the DBBF.

If you do not specify a value for the /BAD\_BLOCKS qualifier, BAD prompts as follows:

```
BAD_BLOCKS =
```

When it prompts, BAD reports any duplicate bad blocks. To terminate the prompting session, type CTRL/Z.

**NOTE:** The term *block* denotes a standard unit of 512 bytes, whereas the term "sector" denotes the physical size of the device sector, which is not always the same for all devices. For example, an RL02 has a sector size of 256 bytes, while an RK07 has a standard sector size of 512 bytes.

Valid bad block location codes follow. You can specify them in any integer combination or radix combination.

---

Code	Meaning
lbn	Specifies the logical block number (LBN) of a single bad block.
lbn:count	Specifies a range of contiguous bad blocks starting at the logical block number (LBN) and continuing for "count" blocks.

Code	Meaning
sec.trk.cyl	Specifies the physical disk address (sector, track, and cylinder) of a single bad sector. This code is valid only for last-track devices.
sec.trk.cyl:count	Specifies a range of bad sectors starting at the specified physical disk address (sector, track, and cylinder) and continuing for "count" sectors. This code is valid only for last-track devices.

### example

§ ANALYZE/MEDIA/EXERCISE/BAD\_BLOCKS=(2) DB1:

The command in this example adds the bad block specification to the DBBF and then tests the medium. The bad block in this example is located at logical block number (LBN) 2.

---

### /EXERCISE

Controls whether the medium should actually be tested. The default is /NOEXERCISE.

### format

**/EXERCISE** [(keyword[,...])]  
**/NOEXERCISE**

### keywords

#### **FULL**

Causes BAD to test the medium using three test patterns (0's, 1's, and "worst case") instead of the default single "worst case" pattern. The FULL keyword can be used only with /EXERCISE. Note that the "worst case" pattern always remains on media tested with the /EXERCISE qualifier.

#### **KEEP**

Ensures the preservation of the current SDBBF. The KEEP keyword is the default when /NOEXERCISE is specified.

#### **NOKEEP**

Causes BAD to create a new SDBBF. The NOKEEP keyword is the default when /EXERCISE is specified. This keyword cannot be used with the /NOEXERCISE qualifier.

#### **PATTERN=(longword[,...])**

Allows users to specify the value of a test pattern to be used as "worst case." Up to an octaword of test pattern data may be specified in decimal (%D), hexadecimal (%X), or octal (%O) radices. The default radix is decimal.

## BAD-4 BAD BLOCK LOCATOR /EXERCISE

The pattern is specified in longwords. If two or more longwords are specified, they must be enclosed in parentheses and separated by commas.

### example

```
$ ANALYZE/MEDIA/NOEXERCISE/BAD_BLOCKS DBB1:
```

The command in this example updates the DBBF without erasing the volume's contents.

---

### /LOG

Specifies whether a message is sent to SYS\$OUTPUT and to SYS\$ERROR indicating the total number of bad blocks detected by BAD. The default is /NOLOG.

### format

**/[NO]LOG**

### example

```
$ ANALYZE/MEDIA/LOG DBB1:
```

Device DBB1: contains a total of 340670 blocks; 11 defective blocks detected.

The command in this example requests BAD to report the total number of bad blocks it detected on the device DBB1.

---

### /OUTPUT

Specifies whether the contents of the DBBF are written to the specified file. If you omit the /OUTPUT qualifier, no output is generated.

When you specify /OUTPUT in conjunction with the /SHOW qualifier, the default keyword for the /SHOW qualifier is AFTER.

### format

**/OUTPUT** *=[file-spec]*

### keyword

#### ***file-spec***

Identifies the output file for storing the results of the medium analysis. If you specify a file type and omit the file name, the default file name ANALYZE is used. The default file type is ANL. If you omit the file-spec, the results are output to SYS\$OUTPUT.

In place of the file-spec, you may specify an output device. In this case, BAD writes the contents of the volume's DBBF to a file called ANALYZE.ANL and queues the file to the output device.

No wildcard characters are allowed in the file specification.

## example

\$ ANALYZE/MEDIA/OUTPUT=BADDBBF.DAT DBA2:

The command in this example writes the contents of the DBBF from DBA2 to the output file BADDBBF.DAT. Note that because /NOEXERCISE is the default, the medium is not tested.

---

## /RETRY

Enables the device driver to retry soft errors. The /RETRY qualifier is used only in conjunction with the /EXERCISE qualifier. The default is /NORETRY.

## format

**/EXERCISE /NORETRY**

**/EXERCISE /RETRY**

## example

\$ ANALYZE/MEDIA/EXERCISE/RETRY DBA0:

The command in this example directs the device driver to retry soft errors.

---

## /SHOW

Lists the contents of the DBBF before and after the medium is exercised (tested).

## format

**/SHOW [(keyword[,...])]**

## keywords

**[NO]BEFORE,[NO]AFTER**

Specifies whether the contents of the DBBF is listed before, after, or before and after the medium is exercised (tested). AFTER is the default.

## example

\$ ANALYZE/MEDIA/EXERCISE/OUTPUT/SHOW=(BEFORE,AFTER) DBA3:

The command in this example lists the contents of the DBBF both before and after the disk DBA3 is exercised and directs the data to the current SYS\$OUTPUT device.





---

## Error Log Utility

The Error Log Utility (ERROR LOG) selectively reports the contents of an error log file.

### format

**ANALYZE/ERROR\_LOG** [/qualifier(s)] [file-spec[,...]]

### parameters

#### **/qualifier(s)**

The function to be performed by the ANALYZE/ERROR\_LOG command.

#### **file-spec[,...]**

Specifies one or more files that contain binary error information to be interpreted for the error log report. You can include wildcard characters in the file specification. If you omit the file specification, the default file is SYS\$ERRORLOG:ERRLOG.SYS

### usage summary

To invoke ERROR LOG, enter the following DCL command:

```
ANALYZE/ERROR_LOG [/qualifier(s)] [file-spec][,...]
```

ERROR LOG does not prompt you. To exit from ERROR LOG, press CTRL/C. You also exit the utility when end-of-file (EOF) is detected. To direct output, use the /OUTPUT, /BINARY, and /REJECTED qualifiers with the ANALYZE/ERROR\_LOG command.

You must have SYSPRV privilege to run ERROR LOG. However, only read access is required to access the file ERRORLOG.SYS. (It is not necessary to rename the file ERRORLOG.SYS to ERRORLOG.OLD before using ERROR LOG.) Do not use the /BINARY qualifier with the /FULL, /BRIEF, /OUTPUT, /REGISTER\_DUMP, or /SUMMARY qualifiers.

## ERR-2 ERROR LOG /BEFORE

### ERROR LOG Qualifiers

The qualifiers for the ANALYZE/ERROR\_LOG command are described in this section.

---

#### /BEFORE

Specifies that only those entries dated earlier than the stated date and time are to be selected for the error report.

#### format

**/BEFORE** [*=date-time*]

#### parameters

##### *date-time*

Limits the error report to those entries dated earlier than the specified time.

#### description

You can specify an absolute time, a delta time, or a combination of absolute and delta times.

If you omit the /BEFORE qualifier or specify /BEFORE without a date or time, all entries are processed.

#### example

```
$ ANALYZE/ERROR_LOG/BEFORE=19-APR-1990:10:00 ERRLOG.OLD;5
```

In this example, the error log report generated for ERRLOG.OLD;5 contains entries that were logged before 10:00 A.M. on April 19, 1990.

---

#### /BINARY

Used to control whether the binary error log records are converted to ASCII text or copied to the specified output file.

#### format

**/BINARY** [*=file-spec*]

**/NOBINARY**

## parameters

### *file-spec*

Specifies the output file selected to contain image copies of the input records.

## description

The /BINARY qualifier creates a binary file that contains copies of the original binary error log entry if the command line also specifies an interval (/SINCE, /BEFORE, or /ENTRY qualifier) or a filter (/INCLUDE or /EXCLUDE qualifier). If no interval or filter is specified, all error log entries are copied.

If you specify /BINARY=file-spec, the selected output file contains image copies of the binary input records (the records are not translated to ASCII). If you omit the device or directory specification, the current device and the default directory are used. If you omit the file name, the file name of the input file is used. If you omit the file type, the default file type is DAT.

Do not use /BINARY with the /FULL, /BRIEF, /OUTPUT, /REGISTER\_DUMP, or /SUMMARY qualifiers. These qualifiers generate an ASCII report; /BINARY generates a binary file.

## example

```
$ ANALYZE/ERROR_LOG/INCLUDE=DBA1/BINARY=DBA1_ERR.DAT ERRLOG.OLD;5
```

In this example, the output file DBA1\_ERR.DAT contains image copies of the entries that apply to DBA1.

---

## /BRIEF

Generates a brief report.

## format

**/BRIEF**

## description

Do not use /BRIEF with the /BINARY qualifier.

## example

```
$ ANALYZE/ERROR_LOG/BRIEF ERRLOG.OLD;97
```

In this example, the error log report generated from ERRLOG.OLD;97 contains minimal information.

## ERR-4 ERROR LOG

### /ENTRY

---

### /ENTRY

Generates an error log report that includes the specified entry range or starts at the specified entry number.

#### format

**/ENTRY** *[(START:decimal-value[,END:decimal-value])]*

#### parameters

*(START:decimal-value[,END:decimal-value])*

The range of entries to be included in the error log report.

#### description

If you specify **/ENTRY** without the entry range or omit the qualifier, the entry range defaults to **START:1,END:end-of-file**.

#### example

```
§ ANALYZE/ERROR_LOG/ENTRY=(START:1,END:18) ERRLOG.SYS
```

In this example, the entry range for the error log report generated from file **ERRLOG.SYS** is limited to entry numbers 1 through 18.

---

### /EXCLUDE

Excludes errors generated by the specified device and error log entry type from the error log report.

#### format

**/EXCLUDE** *=(device-or-entry-type[,...])*

#### parameters

*device-or-entry-type[,...]*

The device and entry type to be excluded from the error log report.

#### description

You can specify one or more devices by device class, device name, or one or more keywords that identify entry types.

### Device Class Keywords

BUSES  
DISKS  
LINE\_PRINTER  
REALTIME  
SYNC\_COMMUNICATIONS  
TAPES  
WORKSTATION

### Examples of Device Name Constructs

DB	Group of devices
DBA1	Specific device/unit number
(DBA1,HSC1\$DUA1,DYA0)	List of devices
(DB,DR,XF)	List of device groups

### Entry Type Keywords

ATTENTIONS	Exclude device attention entries from the output report.
BUGCHECKS	Exclude all types of bugcheck entries from the report.
CONTROL_ENTRIES	Exclude control entries from the report. Control entries include the following entry types: <ul style="list-style-type: none"><li>• System power-fail restarts</li><li>• Time stamps</li><li>• System startups</li><li>• \$SNDErr messages (system service to send messages to error log)</li><li>• Operator messages</li><li>• Network messages</li><li>• ERRLOG.SYS created</li></ul>

## ERR-6 ERROR LOG /EXCLUDE

CPU_ENTRIES	Exclude CPU-related entries from the report. CPU entries include the following entry types: <ul style="list-style-type: none"><li>• SBI alerts/faults</li><li>• Undefined interrupts</li><li>• MBA/UBA adapter errors</li><li>• Asynchronous write errors</li><li>• UBA errors</li></ul>
DEVICE_ERRORS	Exclude device error entries from the report.
ENVIRONMENTAL_ENTRIES	Exclude environmental entries from the report.
MACHINE_CHECKS	Exclude machine check entries from the report.
MEMORY	Exclude memory errors from the report.
SNAPSHOT_ENTRIES	Exclude snapshot entries from the report.
TIMEOUTS	Exclude device timeout entries from the report.
UNKNOWN_ENTRIES	Exclude any entry that had either an unknown entry type or an unknown device type/class.
UNSOLICITED_MSCP	Exclude unsolicited MSCP entries from the output report.
VOLUME_CHANGES	Exclude volume mount and dismount entries from the report.

### example

```
$ ANALYZE/ERROR_LOG/EXCLUDE=(MTA0,DRA5) ERRLOG.OLD
```

In this example, the devices MTA0 and DRA5 are excluded from the error log report for the file ERRLOG.OLD.

---

### /FULL

Generates a full report, which provides all available information for an error log entry. This is the default report format.

### format

**/[NO]FULL**

### description

Do not use /FULL with the /BINARY qualifier.

## example

```
$ ANALYZE/ERROR_LOG ERRLOG.OLD;72
```

The command in this example produces a full report. The default report type is /FULL; it is not necessary to specify it in the command line.

---

## /INCLUDE

Includes errors generated by the specified device and error log entry type in the error log report.

### format

```
/INCLUDE=(device-or-entry-type[,...])
```

### parameters

*device-or-entry-type[,...]*

The device and entry type to be included in the error log report.

### description

You can specify one or more devices by device class, device name, or one or more keywords that identify entry types.

#### Device Class Keywords

BUSES  
DISKS  
LINE\_PRINTER  
REALTIME  
SYNC\_COMMUNICATIONS  
TAPES  
WORKSTATION

#### Examples of Device Name Constructs

DB	Group of devices
DBA1	Specific device/unit number
(DBA1,HSC1\$DUA1,DYA0)	List of devices
(DB,DR,XF)	List of device groups

## ERR-8 ERROR LOG /INCLUDE

### Entry Type Keywords

ATTENTIONS	Include device attention entries in the output report.
BUGCHECKS	Include all types of bugcheck errors in the report.
CONTROL_ENTRIES	Include control entries in the report. Control entries include the following entry types: <ul style="list-style-type: none"><li>• System power-fail restarts</li><li>• Time stamps</li><li>• System startups</li><li>• \$SNDERR messages (system service to send messages to error log)</li><li>• Operator messages</li><li>• Network messages</li><li>• ERRLOG.SYS created</li></ul>
CPU_ENTRIES	Include CPU-related entries in the report. CPU entries include the following entry types: <ul style="list-style-type: none"><li>• SBI alerts/faults</li><li>• Undefined interrupts</li><li>• MBA/UBA adapter errors</li><li>• Asynchronous write errors</li><li>• UBA errors</li></ul>
DEVICE_ERRORS	Include device errors in the report.
ENVIRONMENTAL_ENTRIES	Include environmental entries in the report.
MACHINE_CHECKS	Include machine check errors in the report.
SNAPSHOT_ENTRIES	Include snapshot entries in the report.
MEMORY	Include memory errors in the report.
TIMEOUTS	Include device timeout errors in the report.
UNKNOWN_ENTRIES	Include any entry that had either an unknown entry type or an unknown device type/class.
UNSOLICITED_MSCP	Include unsolicited MSCP entries in the output report.
VOLUME_CHANGES	Include volume mount and dismount entries in the report.



## example

```
$ ANALYZE/ERROR_LOG/INCLUDE=(DISK,VOLUME_CHANGES,DEVICE_ERROR)
```

In this example, the report consists of error log entries for volume and device error information on disks, which are in the default error log file ERRLOG.SYS.

---

## /LOG

Controls whether informational messages that specify the number of entries selected and rejected for each input file are sent to SYS\$OUTPUT. By default, these messages are not displayed.

## format

**/[NO]LOG**

## example

```
$ ANALYZE/ERROR_LOG/LOG ERRLOG.OLD;5
```

In this example, informational messages generated about ERRLOG.OLD;5 are sent to SYS\$OUTPUT.

---

## /OUTPUT

Specifies the output file for the error log report.

## format

**/OUTPUT** [*file-spec*]

## parameters

***file-spec***

The output file selected for the error log report.

## description

If you omit the /OUTPUT qualifier, output is directed to SYS\$OUTPUT. If you specify /OUTPUT=file-spec, the selected output file contains the error log report. If you omit the device or directory specification, the current device and default directory are used. If you omit the file name, the file name of the input file is used. If you omit the file type, the default file type is .LIS.

Do not use /OUTPUT with the /BINARY qualifier.

## ERR-10 ERROR LOG /OUTPUT

### example

```
$ ANALYZE/ERROR_LOG/OUTPUT=ERROR_LOG.LIS ERRLOG.OLD;72
```

In this example, the output file `ERROR_LOG.LIS` contains entries generated from `ERRLOG.OLD;72`.

---

## /REGISTER\_DUMP

Used in conjunction with the `/INCLUDE` qualifier to generate, in a hexadecimal longword format, a report that consists of device register information.

### format

```
/REGISTER_DUMP
```

### description

Use the `/REGISTER_DUMP` qualifier to get a report that lists the hexadecimal contents of the device registers for the device specified by the `/INCLUDE` qualifier. The `/INCLUDE` qualifier must be used with the `/REGISTER_DUMP` qualifier.

`/REGISTER_DUMP` reports register contents for memory, device error, and device timeout entries. There is no translation of any of the device register information.

Do not use `/REGISTER_DUMP` with the `/BINARY` qualifier.

### example

```
$ ANALYZE/ERROR_LOG/INCLUDE=DB/REGISTER_DUMP ERRLOG.OLD;72
```

In this example, the output is in the format of a `REGISTER_DUMP` report containing entries that apply only to the DB device.

---

## /REJECTED

Allows you to specify the name of a file that will contain binary records for rejected entries.

### format

```
/REJECTED [=file-spec]
```

## parameters

### *file-spec*

Specifies the name of the file that is to contain the rejected entries.

## description

The /REJECTED qualifier creates a binary file that contains copies of the original binary error log entry. If the error log entry is rejected because the command line also specifies an interval (/SINCE, /BEFORE, or /ENTRY qualifier) or a filter (/INCLUDE or /EXCLUDE qualifier), the entry is written to the specified file.

Rejected entries are those entries that are not translated because they fall into one of the following categories:

- All entries specified with the /EXCLUDE qualifier
- All entries not specified with the /INCLUDE qualifier
- Any entry that does not occur within the period specified by the /SINCE and /BEFORE qualifiers
- Any entry that is not in the range of entries specified by the /ENTRY qualifier

If you specify /REJECTED=file-spec, the output file contains image copies of the rejected records. If you omit the device or directory specification, the current device and default directory are used. If you omit the file name, the file name of the input file is used. If you omit the file type, the default file type is .REJ.

## example

```
$ ANALYZE/ERROR_LOG/INCLUDE=MTA0/REJECTED=REAL_ERRS.DAT ERRLOG.OLD;5
```

In this example, the output file REAL\_ERRS.DAT contains image copies of all entries from ERRLOG.OLD;5, with the exception of those entries that apply to the MTA0 device.

---

## /SID\_REGISTER

Generates a report consisting of error log entries that occurred on the specified CPU.

## format

**/SID\_REGISTER** [=%X*hexadecimal-value*]

## ERR-12 ERROR LOG /SID\_REGISTER

### parameters

#### **%Xhexadecimal-value**

Specifies the value obtained from the system ID register. Use the \$GETSYI system service to obtain this value, which is unique to each system.

### example

```
$ ANALYZE/ERROR_LOG/SID_REGISTER=%X02006148 ERRLOG.OLD;72
```

In this example, the output consists of only those entries that were logged for the system with the system ID of 02006148 (hexadecimal).

---

### /SINCE

Specifies that only those entries dated later than the stated date and time are to be selected for the report.

### format

**/SINCE** [=date-time]

### parameters

#### **date-time**

Limits the error report to those entries dated later than the specified time.

### description

Only absolute date and time specifications are valid.

If you omit the /SINCE qualifier, all entries are processed. If you specify /SINCE without a date and time, the default is TODAY.

### example

```
$ ANALYZE/ERROR_LOG/SINCE=19-APR-1990:15:00 ERRLOG.OLD;56
```

In this example, the error log report generated from ERRLOG.OLD;56 contains entries that have been logged since 15:00 on April 19, 1990.

## **/STATISTICS**

Generates run-time statistical information.

### **format**

**/STATISTICS**

### **description**

Use the **/STATISTICS** qualifier to generate a report that consists of the page faults, buffered I/O, direct I/O, and CPU time used in the execution of the **ANALYZE/ERROR\_LOG** command.

### **example**

```
$ ANALYZE/ERROR_LOG/STATISTICS ERRLOG.OLD;4
```

In this example, the output generated by this command consists of a full report of all entries in **ERRLOG.OLD;4** and the run-time statistics for the execution of the command.

---

## **/SUMMARY**

Generates an error log report that consists of a statistical summary.

### **format**

**/SUMMARY** [=summary-type[,...]]

**/NOSUMMARY**

### **qualifier parameter**

*summary-type*

The keyword for the selected type of summary.

### **parameters**

#### **Keywords**

<b>DEVICE</b>	Include the device summary section in the report.
<b>ENTRY</b>	Include the summary of entries logged section in the report.
<b>HISTOGRAM</b>	Include the processed entries hour of day histogram in the report.
<b>MEMORY</b>	Include the summary of memory errors section in the report.
<b>VOLUME</b>	Include the volume label section in the report.

## **ERR-14 ERROR LOG**

### **/SUMMARY**

#### **description**

Select the type of summary by specifying one or more keywords.

**NOTE:** If you specify /SUMMARY without a summary type, the report contains all of the summary types listed above. If you want only a summary report, specify both the /NOFULL and the /SUMMARY qualifiers in the command line.

Do not attempt to correlate the error counts reported by the DCL command SHOW ERROR and the /SUMMARY qualifier. A discrepancy in these figures could be due to several system events and would be difficult to track.

Do not use the /BINARY qualifier with /SUMMARY.

#### **example**

```
$ ANALYZE/ERROR_LOG/NOFULL/SUMMARY ERRLOG.OLD;5
```

The output generated by the command in this example consists of a summary report of all entries in ERRLOG.OLD;5.

---

## Exchange Utility

The Exchange Utility (EXCHANGE) allows you to manipulate mass storage volumes written in formats other than those normally recognized by VMS.

### format

**EXCHANGE** *command* [*file-spec*] [*file-spec*]

**EXCHANGE>** *command* [*file-spec*] [*file-spec*]

### parameters

#### *command*

Defines the specific operation to be performed.

#### *file-spec*

Specifies the device name, directory, and file name for the EXCHANGE input or output device. It has the following general form:

device:[directory]filename.filetype;version

device:	The device name can be either a standard VMS device name of the form ddcu: or a logical name that translates to a VMS device name. If the device field is omitted for a reference, the current default device is assumed. When a virtual device is mounted, a name is created for the virtual device and is used as the device name in subsequent EXCHANGE commands.
[directory]	The syntax of the directory subfield is volume specific.
filename	The name field file specification for an input or output file. The exact format allowed for the filename is dependent on the volume format qualifier used.
filetype	The extension field of the file specification.
version	The version number of the file, if supported by the volume type.

### usage summary

You can use EXCHANGE in two ways. You can work interactively (within the utility) by entering "EXCHANGE" at the DCL prompt. This invokes the utility, which responds with the EXCHANGE> prompt. You can then enter any EXCHANGE command. You must invoke the utility and use it interactively if you want to execute more than one EXCHANGE command. However, you can enter a single EXCHANGE command at DCL level.

When you use EXCHANGE at the DCL level, the utility returns you to the DCL prompt after it completes its task. If you are using EXCHANGE interactively, you can return to DCL at any time by typing EXIT or CTRL/Z.

## EXCH-2 Exchange Utility

You can direct output from EXCHANGE operations in several ways. The command qualifier `/[NO]MESSAGE` allows you to control the default display of information from EXCHANGE MOUNT, INITIALIZE, and DISMOUNT operations. When you use the EXCHANGE commands COPY, DELETE, RENAME, or TYPE, include the `/LOG` qualifier to send information about the files being processed to `SYS$OUTPUT`. When you use the EXCHANGE command DIRECTORY, use the `/OUTPUT[=file-spec]` qualifier to direct the output to a specified file. If you specify the `/OUTPUT` qualifier without a file specification, the output is directed to `SYS$OUTPUT`. To send the output to a printer, use the `/PRINTER` qualifier with the DIRECTORY command.



## EXCHANGE Commands

The syntax of each of the EXCHANGE commands is similar to that of the corresponding DCL command. This section describes the functions and provides examples of the EXCHANGE commands.

---

### COPY

Transfers a file or files from an input volume to an output volume. You can use the COPY command to do any of the following:

- Copy a file from a foreign volume to a native volume
- Copy a file from a native volume to a foreign volume
- Copy a file from one foreign volume to another foreign volume
- Convert the format of the file during the transfer
- Copy groups of files from volume to volume
- Give the output file a different name from the input file

#### format

**COPY** *input-file-spec*[, ... ] *output-file-spec*

#### parameters

##### *input-file-spec*[, ... ]

Specifies the names of one or more input files to be copied. If you specify more than one input file, separate them with commas or plus signs. The syntax for input file names depends on the volume format option. You can specify standard VMS wildcards in both Files-11 and foreign file names. COPY supports wildcard directories for Files-11 and DOS-11 input.

##### *output-file-spec*

Specifies the name of the output file, directory, or device to which the input files are to be copied. If the input is a single file, you can specify an explicit output name (which is equivalent to a rename on a copy operation). If the input is more than one file, the output specifier must be one of the following:

- Wildcards (\*, \*.\* or \*.\*;\*) specifying current default device and directory
- An explicit device and/or directory for Files-11 output, such as BB:[EXCHANGE.TMP], with or without wildcards for the file name

## EXCH-4 EXCHANGE COPY

- An explicit device for RT-11 as in DLA2:/VOLUME=RT11
- An explicit device or directory for DOS-11 output, such as TAPE:/VOLUME=DOS11 or TAPE:[11,132]/VOLUME=DOS11

The output file names are constructed according to rules implied by the input and output volume qualifiers. COPY does not concatenate multiple input files into a single output file. Wildcard directories are not permitted. The syntax for input file names depends on the volume format option.

You must specify at least one field in the output file specification; COPY replaces missing fields with the corresponding field of the related input file specification. If the input file has no corresponding field, COPY substitutes null text fields and maximizes version numbers.

The UIC of the output file is the UIC of the current process. For DOS-11 output in UIC format, EXCHANGE uses the current default directory; otherwise, it uses the current process UIC as a directory. You can specify an alternate directory for DOS-11 output in the command.

### qualifiers

#### **/BOOT[=*nn*]**

Copies bootstrap information from a monitor and the handler files to blocks 0 and 2 through 5 of an RT-11 volume, permitting you to use that volume as a system volume. The COPY/BOOT operation does not create any files on the volume; it is intended only to create bootable RT-11 systems.

The /BOOT qualifier implies /VOLUME\_FORMAT=RT11 for both input and output specifications. The output device can be omitted, as it is assumed to be identical to the input device. You cannot combine the /BOOT qualifier with qualifiers other than /LOG. The COPY/BOOT command requires that both the input and output devices be the same volume or virtual device. The file name of the desired monitor must be specified as the input specification.

RT-11 Version 1.0 through Version 3.0 monitors had the system device handler linked into the monitor image. For Version 4.0 of RT-11, the system device handler uses the standard device handler, and the COPY/BOOT command must dynamically link the handler into the bootstrap area. COPY/BOOT finds the default handler for the specific device type and merges the handler with the monitor as it is copied to the boot area.

You can use the two-letter argument *nn* to override the default system device handler. The most frequent use of this option occurs when a diskette is mounted in an RX02 drive, and you want to create a diskette bootable from an RX01 drive. (The diskette must be single density.) The default handler for the RX02 is DY.SYS, and the handler for the RX01 is DX.SYS; therefore, you would use the command COPY/BOOT=DX to

create the bootable RX01 system diskette. Do not specify /BOOT=nn for Version 3.0 RT-11 and earlier systems; instead, choose the monitor file DYMNxx.SYS or DXMNxx.SYS as the source file.

### ***/[NO]LOG***

Controls whether the EXCHANGE command COPY displays the file specifications of each file copied. If you specify /LOG, the system displays the following data for each copy operation: the file specifications of the input and output files, and the number of blocks or the number of records copied (depending on whether the file is copied on a block-by-block or record-by-record basis). The default is /NOLOG.

## **file qualifiers**

### ***/ALLOCATION=n***

Forces the initial allocation of the output file to the number of 512-byte blocks that you specified as n. The /ALLOCATION qualifier is valid only for Files-11 and RT-11 output files.

By default, COPY determines the initial allocation of the output file by the size of the input file. Typically, /ALLOCATION is needed only when you are creating a contiguous file on Files-11 (using /BEST\_TRY\_CONTIGUOUS or /CONTIGUOUS), when the input file is on magnetic tape, or when you want additional space at the end of the file.

If you specify /ALLOCATION, the file's allocated size does not change, unless you also specify /TRUNCATE. When you are unsure of the output size, you might want to specify both /ALLOCATION and /TRUNCATE.

### ***/[NO]BEST\_TRY\_CONTIGUOUS***

Indicates whether the Files-11 output file is to be allocated contiguously on a "best effort" basis; that is, whether EXCHANGE will attempt to place the file on consecutive physical disk blocks. If insufficient contiguous space is available, the file occupies the largest available contiguous space plus additional extents as necessary for the rest of the allocation. You can apply this qualifier only to a Files-11 output file.

The /BEST\_TRY\_CONTIGUOUS qualifier has no effect when you copy files to magnetic tape volumes. When you would like a file from a magnetic tape to be copied contiguously, use both the /ALLOCATION and the /BEST\_TRY\_CONTIGUOUS qualifiers, because the size of the file on magnetic tape cannot be determined until after it is copied to the disk. If you do not know the exact size of the file, you can overestimate the size and specify /TRUNCATE (along with /ALLOCATION and /BEST\_TRY\_CONTIGUOUS) to avoid wasted space.

The default is /NOBEST\_TRY\_CONTIGUOUS.

## EXCH-6 EXCHANGE COPY

### ***/CARRIAGE\_CONTROL=option***

Defines the carriage control attributes of a file, as well as other attributes of the records. The carriage control options are: **CARRIAGE\_RETURN**, which implies carriage return/line-feed control; **FORTTRAN**, which indicates that the first character of each record is to be interpreted as the carriage control specifier; and **NONE**, which indicates that carriage control is not implied.

The default is **/CARRIAGE\_CONTROL=CARRIAGE\_RETURN**.

### ***/[NO]CONTIGUOUS***

Indicates whether the copied file is to be contiguous; that is, stored on consecutive physical blocks on an output disk volume. The **/CONTIGUOUS** qualifier is valid only for Files-11 output files.

The **/CONTIGUOUS** qualifier has no effect when you copy files to magnetic tape volumes. When you would like a file from a magnetic tape to be copied contiguously, use both the **/ALLOCATION** and **/CONTIGUOUS** qualifiers because the size of the file on magnetic tape cannot be determined until after it is copied to the disk. If you do not know the exact size of the file, you can overestimate the size and specify the **/TRUNCATE** qualifier (along with **/ALLOCATION** and **/CONTIGUOUS**) to avoid wasted space.

The default is **/NOCONTIGUOUS**.

### ***/[NO]DELETE***

Controls whether **COPY** deletes existing files of the same name during the copy operation. This qualifier is valid for RT-11 output only; it is equivalent to the RT-11 **COPY** command qualifier **/REPLACE**. In fact, you can use the **EXCHANGE COPY** command qualifier **/REPLACE** to control file deletion, although its function differs from that of **/DELETE** (see the description of the **/REPLACE** qualifier for details on its function).

If you want a message displayed when you delete a file, include the **/LOG** qualifier in your command. To prevent automatic file deletion, use **/NODELETE**.

The default is **/DELETE**. Files with the same name as the output file name are deleted *after* the new file has been copied.

### ***/EXTENSION=n***

Specifies the number of blocks to be added to the output file each time the file is extended. This qualifier is valid for Files-11 output files only.

**EXCHANGE** determines the default extension according to the following hierarchy:

1. An explicit value specified on the **/EXTENSION** qualifier

2. The current process default extension value set by the command SET RMS\_DEFAULT
3. The current system default extension value set at system generation or with the SET RMS\_DEFAULT/SYSTEM command

Use the /EXTENSION qualifier to set an extension quantity with magnetic tape input; EXCHANGE preallocates a file of the correct size when the input is on a directory-structured-device.

### **/[NO]PROTECT**

Determines whether protection is set for an RT-11 output file. The owner UIC of the output file is the UIC of the current process. This qualifier is not valid for Files-11 or DOS-11 output files. Protection attributes for Files-11 output are taken from the current process default protection.

EXCHANGE does not attempt to transfer protection attributes from the input file to the output file, because protection mechanisms of various operating systems do not readily translate to one another.

The default is /NOPROTECT.

### **/RECORD\_FORMAT=(option[, . . . ])**

Defines the internal record structure of a file, as well as other attributes of the records.

### **/[NO]REPLACE**

Requests that if an RT-11 output file already exists with the same file specification as that entered for the output file, the existing file is to be deleted *before* the copy proceeds. COPY allocates new space for the output file. The /REPLACE qualifier is valid for RT-11 output only; it is equivalent to the RT-11 COPY command qualifier /PREDELETE.

By default, COPY creates the new file first and then, after the copy operation is done, deletes the previous file. However, when you use /REPLACE, COPY deletes the previous file *before* it copies the new file. This can be a problem if the input file has been corrupted because the previous version of the file will have been deleted. Therefore, you should use /REPLACE only when there is insufficient room for two copies of the file.

### **/[NO]REWIND**

Determines whether a DOS-11 input magnetic tape reel logically rewinds to the beginning-of-tape mark (BOT) before EXCHANGE searches for the file name specified in the input specifier. This qualifier is valid for DOS-11 magnetic tape only. The default is /NOREWIND.

Use the /REWIND qualifier when you want COPY to search for a file from the logical beginning of the magnetic tape, instead of from the current physical position of the tape.

## EXCH-8 EXCHANGE COPY

### ***/START\_BLOCK=[n]***

For RT-11 volumes, specifies the logical block number where the file is to be placed. This qualifier is especially useful with TU58 tape cassettes, because performance can be significantly enhanced by careful placement of files.

### ***/[NO]SYSTEM***

Controls whether the COPY command copies files that have the file type SYS. Files with a file type of SYS are usually necessary for the operation of an RT-11 system. Only RT-11 volumes handle SYS files in this manner.

The default is /NOSYSTEM; the COPY command does not copy an RT-11 file with the type SYS, whether matched by a wildcard specification or explicitly named. EXCHANGE displays a message whenever it skips over a SYS file during a copy operation.

### ***/TRANSFER\_MODE=option***

Specifies the I/O method to be used in a transfer. This qualifier is useful for all volume formats.

<b>Option</b>	<b>Function</b>
AUTO	Select BLOCK transfer for efficiency if possible
BLOCK	Transfer block by block without looking at records
RECORD	Transfer record by record

The default is the AUTOMATIC transfer mode. In AUTOMATIC mode, EXCHANGE attempts to use a BLOCK transfer whenever possible. BLOCK transfers are possible between RT-11 volumes or between RT-11 and DOS-11 volumes, since the internal file structures are identical. AUTOMATIC does not use the BLOCK transfer if either file specification contains a /RECORD\_FORMAT qualifier.

A BLOCK transfer moves data between devices. Since no interpretation is done on the data, BLOCK transfers are more efficient than RECORD transfers. The block sizes on both devices must be identical. Both input and output must be in BLOCK format. Specifying BLOCK on one parameter implies BLOCK for the other file or device specification.

A BLOCK transfer produces an exact copy of the file. If the output device is Files-11, the file will be a sequential file with fixed-length 512-byte records. This feature is used primarily to avoid any interpretation of the data during the transfer. If the Files-11 file is a sequential file with 512-byte fixed-length records, there is no difference between a /TRANSFER\_MODE=BLOCK transfer and a /RECORD=FIXED=512 transfer.

A RECORD transfer moves the data record by record. A RECORD transfer requires more time than a BLOCK transfer, but it must be used if the input and output record structures differ.

When the /LOG qualifier is used in a COPY command, EXCHANGE displays the size of the file that was transferred. If BLOCK mode was used, the message gives the file size as the number of blocks transferred. If RECORD mode was used, the message displays the number of records.

### ***/[NO]TRUNCATE***

Controls whether COPY truncates an output file at the end-of-file when copying it. The default is /NOTRUNCATE; COPY uses the allocation of the input file to determine the size of the output file.

### ***/VOLUME\_FORMAT=option***

Defines the physical format of the volume to be processed. The default format qualifier is dependent on the device type.

If used, volume format qualifiers must be attached to one or both of the file specification parameters; you cannot attach them directly to the command. A volume format qualifier determines the format of the file name and directory specifications, and often implies certain defaults.

## **description**

COPY transfers a file or files from an input volume to an output volume.

You can create multiple output files by specifying multiple input files. When multiple output files are created, the corresponding field from each input file is used in the output file name.

If you do not specify a version number for Files-11 output, COPY applies a version number as follows:

- The same version number as that of the input file, if the input volume structure supports version numbers and no file exists with the same name and type
- A version number that is one greater than the highest version number of an existing file with the same file name and file type
- Version 1 if neither of the above applies

If you use an asterisk (\*) wildcard character to specify the output file version number, COPY uses the version numbers of the associated input files (if any) as the version numbers of the output files.

Note that ANSI-formatted magnetic tapes do not handle version numbers in the same manner as disks.

EXCHANGE might reformat files during the copy operation. The defaults for reformatting are dependent on the record and volume format qualifiers that are attached to both the input and output file specifications, as well as the type fields of the file specifications.

## EXCH-10 EXCHANGE COPY

The COPY command does not copy a file with the SYS type unless you specify the /SYSTEM qualifier. EXCHANGE displays a message if it passes over one or more SYS files during a copy operation.

EXCHANGE does not copy files with the type BAD if the file specification contains wildcards. EXCHANGE does not display a message when it passes over one or more BAD files during a copy operation. Therefore, to copy a file with the type BAD, specify the file name explicitly instead of using wildcards.

### example

```
EXCHANGE> COPY TEST.DAT DYAO:NEWTST.DAT/VOLUME=RT11
```

The command in this example copies the contents of the file TEST.DAT from the default disk and directory into a file named NEWTST.DAT on an RT-11 diskette (mounted on DYAO). If a file named NEWTST.DAT already exists, the COPY command replaces it. The record formats are variable length on the Files-11 input and ASCII stream on the RT-11 output.

---

## DELETE

Deletes one or more files from a foreign block-addressable mass storage volume. EXCHANGE does not delete files from Files-11 volumes; the DELETE command is supported only on RT-11 volumes.

### format

```
DELETE file-spec[, ... ]
```

### parameters

*file-spec[, ... ]*

Specifies the names of one or more files to be deleted. You can specify wildcard characters in any of the file specification fields.

To delete more than one file, separate the file specifications with commas or plus signs.

The DELETE command does not delete a file with the SYS type unless you specify the /SYSTEM qualifier. EXCHANGE displays a message if it passes over one or more SYS files during a delete operation.

EXCHANGE does not delete files with the type BAD if the file specification contains wildcards. When this happens, you will not receive a warning. Therefore, to delete files with the type BAD, enter their file specifications explicitly.



## qualifiers

### ***/[NO]LOG***

Controls whether the DELETE command displays the file specification of each deleted file. The default is /NOLOG.

### ***/[NO]SYSTEM***

Controls whether the DELETE command deletes files with the file type SYS. Files with the type SYS are usually necessary for the operation of an RT-11 system. Only RT-11 volumes handle .SYS files in this manner.

The default is /NOSYSTEM; the DELETE command does not delete an RT-11 file with the SYS type, whether matched by a wildcard specification or explicitly named. EXCHANGE displays a message whenever it skips a SYS file during a delete operation.

### ***/VOLUME\_FORMAT=option***

Defines the physical format of the volume to be processed. RT-11 volumes are the only volumes on which DELETE is currently supported.

## example

```
EXCHANGE> DELETE DMA0:COMMON.SUM/VOLUME=RT11
```

The command in this example deletes the file COMMON.SUM from the RT-11 device DMA0.

---

## DIRECTORY

Provides a list of files or information about a file or group of files. The files must reside on a foreign volume; EXCHANGE does not list directories of Files-11 volumes.

## format

```
DIRECTORY [file-spec, . . . ]
```

## parameters

### ***file-spec*, . . . ]**

Specifies one or more files to be listed. The /VOLUME\_FORMAT qualifier determines the syntax of a file specification.

To specify more than one file, separate the file specifications with either commas or plus signs. You can use wildcard characters in the directory specification, file name, file type, or version number fields of a file specification.

# EXCH-12 EXCHANGE DIRECTORY

## qualifiers

### ***/[NO]ALL***

Lists all deleted or unused files on an RT-11 volume, in addition to other files selected by the command.

### ***/[NO]BADBLOCKS***

Scans the volume to find any blocks that return read errors. The data on the volume is not modified. If a bad block replacement table is present, the contents of the table are displayed. This is valid for RT-11 volumes only.

### ***/[NO]BLOCKS***

Lists the starting block number of the file. This qualifier is valid only for directories of RT-11 devices. The first block of the device is block number 0. The default is /NOBLOCKS.

### ***/[NO]BRIEF***

Includes only the file name of each file to be listed. Specifying the /BRIEF qualifier is equivalent to specifying /NODATE/NOSIZE. The default is /BRIEF.

### ***/COLUMNS=n***

Lists the files, using the specified number of columns on each line of the display. This qualifier is used in conjunction with the /BRIEF qualifier (either explicitly or by default). The default number of columns is dependent on the volume format and the information requested. The DIRECTORY command attempts to use as many columns as possible. If you request too many columns, DIRECTORY displays a message and reduces the number of columns to the number that fit on the listing.

### ***/[NO]DATE***

Includes the date for each file listed. If you omit this qualifier, the default is /DATE.

### ***/[NO]DELETED***

Lists a directory of files that have been deleted from an RT-11 device, but whose file name information has not been destroyed. The listing includes the file names, types, sizes, creation dates, and starting block numbers (in decimal, unless you also specify the /OCTAL qualifier) of the deleted files. The /DELETED qualifier is valid only with block-addressable volumes in RT-11 format. The default is /NODELETED.

### ***/[NO]FREE***

Includes unused areas in the directory listing. The /FREE qualifier is valid only with RT-11 formatted volumes.

***/FULL***

Lists all the available information for each file. The format of the listing depends on the format of the volume. The */FULL* qualifier overrides the default brief listing format.

***/[NO]OCTAL***

Controls whether numeric information is displayed in decimal or octal format. The default is */NOOCTAL*; numbers are displayed in decimal radix. Dates are always displayed in decimal format.

***/OUTPUT[=file-spec]***

Writes the *DIRECTORY* output to a specified file, rather than to the current *SYS\$OUTPUT* device. If you specify the */OUTPUT* qualifier without a file specification, the output is directed to *SYS\$OUTPUT*. If you omit the file type in the file specification, the default file type is *LIS*. If you specify a file type and omit the file name, the default file name is *EXCHDIRE*. No wildcard characters are allowed in the file specification.

***/OWNER***

Displays information about the owner of a volume and the files on the volume. For *RT-11*, the volume owner is shown. For *DOS-11*, the *UIC* of the file owner is shown.

***/PRINTER***

Queues the command output for printing under the name specified by the */OUTPUT* qualifier. If you specify */PRINTER* without the */OUTPUT* qualifier, the output is directed to a file named *EXCHDIRE.LIS*, which is spooled for printing and then deleted.

***/[NO]SIZE***

Displays the file size in blocks for each file listed. The default is */SIZE*.

***/[NO]SUMMARY***

Lists a summary of the usage of the directory segments for an *RT-11* volume. If a bad block replacement table is present, the contents of the table are displayed.

***/VOLUME\_FORMAT=option***

Defines the physical format of the volume to be processed. The default format is dependent on the device type.

The *EXCHANGE* command *DIRECTORY* is not valid for *Files-11* devices.

# EXCH-14 EXCHANGE DIRECTORY

## description

The output of the DIRECTORY command depends on the volume format and on certain formatting qualifiers and defaults. The following are the formatting qualifiers:

/ALL	/BLOCKS	/BRIEF
/COLUMNS	/DATE	/FULL
/OCTAL	/OWNER	/SIZE

The files that are listed always appear in the order in which they appear in the volume directory or the order in which they reside on a magnetic tape.

## example

```
EXCHANGE> DIRECTORY DLA2:.OBJ/VOLUME=RT11/FULL
```

The command in this example lists all files with the type OBJ on the RT-11 volume mounted on DLA2. The /FULL qualifier causes the file sizes and dates to be listed along with the names.

---

## DISMOUNT

Releases a volume previously accessed by the EXCHANGE command MOUNT.

## format

```
DISMOUNT device-name[:]
```

## parameters

***device-name[:]***

Specifies the name of the device to be dismounted. You can specify a physical device name or a logical name assigned to a physical device name. If you omit a controller designation or a unit number, the defaults are controller A and unit 0, respectively. You can also specify the name of a virtual device.

## qualifiers

***/[NO]MESSAGE***

Controls whether or not EXCHANGE displays a message that the volume was dismounted. The default is determined by the /MESSAGE qualifier on the EXCHANGE command when EXCHANGE was activated.

## description

The DISMOUNT command closes all connections that EXCHANGE maintains to the device. This command does not affect the state of the operating system mount; the device remains accessible to VMS. If you do not use the DISMOUNT command, an implicit DISMOUNT is automatically executed when you exit EXCHANGE.

The DISMOUNT command is valid only with foreign devices.

## example

```
EXCHANGE> MOUNT/FOREIGN MTA0:  
EXCHANGE> COPY MTA0:AVERAGE.FOR/VOLUME=DOS11 *  
EXCHANGE> DISMOUNT MTA0:
```

The first command in this example mounts the tape on the device MTA0. The second command in this example transfers a file from the magnetic tape to the current default directory. The last command releases EXCHANGE's access to the volume; however, the volume is still mounted on the operating system and is accessible to VMS.

---

## EXIT

Terminates execution of EXCHANGE. Control is returned to the DCL command level. You can also use CTRL/Z to exit EXCHANGE.

## format

EXIT

---

## HELP

Displays information about EXCHANGE commands and qualifiers.

## format

HELP [*command* [*qualifier* [*option* [*option*]]]]

## parameters

### *command*

Specifies the name of the EXCHANGE command that you want information about. If you omit the command, HELP displays general information listing all commands recognized by EXCHANGE.

### *qualifier*

Gives the name of the qualifier to be explained.

### *option*

Gives the name of the option to be explained.

## EXCH-16 EXCHANGE HELP

### description

For an overview of EXCHANGE and a listing of the EXCHANGE command names, enter the HELP command with no arguments.

If you enter HELP and the name of an EXCHANGE command, HELP displays a description of the command followed by a list of related qualifiers. For information on any of the related qualifiers, enter the qualifier name at the prompt.

You can also obtain information on any EXCHANGE command qualifier by entering HELP, the command, and the qualifier at the EXCHANGE prompt, as follows:

```
EXCHANGE> HELP COPY/CONTIGUOUS
```

For information on a qualifier with options, enter HELP, the command, the qualifier, and the option at the EXCHANGE prompt.

If you specify an asterisk (\*) in place of any keyword, the HELP command displays all information available at that level.

If you specify an ellipsis ( . . . ) after any keyword, the HELP command displays all information relating to that keyword.

You can specify percent signs and asterisks in the keyword as wildcard characters.

### example

```
EXCHANGE> HELP COPY/VOLUME...
```

The command in this example displays all the help that is available for the COPY qualifier /VOLUME\_FORMAT, including descriptions of each of the options.

---

## INITIALIZE

Formats and writes a label on a foreign mass storage volume. For directory-structured devices, the device directory is also initialized.

### format

```
INITIALIZE device-name [volume-label]  
INITIALIZE/CREATE file-name [volume-label]
```

## parameters

### *device-name*

Specifies the name of the device on which the volume to be initialized is physically mounted.

The device name can also refer to the name of a mounted virtual device to be reinitialized.

### *file-name*

For INITIALIZE/CREATE, file-name refers to the name of a file to be created and initialized as a virtual device.

### *volume-label*

Specifies the identification to be written onto the volume header for RT-11 volumes only. The volume label can contain up to a maximum of 12 alphanumeric characters. The default is *VMS Exchange*. Use quotation marks to specify a volume label with lowercase letters.

## qualifiers

### **/ALLOCATION=*n***

Specifies the allocation of a new virtual device file in terms of 512-byte blocks. The allocation specified is the number you entered as *n*. If you do not specify the /ALLOCATION qualifier when you create a new virtual device file, the default allocation is 494 blocks, the size of a single-density diskette. The maximum allocation is 65,536 blocks.

A virtual device file is usually the size of a standard device supported by both RT-11 and VMS. These sizes are as follows:

Device	Blocks
TU58	512
RX01	494
RX02	494 (single density)
RX50	800
RX02	988 (double density)
RX33	2400
RL02	20480
RK06	27126
RK07	53790

You can also use the /ALLOCATION qualifier to reduce the size of a physical device. For example, if you want to prepare an RL02 disk but have only an RK07 device available, you can initialize the RK07 to a

## EXCH-18    EXCHANGE INITIALIZE

volume of 20,480 blocks. When the RL02 is available, you can transfer the files to the RL02 knowing they will fit on the smaller device.

### ***/BADBLOCKS[=RETAIN]***

Performs a bad block scan of the volume before initialization. A file named FILE.BAD is created on top of each bad block or group of bad blocks encountered on the device, preventing any future use of the bad areas.

If a bad block is found in either the boot block or the volume directory, the volume is not usable and EXCHANGE displays an error message. If the bad block is in a directory segment other than the first, you might be able to use the volume by reinitializing it with a smaller number of segments (see the /SEGMENTS qualifier description).

If you specify /BADBLOCKS=RETAIN, EXCHANGE uses the device's existing bad block information, instead of performing a bad block scan. Therefore, initializing takes less time. If you do not specify RETAIN, EXCHANGE writes a pattern on each block of the volume, then reads each block to verify that the block is usable. EXCHANGE prints a list of the bad blocks found on the device.

RK06, RK07, and RL02 disk volumes support bad block replacement. Therefore, Digital recommends that you use the /REPLACE=RETAIN qualifier for these volumes. If you use the /BADBLOCKS qualifier with a volume initialized previously with the /REPLACE qualifier, EXCHANGE deletes the bad block replacement table and performs a new bad block scan. If you use /BADBLOCKS=RETAIN with such a volume, EXCHANGE uses the FILE.BAD files created during the volume initialization.

### ***/CREATE***

Specifies that a virtual device is to be created and initialized. The specification is a file name; if a file type is not given, EXCHANGE applies the default type of DSK.

### ***/DENSITY=density-value***

Specifies, for magnetic tape volumes, the density in bytes per inch (bpi) at which the tape is to be written.

For magnetic tape volumes, the density value specified can be 800 or 1600, as long as the density is supported by the magnetic tape drive. If you do not specify a density value for a blank tape, the system uses a default of the lowest density supported by the tape drive.

For the RX02 dual-density diskette drive, use the DCL command INITIALIZE/DENSITY=SINGLE or INITIALIZE/DENSITY=DOUBLE to reformat the diskettes to a different density; then use the EXCHANGE command INITIALIZE to create the RT-11 directory structure.



**NOTE:** Diskettes formatted in double density cannot be read or written by the console block storage device (an RX01 drive) of a VAX-11/780 until they have been reformatted in single density.

***/EXTRA\_WORDS=n***

Specifies, for RT-11 volumes, the number of extra words to add to each directory entry, in addition to the required seven words. The ability to increase the length of directory entries is useful for some RT-11 applications. Increasing the size of the directory entries reduces the number of entries that fit in each directory segment.

***/[NO]MESSAGE***

Controls whether or not EXCHANGE displays a message that the volume was initialized. The default is determined by the /MESSAGE qualifier entered with the EXCHANGE command when EXCHANGE was activated.

***/REPLACE=RETAIN***

Retains, when an RT-11 volume is initialized, the bad block replacement table and any existing FILE.BAD files.

The RETAIN option is required; EXCHANGE cannot build a replacement table for a volume. The RT-11 system builds and uses the table based on specific hardware error conditions. The VMS I/O system is different, and cannot be relied upon to generate exactly the same error conditions. Therefore, it is not possible for EXCHANGE to generate the same replacement table that would be generated by RT-11.

If no replacement table is present, the qualifier /REPLACE=RETAIN is equivalent to /BADBLOCKS=RETAIN.

***/SEGMENTS=n***

Defines, for RT-11 volumes, the number of 2-block directory segments to allocate for the directory. The number of segments in the directory establishes the number of files that can be stored on a device. The system allows a maximum of 72 files per directory segment and 31 directory segments per device. The argument *n* represents the number of segments; the valid range for *n* is from 1 to 31 (decimal). The default values for *n* depend on the device type, as follows:

Device	Segments
TU58	1
RX01	1
RX02	1 (single density)

## EXCH-20 EXCHANGE INITIALIZE

---

Device	Segments
RX02	4 (double density)
RX50	4
RX33	16
RL02	16
RK06	16
RK07	31

---

### ***/VOLUME\_FORMAT=option***

Defines the physical format of the volume to be processed.

The EXCHANGE command INITIALIZE is not valid for Files-11 devices.

### **description**

The EXCHANGE command INITIALIZE erases all files from a volume. After initialization, the volume directory contains no files. DOS-11 magnetic tapes and RT-11 block-addressable devices can be initialized.

The device must be mounted with the /FOREIGN qualifier.

### **example**

```
$ MOUNT/FOREIGN DLA2:
%MOUNT-I-MOUNTED,           mounted on DLA2
$ EXCHANGE
EXCHANGE> INITIALIZE DLA2:
%EXCHANGE-S-INITIALIZED, the RT--11 volume _DLA2: has been initialized
```

The command in this example initializes the volume mounted on the RL02 drive DLA2. Since DLA2 is a block-addressable device mounted with the /FOREIGN qualifier, RT-11 is the default format. EXCHANGE physically scans all blocks of the volume, builds a bad block replacement table, and displays a message indicating that it failed to turn up any bad blocks.

---

## **MOUNT**

Makes a foreign volume and the files or data it contains available for processing by EXCHANGE. The EXCHANGE command MOUNT enters the device into internal tables maintained by EXCHANGE.

### **format**

**MOUNT** *device-name*

**MOUNT/VIRTUAL** *device-name file-name*

## parameters

### *device-name*

Specifies the physical device name or logical name of the device on which the volume is to be mounted. For MOUNT/VIRTUAL, the device-name parameter supplies a name for the virtual device.

### *file-name*

For MOUNT/VIRTUAL only, the file-name parameter gives the name of the file containing the image of the foreign volume.

## qualifiers

### ***/[NO]DATA\_CHECK[=(READ,WRITE)]***

Determines whether EXCHANGE performs a second operation after every I/O operation to verify that the data was correctly transferred. If you specify /DATA\_CHECK=WRITE, after every write operation EXCHANGE rereads the data that was just written and compares it with the original data. If you specify /DATA\_CHECK=READ, EXCHANGE reads each block of data twice and verifies that both read operations received identical data.

It is usually more efficient to use the /DATA\_CHECK option on the DCL command MOUNT than to use the option on the EXCHANGE command MOUNT. If you mount a device with the DCL command MOUNT/FOREIGN/DATA\_CHECK, VMS can use features in the device hardware and device driver to perform the redundant I/O operations.

The RX01 and RX02 diskette drives do not contain the necessary features for the operating system to perform data checking. If you use the DCL command MOUNT/DATA\_CHECK with a diskette, the system is unable to perform data checking (no warning message is displayed). EXCHANGE is able to recognize, however, that a diskette was mounted with the data checking option; in this case, EXCHANGE performs the software data checking internally, even if you have not specified an explicit MOUNT/DATA\_CHECK command.

If you specify the /DATA\_CHECK qualifier without an option, the default is /DATA\_CHECK=WRITE.

### ***/FOREIGN***

Indicates that the volume is not in the standard format used by the VMS operating system; that is, a magnetic tape volume is not in the standard ANSI format, or a disk volume is not in Files-11 format. The EXCHANGE command MOUNT mounts only foreign volumes. The /FOREIGN qualifier is the default. You must use the DCL command MOUNT to mount VMS volumes.

## EXCH-22    EXCHANGE           MOUNT

The default protection applied to foreign volumes is RWLP (Read, Write, Logical I/O, Physical I/O) for the system and owner. If you mount a volume currently in Files-11 format with the /FOREIGN qualifier, you must have the user privilege to override volume protection (VOLPRO), or your UIC must match the UIC on the volume.

### ***/[NO]MESSAGE***

Controls whether EXCHANGE displays a message indicating that the volume was mounted. The default is determined by the /MESSAGE qualifier specified with the EXCHANGE command when EXCHANGE was invoked.

### ***/VIRTUAL***

Mounts a Files-11 file as a virtual device. When you specify /VIRTUAL, the MOUNT command requires two parameters. The first parameter is a device name assigned as the name of the virtual device. The second parameter is the name of the Files-11 file that is the image of a foreign volume.

### ***/VOLUME\_FORMAT=option***

Defines the physical format of the volume to be processed.

### ***/[NO]WRITE***

Controls whether the volume can be written. You can specify /NOWRITE to protect files by providing read-only access. Specifying /NOWRITE is equivalent to write-locking the device.

The default is /WRITE. If /WRITE is specified (either explicitly or by default) and the volume itself is write-locked, EXCHANGE displays a message to inform you that the volume is write-locked.

## **description**

The EXCHANGE command MOUNT enters the description of the foreign volume in internal tables maintained by EXCHANGE. This command is different from the DCL command MOUNT, which enters the device in tables maintained by the VMS operating system.

A virtual volume must be explicitly mounted with the MOUNT/VIRTUAL command.

If an EXCHANGE command is given on an unmounted foreign volume, EXCHANGE attempts to execute an implied MOUNT/FOREIGN/WRITE-/NODATACHECK on the device. This feature enables EXCHANGE to operate in the single-command DCL mode.

If a MOUNT/FOREIGN (either implied or explicit) command is given for a foreign device that has not been mounted on the VMS system, EXCHANGE issues the equivalent of the DCL command MOUNT/FOREIGN and attempts to make the volume known to the

operating system. Any volume mounted in this way remains mounted after EXCHANGE exits.

When EXCHANGE issues the MOUNT/FOREIGN command, the system checks the following:

- That the device has not been allocated to another user
- That a volume is physically loaded on the specified device
- For magnetic tapes, the volume accessibility field of the VOL1 label

### example

```
EXCHANGE> MOUNT MT:
%EXCHANGE-I-MOUNTED, MATH06 mounted on _MTA0:
```

The command in this example requests that the magnetic tape loaded on the device MTA0 be mounted as a foreign volume. The tape label is displayed, since the tape has been previously initialized as an ANSI-labeled tape with the label MATH06. This tape cannot be accessed as a Files-11 tape; it should be reinitialized as a DOS-11 tape during the current EXCHANGE session.

## RENAME

Changes the file specification of an existing file on an RT-11 volume.

### format

```
RENAME input-file-spec output-file-spec
```

### parameters

#### *input-file-spec*

Specifies the names of one or more files whose specifications are to be changed.

You can use wildcard characters in the file name and file type specification; if you do, all files that satisfy the specified fields are renamed.

#### *output-file-spec*

Provides the new file specification to be applied to the input file. The RENAME command uses the file name and file type of the input file specification to provide defaults for nonspecified fields in the output file.

You can specify an asterisk (\*) in place of the file name or file type of the output file; the RENAME command uses the corresponding field in the input file specification to name the output file. Specifying wildcard characters in corresponding fields of the input and output file specifications results in multiple rename operations.

## EXCH-24 EXCHANGE RENAME

You can omit the device name from the output specification. EXCHANGE uses the device name specified for the input, since it is not possible to rename a file from one device to another.

### qualifiers

#### ***/[NO]LOG***

Controls whether the RENAME command displays the file specification of each file that it renames. The default is /NOLOG.

#### ***/[NO]PROTECT***

Determines whether protection is set for an RT-11 output file. The default is /NOPROTECT.

This qualifier is not valid for Files-11 or DOS-11 output files. Protection attributes for Files-11 output are taken from the current process default protection.

EXCHANGE does not attempt to transfer protection attributes from the input file to the output file. Protection mechanisms of various operating systems do not readily translate to one another.

The owner UIC of the output file is the UIC of the current process.

#### ***/[NO]SYSTEM***

Controls whether the RENAME command renames files that have the file type SYS. These files are usually files necessary for the operation of an RT-11 system. Only RT-11 volumes handle SYS files in this manner.

The default is /NOSYSTEM; the RENAME command does not rename an RT-11 file with the type SYS, whether it is matched by a wildcard specification or is named explicitly. EXCHANGE displays a message when it skips an SYS file during a rename operation.

EXCHANGE handles files with the file type BAD in a similar manner; that is, the rename operation skips BAD files. However, EXCHANGE does not warn that BAD files are being skipped, and the /SYSTEM qualifier has no effect on BAD files. To rename a file with the type BAD, specify the file explicitly instead of using wildcards.

#### ***/VOLUME\_FORMAT=option***

Defines the physical format of the volume to be processed. EXCHANGE supports the RENAME command on RT-11 volumes only.

### example

```
EXCHANGE> RENAME DMA0:AVERAG.OBJ MEAN
```

The command in this example changes the file name of the file AVERAG.OBJ to MEAN.OBJ.

---

**SHOW**

Displays the devices currently mounted by EXCHANGE.

**format****SHOW****example**

```
EXCHANGE> MOUNT DBA0:
%EXCHANGE-I-VMSMOUNT, a "$ MOUNT /FOREIGN DBA0:" command was done by Exchange
%EXCHANGE-S-MOUNTED, the RT--11 volume _DBA0: has been mounted
EXCHANGE> MOUNT DLA2:
%EXCHANGE-I-VMSMOUNT, a "$ MOUNT /FOREIGN DLA2:" command was done by Exchange
%EXCHANGE-S-MOUNTED, the RT--11 volume _DLA2: has been mounted
EXCHANGE> INITIALIZE/CREATE WRKD:[USER]VIRT.DSK
%EXCHANGE-S-INITIALIZED, the RT--11 volume WRKD:[USER]VIRT.DSK;1 has been
initialized
EXCHANGE> MOUNT/VIRTUAL DISK: VIRT.DSK
%EXCHANGE-S-MOUNTVER, the RT--11 volume DISK: has been mounted
      using the file WRKD:[USER]VIRT.DSK;1
```

```
EXCHANGE> SHOW
```

```
Mounted volumes:
  volume format:      RT--11
  volume class:      disk (virtual volume)
  virtual file name:  WRKD:[USER]VIRT.DSK;1
  volume size:       494 blocks

  _DLA2:
    volume format:    RT--11
    volume class:     disk
    physical device name:  _DLA2:
    volume size:      20480 blocks

  _DBA0:
    volume format:    RT--11
    volume class:     disk
    physical device name:  _DBA0:
    volume size:      65535 blocks
```

```
EXCHANGE>
```

The MOUNT commands in this example mount foreign devices on drives DBA0 and DLA2. The SHOW command displays all devices currently mounted by EXCHANGE.

## TYPE

Displays the contents of a file or group of files on the current output device.

### format

**TYPE** *file-spec[, ... ]*

### parameters

***file-spec[, ... ]***

Specifies the names of one or more input files to be copied. If you specify more than one input file, separate them with either commas or plus signs. You can specify standard VMS wildcards in file names, both Files-11 and foreign. You can use wildcard directories with Files-11 and DOS-11 input.

The syntax for the file names is dependent on the particular volume format option present or implied.

### qualifiers

***/[NO]LOG***

Controls whether TYPE displays the file specifications of each file displayed.

If you specify /LOG, the TYPE command displays the following for each copy operation:

- File specifications of the input and output files
- Number of blocks or the number of records copied (depending on whether the file is copied on a block-by-block or record-by-record basis)

***/RECORD\_FORMAT=(option[, ... ])***

Defines the internal record structure of a file and other attributes of the records.

***/[NO]REWIND***

Controls whether the DOS-11 input magnetic tape reel logically rewinds to the beginning-of-tape mark before EXCHANGE searches for the file name given in the input specifier.

Use this qualifier only for DOS-11 magnetic tape devices. The default is /NOREWIND; you should use /REWIND when you want TYPE to start searching for a file at the beginning of the magnetic tape rather than at the current position.



***/VOLUME\_FORMAT=option***

Defines the physical format of the volume to be processed. The default format qualifier is dependent on the device type.

**example**

```
EXCHANGE> TYPE DYAO:BEAM.RAT/VOLUME=RT11/RECORD=STREAM
```

The command in this example copies the RT-11 file to the current SYS\$OUTPUT device. The two qualifiers are actually the defaults if DYAO was mounted as a foreign volume.



---

## Install Utility

Use the Install Utility (INSTALL) to enhance the performance of selected executable and shareable images, to assign enhanced privileges to images, and to support user-written system services. The system stores the name and attributes of installed images on known file lists.

### format

**INSTALL** [*command*]

### parameter

#### *command*

Specifies an INSTALL command. This parameter is optional. If no command is specified, the utility displays its prompt and waits for command input.

### usage summary

To invoke INSTALL, enter the DCL command INSTALL at the DCL prompt as follows:

```
$ INSTALL
```

The utility responds with the following prompt:

```
INSTALL>
```

You can then perform INSTALL operations by entering the appropriate INSTALL commands. Alternatively, you can enter a single INSTALL command on the same line as the command that invokes the utility, for example:

```
$ INSTALL LIST/FULL SYS$SYSTEM:LOGINOUT
```

To exit from the Install Utility, enter the EXIT command at the INSTALL> prompt or press CTRL/Z. Either method returns control to the DCL command level.

**The Install Utility requires that you have the CMKRNL privilege to invoke it. It requires the SYSGBL privilege to create system global sections and the PRMGBL privilege to create permanent global sections.**

## INS-2 INSTALL ADD

### INSTALL Commands

This section describes the INSTALL commands and provides examples of their use.

---

#### ADD

Installs the specified image file as a known image.

#### format

**ADD** *file-spec*

#### parameter

##### ***file-spec***

Names the file specification of an image to be installed as a known image. The file specification must name an existing executable or shareable image. If you omit the device and directory specification, the default SYS\$SYSTEM is used. The default file type is EXE.

The highest existing version of the file is used by default. However, you can specify another version of the file as the known version of the image. Even if other versions of the file exist, the version that you specify will be the version that satisfies all known file lookups for the image.

#### qualifiers

##### ***/[NO]ACCOUNTING***

Allows you to enable image-level accounting for selected images when image accounting is disabled on the system (with the DCL command SET ACCOUNTING/DISABLE=IMAGE). When image accounting is enabled on the system, it logs all images. The /NOACCOUNTING qualifier has no effect.

##### ***/[NO]EXECUTE\_ONLY***

The /EXECUTE\_ONLY qualifier is only meaningful to main programs. It allows the image to activate shareable images to which the user has EXECUTE access but has no READ access. All shareable images referenced by the program must be installed, and VMS RMS uses "trusted" logical names, those created for use in EXEC or KERNEL mode.

You may not specify this qualifier for an executable image linked with the /TRACEBACK qualifier.

##### ***/[NO]HEADER\_RESIDENT***

Installs the file as a known image with a permanently resident header (native mode images only). The image is made permanently open even if /OPEN is not specified.

***/[NO]LOG***

Lists the newly added known file entry along with any associated global sections created by the installation.

***/[NO]OPEN***

Installs the file as a permanently open known image.

***/[NO]PRIVILEGED[=(priv-name[,...])]***

Installs the file as a known image with the privileges specified (executable images only). Then, if the image is not located on the system volume, the image is made permanently open even if /OPEN is not specified.

You can specify one or more of the following privilege names:

ACNT	ALLSPOOL	ALTPRI
BUGCHK	BYPASS	CMEEXEC
CMKRNL	DETACH	DIAGNOSE
EXQUOTA	GROUP	GRPNAM
GRPPRV	LOG_IO	MOUNT
NETMBX	OPER	PFNMAP
PHY_IO	PRMCEB	PRMGBL
PRMMBX	PSWAPM	READALL
SECURITY	SETPRV	SHARE
SHMEM	SYSGBL	SYSLCK
SYSNAM	SYSPRV	TMPMBX
VOLPRO	WORLD	

You may not specify this qualifier for an executable image linked with the /TRACEBACK qualifier.

***/[NO]PROTECTED***

Installs the file as a known image that is protected from user-mode and supervisor-mode write access. You can only write into the image from EXEC or KERNEL mode. The /PROTECTED qualifier together with the /SHARE qualifier are used to implement user-written services, which become privileged shareable images.

***/[NO]PURGE***

Specifies that the image can be removed by a PURGE operation; if you do not specify /PURGE, it can be removed only by a DELETE or REMOVE operation. /NOPURGE is the default form of the qualifier.

***/[NO]SHARED***

Installs the file as a shared known image and causes creation of global sections for the image. The image is made permanently open even if /OPEN is not specified.

## INS-4 INSTALL ADD

### ***/[NO]WRITABLE***

Installs the file as a writable known image as long as you also specify the /SHARED qualifier. The /WRITABLE qualifier is automatically negated if the /NOSHARED qualifier is specified.

### **example**

```
INSTALL> ADD/OPEN/PRIVILEGED=(GROUP,GRPNAM) GRPCOMM
```

The command in this example installs the image file GRPCOMM as a permanently open known image with the privileges GROUP and GRPNAM.

Any process running GRPCOMM receives the GROUP and GRPNAM privileges for the duration of the execution of GRPCOMM. The full name of GRPCOMM is assumed to be SYS\$SYSTEM:GRPCOMM.EXE.

---

## **CREATE**

Installs the specified image file as a known image. The CREATE command is synonymous with the ADD command.

---

## **DELETE**

Deletes a known image.

### **format**

```
DELETE file-spec
```

### **parameter**

*file-spec*

Names the file specification of an image installed as a known image.

### **description**

The DELETE command deletes an entry from the known image file list. The image's entry on the known file list and any global sections created for the image are deleted. The image itself (that is, the image file) remains unaffected. Writable global sections are written back to disk upon their removal as known images.

If a process is accessing global sections when the DELETE command is entered, the global sections are deleted only after the operation initiated by the process completes. However, once the command is entered, no additional processes can access the global sections because they are "marked for deletion."

The DELETE command is identical to the REMOVE command.

## example

```
INSTALL> DELETE WRKDS:[MAIN]STATSHR
```

The command in this example deletes the entry for the image STATSHR from the known file list.

---

## EXIT

Terminates INSTALL and returns control to the DCL command level. You can also exit from INSTALL by pressing CTRL/Z.

## format

**EXIT**

---

## HELP

Displays information about how to use INSTALL.

## format

**HELP** *[command]*

## parameter

### *command*

Specifies the name of a command for which help information is to be displayed. If you omit a command name, a list of commands is displayed, and you are prompted for a command name.

---

## LIST

Displays a description of each specified known image or (if no file is specified) all known images.

## format

**LIST** *[file-spec]*

## parameter

### *file-spec*

Names the file specification of an image installed as a known image. If you omit the file specification, INSTALL displays all known file entries.

# INS-6 INSTALL LIST

## qualifiers

### **/FULL**

Displays a multiline description of the specified known image, including the number of accesses, the number of concurrent accesses, and the number of global sections created. The **/FULL** qualifier with the **/GLOBAL** qualifier shows information on global sections, plus owner and protection codes and access control entries, if set.

### **/GLOBAL**

Lists global sections for any specified shared image, or if you omit the file specification, lists all global sections.

### **/STRUCTURE**

Lists addresses of known file entry data structures.

### **/SUMMARY**

Used with the **/GLOBAL** qualifier; displays a summary of global section and global page usage on the system, for local and shared memory global sections.

## description

You can use the **LIST** command with the **/FULL** qualifier to display information that is useful in “tuning” the known file database. For example, a high entry access count for an image may indicate that system performance could benefit if the image were installed **/OPEN**. Similarly, high entry access counts for an image may indicate that installing the image **/SHARED**—that is, with global sections—could improve performance.

## example

```
INSTALL> LIST/FULL LOGINOUT
```

The command in this example displays a multiline description of the known image **LOGINOUT**.

```
DISK$VAXVMSRL5:<SYSO.SYSEXE>.EXE
```

```
LOGINOUT;3      Open Hdr   Shar Priv
  Entry access count      = 44 ①
  Current / Maximum shared = 3 / 5 ②
  Global section count    = 2 ③
  Privileges = CMKRNL SYSNAM TMPMBX EXQUOTA SYSPRV ④
```

- ① Number of times known file entry has been accessed by this node since it became known.
- ② The first number indicates the current count of concurrent accesses of the known file. The second number indicates the highest count of concurrent accesses of the file since it was installed. This number appears only if the image is installed with the **/OPEN** qualifier.



- ③ Number of global sections created for the known file; appears only if the image is installed with the /SHARED qualifier.
- ④ Translation of the privilege mask; appears only if the image is installed with privileges.

---

## PURGE

Deletes all known file entries for images installed without the /NOPURGE qualifier.

### format

**PURGE**

### description

The PURGE command deletes all known file entries for images installed without the /NOPURGE qualifier.

If a process is accessing global sections when the PURGE command is entered, the global sections are deleted only after the operation initiated by the process completes. However, once the command is entered, no additional processes can access the global sections because they are “marked for deletion.”

---

## REMOVE

Removes an entry from the known image file list. The REMOVE command is synonymous with the DELETE command.

---

## REPLACE

Associates a known image with the latest version of the image file, or modifies the attributes of an installed image.

### format

**REPLACE** *file-spec*

### parameter

*file-spec*

Names the file specification of an image installed as a known image.

## qualifiers

### ***/[NO]ACCOUNTING***

Allows you to enable image-level accounting for selected images when image accounting is disabled on the system (with the DCL command SET ACCOUNTING/DISABLE=IMAGE). When image accounting is enabled on the system, it logs all images. The /NOACCOUNTING qualifier has no effect.

### ***/[NO]EXECUTE\_ONLY***

The /EXECUTE\_ONLY qualifier is meaningful only to main programs. It allows the image to activate shareable images to which the user has EXECUTE access but has no READ access. All shareable images referenced by the program must be installed, and VMS RMS uses "trusted" logical names, those created for use in EXEC or KERNEL mode.

You may not specify this qualifier for an executable image linked with the /TRACEBACK qualifier.

### ***/[NO]HEADER\_RESIDENT***

Installs the file as a known image with a permanently resident header (native mode images only). The image is made permanently open even if /OPEN is not specified.

### ***/[NO]LOG***

Lists the newly created known file entry along with any associated global sections created by the installation.

### ***/[NO]OPEN***

Installs the file as a permanently open known image.

### ***/[NO]PRIVILEGED[=(priv-name[,...])]***

Installs the file as a known image with the privileges specified (executable images only). Then, if the image is not located on the system volume, the image is made permanently open even if /OPEN is not specified. For a complete listing of privileges, see the ADD command.

You may not specify this qualifier for an executable image linked with the /TRACEBACK qualifier.

### ***/[NO]PROTECTED***

Installs the file as a known image that is protected from user-mode and supervisor-mode write access. You can only write into the image from EXEC or KERNEL mode. The /PROTECTED qualifier together with the /SHARE qualifier are used to implement user-written services, which become privileged shareable images.

***/[NO]PURGE***

Specifies that the image can be removed by a PURGE operation; if you do not specify /PURGE, it can be removed only by a DELETE or REMOVE operation. (/NOPURGE is the default form of the qualifier.)

***/[NO]SHARED***

Installs the file as a shared known image and causes creation of global sections for the image. The image is made permanently open even if /OPEN is not specified.

***/[NO]WRITABLE***

Installs the file as a writable known image as long as you also specify the /SHARED qualifier. The /WRITABLE qualifier is automatically negated if the /NOSHARED qualifier is specified.

**description**

The REPLACE command updates a known file to the latest version found in the specified directory.

You can use the REPLACE command to modify the attributes of currently installed images. Either specify new qualifiers, or change the value of qualifiers used when installing the image with the ADD (or CREATE) command. If you specify no qualifiers, the new image retains the same attributes as the old one. If the old image was installed with the /SHARED qualifier, the global sections are recreated, probably with new identifiers.

If a process is accessing global sections when the REPLACE command is entered, the global sections are deleted only after the operation initiated by the process completes. However, once the command is entered, no additional processes can access the global sections because they are "marked for deletion."

**example**

```
INSTALL> REPLACE GRPCOMM /ACCOUNTING/NOOPEN
```

The command in this example replaces the known image GRPCOMM with the latest version of the image, while enabling image accounting and removing the OPEN attribute for this version.

The full name of the file specification is assumed to be SYS\$SYSTEM:GRPCOMM.EXE.



---

## LAT Control Program Utility

The LAT Control Program (LATCP) allows you to control and obtain information from the LAT port driver (LTDRIVER) on a VMS node.

### format

**RUN SYS\$SYSTEM:LATCP**

### usage summary

To invoke LATCP, type RUN SYS\$SYSTEM:LATCP at the DCL command prompt. At the LCP> prompt, you can enter any of the LATCP commands described in the following section.

To exit from LATCP, enter the LATCP command EXIT at the LCP> prompt or press CTRL/Z.

**Use of LATCP requires the CMKRNL privilege.**

## LAT-2 LATCP CREATE LINK

### LATCP Commands

This section describes the following LATCP commands and provides examples of their use.

---

#### CREATE LINK

Creates the Ethernet links that you want a VMS service node to use.

#### format

```
CREATE LINK link-name
```

#### parameter

##### *link-name*

Specifies a name for an Ethernet link. A link name can have up to 16 ASCII characters. (See the CREATE SERVICE command for a list of legal characters.) You can create a maximum of two links on your node. Use the SHOW CHARACTERISTICS command for a list of the link names that are defined for your node.

#### qualifiers

##### ***/[NO]DECNET***

Directs LAT protocol to use the DECnet Ethernet address (/DECNET) or the hardware address (/NODECNET) when starting the Ethernet controller. The default is /DECNET.

##### ***/DEVICE=device-name***

Specifies the Ethernet controller device name for an Ethernet link; for example, XEB0. Only one Ethernet link can be associated with an Ethernet controller. If you enter the CREATE LINK command without the /DEVICE qualifier, LATCP attempts to find an available controller. You can specify a default device name by defining the LAT\$DEVICE logical name.

##### ***/ENABLE=(group-code[,...])***

Specifies the service groups that can be used on the link. There can be up to 256 groups, numbered from 0 through 255. If you specify only one group, you can omit the parentheses.

By default, no groups are enabled for a link. In this case, the groups that you enabled for the service node with SET NODE or START NODE apply to the link.

If you enable groups with this qualifier, only the specified groups apply to the link; the groups enabled for your service node do not apply.

***/[NO]LOG***

Specifies whether the link characteristics are displayed when this command is executed. The default is */LOG*.

**example**

```
LCP> CREATE LINK Network_A /DEVICE=XEB0: /ENABLE=(1,2)
```

The **CREATE LINK** command in this example creates a link to the Ethernet network named *Network\_A*. It specifies the Ethernet controller device *XEB0* for that link. The command enables groups 1 and 2 for the *Network\_A* Ethernet link.

---

**CREATE PORT**

Creates a logical port on a VMS service node that connects with either a remote device on a terminal server or an application program.

**format**

```
CREATE PORT port-name
```

**parameter*****port-name***

Specifies the port name in the form *LTAn:*, where *n* is a unique number from 1 through 9999. If the port you specify already exists, LATCP returns an error message.

**qualifiers*****/APPLICATION***

Specifies that a logical port on a VMS service node will be used to connect to a remote device (typically a printer) on a terminal server. The default port type is */APPLICATION*.

***/DEDICATED***

Specifies that a logical port on a VMS node is reserved for an application service. When terminal server users request a connection to this service name, they are connected to the dedicated port, provided the application program has assigned a channel to the port. See the *VMS I/O User's Reference Volume* for a description of programming an application service.

After creating a dedicated port on a VMS service node, use the **SET PORT /DEDICATED /SERVICE** command to map this port to a service.

***/[NO]LOG***

Specifies whether characteristics of the ports on your service node are displayed when this command is executed. The default is */LOG*.

## LAT-4 LATCP CREATE PORT

### example

```
LCP> CREATE PORT LTA27: /APPLICATION
```

The CREATE PORT command in this example creates an applications port named LTA27 on a VMS service node. It is mapped to a remote device on a terminal server.

---

## CREATE SERVICE

Creates a service on a VMS service node.

### format

```
CREATE SERVICE service-name
```

### parameter

#### ***service-name***

Specifies a LAT service name. You can specify as many as eight service names for your node. By default, a service name is the translation of the SYS\$NODE logical name.

The service name can be from 1 to 16 ASCII characters.

### qualifiers

#### ***/IDENTIFICATION="identification-string"***

Describes a VMS service offered or delivers a message to terminal servers on the Ethernet. By default, the identification string is a translation of SYS\$ANNOUNCE. A VMS service node advertises its services at regular intervals, established in the SET NODE command.

An identification string can have up to 64 ASCII characters but cannot begin with an ampersand (&). Nonprintable characters are translated as spaces. Enclose the string in quotation marks (").

#### ***/LINK=(link-name[,...])***

Specifies the Ethernet link on which you want to offer the service. If you specify one link, you can omit the parentheses. This link must have been created, either explicitly with the CREATE LINK command or implicitly with the START NODE command. By default, a service is offered on all the Ethernet links defined for your node. In most cases, you should offer services over all of the Ethernet links. The SHOW CHARACTERISTICS command displays the links that are currently defined for your node.

You can use this qualifier to limit the users of a service to a particular Ethernet link.



***/[NO]LOG***

Specifies whether the characteristics for your service node are displayed when this command is executed. The default is /LOG.

***/[NO]STATIC\_RATING=*rating****

Enables or disables dynamic service ratings.

**example**

```
LCP> CREATE SERVICE SALES /LINK=(Network_A,Network_B) -  
_LCP> /STATIC_RATING=195
```

The CREATE SERVICE command in this example creates the service "SALES" on a VMS service node. The service will be offered on the Ethernet links named Network\_A and Network\_B. This command also assigns a static rating of 195 so terminal servers can assess the availability of services on the node.

---

**DELETE PORT**

Deletes a logical port from a VMS service node.

**format**

**DELETE PORT** *port-name*

**parameter**

***port-name***

Specifies the name of the applications port or the dedicated port that you want to delete. An applications port connects to a remote device on a terminal server, whereas a dedicated port connects to a special VMS service.

The port must have been created with the CREATE PORT command. Use the SHOW PORTS command for a list of the applications ports and the dedicated ports that are defined for your service node.

**example**

```
LCP> DELETE PORT LTA27:
```

The DELETE PORT command in this example deletes the applications port LTA27. The port was created with the CREATE PORT command.

## DELETE SERVICE

Deletes a service that your VMS service node currently offers.

### format

**DELETE SERVICE** *service-name*

### parameter

#### *service-name*

Specifies the name of the service, as displayed by the SHOW CHARACTERISTICS command. By default, the service name is the translation of SYS\$NODE.

### qualifiers

#### **/[NO]LOG**

Specifies whether the characteristics for your service node are displayed when this command is executed. The default is /LOG.

### example

LCP> DELETE SERVICE SALES

The DELETE SERVICE command in this example removes the service SALES from your service node. The service is no longer available to server users.

---

## EXIT

Stops execution of LATCP and returns control to the DCL command level. You can also type CTRL/Z to exit at any time.

### format

**EXIT**

---

## HELP

Provides online help information for using the LATCP commands.

### format

**HELP** [*command-name*]

## SET COUNTERS/ZERO

The SET COUNTERS/ZERO command resets the service node event/error counters. The /ZERO qualifier is required.

### format

SET COUNTERS/ZERO

---

## SET LINK

Changes the characteristics of Ethernet links.

### format

SET LINK *link-name*

### parameter

#### *link-name*

Specifies the name for an Ethernet link. A link name can have up to 16 ASCII characters. (See the CREATE SERVICE command for a list of legal characters.) The SHOW CHARACTERISTICS command displays the names of the links defined for a VMS service node.

### command qualifiers

#### ***/[NO]DECNET***

Directs LAT protocol to use the DECnet Ethernet address (/DECNET) or the hardware address (/NODECNET) when starting the Ethernet controller. The default is /DECNET. Note that you cannot change the characteristics of an active link.

#### ***/DEVICE=device-name***

Specifies the Ethernet controller device name for the link; for example, XEA0. Only one link can be associated with any Ethernet controller and its related Ethernet cable. You cannot change the device for an active link.

#### ***/DISABLE=(group-code[,...])***

Removes previously enabled groups associated with a link.

#### ***/ENABLE=(group-code[,...])***

Specifies additional groups that you want enabled for a link. If there is only one group, you can omit the parentheses. There are 256 groups, numbered from 0 through 255. See the SET NODE command for more information on groups.

## LAT-8 LATCP SET LINK

### ***/[NO]LOG***

Specifies whether to display link characteristics when the command executes. The default is */LOG*.

### **example**

```
LCP> SET LINK Network_A /ENABLE=(8,11)
```

The **SET LINK** command in this example assigns the groups 8 and 11 to the Ethernet link, *Network\_A*.

---

## **SET NODE**

Specifies the LAT characteristics of a VMS service node.

### **format**

**SET NODE** *node-name*

### **parameter**

#### ***node-name***

Specifies a name for a VMS service node. By default, the node name is the translation of *SYS\$NODE*. A LAT service node name should be the same as the DECnet node name. If the VMS service node is not running DECnet but will be in the future, it is recommended that you define *SYS\$NODE* and use it for both DECnet and LAT node names.

A node name can be from 1 to 16 ASCII characters.

### **qualifiers**

#### ***/DISABLE=(group-code[,...])***

Removes previously enabled groups associated with your service node. If you enter one group code, you can omit the parentheses. The **SHOW CHARACTERISTICS** command displays the groups enabled for your service node.

#### ***/ENABLE=(group-code[,...])***

Gives the listed groups access to your service node. A network manager organizes terminal server nodes into groups, based on the number of terminal server nodes in the LAT network. Groups subdivide the LAT network, limiting the number of terminal server nodes that can connect with a given VMS service node.

There can be as many as 256 groups, numbered 0 through 255. By default all terminal server nodes belong to group 0. If you enter one group code, you can omit the parentheses. Use the **SHOW CHARACTERISTICS** command for a list of the groups enabled for your service node.

***/IDENTIFICATION="identification-string"***

Describes a service offered by a VMS service node or delivers a message to terminal servers on the Ethernet. By default, the identification string is the translation of SYS\$ANNOUNCE. A VMS service node advertises its services at regular intervals, established in the SET NODE command.

An identification string can have up to 64 ASCII characters but cannot begin with an ampersand (&). Nonprintable characters are translated as spaces. Enclose the string in quotation marks ("").

***/[NO]LOG***

Specifies whether your service node characteristics are displayed when this command is executed. The default is /LOG.

***/MULTICAST\_TIMER=seconds***

Specifies the time, in seconds, between multicast messages sent by a VMS service node. A multicast message, established with the /IDENTIFICATION qualifier, advertises the services offered by a VMS service node. The minimum value is 10 seconds; the maximum is 255 seconds. The default value is 60.

**example**

```
LCP> SET NODE DUKE /IDENT="NODE DUKE, SALES VAXCLUSTER"
```

The SET NODE command in this example specifies that the announcement "NODE DUKE, SALES VAXCLUSTER" is multicast from node DUKE.

---

**SET PORT**

Logically associates an applications port on a VMS service node with a remote port on a terminal server that supports a device. Alternatively, it creates a logical port on a VMS service node that is dedicated to a specific service.

**format**

**SET PORT** *port-name*

**parameter**

***port-name***

Specifies the name of the port. A port name must be in the form LTA*n*;, where *n* is a unique number from 1 through 9999.

LAT-10 LATCP  
SET PORT

**qualifiers**

***/APPLICATION***

Specifies that a port on a VMS service node is an applications port, logically associated with a port on a remote terminal server. The terminal server port supports a device, for example, a printer. If the port is used to support a printer, the print queue is established in a startup command procedure, as described in *Guide to Maintaining a VMS System*.

The port must have been created with the CREATE PORT command.

***/DEDICATED***

Specifies that a port on a VMS service node functions as a dedicated logical port through which terminal server users connect with a special service. The /DEDICATED qualifier requires the /SERVICE qualifier.

To create a special service, create the service and define the dedicated port (CREATE PORT/DEDICATED) in LTLOAD.COM, which is executed in SYSTARTUP\_V5.COM. Then run the application program. Within the program, allocate dedicated ports with the same name as those defined in LTLOAD.COM. See *Guide to Setting Up a VMS System* and *VMS I/O User's Reference Volume* for further information.

***/LINK=(link-name[,...])***

Specifies the name of the Ethernet link that the applications port uses. If you use the SET PORT command and do not specify a link name, and no link has been defined, LATCP creates a default link name called LAT\$LINK and assigns an Ethernet controller device to this link. To look at the links defined for your node, use the SHOW CHARACTERISTICS command.

***/[NO]LOG***

Specifies whether or not to display the characteristics of the ports on your service node when this command is executed. The default is /LOG.

***/NODE=remote-node-name***

Specifies the name of a terminal server that supports a remote device and is logically associated with an applications port on your VMS service node.

***/PORT=remote-port-name***

Specifies the name of the remote port on a terminal server that supports a remote device and is logically associated (mapped) with an applications port on a VMS service node.

***/[NO]QUEUED***

Specifies queued or nonqueued access to the server port. The default is /QUEUED.

***/SERVICE=service-name***

Specifies either: (1) the name of the remote service offered at the terminal server port that is to be associated with an applications port (/APPLICATION) for a device, or (2) a service name for an application program being offered on a dedicated port (/DEDICATED) on a VMS service node.

**example**

```
LCP> SET PORT LTA28: /NODE=TLAT2 /PORT=PORT_7 /LINK=Network_B
```

The SET PORT command in this example associates the applications port LTA28 with the port named PORT\_7 on the terminal server named TLAT2. The applications port uses the Ethernet named Network\_B.

---

**SET SERVICE**

Dynamically changes the characteristics of a service.

**format**

**SET SERVICE** *service-name*

**parameter*****service-name***

Specifies the service whose characteristics are to be modified. If a service name is omitted, the default service name is the translation of SYS\$NODE.

**qualifiers*****/IDENTIFICATION="identification-string"***

Provides a new description of a VMS service or delivers a message to terminal servers on the Ethernet. By default, the identification string is the translation of SYS\$ANNOUNCE. A VMS service node advertises its services at regular intervals, established in the SET NODE command.

An identification string can have up to 64 ASCII characters but cannot begin with an ampersand (&). Nonprintable characters are translated as spaces. Enclose the string in quotation marks.

***/LINK=(link-name[,...])***

Specifies which links offer the service. Unless you specify a link name for a service, the service is offered on all active Ethernet links. The SHOW CHARACTERISTICS command displays links that are defined for a VMS service node.

***/[NO]LOG***

Specifies whether or not to display the qualifier values used in this command when this command is executed. The default is /LOG.





---

## SHOW COUNTERS

Displays performance and error statistics for a VMS service node.

### format

**SHOW COUNTERS**

### qualifiers

**/DEVICE**

Displays the Ethernet device counters. This information is the sum of all Ethernet counters for a particular controller on your node, including LAT and DECnet. If you have more than one Ethernet controller device on your node, use the /LINK qualifier to specify the link name of the controller device for which you want the counters.

### example

LCP>SHOW COUNTERS /NODE

The SHOW COUNTERS command in this example generates the following type of display:

```
LCP Node Counters
127597  Receive frames
         0  Receive errors
         3  Receive duplicates
161885  Transmit frames
         0  Transmit errors
00000000 Last transmit failure code
         28 Retransmissions
         6  Circuit timeouts
         0  Protocol errors
00000000 Protocol bit mask
         0  Resource errors
         0  No transmit buffer
         0  Unit timeouts
         0  Solicitation failures
         0  Discarded output bytes
```

---

## SHOW PORTS

Displays the characteristics of ports on a VMS service node.

### format

**SHOW PORTS** [*port-name*]

# LAT-14 LATCP

## SHOW PORTS

### parameter

#### ***port-name***

Specifies the name of the port for which information is displayed. The SHOW PORTS command without a port name displays the characteristics for all LTA $n$  ports on a service node.

Do not use the /APPLICATION, /DEDICATED, or /INTERACTIVE qualifiers with a specific port name.

### qualifiers

#### ***/APPLICATION***

Generates a display of all applications ports.

#### ***/DEDICATED***

Generates a display of all dedicated ports.

#### ***/INTERACTIVE***

Generates a display of all LAT interactive ports.

### example

```
LCP> SHOW PORTS
```

The SHOW PORTS command in this example produces the following type of display:

```
Local Port Name = LTA27:    <interactive>
    Actual Remote Node Name = TLAT1
    Actual Remote Port Name = PORT_7
    Link Name = Network_A

Local Port Name = LTA28:    <application>
    Specified Remote Node Name = TLAT2
    Specified Remote Port Name = PORT_7
    Specified Remote Service Name = PRINTER
    Actual Remote Node Name = TLAT2
    Actual Remote Port Name = PORT_7
    Link Name = Network_B

Local Port Name = LTA29:    <dedicated>
    Specified Service Name   = GRAPHICS
    Link Name = Network_A
```

The first port the example displays is the interactive port LTA27, which is connected via LAT Port\_7 on the TLAT1 server. The Ethernet link is Network\_A. In this display the presence of the actual values indicates an established connection.

The second port the example displays is the LTA28 applications port. This port is mapped to the following:

- The remote server TLAT2
- The remote port 7
- The remote service PRINTER

The presence of the actual values in the display indicates an established connection. The Ethernet link is Network\_B.

The third port the example displays is LTA29, a dedicated port on a VMS service node that offers the service GRAPHICS to terminal server users on the Network\_A Ethernet.

---

## SHOW SERVERS

Displays the characteristics of terminal servers known to a VMS service node, and indicates which Ethernet link the servers use to access the VMS node.

### format

**SHOW SERVERS**

### qualifiers

***/INACTIVE***

Displays the cumulative counters for all servers known to your service node. To obtain a display of the current counters, enter the SHOW COUNTERS/SERVER command.

### example

LCP>SHOW SERVERS

The SHOW SERVERS command in this example produces the following display:

```
LCP Server Characteristics for TLAT1
Ethernet address = 08-00-2B-02-F2-EC
Server is active
Link Name = Network_A      Active users = 1
```

## START NODE

Starts the LAT port driver and sets service node characteristics. This command also activates specific links on a VMS service node.

### format

**START NODE** [*node-name*]

### parameter

#### ***node-name***

Specifies the name you choose for a VMS service node. The default is the translation of SYS\$NODE. A node name should be the same as the DECnet node name. The node name can be from 1 to 16 characters long.

### qualifiers

#### ***/[NO]DECNET***

Directs the LAT protocol to use the DECnet Ethernet address (/DECNET) or the hardware address (/NODECNET) when starting the Ethernet controller. The default is /DECNET.

The /NODECNET qualifier can help improve performance when you have two Ethernet controllers on a VAX processor. You can restrict LAT traffic to one Ethernet controller and DECnet traffic to the other. Note that once you start the LAT protocol using the /NODECNET qualifier, you cannot start DECnet on the same Ethernet link without stopping the LAT port driver and restarting it.

#### ***/DISABLE=(group-code[,...])***

Removes previously enabled groups associated with a VMS service node.

#### ***/ENABLE=(group-code[,...])***

Gives listed groups access to a VMS service node. There are 256 groups, numbered from 0 through 255. By default, group 0 is enabled. If you enter only one group code, you can omit the parentheses.

#### ***/IDENTIFICATION="identification-string"***

Describes a VMS service offered or delivers a message to terminal servers on the Ethernet. By default, the identification string is the translation of SYS\$ANNOUNCE. A VMS service node advertises its services at regular intervals, established in the SET NODE command.

An identification string can have up to 64 ASCII characters but cannot begin with an ampersand (&). Nonprintable characters are translated as spaces. Enclose the string in quotation marks.

***/LINK=(link-name[,...])***

Specifies the name(s) of the link(s) that you want activated on a VMS service node. If you do not specify a link name, all defined links on your node are started. If you supply only one link name, you can omit the parentheses.

***/[NO]LOG***

Specifies whether to display your service node characteristics when this command is executed. */NOLOG* prevents the display. The default is */LOG*.

***/MULTICAST\_TIMER=seconds***

Specifies the time, in seconds, between the multicast messages sent by your service node. The minimum value is 10 seconds; the maximum is 255 seconds. The default value is 60.

**example**

```
LCP> START NODE DUKE /LINK=Network_A
```

The **START NODE** command in this example starts node **DUKE** and activates the **Network\_A** Ethernet link on node **DUKE**.

---

**STOP NODE**

Deactivates a specific Ethernet link on a VMS service node, or shuts down the LAT port driver on a VMS node, terminating sessions for all links.

**format**

**STOP NODE**

**qualifiers**

***/LINK=(link-name[,...])***

Specifies the name of the Ethernet link that you want to stop. Use this qualifier only if you want to stop a specific link.

***/[NO]LOG***

Specifies whether to display a confirmation message on the user's terminal when you shut down the LAT port driver. (Note that the actual shutdown takes a few seconds if the driver has to terminate active sessions.) The default is */LOG*.

**example**

```
LCP> STOP NODE /LINK=Network_A
```

The **STOP NODE** command in this example deactivates the **Network\_A** Ethernet link on a VMS service node.



---

## Mount Utility

The Mount Utility (MOUNT) allows you to make a disk or magnetic tape volume available for processing.

### format

**MOUNT** *device-name[:][,...]* [*volume-label[,...]*] [*logical-name[:]*]

### command parameters

***device-name[:][,...]***

Specifies the physical device name or logical name of the device on which the volume is to be mounted. On a system where volumes are not connected to Hierarchical Storage Controllers (HSCs), use the following format:

**ddcu:**

The **dd** describes the device type of the physical devices used. For example, an RA60 disk drive is device type **DJ**, and an RA80 or RA81 disk drive is device type **DU**. The **c** identifies the controller, and the **u** identifies the unit number of the device.

On a system with Hierarchical Storage Controllers (HSCs), use one of the following formats:

**node\$ddcu:**

**allocation-class\$ddcu:**

If your devices are dual ported to HSCs, use the allocation-class format. For example, \$125\$DUA23 represents an RA80 or RA81 disk with unit number 23. The disk's allocation class is \$125\$. The **c** part of the format is always A for HSC disks. TROLL\$DJA12 represents an RA60 disk with unit number 12. The device is connected to an HSC named TROLL.

Device names can be generic so that if no controller or unit number is specified, the system attempts to mount the first available device that satisfies those specified components of the device name(s). If no volume is physically mounted on the specified device, MOUNT displays a message requesting that you place the volume in the device; after you place the volume in the named drive, MOUNT then completes the operation.

If you specify more than one device name for a disk or magnetic tape volume set, separate the device names with either commas or plus signs. For a magnetic tape volume set, you can specify more volume labels than device names or more device names than volumes.

### ***volume-label[,...]***

Specifies the label on the volume. For disk volumes, labels can have from 1 through 12 characters; for magnetic tape volumes, labels can have from 0 through 6 characters.

If you specify more than one volume label, separate the labels with either commas or plus signs. The volumes must be in the same volume set and the labels must be specified in ascending order according to relative volume number.

When you mount a magnetic tape volume set, the number of volume labels need not equal the number of device names specified. When a magnetic tape reaches the end-of-tape (EOT) mark, the system requests the operator to mount the next volume on one of the devices. The user is not informed of this request; only the operator is informed.

When you mount a disk volume set, each volume label specified in the list must correspond to a device name in the same position in the device name list.

The volume-label parameter is not required when you mount a volume with the /FOREIGN or /NOLABEL qualifier or when you specify /OVERRIDE=IDENTIFICATION. To specify a logical name when you enter either of these qualifiers, type any alphanumeric characters in the volume-label parameter position.

### ***logical-name[:]***

Defines a 1- through 255-alphanumeric character string logical name to be associated with the volume.

If you do not specify a logical name, the MOUNT command assigns the default logical name DISK\$volume-label to individual disk drives; it assigns the default logical name DISK\$volume-set-name to the device on which the root volume of a disk volume set is mounted. Note that if you specify a logical name in the mount request that is different from DISK\$volume-label or DISK\$volume-set-name, then two logical names are associated with the device.

If you do not specify a logical name for a magnetic tape drive, the MOUNT command assigns only one logical name, TAPE\$volume-label, to the first magnetic tape device in the list. No default logical volume set name is assigned in this case.

The MOUNT command places the name in the process logical name table, unless you specify /GROUP or /SYSTEM. In the latter cases, it places the logical names in the group or system logical name table.

**NOTE:** Avoid assigning a logical name that matches the file name of an executable image in SYS\$SYSTEM. Such an assignment prohibits you from invoking that image.



If the logical name of a volume is in a process-private table, then the name is not deleted when the volume is dismounted.

### **usage summary**

To invoke the Mount Utility, enter the command MOUNT at the DCL prompt, followed by the device name, volume label, and logical name. If you omit a parameter, MOUNT prompts you for it. You must include a device name and a volume label (unless you specify /OVERRIDE=IDENTIFICATION or use the /FOREIGN or /NOLABEL qualifier); the logical name is optional.

The Mount Utility returns you to the DCL level after it either successfully completes the operation or fails, generating an error message. If you press CTRL/Y or CTRL/C, MOUNT aborts the operation and returns you to the DCL prompt.

You can direct output from MOUNT operations with the /COMMENT and /MESSAGE qualifiers. When the mount operation requires operator assistance, use /COMMENT to specify additional information to be included with the operator request. The /COMMENT text string is sent to the operator log file and to SYS\$OUTPUT. The string must contain no more than 78 characters.

Use the /MESSAGE qualifier (this is the default) to send mount request messages to your current SYS\$OUTPUT device. If you specify /NOMESSAGE during an operator-assisted mount, messages are not sent to SYS\$OUTPUT; the operator sees them, however, if an operator terminal is enabled to receive messages.

Many MOUNT qualifiers require special privileges. Some qualifiers require different privileges according to which qualifier keyword you specify. See the individual qualifiers for details.

## **MOUNT-4 MOUNT /ACCESSED**

### **MOUNT Qualifiers**

The following pages describe the Mount Utility qualifiers. The qualifiers are listed alphabetically and include examples, as needed. There are no subcommands for the Mount Utility.

---

#### **/ACCESSED**

Specifies, for disk volumes, the approximate number of directories that will be in use concurrently on the volume.

#### **format**

*/ACCESSED=n device-name*

#### **qualifier value**

*n*

Specifies the approximate number of directories that will be in use concurrently on the volume. Specify a value from 0 through 255 to override the default that was specified when the volume was initialized.

You need the user privilege OPER to use /ACCESSED.

#### **example**

```
$ MOUNT/ACCESSED=150 DBA1 WORK
```

This command requests the volume labeled WORK to be mounted on DBA1, specifying 150 as the number of active directories on the volume.

---

#### **/ASSIST**

Directs the mount operation to allow operator or user intervention if the mount request fails.

#### **format**

*/ASSIST device-name*

*/NOASSIST device-name*

#### **description**

When you specify the /ASSIST qualifier, MOUNT notifies the user and certain classes of operator if a failure occurs during the mount operation. If a failure occurs, the operator or user can either abort the operation or correct the error condition to allow the operation to continue.

The operator-assist messages are sent to all operator terminals that are enabled to receive messages; magnetic tape mount requests go to TAPE and DEVICE operators, and disk mount requests go to DISK and DEVICE operators. Thus, if you need operator assistance while mounting a disk device, a message is sent to DISK operators.

Any operator reply to a mount request is written to SYS\$OUTPUT to be displayed on the user's terminal or written in a batch job log.

If no operator terminal is enabled to receive and respond to a mount assist request, a message is displayed informing the user of the situation. If a volume is placed in the requested drive, no additional operator response is necessary. If the mount request originates from a batch job and no operator terminal is enabled to receive messages, the mount is aborted.

The default is /ASSIST and can be overridden by /NOASSIST.

### example

```
$ MOUNT/NOASSIST DMA0: DOC WORK
%MOUNT-I-MOUNTED, DOC          mounted on _NODE$DMA0:
```

This command mounts an RK07 volume labeled DOC and assigns the logical name WORK. The /NOASSIST qualifier signals MOUNT that no operator intervention is necessary.

---

### /AUTOMATIC

Determines whether MOUNT enables or disables automatic volume switching and labeling for magnetic tape.

### format

**/AUTOMATIC** *device-name*  
**/NOAUTOMATIC** *device-name*

### description

The default is /AUTOMATIC. If you have multiple magnetic tape drives allocated to a volume set, the Magnetic Tape Ancillary Control Process (MTAACP) performs the volume switch by sequentially selecting the next available drive allocated to the volume set. The MTAACP expects the next reel of the volume set to be loaded on that drive.

If the MTAACP is writing to the volume set, it creates a label and initializes the magnetic tape with that label and the protections established for the first magnetic tape of the volume set. If it is reading from the volume set, the MTAACP generates the label and attempts to mount the next magnetic tape with that label. If the drive has the wrong magnetic tape (or no magnetic tape) loaded, the MTAACP sends a message to the operator's console to prompt for the correct magnetic tape.

## MOUNT-6 MOUNT /AUTOMATIC

The label generated by the MTAACP fills the 6-character volume identifier field. The first four characters of the field contain the first four characters of the label specified in the MOUNT command, padded with an underscore when the label is not at least four characters. The fifth and sixth characters contain the relative volume number for this reel in the volume set.

If you specify /NOAUTOMATIC, the MTAACP requires operator intervention to switch to the next drive during end-of-tape processing, and requires that the operator specify a label for each new reel added to a volume set.

### example

```
$ MOUNT/NOAUTOMATIC MTA0: ABCD, EFGH
```

This command instructs MOUNT not to generate its own label for the second volume, but to use the ones supplied with the MOUNT command. If the second volume is not already labeled, then the operator must use REPLY/INIT and supply the second label.

---

## /BIND

Creates a volume set of one or more disk volumes or adds one or more volumes to an existing volume set.

### format

*/BIND=volume-set-name device-name[,...] volume-label[,...]*

### keyword

#### ***volume-set-name***

Specifies a 1- through 12-alphanumeric-character name identifying the volume set.

### description

You must specify the /BIND qualifier when you first create the volume set or each time you add a volume to the set. To dismount an individual volume of the volume set, you must use the DISMOUNT qualifier /UNIT. Otherwise, dismounting an individual volume dismounts the entire volume set.

When you create a volume set, the volumes specified in the volume-label list are assigned relative volume numbers based on their position in the label list. The first volume specified becomes the root volume of the set.

When you add a volume or volumes to a volume set, the first volume label specified must be that of the root volume, or the root volume must already be on line.

Note that if you attempt to create a volume set from two or more volumes that already contain files and data, the file system does not issue an error message when you issue the MOUNT/BIND command. However, the volumes are unusable as a volume set because the directory structures are not properly bound.

### example

```
$ MOUNT/BIND=LIBRARY DMA0:,DMA1:,DMA2: BOOK1,BOOK2,BOOK3
```

This command creates a volume set named LIBRARY. This volume set consists of the volumes labeled BOOK1, BOOK2, and BOOK3, which are mounted physically on devices DMA0, DMA1, and DMA2, respectively.

---

### /BLOCKSIZE

Specifies the default block size for magnetic tape volumes.

### format

**/BLOCKSIZE=*n device-name***

### qualifier value

*n*

Specifies the default block size value for magnetic tape volumes. Valid values are in the range 20 through 65,532 for VMS RMS operations, and 18 through 65,534 for non-VMS RMS operations. By default, records are written to magnetic tape volumes in 2048-byte blocks. For foreign or unlabeled magnetic tapes, the default is 512 bytes.

### description

You must specify /BLOCKSIZE in two situations:

- When mounting magnetic tapes that do not have HDR2 labels. For these magnetic tapes, you must specify the block size. For example, you must specify /BLOCKSIZE=512 to mount an RT-11 magnetic tape.
- When mounting magnetic tapes that contain blocks whose size exceeds the default block size (2048 bytes). In this case, specify the size of the largest block for the block size.

### example

```
$ MOUNT/FOREIGN/BLOCKSIZE=1000 MTA1:
```

In this example, the /BLOCKSIZE qualifier specifies a block size of 1000 bytes; the default for a magnetic tape mounted with the /FOREIGN qualifier is 512.

## **/CACHE**

For disks, controls whether caching limits established at system generation time are disabled or overridden. With the `TAPE_DATA` option, enables write caching for the tape controller specified (if the tape controller supports write caching).

### **format**

**/CACHE=(keyword[,...])**

**/NOCACHE**

### **keywords**

***EXTENT[=n]***

***NOEXTENT***

Enables or disables extent caching. To enable extent caching, you must have the operator user privilege (OPER) and you must specify *n*, the number of entries in the extent cache. Note that `NOEXTENT` is equivalent to `EXTENT=0`; both disable extent caching.

***FILE\_ID[=n]***

***NOFILE\_ID***

Enables or disables file identification caching. To enable file identification caching, you must have the operator user privilege (OPER) and you must specify *n*, the number of entries, as a value greater than 1. Note that `NOFILE_ID` is equivalent to `FILE_ID=1`; both disable file identification caching.

***LIMIT=n***

Specifies the maximum amount of free space in the extent cache in one-thousandths of the currently available free space on the disk.

***QUOTA[=n]***

***NOQUOTA***

Enables or disables quota caching. To enable quota caching, you must have the operator user privilege (OPER) and you must specify *n*, the number of entries in the quota cache. Normally *n* is set to the maximum number of active users expected for a disk with quotas enabled. Both `NOQUOTA` and `QUOTA=0` disable quota file caching.

***TAPE\_DATA***

Enables write caching for a magnetic tape device if the tape controller supports write caching. `/NOCACHE` is the default for mounting tape devices. You must specify `TAPE_DATA` to enable write caching. If the tape controller does not support write caching, the keyword is ignored.

### **WRITETHROUGH**

Disables writeback caching, which writes only the file headers of files open for write when the files are closed. Thus, if you specify the WRITETHROUGH keyword, file headers are written to the disk on every file header operation.

### **description**

Used with the disk options, the /CACHE qualifier overrides one or more of the present disk caching limits established at system generation time. Used with the TAPE\_DATA option, the /CACHE qualifier enables write caching for the tape controller specified.

If you specify more than one option, separate them by commas and enclose the list in parentheses. The options [NO]EXTENT, [NO]FILE\_ID, LIMIT, and [NO]QUOTA apply only to a disk device. The option TAPE\_DATA applies only to a tape device.

If you specify /NOCACHE for a disk device, all caching is disabled for this volume. Note that the /NOCACHE qualifier is equivalent to /CACHE=(NOEXTENT,NOFILE\_ID,NOQUOTA,WRITETHROUGH).

If you specify /NOCACHE for a magnetic tape device, the tape controller's write cache is disabled for this volume. This is the default for the TAPE\_DATA option.

### **example**

```
$ MOUNT/CACHE=(EXTENT=60,FILE_ID=60,QUOTA=20,WRITETHROUGH) -
_ $ DMA0: FILES WORK
%MOUNT-I-MOUNTED, FILES          mounted on _NODE$DMA0:
```

This command mounts an RK07 device labeled FILES and assigns the logical name WORK. The /CACHE qualifier enables an extent cache of 60 entries, a file identification cache of 60 entries, and a quota cache of 20; it disables writeback caching.

### **/CLUSTER**

Specifies that after the volume is successfully mounted on the local node, or if it is already mounted /SYSTEM on the local node, it is to be mounted on every other node in the existing VAXcluster (that is, the volume is mounted clusterwide).

### **format**

**/CLUSTER** *device-name*

# MOUNT-10 MOUNT /CLUSTER

## description

Only system or group volumes can be mounted clusterwide. If you specify the /CLUSTER qualifier with neither the /SYSTEM nor the /GROUP qualifier, the default is /SYSTEM. Note that you must use a cluster device-naming convention. Use either *node\$device-name* or *allocation-class\$device-name* as required by your configuration.

You need the user privileges GRPNAM and SYSNAM, respectively, to mount group and system volumes clusterwide.

If the system is not a member of a VAXcluster, the /CLUSTER qualifier has no effect.

## example

```
$ MOUNT/CLUSTER DOPEY$DMA1: SNOWWHITE DWARFDISK
%MOUNT-I-MOUNTED, SNOWWHITE          mounted on _DOPEY$DMA1:
$ SHOW DEVICE/FULL DWARFDISK:
```

Disk \$2\$DMA1: (DOPEY), device type RK07, is online, mounted, file-oriented device, shareable, served to cluster via MSCP Server, error logging is enabled.

Error count	0	Operations completed	159
Owner process	""	Owner UIC	[928,49]
Owner process ID	00000000	Dev Prot	S:RWED,O:RWED,G:RW,W:R
Reference count	1	Default buffer size	512
Total blocks	53790	Sectors per track	22
Total cylinders	815	Tracks per cylinder	3
Allocation class	2		

Volume label	"SNOWWHITE"	Relative volume number	0
Cluster size	3	Transaction count	1
Free blocks	51720	Maximum files allowed	6723
Extend quantity	5	Mount count	7
Mount status	System	Cache name	"_\$255\$DWARF1:XQPCACHE"
Extent cache size	64	Maximum blocks in extent cache	5172
File ID cache size	64	Blocks currently in extent cache	0
Quota cache size	25	Maximum buffers in FCP cache	349

Volume status: subject to mount verification, file high-water marking, write-through caching enabled.

Volume is also mounted on DOC, HAPPY, GRUMPY, SLEEPY, SNEEZY, BASHFUL.

This MOUNT/CLUSTER command mounts the volume SNOWWHITE on DOPEY\$DMA1, then proceeds to mount the volume clusterwide. The SHOW DEVICE/FULL command displays information about the volume, including the other nodes on which it is mounted.



---

## /COMMENT

Specifies additional information to be included with the operator request when the mount operation requires operator assistance.

### format

*/COMMENT="string" device-name*

### keyword

#### *string*

Specifies the text string that is output to the operator log file and the current SYS\$OUTPUT device. The string must contain no more than 78 characters.

### example

```
$ MOUNT DY1: TESTSYS/COMMENT="Volume in cabinet 6."  
%MOUNT-I-OPRQST, Please mount volume TESTSYS in device _DY1:  
Volume in cabinet 6.  
%MOUNT-I-MOUNTED TESTSYS mounted on _DY1:  
%MOUNT-I-OPRQSTDON, operator request canceled - mount  
completed successfully
```

This command requests the operator to mount the disk volume TESTSYS on the device DY1. Notice that the /COMMENT qualifier is used to inform the operator of the location of the volume. After the operator places the volume in DY1, MOUNT retries the operation. After the operation completes, the operator request is canceled.

---

## /CONFIRM

Applicable only if you have the volume shadowing option.

---

## /COPY

Applicable only if you have the volume shadowing option.

## MOUNT-12 MOUNT

### /DATA\_CHECK

---

### /DATA\_CHECK

Overrides the read-check or write-check option (or both) specified for a volume when it was initialized.

#### format

*/DATA\_CHECK[(keyword[,...])] device-name*

#### keywords

##### **READ**

Performs checks following all read operations.

##### **WRITE**

Performs checks following all write operations.

#### description

You can specify either or both of the keywords. If you specify more than one keyword, separate them by commas and enclose the list in parentheses.

If you specify the /DATA\_CHECK qualifier without specifying a keyword, MOUNT defaults to /DATA\_CHECK=WRITE.

#### example

```
$ MOUNT/DATA_CHECK=READ CLEMENS$DBA2: SAM BOOK
```

This command mounts a volume labeled SAM on CLEMENS\$DBA2 and assigns the logical name BOOK. The /DATA\_CHECK=READ qualifier overrides a previous INITIALIZE/DATA\_CHECK=WRITE specification, so that subsequent read operations on BOOK are subject to data-checking operations.

---

### /DENSITY

Specifies the density (in bpi) at which a foreign or unlabeled magnetic tape is to be written.

#### format

*[/FOREIGN][/NOLABEL]/DENSITY=*n* device-name*

#### qualifier value

##### ***n***

Specifies a density of 800 bpi, 1600 bpi, or 6250 bpi, if supported by the magnetic tape drive. If you do not specify a density for a magnetic tape that was previously written, the density defaults to that of the first record on the volume.

## description

The specified density is used only if you specify /FOREIGN or /NOLABEL and the first operation performed on the magnetic tape is a write.

If you specify /LABEL, or if the first operation on the magnetic tape is a read, the magnetic tape is read or written at the density at which the first record on the magnetic tape is recorded. The default is /LABEL.

## example

```
$ MOUNT/FOREIGN/DENSITY=1600 MFA0: TAPE
```

This command mounts a foreign magnetic tape on drive MFA0 and assigns the logical name TAPE. The /DENSITY qualifier specifies that the magnetic tape is to be written at a density of 1600 bpi.

---

## /EXTENSION

Specifies the number of blocks by which disk files are to be extended on the volume unless otherwise specified by an individual command or program request.

## format

*/EXTENSION=n device-name*

## qualifier value

*n*

Specifies a value from 0 through 65,535 to override the value specified when the volume was initialized.

## example

```
$ MOUNT/EXTENSION=64 DBA0: DOC WORK
```

This command mounts a volume labeled DOC on DBA0, assigns the logical name WORK, and specifies a default block extent of 64 for the files on WORK.

## MOUNT-14 MOUNT /FOREIGN

---

### **/FOREIGN**

Indicates that the volume is not in the standard format used by the VMS operating system.

#### **format**

**/FOREIGN** *device-name*

#### **description**

You should use the **/FOREIGN** qualifier when a magnetic tape volume is not in the standard ANSI format, or when a disk volume is not in Files-11 format.

If you mount a volume with the **/FOREIGN** qualifier, the program you use to read the volume must be able to process the labels on the volume, if any. The VMS operating system does not provide an ancillary control process (ACP) to process the volume.

You must mount DOS-1 and RT-11 volumes with the **/FOREIGN** qualifier and process them with the Exchange Utility (**EXCHANGE**).

The default protection applied to foreign volumes is RWLP (Read, Write, Logical I/O, Physical I/O) for the system and owner. If you also specify **/GROUP**, group members are also given RWLP access. If you specify **/SYSTEM** or **/SHARE**, the group and world are both given RWLP access. If you mount a volume currently in Files-11 format with the **/FOREIGN** qualifier, you must have the user privilege VOLPRO, or your UIC must match the UIC on the volume.

#### **example**

```
$ MOUNT/FOREIGN MTA1: TAPE
```

This command mounts a foreign magnetic tape on drive MTA1.

---

### **/GROUP**

Makes the volume available to other users with the same group number in their UICs as the user entering the MOUNT command.

#### **format**

**/GROUP** *device-name*

## description

The logical name for the volume is placed in the group logical name table. You must have the user privilege GRPNAM to use the /GROUP qualifier.

Note that if the volume is owned by a group other than yours, access may be denied because of the volume protection.

## example

```
$ MOUNT/GROUP DB1:, DB2:, DB3: -  
_ $ PAYVOL1,PAYVOL2,PAYVOL3 PAY
```

This command mounts and makes available on a group basis the volume set consisting of volumes labeled PAYVOL1, PAYVOL2, and PAYVOL3. The logical name PAY is assigned to the set; anyone wanting to access files on these volumes can refer to the set as PAY.

```
$ MOUNT/GROUP/BIND=MASTER_PAY -  
_ $ DB4: PAYVOL4
```

This command adds the volume labeled PAYVOL4 to the existing volume set MASTER\_PAY. The root volume for the volume set must be on line when you enter this command.

---

## /HDR3

Controls whether ANSI standard header label 3 is written on a magnetic tape volume.

## format

```
/HDR3 device-name  
/NOHDR3 device-name
```

## description

By default, header label 3 is written. You can specify the /NOHDR3 qualifier to write magnetic tapes that are to be used on other systems that do not process HDR3 labels correctly.

## example

```
$ INITIALIZE MTA0: ABCD  
$ MOUNT/NOHDR3 MTA0: ABCD
```

The INITIALIZE and MOUNT commands prepare an ANSI-formatted magnetic tape for processing. The /NOHDR3 qualifier specifies that no HDR3 labels are to be written, thus creating a magnetic tape that can be transported to systems that do not process implementation-dependent labels correctly.

**MOUNT-16 MOUNT**  
**/INITIALIZE=CONTINUATION**

---

**/INITIALIZE=CONTINUATION**

Specifies that any volume added to the magnetic tape volume set is initialized before you can write to the volume.

**format**

**/INITIALIZE=CONTINUATION** *device-name*

**example**

\$ MOUNT/INITIALIZE=CONTINUATION MTA0: ABCD

This **/INITIALIZE=CONTINUATION** qualifier instructs the **MOUNT** command to assign its own continuation label. In this case, the operator can enter the command **REPLY/TO=n**, and the system assigns a label derived from the original. It uses the label specified in the **MOUNT** command and adds the appropriate number (ABCD02, ABCD03, and so forth).

---

**/LABEL**

Indicates that the volume is in the standard format used by the VMS operating system; that is, a magnetic tape volume is in the standard ANSI format, or a disk volume is in Files-11 format.

**format**

**/LABEL** *device-name*  
**/NOLABEL** *device-name*

**description**

The default is **/LABEL**.

Note that **/NOLABEL** is equivalent to **/FOREIGN**.

**example**

\$ MOUNT/LABEL MFA1: TAPE

This command mounts an ANSI-labeled magnetic tape on MFA1 and assigns the logical name TAPE.

## **/MESSAGE**

Causes mount request messages to be sent to your current SYS\$OUTPUT device.

### **format**

**/MESSAGE** *device-name*  
**/NOMESSAGE** *device-name*

### **description**

If you specify **/NOMESSAGE** during an operator-assisted mount, messages are not output to SYS\$OUTPUT; the operator sees them, however, provided an operator terminal is enabled.

The default is **/MESSAGE**.

### **example**

```
$ MOUNT/NOMESSAGE DLA0: SLIP DISC
```

In this example an RL02 device labeled SLIP is mounted on drive DLA0 and is assigned the logical name DISC. The **/NOMESSAGE** qualifier disables the broadcast of mount request messages to the user terminal.

---

## **/MOUNT\_VERIFICATION**

Specifies that the device is a candidate for mount verification.

### **format**

**/MOUNT\_VERIFICATION** *device-name*  
**/NOMOUNT\_VERIFICATION** *device-name*

### **description**

The **/MOUNT\_VERIFICATION** qualifier affects Files-11 Structure Level 2 disks, and as of VMS Version 5.0 affects foreign and ANSI-labeled magnetic tape volumes. The default is **/MOUNT\_VERIFICATION**.

### **example**

```
$ MOUNT/CACHE=(NOEXTENT,NOFILE_ID,NOQUOTA,WRITETHROUGH) -  
_ $ /NOMOUNT_VERIFICATION DMA0: FILES WORK  
%MOUNT-I-MOUNTED, FILES          mounted on _NODE$DMA0:
```

This command mounts an RK06 or RK07 device labeled FILES and assigns the logical name WORK. The **/CACHE** qualifier disables extent caching, file identification caching, quota caching, and writeback caching; the **/NOMOUNT\_VERIFICATION** qualifier disables mount verification.

## MOUNT-18 MOUNT /MULTI\_VOLUME

---

### /MULTI\_VOLUME

For foreign or unlabeled magnetic tape volumes, determines whether you override MOUNT volume-access checks. Use /MULTI\_VOLUME to override access checks on volumes that do not contain labels that MOUNT can interpret. If you have software produced before VMS Version 5.0 that processes multiple-volume, foreign-mounted tape volumes without specifically mounting and dismounting each reel, you may now need to mount the first volume with the /MULTI\_VOLUME qualifier.

### format

**/MULTI\_VOLUME** *device-name*  
**/NOMULTI\_VOLUME** *device-name*

### description

Use this qualifier when a utility that supports multiple-volume, foreign-mounted magnetic tape sets needs to process subsequent volumes, and these volumes do not contain labels that the VMS Mount Utility can interpret.

As of VMS Version 5.0, by default, all tape volumes are subject to the complete access checks of the VMS Mount Utility (MOUNT). Some user-written and vendor-supplied utilities used prior to VMS Version 5.0 may mount only the first tape in a foreign tape set. To make these utilities compatible with VMS Version 5.0, you should alter them to perform explicit calls to the \$MOUNT and \$DISMOU system services for each reel in the set. As an alternative, you can now mount the magnetic tape sets to be used by these utilities with the /MULTI\_VOLUME qualifier.

You must specify the /FOREIGN qualifier with the /MULTI\_VOLUME qualifier and you must have the user privilege VOLPRO. The default is /NOMULTI\_VOLUME.

**NOTE:** The VMS Backup Utility has been modified for VMS Version 5.0 to explicitly perform calls to the \$MOUNT and \$DISMOU system services on each reel of a foreign-mounted magnetic tape set.

### example

```
$ MOUNT/FOREIGN/MULTI_VOLUME MUA0:
```

This command mounts a tape volume set. MOUNT performs an access check on the first volume in the set and proceeds without checks to subsequent reels as they are needed for processing.



## **/OVERRIDE**

Inhibits one or more protection checks that the MOUNT command performs.

### **format**

*/OVERRIDE=(keyword[,...]) device-name*

### **keywords**

#### **ACCESSIBILITY**

For magnetic tapes only. If the installation allows, this keyword overrides any character in the Accessibility Field of the volume. The necessity of this keyword is defined by the installation. That is, each installation has the option of specifying a routine that the magnetic tape file system will use to process this field. By default, VMS provides a routine that checks this field in the following manner:

- If the magnetic tape was created on a version of VMS that conforms to Version 3 of ANSI, then you must use this keyword to override any character other than an ASCII space.
- If a VMS protection is specified and the magnetic tape conforms to an ANSI standard that is higher than Version 3, then you must use this keyword to override any character other than an ASCII 1.

To use the ACCESSIBILITY keyword, you must have the user privilege VOLPRO or own the volume.

#### **EXPIRATION**

For magnetic tapes only. Allows you to override the expiration dates of a volume and its files. Use this keyword when the expiration date in the first file header label of any file that you want to overwrite has not been reached. You must have the user privilege VOLPRO or your UIC must match the UIC written on the volume.

#### **IDENTIFICATION**

Overrides processing of the volume identifier in the volume label. Use this keyword to mount a volume for which you do not know the label. Only the volume identifier field is overridden. Volume protection, if any, is preserved. The volume must be mounted /NOSHARE (either explicitly or by default).

#### **LOCK**

Directs MOUNT not to write-lock the volume as a consequence of certain errors encountered while mounting it. Use this keyword when you are mounting a damaged volume to be repaired using the Verify Utility. You must have VOLPRO privilege or own the volume to use the LOCK keyword.

## **MOUNT-20 MOUNT /OVERRIDE**

### ***OWNER\_IDENTIFIER***

For magnetic tapes only. Overrides the processing of the owner identifier field. Use this keyword to interchange protected magnetic tapes between VMS and other Digital operating systems.

### ***SETID***

For magnetic tapes only. Prevents MOUNT from checking the file-set identifier in the first file header label of the first file on a continuation volume. Use this keyword only for ANSI-labeled volumes on which the file-set identifier of the first file on a continuation volume differs from the file-set identifier of the first file of the first volume that was mounted.

### ***SHADOW\_MEMBERSHIP***

Applicable only if you have the volume shadowing option.

If you specify more than one keyword, separate them with commas and enclose the list in parentheses.

You need the user privileges OPER and VOLPRO to specify /OVERRIDE=(ACCESSIBILITY, EXPIRATION) along with the /FOREIGN qualifier; otherwise, the magnetic tape is not read.

## **example**

```
$ MOUNT/OVERRIDE=IDENTIFICATION MFA0:
```

This command overrides the volume identification field, thus mounting a magnetic tape on MFA0 without a label specification.

---

## **/OWNER\_UIC**

Requests that the specified UIC be assigned ownership of the volume while it is mounted, overriding the ownership recorded on the volume. Or, if you are mounting a volume using the /FOREIGN qualifier, requests an owner UIC other than your current UIC.

## **format**

**/OWNER\_UIC=***uic device-name*

## **keyword**

### ***UIC***

Specifies the user identification code (UIC) in the following format:

**[group,member]**

You must use brackets in the UIC specification. The group number is an octal number in the range 0 through 37776; the member number is an octal number in the range 0 through 17776.

To use the /OWNER\_UIC qualifier for a Files-11 volume you must have the user privilege VOLPRO, or your UIC must match the UIC written on the volume.

### example

```
$ MOUNT/OWNER_UIC=[016,360] DRA3: WORK
```

This command mounts a disk device labeled WORK on DRA3 and assigns an owner UIC of [016,360].

---

## /PROCESSOR

For magnetic tapes and Files-11 Structure Level 1 disks, requests that the MOUNT command associate an Ancillary Control Process (ACP) to process the volume. The /PROCESSOR qualifier causes MOUNT to override the default manner in which ACPs are associated with devices.

For Files-11 Structure Level 2 disks, controls block cache allocation.

### format

*/PROCESSOR=keyword device-name*

### keywords

#### **UNIQUE**

For magnetic tape and Files-11 Structure Level 1 disks, creates a new process to execute a copy of the default ACP image for the specified device type or controller.

For Files-11 Structure Level 2 disks, allocates a separate block cache.

#### **SAME:device**

For magnetic tape and Files-11 Structure Level 1 disks, uses the same ACP process currently being used by the device specified.

For Files-11 Structure Level 2 disks, takes the block cache allocation from the specified device.

#### **filespec**

Creates a new process to execute the ACP image specified by the file specification (for example, a modified or a user-written ACP). You cannot use wildcard characters, or node and directory names in the file specification.

To use this keyword, you need CMKRNL and OPER privilege.

You must have the operator user privilege OPER to use the /PROCESSOR qualifier.

## MOUNT-22 MOUNT /PROCESSOR

### example

```
$ MOUNT/PROCESSOR=SAME:MTA1: MFA0:
```

This command directs MOUNT to mount a magnetic tape on MFA0 using the same ACP process currently associated with MTA1.

---

### /PROTECTION

Specifies the protection code to be assigned to the volume.

### format

*/PROTECTION=code device-name*

### keyword

#### *code*

Specifies the protection code according to the standard syntax rules for specifying protection. If you omit a protection category, that category of user is denied all access.

If you do not specify a protection code, the default is the protection that was assigned to the volume when it was initialized.

### description

If you specify the /PROTECTION qualifier when you mount a volume with the /SYSTEM or /GROUP qualifier, the specified protection code overrides any access rights implied by the other qualifiers.

If you specify the /FOREIGN qualifier, the Execute and Delete access codes are synonyms for Logical and Physical. You can, however, specify the access codes P (Physical I/O) or L (Logical I/O), or both, to restrict the nature of input/output operations that different user categories can perform.

To use the /PROTECTION qualifier on a Files-11 volume, you must have the user privilege VOLPRO or your UIC must match the UIC written on the volume.

### example

```
$ MOUNT/PROTECTION=(SYSTEM:RWE,O:RWED,G:RE,W:R) DBA1: WORKDISK
```

This command mounts a device labeled WORKDISK on DBA1 and assigns a protection code. Access to the volume will be READ, WRITE, and EXECUTE for SYSTEM users; READ, WRITE, EXECUTE, and DELETE for OWNER; READ and EXECUTE for GROUP users; and READ-only for users in the WORLD category.

## **/QUOTA**

Controls whether or not quotas are to be enforced on the specified disk volume.

### **format**

**/QUOTA** *device-name*  
**/NOQUOTA** *device-name*

### **description**

The default is **/QUOTA**, which enforces the quotas for each user. The **/NOQUOTA** qualifier inhibits this checking. To specify the **/QUOTA** qualifier, you must have the user privilege **VOLPRO** or your UIC must match the UIC written on the volume.

### **example**

```
$ MOUNT/OWNER_UIC=[016,360]/NOQUOTA DRA3: WORK
```

This command specifies that the disk volume labeled **WORK** on **DRA3** has an owner UIC of **[016,360]** and no quotas enforced.

---

## **/REBUILD**

Controls whether or not **MOUNT** performs a rebuild operation on a disk volume.

### **format**

**/REBUILD** *device-name*  
**/NOREBUILD** *device-name*

### **description**

If a disk volume is improperly dismounted (such as during a system failure), you must rebuild it to recover any caching limits that were enabled on the volume at the time of the dismount. By default, **MOUNT** attempts the rebuild. For a successful rebuild operation that includes reclaiming all of the available free space, you must mount *all* of the volume set members.

The rebuild may consume a considerable amount of time, depending on the number of files on the volume and, if quotas are in use, on the number of different file owners.

## MOUNT-24 MOUNT /REBUILD

The following caches may have been in effect on the volume before it was dismounted:

- Preallocated free space (EXTENT cache)
- Preallocated file numbers (FILE\_ID cache)
- Disk quota usage caching (QUOTA cache)

If caching was in effect for preallocated free space or file numbers, the rebuild time is directly proportional to the greatest number of files that ever existed on the volume at one time. If disk quota caching was in effect, you can expect additional time that is proportional to the square of the number of entries in the disk quota file.

If none of these items were in effect, the rebuild is not necessary and does not occur.

If you use the /NOREBUILD qualifier, devices can be returned to active use immediately. You can then perform the rebuild later with the DCL command SET VOLUME/REBUILD.

### example

```
$ MOUNT/REBUILD NODE$DBA2: WORKDISK
%MOUNT-I-MOUNTED, WORKDISK          mounted on _NODE$DBA2:
%MOUNT-I-REBUILD, volume was improperly dismounted; rebuild in progress
```

In this example, the volume WORKDISK is mounted on NODE\$DBA2. Because the volume is found to have been improperly dismounted and the /REBUILD qualifier is in effect, MOUNT displays a message and proceeds to rebuild the volume.

---

### /RECORDSIZE=*n*

Specifies the number of characters in each record of a magnetic tape volume.

### format

*/RECORDSIZE=*n* device-name*

### qualifier value

*n*

Specifies the block size in the range 20 through 65,532 bytes if you are using VMS RMS, or 18 through 65,534 bytes if you are not using VMS RMS.

## description

You typically use this qualifier with the /FOREIGN and /BLOCKSIZE qualifiers to read or write fixed-length records on a block structured device. In this case, the record size must be less than or equal to the block size specified or used by default.

Use the /RECORDSIZE qualifier when mounting magnetic tapes without HDR2 labels (such as RT-11 magnetic tapes) to provide VMS RMS with default values for the maximum record size.

## example

```
$ MOUNT/FOREIGN/BLOCKSIZE=512/RECORDSIZE=512 MTA0:
```

In this example the magnetic tape is mounted on MTA0 with a default block size and record size of 512 characters.

---

## /SHADOW

Applicable only if you have the volume shadowing option.

---

## /SHARE

Specifies, for a disk volume, that the volume is shareable.

## format

*/SHARE device-name*

*/NOSHARE device-name*

## description

If another user has already mounted the volume shareable, and you request it to be mounted with the /SHARE qualifier, any other qualifiers you enter are ignored.

By default, a volume is not shareable, and the MOUNT command allocates the device on which it is mounted.

If you previously allocated the device and specify the /SHARE qualifier, the MOUNT command deallocates the device so that other users can access it.

## MOUNT-26 MOUNT /SHARE

### example

```
$ MOUNT/NOMESSAGE/SHARE DLA0: SLIP DISC
```

This command mounts the device labeled SLIP on DLA0, disables broadcasting of MOUNT messages, specifies that the volume is shareable, and assigns the logical name DISC.

---

### /SYSTEM

Makes the volume public, that is, available to all users of the system, as long as the UIC-based volume protection allows them access.

### format

*/SYSTEM device-name*

### description

The logical name for the device is placed in the system logical name table. You must have the user privilege SYSNAM to use the /SYSTEM qualifier.

When you mount a volume with the /SYSTEM qualifier in a VAXcluster, you must use a volume label that is unique clusterwide, even if the specified volume is not mounted clusterwide.

### example

```
$ MOUNT/NOMESSAGE/SYSTEM DUA1: SLIP SACH
```

This command mounts the volume labeled SLIP on DUA1 with mount messages disabled. The volume is made available systemwide. MOUNT also assigns the logical name SACH.

---

### /UNLOAD

Controls whether or not the disk or magnetic tape volume or volumes specified in the MOUNT command are unloaded when they are dismounted. The default is /UNLOAD.

### format

*/UNLOAD device-name*

*/NOUNLOAD device-name*



## example

```
$ MOUNT/NOUNLOAD DBA1: OFFENS STRAT
```

In this example, the volume labeled OFFENS is mounted on DBA1 with the /NOUNLOAD qualifier so that it can be dismounted without being physically unloaded. MOUNT also assigns the logical name STRAT.

---

## /WINDOWS

Specifies the number of mapping pointers to be allocated for file windows.

### format

*/WINDOWS=n device-name*

### qualifier value

*n*

Specifies a value from 7 through 80 that overrides the default value specified when the volume was initialized.

### description

When a file is opened, the file system uses the mapping pointers to access data in the file. Use MOUNT/WINDOWS to override the default value specified when the volume was initialized. If no value was specified at volume initialization, the default number of mapping pointers is 7.

You must have the operator user privilege (OPER) to use the /WINDOWS qualifier.

## example

```
$ MOUNT/SYSTEM/WINDOWS=25 DBA2: GONWITH THE_WINDOW
```

This command makes the volume labeled GONWITH on DBA2 available systemwide and assigns the logical name THE\_WINDOW. You override the default number of mapping pointers by specifying a value of 25 for the /WINDOWS qualifier.

**MOUNT-28 MOUNT  
/WRITE**

---

**/WRITE**

Controls whether the volume can be written.

**format**

**/WRITE** *device-name*

**/NOWRITE** *device-name*

**description**

By default, a volume is considered read/write when it is mounted. You can specify **/NOWRITE** to provide read-only access to protect files. This is equivalent to write-locking the device.

**example**

```
$ MOUNT/CLUSTER/NOWRITE NODE$DBA1: BOOKS
```

This command mounts a volume labeled **BOOKS** on **NODE\$DBA1** and then proceeds to mount it on each node in the existing **VAXcluster**. The **/NOWRITE** qualifier makes the volume available for read-only access.

---

## NCP Utility

The Network Control Program (NCP) is a DECnet-VAX utility that accepts terminal commands to configure, control, monitor, and test a DECnet network.

### format

#### **RUN SYS\$SYSTEM:NCP**

To invoke NCP, enter the following DCL command:

```
$ RUN SYS$SYSTEM:NCP
```

NCP returns the following prompt:

```
NCP>
```

Alternatively, you can execute a single NCP command by using a DCL string assignment statement. For example:

```
$ NCP=="$NCP"  
$ NCP SHOW STATUS KNOWN LINES
```

NCP executes the SHOW KNOWN LINES command and returns control to DCL.

To exit from an NCP session, type EXIT or press CTRL/Z after the NCP> prompt.

Output for the SHOW and LIST commands is normally displayed on the default output device, SYS\$OUTPUT. Alternatively, you may direct output to a specified file using the TO qualifier with the SHOW or LIST command.

You can use the asterisk (\*) and the percent sign (%) as wildcard characters in an NCP command line to refer to a group of NCP components by a general name, rather than specifying each component name individually.

The wildcard characters can be used to represent the following component names:

- Node name
- Node address
- Circuit name
- Line name
- Object name
- Events

The asterisk wildcard represents one or more characters, while the percent sign represents a single character.

## **NCP-2    NCP Utility**

You need certain privileges to use most NCP commands. The SET and DEFINE commands require the operator privilege (OPER), and the LIST command requires the system privilege (SYSPRV). The SHOW command does not require any privileges.

## NCP Commands

The following section presents the NCP commands in alphabetical order.

**NOTE:** The following section contains only a subset of the complete set of NCP commands. Also, this section includes only the most commonly used command parameters and qualifiers for the commands listed.

You can abbreviate any command verb, component, parameter, or qualifier as long as the abbreviation is not ambiguous. Certain words provide syntactic clarity but are optional. If omission of a word in an NCP command line produces an unambiguous result, that word is optional.

For convenience, commands that have the same components and parameters, yet different command verbs—depending upon whether they access the volatile or the permanent database—are listed together. Examples of commands listed together are SET and DEFINE, and SHOW and LIST, where SET, and SHOW verbs apply to the volatile database and DEFINE and LIST verbs apply to the permanent database. When two commands are grouped together, components and parameters are described for the command that accesses the volatile database. Typically, the actions described for the volatile database also apply to the permanent database.

When you issue NCP commands, many components, parameters, and qualifiers require you to supply additional information. For the most part, their syntax follows a standard set of rules. Exceptions to these rules are documented in the description of the component, parameter, or qualifier to which they apply.

The syntax of the various component-name, parameter, and qualifier values is summarized below. In the following list, all numeric values are in decimal and have a range of 0 to 65,535 unless otherwise specified:

area-number	A decimal value in the range 1 to 63 to be specified in the beginning of the <i>node-address</i> and separated from the <i>node number</i> by a period. If you do not specify an area number, the area number of the executor is used. The default area number for the executor is 1.
-------------	---

# NCP-4 NCP NCP Commands

**circuit-id** A string of characters whose exact syntax is that for a DECnet circuit identification.

Circuit identification takes one of the following formats:

dev-c  
dev-c-u

**dev** Is a device name. Refer to the description of device-type for a list of device mnemonic names.

**c** Is a decimal number (0 or a positive integer) designating the device's hardware controller.

**u** Is a decimal unit or circuit number (0 or a positive integer) included only if more than one unit is associated with the controller.

**count** A decimal numeric value.

**device-type** A string of characters representing the mnemonic name for the device. Devices supported by DECnet-VAX include the following:

BNA DMF TT  
CI DMP TX  
DMB QNA  
DMC SVA

**E-address** A string of 12 hexadecimal digits, represented by 6 bytes separated by hyphens (for example, AA-00-04-00-AB-04). The string indicates an Ethernet address. The bytes are ordered from left to right as transmitted and received on the Ethernet.

**event-list** A list of event types for a given class in the format class.type. When specifying an event list, you may specify only one class; however, you can specify a range of types by using commas and hyphens, for example, 4.3-5,7-10. The following table provides examples of these formats.

<b>Event List</b>	<b>Meaning</b>
4.4	Identifies event class 4, type 4.
4.5-7	Identifies event class 4, types 5 through 7.
4.5,7-9,11	Identifies event class 4, types 5, 7 through 9, and 11. Note that types must be specified in ascending order.

**filespec** A VMS file specification string in the following general format:

node-spec::device:[directory]filename.type;version

Logical names are permitted. For a file in your current directory, you need specify only a file name of up to 39 alphanumeric characters, optionally followed by a period and a file type of up to 39 alphanumeric characters.

NCP    NCP-5  
NCP Commands

hex-password	A string of up to 8 hexadecimal digits.
id-string	A string of up to 32 characters. If the string includes spaces or tabs, enclose it within quotation marks.
line-id	A string of characters whose exact syntax is that for a DECnet line identification. For VMS operating systems, <i>line-id</i> takes one of the following formats:  dev-c dev-c-u  dev    Is a device name. Refer to the description of <i>device-type</i> for a list of device mnemonic names.  c      Is a decimal number (0 or a positive integer) designating the device's hardware controller.  u      Is a decimal unit or line number (0 or a positive integer) included if the device is a multiple unit line controller.
milliseconds	A decimal numeric value.
node-address	A numeric value in the range 1.1 to 63.1023, composed of an area number to the left of the period followed by a node number to the right of the period. (The node number indicates the address of the node within the specified area.) If the area number is not supplied, the area number of the executor node is used. The default area number for the executor is 1.
node-id	Either a <i>node-name</i> or a <i>node-address</i> .
node-name	A string of up to six alphanumeric characters containing at least one alphabetic character.
node-spec	A <i>node-id</i> followed by optional access control information as specified for VMS operating systems in the following format:  node-id"user password account"
node-type	A string of characters consisting of one of the following:  Routing III Nonrouting III Routing IV Nonrouting IV Area
number	A decimal numeric value.
object-name	A string of up to 12 printable characters.
password	A string of up to 39 printable characters.
privilege-list	A list of VMS privilege names delimited by space characters.
seconds	A decimal numeric value.
user-id	A string of up to 39 alphanumeric and hyphen characters.

## NCP-6 NCP

### COPY KNOWN NODES

Quotation mark delimiters are valid for the *node-spec* format. In addition, you can use quotation marks as delimiters when providing receive and transmit passwords for the SET NODE and DEFINE NODE commands. For example:

```
NCP>SET NODE TRANSMIT PASSWORD "HI VAX"
```

Also, use quotation marks to delimit the software identification string specified for the IDENTIFICATION parameter of the SET EXECUTOR command. For example:

```
NCP>SET EXECUTOR IDENTIFICATION "VMS HOST SYSTEM"
```

---

## COPY KNOWN NODES

The COPY KNOWN NODES command updates the node database on the local node. You can copy the volatile or permanent node database from a remote node to either or both the volatile and permanent node databases on the local node. You also have the option of clearing or purging the node database on the local node before beginning the copy operation.

Only the node name and node address are copied. A node entry will not be copied into the node database if it would result in the association of two names with one address or two addresses with one name.

The TELL prefix cannot be used with this command.

### format

```
COPY node-component parameter [qualifier] [...]
```

### node component

#### **KNOWN NODES**

Indicates that names and addresses of all known nodes stored in the database of the specified remote node are to be copied.

### command parameter

#### **FROM node-id**

Specifies the remote node from which node database information is to be copied. The remote node can be any node in the network to which you have access. The word FROM is optional.

### qualifiers

#### **USING option**

Specifies the node database on the remote node from which the information is to be copied. There are two possible options:

**VOLATILE**            Indicates that the volatile database on the remote node is to be copied.



**PERMANENT** Indicates that the permanent database on the remote node is to be copied.

The default is **VOLATILE**.

***TO option***

Specifies the node database on the local node to which the information is to be copied. There are three possible options:

**VOLATILE** Indicates that the information is to be copied to the volatile database on the local node.

**PERMANENT** Indicates that the information is to be copied to the permanent database on the local node.

**BOTH** Indicates that the information is to be copied to both the volatile and permanent databases on the local node.

The default is **VOLATILE**.

***WITH option***

Clears or purges the node database on the local node before the copy operation is performed. Retains the executor node characteristics and the name and address of the remote node from which the node information is to be copied. The node database to be cleared or purged is the local database to which the information will be copied. There are two options:

**CLEAR** Clears the volatile node database at the local node.

**PURGE** Purges the permanent node database at the local node.

Note that you can actually specify either **CLEAR** or **PURGE** for either database or for both databases.

If you do not specify the **WITH** qualifier, the node entries copied are added to the existing node database(s).

**example**

```
NCP>LIST KNOWN NODES
```

```
Known Node Permanent Summary as of 30-DEC-1988 13:50:20
```

```
Executor node = 2.20 (ROBIN)  
State = on
```

```
Remote node = 2.21 (THRUSH)  
No information available
```

```
Remote node = 2.22 (LARK)  
No information available
```

```
NCP>TELL LARK LIST KNOWN NODES
```

```
Known Node Permanent Summary as of 30-DEC-1989 13:50:20
```

```
Executor node = 2.22 (LARK)  
State = on
```

## NCP-8 NCP

### COPY KNOWN NODES

Remote node = 2.20 (ROBIN)  
No information available

Remote node = 2.23 (DOVE)  
No information available

NCP>COPY KNOWN NODES FROM LARK USING PERMANENT -  
\_ TO PERMANENT WITH PURGE

%NCP-I-SUCCESS - Success  
Remote node = 2.21 (THRUSH)  
%NCP-I-RECDELETE, Database entry deleted  
%NCP-I-SUCCESS - Success  
Remote node = 2.22 (LARK)  
%NCP-I-RECDELETE, Database entry deleted  
%NCP-I-SUCCESS - Success  
Executor node = 2.20 (ROBIN)  
%NCP-I-RECDELETE, Database entry deleted

NCP>LIST KNOWN NODES

Known Node Permanent Summary as of 30-DEC-1989 14:01:05

Executor node = 2.20 (ROBIN)  
State = on

Remote node = 2.22 (LARK)  
No information available

Remote node = 2.23 (DOVE)  
No information available

This copy command copies the node information from the permanent node database on node LARK into the permanent node database on the local node (ROBIN). The node database is purged before the copy operation is begun.

---

## SET/DEFINE CIRCUIT

The SET CIRCUIT command creates or modifies circuit parameters in the volatile database on the local node. The DEFINE CIRCUIT command creates or modifies circuit parameters in the permanent database on the local node. The circuit must be in the OFF state before you modify any parameters other than COST, COUNTER TIMER, STATE, or VERIFICATION.

### format

**SET** *circuit-component parameter [...]*  
**DEFINE** *circuit-component parameter [...]*

## circuit components

### ***CIRCUIT circuit-id***

Identifies the circuit whose parameters are to be updated.

### ***KNOWN CIRCUITS***

Indicates that parameters for all known circuits are to be updated.

## command parameters

### ***COST cost***

Specifies the routing cost of the circuit. Messages travel between nodes along the path with the smallest total cost. The *cost* value must be a decimal integer in the range 1 to 25. The default value is 10.

### ***COUNTER TIMER seconds***

Specifies the number of seconds that the circuit counter timer will run. When the counter timer expires, a circuit counter logging event occurs. The *seconds* value must be a decimal integer in the range 0 to 65,535. If no value is set for COUNTER TIMER, the circuit counters are not logged automatically.

### ***HELLO TIMER seconds***

Specifies the frequency of Routing Hello messages sent to adjacent nodes on the circuit. The *seconds* value must be a decimal integer in the range 0 to 8191. The default value is 15. The value of the read-only circuit parameter LISTEN TIMER is three times the value of the HELLO TIMER parameter.

### ***MAXIMUM ROUTERS number***

Applies only to Ethernet circuits. Specifies the maximum number of routers (other than the executor node) allowed by the Routing layer on this circuit. Use a number in the range 1 to 33. The default value is 33.

### ***ROUTER PRIORITY number***

Applies only to Ethernet circuits. Specifies the priority this router (the executor node on this circuit) is to have in the selection of a designated router for this circuit. Use a value in the range 0 to 127. The default is 64.

### ***SERVICE service-mode***

Specifies whether service operations (loading and loop testing) are allowed for the circuit. There are two possible modes:

- |          |  |
|----------|--|
| DISABLED | The circuit cannot be put into service state and cannot perform service functions. The default mode is DISABLED. |
| ENABLED  | The circuit can be put into service state and perform service functions.   |

## NCP-10 NCP SET/DEFINE CIRCUIT

### **STATE circuit-state**

Specifies the circuit's operational state. There are three possible states:

OFF	The circuit is not in use.
ON	The circuit is available for normal use or service functions.
SERVICE	The circuit is available for service functions only.

### **TRANSMIT TIMER milliseconds**

Defines the number of milliseconds to delay between data message transmits. The *milliseconds* value must be a decimal integer up to 65,535. The default is 0.

### **VERIFICATION option**

Applies only to synchronous and asynchronous circuits. Requires the remote node to send its routing initialization password. There are three options:

DISABLED	Does not require the remote node to send its routing initialization password. This is the default.
ENABLED	Requires the remote node to send its routing initialization password.
INBOUND	Applies to any DDCMP point-to-point circuit. Specifies that the executor node expects to receive a routing initialization password for verification from a remote node before a connection is made between the nodes. The executor is prohibited from sending its routing initialization password to the remote node. This parameter is specified automatically for dynamic asynchronous DDCMP circuits. If you specify the VERIFICATION INBOUND parameter for a circuit, you must specify the INBOUND node parameter (by using the SET/DEFINE NODE command) for the remote node.

## **example**

```
NCP>SET CIRCUIT UNA-0 STATE ON MAXIMUM ROUTERS 5
```

This command sets Ethernet circuit UNA-0 to ON and sets the maximum number of routers permitted on the circuit to 5.

---

## **SET/DEFINE EXECUTOR**

The SET EXECUTOR command creates or modifies parameters in the volatile database that controls the network on the local node. The DEFINE EXECUTOR command creates or modifies parameters in the permanent database that controls the network on the local node.

After the local node's state is set to ON, you cannot change the ADDRESS, ALIAS NODE, ALIAS INCOMING, BUFFER SIZE, NAME, or TYPE parameter for the local node. If the local node whose state is ON is connected to an Ethernet circuit whose state is ON, you cannot change the MAXIMUM CIRCUITS parameter for the local node.

The SET EXECUTOR command cannot be used with the TELL prefix.

## format

SET EXECUTOR *parameter* [...]  
DEFINE EXECUTOR *parameter* [...]

## command parameters

### **ADDRESS** *node-address*

Establishes a node address for the local node, in the following format:

area-number.node-number

where:

<b>area-number</b>	Is in the range 1 to 63.
<b>node-number</b>	Is in the range 1 to 1023.

If you do not specify *area-number*, the default value is 1. You need not supply the area number in the *node-address* if your node is in area 1. When you configure the local node, this parameter is required.

### **ALIAS INCOMING** *option*

Specifies whether the local node accepts incoming connect requests directed to the alias node identifier specified for the local node. The alias node identifier is described under the ALIAS NODE parameter. There are two options for ALIAS INCOMING:

DISABLED	Specifies that the local node will not accept incoming connect requests directed to the alias node identifier.
ENABLED	Specifies that the local node will accept incoming connect requests directed to the alias node identifier. This is the default if an alias node identifier has been specified.

### **ALIAS MAXIMUM LINKS** *number*

Specifies the maximum number of logical links for the local node that can use the alias node identifier. The alias node identifier is described under the ALIAS NODE parameter. The maximum value for ALIAS MAXIMUM LINKS is 200. The default value is 32.

### **ALIAS NODE** *node-id*

Establishes a cluster alias node identifier for use by the local node. The *node-id* is a DECnet node identifier that can be either a node name or a node address. This alias permits the local node to be associated with a cluster node identifier common to some or all nodes in the cluster, in addition to its own unique *node-id*. If you do not specify this parameter, the local node is not associated with a cluster alias node identifier. If a node name is to be used as the alias *node-id*, the node name must previously have been defined in the database.

**NCP-12    NCP**  
**SET/DEFINE EXECUTOR**

***AREA MAXIMUM COST number***

Applies only to an executor node whose type is AREA. Specifies the maximum total path cost allowed from the executor to any other level 2 routing node (area router). You can specify a decimal value in the range 1 to 1022. The default is 1022.

***AREA MAXIMUM HOPS number***

Applies only to an executor node whose type is AREA. Specifies the maximum number of routing hops allowable from the executor to any other level 2 routing node. You can specify a decimal value in the range 1 to 30. The default is 30.

***BROADCAST ROUTING TIMER seconds***

Specifies the maximum amount of time allowed between routing updates on Ethernet circuits. When the timer expires before a routing update occurs, a routing update is forced. The routing update produces a routing configuration message for each adjacent node. You can specify a number in the range 1 to 65,535. The default is 40.

***BUFFER SIZE number***

Specifies in bytes the size of the receive buffers, thereby controlling the maximum size of NSP message segments that can be received and forwarded. (The size includes protocol overhead down to and including the End Communication layer, but does not include the Data Link layer overhead.) This buffer size applies to all circuits known to the executor. Use a value up to a maximum of 65,535. The default value is equal to the value of the SEGMENT BUFFER SIZE, if specified; otherwise the default is 576.

***COUNTER TIMER seconds***

Specifies a timer whose expiration causes a node counter logging event.

***DEFAULT ACCESS option***

Assigns the default access to all nodes that do not have a specific node ACCESS entry in the volatile database. There are four options:

BOTH	Allows incoming and outgoing logical link connections. This is the default.
INCOMING	Allows logical link connections from the remote node.
NONE	Does not allow incoming or outgoing logical link connections to this node.
OUTGOING	Allows the local node to initiate connections to the remote node, but does not allow connections from the remote node.

If you have OPER privilege on the local system, you can override the default access restriction specified in this parameter.

***DELAY FACTOR number***

Specifies the number by which to multiply one-sixteenth of the estimated round trip delay to a node to set the retransmission timer to that node. Use a number up to a maximum of 255. If you do not set this parameter, the default value is 80.

***DELAY WEIGHT number***

Specifies the weight to apply to a new round-trip delay data point when updating the estimated round-trip delay to a node. Use a number in the range up to a maximum of 255. If you do not set this parameter, the default value is 5.

***IDENTIFICATION id-string***

Specifies a text string that describes the executor node (for example, "VMS Host System"). The string can be a maximum of 32 characters. If it contains blanks or tabs, you must enclose the string in quotation marks. If you do not set this parameter, the default value is DECnet-VAX V5.n VMS V5.n.

***INACTIVITY TIMER seconds***

Specifies the maximum duration of inactivity (no data in either direction) on a logical link before the node checks to see if the logical link still works. If you do not set this parameter, the default value is 60.

***INCOMING PROXY option***

Indicates whether proxy login requests present on incoming logical links are to be honored. There are two options for INCOMING PROXY:

DISABLED

Ignores all incoming proxy requests and instead relies exclusively on access control information supplied in the connect requests to validate the logical link.

ENABLED

Invokes the appropriate proxy, based on the source user, source node, and supplied access control information (if any). This is the default.

Note that proxy access characteristics established in the object database take preference over the proxy access characteristics established in the executor database.

***INCOMING TIMER seconds***

Specifies the maximum amount of elapsed time between the time a connection is received for a process and the time that process accepts or rejects the connection. For very busy systems, use a value in the range of 45 to 60 seconds. Otherwise use a value of 30 seconds. The default value is 45.

***MAXIMUM ADDRESS number***

Defines the largest node address and, consequently, the greatest number of nodes that can be addressed by the local node. Use as small a number as possible. The default value is 1023.

**MAXIMUM AREA number**

Applies only to an executor node whose type is AREA. Specifies the largest area number and, therefore, the greatest number of areas that can be known about by the executor node's Routing layer. You can specify a decimal value up to a maximum of 63. The default is 63.

**MAXIMUM BROADCAST NONROUTERS number**

Specifies the maximum total number of nonrouting nodes (end nodes) the executor node can have on its Ethernet circuits. Use a number up to a maximum of 65,535. The default value is 64.

**MAXIMUM BROADCAST ROUTERS number**

Specifies the maximum total number of routers the executor node can have on its Ethernet circuits. Use a number up to a maximum of 65,535. The default value is 32.

**MAXIMUM BUFFERS number**

Specifies the maximum number of buffers in the transmit buffer pool. DECnet normally allocates only what it needs. At minimum, use a value that is 15 times the square root of the number of lines. Increase this value if you experience congestion loss. The default value is 100.

**MAXIMUM CIRCUITS number**

Defines the maximum number of routing circuits that the local node can use. The number must be in the range 1 to 127. The default value is 16.

**MAXIMUM COST number**

Specifies the maximum total path cost allowed from the local node to any node. The path cost is the sum of the circuit costs along a path between two nodes. Use as small a number as possible in the range of 1 to 1022. The default is 1022.

**MAXIMUM HOPS number**

Specifies the maximum routing hops from the local node to any other reachable node. A hop is the logical distance over a circuit between two adjacent nodes. Use as small a number as possible in the range of 1 to 30, and be sure that this value is less than or equal to the MAXIMUM VISITS parameter. The default value is 30.

**MAXIMUM LINKS number**

Specifies the maximum logical link count for the local node. A reasonable range for most networks is 25 to 50. The maximum value for MAXIMUM LINKS is 3885. The default value of MAXIMUM LINKS is 32.

**MAXIMUM PATH SPLITS number**

Indicates the maximum number of equal cost paths to a given destination node among which the packet load may be split. The default value is 1.



**MAXIMUM VISITS number**

Specifies the maximum number of nodes a message can visit before it is received by the destination node. Use a number in the range of the value of the MAXIMUM HOPS parameter to 63. You should specify a number that is twice the MAXIMUM HOPS value. The default value is 63.

**NAME node-name**

Specifies the node name to be associated with the executor node identification. You can assign only one name to a node address or node identification.

**NONPRIVILEGED item**

Specifies nonprivileged inbound access control information for the node. Associate any of the following parameters with the NONPRIVILEGED parameter:

ACCOUNT account	Identifies the account for the default nonprivileged DECnet account on the executor node.
PASSWORD password	Identifies the password for the default nonprivileged DECnet account on the executor node.
USER user-id	Identifies the user name for the default nonprivileged DECnet account on the executor node.

**OUTGOING PROXY option**

Indicates whether proxy login may be used on outgoing connect requests. There are two options for OUTGOING PROXY.

DISABLED	Specifies that proxy login is not requested on any outgoing logical links.
ENABLED	Specifies that proxy login is requested on outgoing logical links. This is the default.

Note that proxy access characteristics established in the object database take preference over the proxy access characteristics established in the executor database.

**OUTGOING TIMER seconds**

Specifies the timeout value for the elapsed time between the moment a connection is requested and the moment that connection is acknowledged by the destination node. A value in the range of 30 to 60 seconds is recommended. The default is 45.

**PATH SPLIT POLICY policy**

Specifies the policy for equal cost path splitting of network traffic. There are two values for PATH SPLIT POLICY:

**NCP-16    NCP  
SET/DEFINE EXECUTOR**

<b>INTERIM</b>	Specifies that traffic will be split over all equal cost paths while forcing packets for individual network sessions to follow the same paths in order to guarantee that packets will be received by the destination node in the correct order. The INTERIM value should be set if some of the nodes in the network do not support out-of-order packet caching. (DECnet-VAX Version 4.5 and lower DECnet-VAX versions do not support out-of-order packet caching.)
<b>NORMAL</b>	Specifies that all traffic will be split equally over all equal cost paths to a destination node. All destination nodes must support out-of-order packet caching (supported by DECnet-VAX Version 4.6 or higher); otherwise, network performance may suffer. This is the default.

***PIPELINE QUOTA quota***

Specifies the maximum number of bytes of nonpaged pool that DECnet will use for transmission over logical links. Use this parameter for multibuffering at the NSP level. The default value is 10000 bytes. For satellite communications, a value of 6000 or greater is recommended.

***PRIVILEGED item***

Specifies privileged inbound access control information for the node. Associate any of the following parameters with the PRIVILEGED parameter:

<b>ACCOUNT account</b>	Identifies the account for the default privileged DECnet account on the executor node.
<b>PASSWORD password</b>	Identifies the password for the default privileged DECnet account on the executor node.
<b>USER user-id</b>	Identifies the user name for the default privileged DECnet account on the executor node.

These parameters are not needed unless the PRIVILEGES parameter is used explicitly in the object database.

***RETRANSMIT FACTOR number***

Defines the maximum number of times any given message (except a connect initiate message) will be retransmitted before the logical link is disconnected. If you do not set this parameter, the default value is 10.

***ROUTING TIMER seconds***

Specifies the maximum amount of elapsed time before a routing update is forced on non-Ethernet circuits. The routing update produces a routing configuration message for each adjacent node. You can use a number up to a maximum of 65,535. If you do not set this parameter, the default value is 600.

**SEGMENT BUFFER SIZE *number***

Specifies in bytes the maximum size of transmit buffers, thereby controlling the maximum size NSP message segment that can be transmitted. (This value is the maximum size message the End Communications layer can transmit; it does not include Data Link layer overhead.) Use a value up to a maximum of 65,535. The default value is equal to the value of BUFFER SIZE, if specified; otherwise, the default is 576.

The SEGMENT BUFFER SIZE is always less than or equal to the BUFFER SIZE. The two values are normally equal but may differ to permit the network manager to alter buffer sizes on all nodes without interruption of service.

**STATE *node-state***

Specifies the operational state of the local node. There are four possible states:

OFF	Allows no new logical links, terminates existing links, and stops route-through traffic.
ON	Allows logical links.
RESTRICTED	Allows no new inbound links from other nodes.
SHUT	Allows no new logical links, does not destroy existing links, and goes to the OFF state when all logical links are disconnected.

If you have OPER privilege, you can override the state value specified in this parameter.

**TYPE *node-type***

Indicates the type of the executor node. There are three possible node types:

AREA  
NONROUTING IV  
ROUTING IV

The default depends upon the DECnet-VAX license registered. If the full function kit is installed, the default is ROUTING IV; if the end node kit is installed, the default (and only possible value) is NONROUTING IV.

A routing node has full routing capability. A nonrouting node (or end node) can deliver packets to or receive them from any node, but cannot route packets from other source nodes through to destination nodes.

An area node is a level 2 router that can route packets between areas.

**NCP-18    NCP**  
**SET/DEFINE EXECUTOR**

**example**

```
NCP>SET NODE 2.13 NAME CLUSTR
```

```
.  
.  
.
```

```
NCP>SET EXECUTOR ALIAS NODE CLUSTR
```

The **SET NODE** command establishes a node address 2.13 with the associated node name **CLUSTR**. The **SET EXECUTOR ALIAS NODE** command then establishes the node name **CLUSTR** as the alias node identifier.

---

**SET/DEFINE LINE**

The **SET LINE** command creates or modifies line parameters in the volatile database on the local node. The **DEFINE LINE** command creates or modifies line parameters in the permanent database on the local node. A line must be in the **OFF** state in order for all but the **COUNTER**, **TIMER**, **SERVICE TIMER**, and **STATE** parameters to be changed.

**format**

```
SET    line-component parameter [...]  
DEFINE line-component parameter [...]
```

**line components**

***LINE line-id***

Identifies the line for which specified parameters are to be created or modified in the volatile database.

***KNOWN LINES***

Indicates that the specified parameters for all known lines are to be created or modified in the volatile database.

**command parameters**

***BUFFER SIZE number***

Specifies in bytes the size of receive buffers for the specified line, thereby controlling the maximum size of NSP message segments that can be received from or forwarded to an adjacent node that has accepted the line buffer size. Use a value up to a maximum of 65,535. For Ethernet lines, a default value of 1498 bytes is provided. For all other types of line, the default is the executor **BUFFER SIZE** value (as specified in the **SET EXECUTOR** command).

You can use the line parameter **BUFFER SIZE** to increase the size of NSP messages for logical links over this line.

***CLOCK clock-mode***

Applies only to synchronous DDCMP lines. Specifies the hardware clock mode for the line. There are two values for *clock-mode*:

- EXTERNAL    For normal clock operating mode. The clock signal is supplied externally to the controller.
- INTERNAL    For use of the clock in test mode. Setting this value causes the line device to supply a clock signal that will allow all transmitted messages to be looped back from outside the device. Note that, in order to use this parameter, the operator may have to connect a loopback plug in place of the normal line.

***CONTROLLER mode***

Specifies the controller mode for the line. There are two possible modes:

- LOOPBACK    Internal device loopback mode
- NORMAL        Normal operating mode, which is the default

***COUNTER TIMER seconds***

Specifies a timer whose expiration causes a line counter logging event. Specify a decimal integer up to a maximum of 65,535.

***DUPLEX mode***

Does not apply to Ethernet lines. Specifies the hardware duplex mode of the line. There are two possible modes:

- FULL    Full-duplex (default)
- HALF    Half-duplex

***HANGUP option***

Applies only to asynchronous DDCMP lines. Indicates whether the modem signals are dropped when the line is shut down. There are two possible options:

- DISABLED    Indicates that modem signals should not be dropped when the line is shut down. This is the default for static asynchronous DDCMP lines.
- ENABLED    Indicates that modem signals should be dropped when the line is shut down.

This parameter is supplied automatically for dynamic asynchronous DDCMP lines. The default is HANGUP ENABLED if the /HANGUP qualifier was specified for the DCL command SET TERMINAL, and HANGUP DISABLED if /NOHANGUP was specified.

***LINE SPEED number***

Applies only to asynchronous DDCMP lines. Specifies the speed of the line in baud. This parameter must be set to the same value on both sides of an asynchronous DDCMP connection. It is specified automatically for dynamic asynchronous DDCMP lines. If not specified, the value of this parameter is equal to the current speed of the line.

**NCP-20    NCP**  
**SET/DEFINE LINE**

***PROTOCOL protocol-name***

Defines the Data Link protocol to be used on this line. The following values can be used for *protocol-name*:

DDCMP CONTROL	Specifies this line as a multipoint control station. You can specify multiple circuits for CONTROL lines, but each circuit must have a unique physical tributary address.
DDCMP DMC	Specifies that this line is in DMC emulator mode. DMC is similar to POINT, except that DMC uses an older version of DDCMP (Version 3.2). This protocol should be set for the local line when the remote line is a DMC. Note that this protocol is valid only when a DMP11 or DMV11 is being used.
DDCMP POINT	Defines this line as one end of a point-to-point DDCMP connection. You may specify only one circuit per POINT line.
DDCMP TRIBUTARY	Specifies that this line is a tributary end of a DDCMP multipoint group. You may specify only one circuit per TRIBUTARY line.
ETHERNET	Specifies that this line uses the Ethernet protocol.

Default line protocols based on line names are as follows:

BNA	ETHERNET
CI	No protocol specified
DMB	DDCMP POINT
DMC/DMR	DDCMP POINT
DMF	DDCMP POINT
DMP/DMV	DDCMP POINT
DPV	LAPB
QNA	ETHERNET
SVA	ETHERNET
UNA	ETHERNET

***RECEIVE BUFFERS number***

Specifies the length of the line's receive queue. Use a value in the range 1 to 32. A value in the range 2 to 4 is adequate for line speeds of less than 56 kilobits/second. Line speeds of 1 megabit/second may require eight or more buffers depending on the observed error rate.

***STATE line-state***

Specifies the line's operational state. The possible states include the following:

OFF	The line is not in use.
ON	The line is available for normal use or service functions.

***SWITCH option***

Applies only to asynchronous DDCMP lines. Forces the line currently being used as a DECnet asynchronous communications line to be converted back to a terminal line. There are two values for *option*:

- DISABLED    The line is not switched to a terminal line. This is the default for static lines.
- ENABLED    The line is switched to a terminal line after it is disconnected from the network (when the channel to the network is deassigned). This is the default for dynamic lines.

***TRANSMIT PIPELINE count***

Applies only to DMR11 lines. Specifies the maximum number of DDCMP messages for which outstanding acknowledgments are allowed. Specify a value in the range 1 to 32. By default, the value for outstanding DDCMP messages is 7. To avoid excessive use of system memory, do not arbitrarily set this value higher than necessary.

**example**

```
NCP>SET LINE UNA-0 STATE ON
```

This command sets Ethernet line UNA-0 to the ON state.

---

**SET/DEFINE NODE**

The SET NODE command creates or modifies node parameters in the volatile database on the local node. The DEFINE NODE command creates or modifies node parameters in the permanent database on the local node.

**format**

```
SET  node-component parameter [...]  
DEFINE node-component parameter [...]
```

**node components**

***NODE node-id***

Identifies the node (local or remote) for which specified parameters are to be created or modified in the database.

***KNOWN NODES***

Indicates that the specified parameters for all known nodes are to be created or modified in the database.

## command parameters

### ***ACCESS option***

Specifies the allowed logical link connections for the node. There are four options:

BOTH	Allows incoming and outgoing logical link connections. This is the default.
INCOMING	Allows logical link connections from the remote node.
NONE	Does not allow incoming or outgoing logical link connections to this node.
OUTGOING	Allows the local node to initiate connections to the remote node, but does not allow connections from the remote node.

If you have OPER privilege, you can override the access restriction specified in this parameter.

### ***ADDRESS node-address***

Specifies the address of the node to which you want the database entry to refer.

### ***COUNTER TIMER seconds***

Specifies a timer whose expiration causes a node counter logging event.

### ***CPU cpu-type***

Identifies the node's CPU type. There are four possibilities:

DECSYSTEM1020  
PDP11  
PDP8  
VAX

### ***HARDWARE ADDRESS E-address***

Identifies the Ethernet address originally assigned to the Ethernet controller for the system on the adjacent node. Used during operations to communicate with the system before the system has set up its physical address.

### ***INBOUND node-type***

Required for nodes when the VERIFICATION INBOUND parameter is specified for the circuit over which the connection is to be made. Specifies the type of the node. The *node-type* is checked by the executor node if the specified node attempts to form a dynamic connection with the executor node. If VERIFICATION INBOUND is not specified for the circuit, the INBOUND parameter for the node is ignored. There are two possible node types:



- ENDNODE    Allows the remote node to be connected only if it is configured as an end node.
- ROUTER     Allows the remote node to be connected whether it is configured as an end node or a router.

***NAME node-name***

Specifies the node name to be associated with the node identification. You can assign only one name to a node address or line identification.

***NONPRIVILEGED item***

Specifies nonprivileged inbound access control information for the node. Associate any of the following parameters with the NONPRIVILEGED parameter:

- ACCOUNT account                      Identifies the account for the default nonprivileged DECnet account on the designated node.
- PASSWORD password                    Identifies the password for the default nonprivileged DECnet account on the designated node.
- USER user-id                            Identifies the user name for the default nonprivileged DECnet account on the designated node.

***PRIVILEGED item***

Specifies privileged inbound access control information for the node. Associate any of the following parameters with the PRIVILEGED parameter:

- ACCOUNT account                      Identifies the account for the default privileged DECnet account on the designated node.
- PASSWORD password                    Identifies the password for the default privileged DECnet account on the designated node.
- USER user-id                            Identifies the user name for the default privileged DECnet account on the designated node.

***RECEIVE PASSWORD password***

Does not apply to nodes on an Ethernet circuit. Defines the password (1 to 8 characters) that is expected from the remote node during a routing initialization sequence. You use this parameter only if verification is enabled or set to INBOUND for the circuit.

***TRANSMIT PASSWORD password***

Does not apply to nodes on an Ethernet circuit. Specifies a password (1 to 8 characters) sent to the remote node during a routing initialization sequence. This parameter is used only if the VERIFICATION parameter has been set to ENABLED or INBOUND for the circuit.

## NCP-24 NCP SET/DEFINE NODE

### example

```
NCP>SET NODE 14 ADDRESS 2
```

This command associates the information for node 1.14 with a new node whose address is 1.2. The executor is assumed to be in area 1.

---

## SET/DEFINE OBJECT

The SET OBJECT command creates or modifies object parameters in the volatile database on the local node. The DEFINE OBJECT command creates or modifies object parameters in the permanent database on the local node.

### format

```
SET object-component parameter [...]  
DEFINE object-component parameter [...]
```

### object component

#### **OBJECT *object-name***

Identifies the object for which specified parameters are to be created or modified in the database.

### command parameters

#### **ACCOUNT *account***

Identifies the default user's account for access control on inbound connects to the object when no access control is specified by the remote node.

#### **ALIAS INCOMING *option***

Specifies how a particular object responds to incoming connect requests directed to the alias node address. You establish the alias node address using the SET EXECUTOR command. There are two options for ALIAS INCOMING.

**DISABLED** Does not allow a specified object to receive incoming connect requests that have been directed to the alias node address. An object whose resources are not accessible clusterwide should have ALIAS INCOMING disabled. If an attempt is made to connect to an object that does not have ALIAS INCOMING enabled, the status message NO SUCH OBJECT is returned.

**ENABLED**      Allows a specified object to receive incoming connect requests that have been directed to the alias node address. An object such as PHONE, which uses a protocol that depends on multiple links, should not be enabled for ALIAS INCOMING. By default, if an alias node identifier has been specified, ALIAS INCOMING is enabled for all objects except for PHONE.

***ALIAS OUTGOING option***

Specifies whether a particular object uses the alias node identifier specified in the SET EXECUTOR command in its outgoing connect requests and other protocols. Specify either of the following two options:

**DISABLED**      Does not allow a specified object to use the alias node address in its outgoing connect requests.

**ENABLED**      Allows a specified object to use the alias node address in its outgoing connect requests. An object such as PHONE, which uses a protocol that depends on multiple links, should not have the ALIAS OUTGOING parameter enabled. By default, only the object MAIL has ALIAS OUTGOING enabled.

***FILE filespec***

Specifies the command file containing the command procedure used to start the indicated object. If not specified, the default is SYS\$SYSTEM:object-name.COM. When you specify an object for the first time, this parameter is mandatory.

***NUMBER number***

Specifies the object number. Use a number up to a maximum of 255, except for those reserved. See Table NCP-1 for a list of reserved object numbers. When you specify an object for the first time, this parameter is mandatory.

***PASSWORD password***

Identifies the default user's password for access control on inbound connects to the object when no access control is specified by the remote node. This password must match the password established for the account.

***PRIVILEGES privilege-list***

Specifies those privileges normally required by the object. A user with those privileges may be supplied with default outbound privileged access control information when connecting to the object.

***PROXY option***

Assigns the proxy login access defaults to individual objects. Specify one of the following four options:

**BOTH**            Allow both incoming and outgoing proxy login access. This is the default option.

**NCP-26    NCP**  
**SET/DEFINE OBJECT**

- INCOMING    Allows proxy login to the object.
- NONE        Does not allow incoming or outgoing proxy login access.
- OUTGOING    Allows the object to initiate proxy login.

***USER user-id***

Identifies the default user's identification for access control on inbound connects to the object when no access control is specified by the remote node.

**description**

A DECnet object is identified by object name and object type. (The type is specified in the NUMBER parameter.)

The privilege list in the SET/DEFINE OBJECT command is used to validate the user privileges for outbound connections to that object. The access control information is used as the default access control for inbound connections.

Table NCP-1 lists the object type codes used with the SET OBJECT and DEFINE OBJECT commands. All values in Table NCP-1 are expressed in decimal.

**Table NCP-1: Object Type Codes**

Code	Object Type	
	Mnemonic	Description
0	TASK	User program
1-16		Reserved for Digital use
17	FAL	File Access Listener for remote file and record access
18	HLD	Host loader for RSX-11S downline task loading requests
19	NML	Network Management Listener object
20		RSTS/E media transfer program (NETCPY)
21-22		Reserved for Digital use
23	REMACP	Network terminal handler (host side)
24		Network terminal handler (terminal side)
25	MIRROR	Loopback mirror

(continued on next page)

**Table NCP-1 (Cont.): Object Type Codes**

Code	Object Type	
	Mnemonic	Description
26	EVL	Event receiver
27	MAIL	VMS Mail Utility
28		Reserved for Digital use
29	PHONE	VMS Phone Utility and RSX-11M/M-PLUS Phone Utility
30-41		Reserved for Digital use
42	CTERM	Network terminal handler
43-62		Reserved for Digital use
63	DTR	DECnet Test Receiver object
64-127		Reserved for Digital use
128-255		Reserved for customer use

### example

```
NCP>SET OBJECT NML -
_ PRIVILEGES OPER DIAGNOSE -
_ USER NET_NONPRIV -
_ PASSWORD NET_NONPRIV
```

This command establishes default access control information for the NML object and sets those privileges required to connect to this object.

---

### SHOW/LIST CIRCUIT

The SHOW CIRCUIT command displays circuit information from the volatile database available to the local node or DTE. The LIST CIRCUIT command displays circuit information from the permanent database available to the local node or DTE.

#### format

```
SHOW circuit-component parameter [qualifier] [...]
LIST circuit-component parameter [qualifier] [...]
```

#### circuit components

##### **ACTIVE CIRCUITS**

Indicates that information for all active circuits is to be displayed.

##### **CIRCUIT *circuit-id***

Identifies a particular circuit for which information is to be displayed.

**NCP-28    NCP**  
**SHOW/LIST CIRCUIT**

***KNOWN CIRCUITS***

Indicates that information for all known circuits is to be displayed.

**command parameters**

***CHARACTERISTICS***

Indicates that static circuit information is to be displayed.

***COUNTERS***

Indicates that circuit error and performance statistics are to be displayed.

***STATUS***

Indicates that dynamic circuit information is to be displayed, including end node adjacencies and routing node adjacencies.

***SUMMARY***

Indicates that dynamic circuit information is to be displayed, including the routing adjacencies available to this circuit. **SUMMARY** is the default display type.

**qualifiers**

***ADJACENT NODE node-id***

Indicates that the display of a list of circuits is to be restricted to those circuits leading to the specified adjacent node.

***TO filespec***

Specifies the output file. If none is specified, **SYS\$OUTPUT** is the default.

**interpreting the display**

***Adjacent node node-id***

This read-only parameter indicates an adjacent node on the circuit. There can be many adjacent nodes on an Ethernet circuit.

***Block size number***

This read-only parameter is the block size in bytes for the adjacent node, as negotiated with the adjacent Routing layer during routing initialization over the circuit.

***Designated router node-id***

This read-only value is the Routing layer identification of the node that is to be used for routing to nonrouting nodes (end nodes) on this circuit.

***Listen timer seconds***

This read-only parameter determines the maximum time allowed to elapse before a message (a Routing Hello message or a user message) is received from an adjacent node on the circuit. The value can be up to a maximum of 65,535. Note that the **LISTEN TIMER** value is three times that of the **HELLO TIMER** circuit parameter.

**Loopback name**

This read-only parameter is the node name associated with a circuit for loopback testing. It identifies the circuit to be used for all traffic to the loop node.

**Substate**

This read-only value is the operational substate of the circuit. The substate is displayed as a tag on the STATE parameter (for example, ON-SYNCHRONIZING). Possible substate values are as follows:

- Synchronizing
- Starting
- Reflecting
- Looping
- Loading
- Dumping
- Triggering
- Autoservice
- Autoloading
- Autodumping
- Autotriggering
- Failed

**example**

```
NCP>SHOW ACTIVE CIRCUITS CHARACTERISTICS
```

```
Active Circuit Volatile Characteristics as of 30-DEC-1989 15:39:21
```

```
Circuit = DMC-0
```

```
State                = on
Service              = enabled
Cost                 = 12
Hello timer          = 15
Listen timer         = 30
Maximum buffers      = 255
Verification         = disabled
Adjacent node        = 3.5 (TRNTO)
Listen timer         = 30
```

```
Circuit = UNA-0
```

```
State                = on
Designated router    = 2.20 (ROBIN)
Cost                 = 1
Maximum routers allowed = 33
Router priority      = 64
Hello timer          = 15
Verification         = disabled
Adjacent node        = 2.22 (LARK)
Listen timer         = 45
```

## NCP-30 NCP

### SHOW/LIST CIRCUIT

```
Circuit = UNA-0
Adjacent node      = 2.23 (DOVE)
Listen timer      = 45
Circuit = UNA-0
Adjacent node      = 2.20 (ROBIN)
Listen timer      = 45
Circuit = UNA-0
```

This command displays circuit characteristics for all circuits whose states are ON.

---

## SHOW/LIST EXECUTOR

The SHOW EXECUTOR command displays local node information from the volatile database. The LIST EXECUTOR command displays local node information from the permanent database.

### format

```
SHOW EXECUTOR parameter [qualifier]
LIST EXECUTOR parameter [qualifier]
```

### command parameters

#### **CHARACTERISTICS**

Indicates that static local node information is to be displayed.

#### **COUNTERS**

Indicates that local node error and performance statistics are to be displayed.

#### **STATUS**

Indicates that dynamic local node information is to be displayed.

#### **SUMMARY**

Indicates that only the most useful local node information is to be displayed. This is the default display type.

### qualifier

#### **TO filespec**

Specifies the output file. If none is specified, SYS\$OUTPUT is the default.

### interpreting the display

#### **Active links number**

This read-only parameter represents the number of active logical links from the executor to the destination node.

#### **Delay seconds**

This read-only parameter is the average round-trip delay in seconds from the executor to the destination node.



**Management version n.n.n**

This read-only parameter identifies the version number of the Network Management layer. The format of the number consists of the version number, the Engineering Change Order (ECO) number, and the user ECO number (for example, V3.0.0).

**NSP version n.n.n**

This read-only parameter identifies the version number of the End Communication layer. The format for the number is the same as for the management version number.

**Physical address E-address**

This read-only parameter is the Ethernet address that identifies the executor node.

**Routing version n.n.n**

This read-only parameter identifies the version number of the Routing layer. The format for the number is the same as for the management version number.

**example**

NCP>SHOW EXECUTOR CHARACTERISTICS

Node Volatile Characteristics as of 30-DEC-1989 15:37:32

Executor node = 2.11 (BOSTON)

Identification	= DECnet--VAX V5.0, VMS V5.0
Management version	= V4.0.0
Incoming timer	= 45
Outgoing timer	= 45
Incoming Proxy	= Enabled
Outgoing Proxy	= Enabled
NSP version	= V4.0.0
Maximum links	= 128
Delay factor	= 80
Delay weight	= 5
Inactivity timer	= 60
Retransmit factor	= 10
Routing version	= V2.0.0
Type	= routing IV
Routing timer	= 600
Broadcast routing timer	= 40
Maximum address	= 1023
Maximum circuits	= 16
Maximum cost	= 1022
Maximum hops	= 15
Maximum visits	= 63
Maximum area	= 63
Max broadcast nonrouters	= 64
Max broadcast routers	= 32
Maximum path splits	= 1
Area maximum cost	= 1022
Area maximum hops	= 30
Maximum buffers	= 100
Buffer size	= 576

## NCP-32 NCP

### SHOW/LIST EXECUTOR

Default access = incoming and outgoing  
Pipeline quota = 1500  
Alias incoming = Enabled  
Alias maximum links = 32  
Alias node = 2.10 (CLUSTR)  
Path split policy = Normal

This command displays local node characteristics. This display shows values that you have set for the local node. In addition, it provides supplemental information about the software versions of NML, NSP, and Routing.

---

## SHOW/LIST LINE

The SHOW LINE command displays line information from the volatile database available to the local node. The LIST LINE command displays line information from the permanent database available to the local node.

### format

**SHOW** *line-component parameter [qualifier]*

**LIST** *line-component parameter [qualifier]*

### line components

#### **ACTIVE LINES**

Indicates that information for all active lines is to be displayed.

#### **KNOWN LINES**

Indicates that information for all known lines is to be displayed.

#### **LINE line-id**

Identifies a particular line for which information is to be displayed.

### command parameters

#### **CHARACTERISTICS**

Indicates that static line information is to be displayed.

#### **COUNTERS**

Indicates that line error and performance statistics are to be displayed.

#### **STATUS**

Indicates that dynamic line information is to be displayed.

#### **SUMMARY**

Indicates that only the most useful line information is to be displayed. This is the default display type.

## qualifier

### *TO filespec*

Specifies the output file. If none is specified, SYS\$OUTPUT is the default.

## interpreting the display

### *Hardware address E-address*

This read-only parameter is the Ethernet address associated with the line device hardware.

### *Substate*

This read-only value is the operational substate of the line. The substate is displayed as a tag on the STATE parameter (for example, ON-SYNCHRONIZING). Possible substate values are as follows:

- Synchronizing
- Starting
- Reflecting
- Looping
- Loading
- Dumping
- Triggering
- Autoservice
- Autoloading
- Autodumping
- Autotriggering
- Failed

## example

```
NCP>SHOW KNOWN LINES STATUS
```

```
Known Line Volatile Status as of 30-DEC-1989 10:21:27
```

Line	State
DMC-0	on
DMC-1	on
DUP-0	on
UNA-0	on

This command displays status information for all known lines connected to the local node. This display shows the current state of the line.

## SHOW/LIST NODE

The SHOW NODE command displays node information from the volatile database available to the local node. The LIST NODE command displays node information from the permanent database available to the local node.

### format

**SHOW** *node-component parameter [qualifier]*

**LIST** *node-component parameter [qualifier]*

### node components

#### **ACTIVE NODES**

For a routing node, indicates that information about all reachable nodes is to be displayed. For a nonrouting node (end node), indicates that information about the executor is to be displayed. Optionally, you can associate the following CIRCUIT parameter with this parameter:

CIRCUIT circuit-id Specifies that the display of a list of nodes is to be restricted to those nodes adjacent to the specified circuit.

#### **ADJACENT NODES**

Indicates that information about all adjacent nodes is to be displayed. Adjacent nodes are those the executor perceives Routing can reach that are separated from the executor by a single circuit. Each occurrence of a node on a different circuit appears as a separate adjacent node. Optionally, you can associate the following CIRCUIT parameter with this parameter:

CIRCUIT circuit-id Specifies that the display of a list of nodes is to be restricted to those nodes adjacent to the specified circuit.

#### **KNOWN NODES**

Indicates that information about all known nodes is to be displayed. Optionally, you can associate the following CIRCUIT parameter with this parameter:

CIRCUIT circuit-id Specifies that the display of a list of nodes is to be restricted to those nodes adjacent to the specified circuit.

#### **LOOP NODES**

Indicates that information about all loop nodes is to be displayed.

#### **NODE node-id**

Identifies a particular node about which information is to be displayed.

## command parameters

### ***CHARACTERISTICS***

Indicates that static node information is to be displayed.

### ***COUNTERS***

Indicates that node error and performance statistics are to be displayed.

### ***STATUS***

Indicates that dynamic node information is to be displayed.

### ***SUMMARY***

Indicates that only the most useful node information is to be displayed.  
This is the default display type.

## qualifier

### ***TO filespec***

Specifies the output file. If none is specified, SYS\$OUTPUT is the default.

## interpreting the display

### ***Active links number***

This read-only parameter represents the number of active logical links from the executor to the destination node.

### ***Circuit circuit-id***

This read-only parameter identifies the circuit used to get to a remote node.

### ***Cost number***

This read-only parameter represents the total cost over the current path to the destination node. The DECnet Routing layer routes messages (data) along the path between two nodes with the smallest cost. Cost is a positive integer value.

### ***Delay seconds***

This read-only parameter is the average round-trip delay in seconds from the executor to the destination node.

### ***Hops number***

This read-only parameter indicates the number of hops from the executor node to a destination node. A hop is a value assigned by the Routing layer that represents the logical distance between two nodes on a network.

### ***Management version n.n.n***

This read-only parameter identifies the version number of the Network Management layer. The format of the number consists of the version number, the Engineering Change Order (ECO) number, and the user ECO number (for example, V3.0.0).

**NCP-36 NCP  
SHOW/LIST NODE**

***Next node node-id***

This read-only parameter indicates the address and name of the next node on the circuit used to get to the node whose status is being displayed. Knowing which node is the partner on the next hop of the path to the destination node aids in tracing the path to that destination over a large number of hops.

***NSP version n.n.n***

This read-only parameter identifies the version number of the End Communication layer. The format for the number is the same as for the management version number.

***Physical address E-address***

This read-only parameter is the Ethernet address that identifies the executor node.

***Routing version n.n.n***

This read-only parameter identifies the version number of the Routing layer. The format for the number is the same as for the Management version number.

***Type node-type***

This read-only parameter indicates the type of the specified node. The values of *node-type* are as follows:

- Phase II
- Routing III
- Nonrouting III
- Routing IV
- Nonrouting IV
- Area

If the specified node is not adjacent to the local node, the *node-type* will be blank.

**example**

```
NCP>SHOW ACTIVE NODES CHARACTERISTICS
```

```
Active Node Volatile Characteristics as of 30-DEC-1989 13:38:34
```

```
Executor node = 2.11 (BOSTON)
```

NCP NCP-37  
SHOW/LIST NODE

Identification = DECnet--VAX V5.0, VMS V5.0  
Management version = V4.0.0  
Incoming timer = 45  
Outgoing timer = 45  
Incoming Proxy = Enabled  
Outgoing Proxy = Enabled  
NSP version = V3.2.0  
Maximum links = 128  
Delay factor = 80  
Delay weight = 5  
Inactivity timer = 60  
Retransmit factor = 10  
Routing version = V2.0.0  
Type = routing IV  
Routing timer = 600  
Maximum address = 1023  
Maximum circuits = 16  
Maximum cost = 1022  
Maximum hops = 15  
Maximum visits = 63  
Maximum area = 63  
Max broadcast nonrouters = 64  
Max broadcast routers = 32  
Maximum path splits = 1  
Area maximum cost = 1022  
Area maximum hops = 30  
Maximum buffers = 100  
Buffer size = 576  
Default access = incoming and outgoing  
Pipeline quota = 1500  
Alias incoming = Enabled  
Alias maximum links = 32  
Alias node = 2.10 (CLUSTER)  
Path split policy = Normal

Remote node = 3.5 (TRNTO)

Nonprivileged user id = NETNONPRIV

Remote node = 11.9 (DALLAS)

Nonprivileged user id = NETNONPRIV

Remote node = 12.34 (MYNODE)

Inbound = router

Remote node = 2.13 (KANSAS)

Nonprivileged user id = NETNONPRIV

Remote node = 2.17 (NYC)

Nonprivileged user id = NETNONPRIV

Loop node = 0 (TESTER)

**This command displays characteristics for all active nodes. This display shows values that you have set for both the local node and remote nodes.**

## SHOW/LIST OBJECT

The SHOW OBJECT command displays object information from the volatile database available to the local node. The LIST OBJECT command displays object information from the permanent database available to the local node.

### format

**SHOW** *object-component parameter [qualifier]*

**LIST** *object-component parameter [qualifier]*

### object components

#### **KNOWN OBJECTS**

Indicates that information about all known objects is to be displayed.

#### **OBJECT** *object-name*

Identifies a particular object about which information is to be displayed.

### command parameters

#### **CHARACTERISTICS**

Indicates that static object information is to be displayed. The SHOW OBJECT CHARACTERISTICS command displays only those parameters that you have defined.

#### **STATUS**

Indicates that dynamic object information is to be displayed.

#### **SUMMARY**

Indicates that only the most useful object information is to be displayed. This is the default display type.

### qualifier

#### **TO** *filespec*

Specifies the output file. If none is specified, SYS\$OUTPUT is the default.

### comments

This command is a system-specific network management command; therefore, an error occurs if you execute this command at a node other than a DECnet-VAX node, because objects may have different characteristics on different nodes.



**example**

NCP>SHOW OBJECT MAIL CHARACTERISTICS

Object Volatile Characteristics as of 30-DEC-1989 13:46:22

Object =	MAIL	
Number		= 27
File id		= MAIL.EXE
User id		= NETNONPRIV
Proxy access		= outgoing
Alias outgoing		= Enabled
Alias incoming		= Enabled

**This command displays object characteristics for the MAIL object. This display shows values that you have set for the object.**



---

## System Generation Utility

The System Generation Utility (SYSGEN) is a system management tool that performs certain privileged system configuration functions. With SYSGEN, you can create and modify system parameters, load device drivers, and create additional page and swap files.

### format

**RUN SYS\$SYSTEM:SYSGEN**

### usage summary

To invoke SYSGEN, type RUN SYS\$SYSTEM:SYSGEN at the DCL command prompt. At the SYSGEN> prompt, enter any of the SYSGEN commands described in the following section.

To exit from SYSGEN, enter the SYSGEN command EXIT at the SYSGEN> prompt or press CTRL/Z. You can direct output from a SYSGEN session to an output file using the SET/OUTPUT command. By default, output is written to SYS\$OUTPUT.

**NOTE:** Digital recommends the use of the AUTOGEN command procedure when modifying system parameters, loading device drivers, or creating additional page and swap files. Refer to the *Guide to Setting Up a VMS System* for a description of AUTOGEN.

## SGN-2 SYSGEN AUTOCONFIGURE

### SYSGEN Commands

This section explains SYSGEN commands and provides examples of their use.

---

### AUTOCONFIGURE

Automatically connects devices that are physically attached to the system and loads their drivers.

Use of the AUTOCONFIGURE command requires the CMKRNL privilege.

#### format

**AUTOCONFIGURE** *adapter-spec*

**AUTOCONFIGURE ALL**

#### parameter

##### ***adapter-spec***

Specifies the adapter specification (backplane interconnect arbitration line) or slot number of the single UNIBUS or MASSBUS adapter that is to be configured. The adapter specification can be expressed as an integer or with one of the names listed by the SYSGEN command SHOW/ADAPTER.

You can specify AUTOCONFIGURE ALL to configure all standard devices attached to the system.

#### qualifiers

##### ***/EXCLUDE=(device-name[,...])***

Specifies the device types that you do not want automatically configured.

Do not use this qualifier with the /SELECT qualifier.

##### ***/LOG***

Produces a display of the controller and its units on the current SYS\$OUTPUT device after they have been successfully autoconfigured. Each controller and its associated units are displayed only after AUTOCONFIGURE has found the next controller. Therefore, the error message displays precede the display of the controller and units that caused the error.

##### ***/SELECT=(device-name[,...])***

Specifies the device types that you want automatically configured.

Table SGN-1 shows device-type codes you can specify. You can include a controller designation but not a unit number. If the controller designation is omitted, all devices of the specified type are selected. The device-name specification defaults to all devices on the adapter.

Do not use /SELECT with the /EXCLUDE qualifier.

**Table SGN-1: Device Type Codes**

Code	Device Type
CR	Card Reader
CS	Console Storage Device
DB	RP05, RP06 Disk
DD	TU58 Cartridge Tape
DJ	RA60 Disk
DL	RL02 Cartridge Disk
DM	RK06, RK07 Cartridge Disk
DQ	RL02 Cartridge Disk, R80 Disk
DR	RM03, RM05, RM80, RP07 Disk
DU	UDA Disk
DX	RX01 Floppy Diskette
DY	RX02 Floppy Diskette
LA	LPA11-K Laboratory Peripheral Accelerator
LC	Line Printer on DMF32
LP	Line Printer on LP11
MB	Mailbox
MF	TU78 Magnetic Tape
MS	TS11 Magnetic Tape
MT	TE16, TU45, TU77 Magnetic Tape
MU	Tape Class Driver
NET	Network Communications Logical Device
NL	System "Null" Device
OP	Operator's Console
PA	Computer Interconnect (CI)
PT	TU81 Magnetic Tape
PU	UDA-50
RT	Remote Terminal
TT	Interactive Terminal on DZ11
TX	Interactive Terminal on DMF32, DMZ32, DHU11, or DMB32
XA	DR11-W General Purpose DMA Interface

(continued on next page)

## SGN-4 SYSGEN AUTOCONFIGURE

**Table SGN-1 (Cont.): Device Type Codes**

Code	Device Type
XD	DMP-11 Synchronous Communications Line
XF	DR32 Interface Adapter
XG	DMF32 Synchronous Communications Line
XI	DR Interface on DMF-32
XJ	DUP11 Synchronous Communications Line
XM	DMC11 Synchronous Communications Line

### example

```
SYSGEN> AUTOCONFIGURE ALL/SELECT=(TT,MTA,LP)
```

The command in this example automatically configures all terminals, all magnetic tape units on controller A, and all line printers.

---

## CONFIGURE

Requests UNIBUS device names and issues the set of CSR and vector addresses that AUTOCONFIGURE will use.

### format

**CONFIGURE**

### qualifiers

#### ***/INPUT=file-spec***

Specifies the name of an input file from which previously prepared data is read. By default, input data is read from SYS\$INPUT.

#### ***/OUTPUT=file-spec***

Specifies the name of an output file to which output from CONFIGURE is written. By default, output is directed to SYS\$OUTPUT. The default file type is LIS.

#### ***/NOJRESET***

Controls whether controller names are reset. The /NORESET qualifier is useful with multiple UNIBUS systems. When you specify /NORESET, it is not necessary to specify the second parameter (p) on subsequent CONFIGURE commands, since the controller names are not reset. By default, if you omit /NORESET, the controller names are reset.

### example

```
SYSGEN> CONFIGURE
DEVICE> DZ11,3,2
DEVICE> LP11
DEVICE> DMC11,2
DEVICE> CTRLZ
```

The system displays the following data:

Device:	RK611	Name:	DMA	CSR:	777440	Vector:	210	Support:	yes
Device:	LP11	Name:	LPA	CSR:	777514	Vector:	200	Support:	yes
Device:	DMC11	Name:	XMA	CSR:	760070*	Vector:	300*	Support:	yes
Device:	DMC11	Name:	XMB	CSR:	760100*	Vector:	310*	Support:	yes
Device:	DZ11	Name:	TTC	CSR:	760120*	Vector:	320*	Support:	yes
Device:	DZ11	Name:	TTD	CSR:	760130*	Vector:	330*	Support:	yes
Device:	DZ11	Name:	TTE	CSR:	760140*	Vector:	340*	Support:	yes

\* Indicates a floating address

This example illustrates the use of the CONFIGURE command to calculate the UNIBUS CSR and vector addresses. The support field in the display indicates whether Digital includes the supported driver for this device with the VMS operating system.

## CONNECT/ADAPTER=adapter-spec

Connects a hardware device and loads its driver if the driver is not already loaded. The adapter specification is the name of the UNIBUS or MASSBUS adapter to which the device is attached. The value can be expressed as an integer or as one of the names listed by the SYSGEN command SHOW/ADAPTER.

Use of the CONNECT/ADAPTER=adapter-spec command requires the CMKRNL privilege.

### format

**CONNECT/ADAPTER=adapter-spec** *device*

### parameter

#### *device*

Specifies the name of the hardware device to be connected. It should be specified in the following form: device-type, controller, unit. For example, LPA0 specifies the line printer (LP) on controller A at unit number 0. When specifying the device name, do *not* follow it with a colon (:).

## qualifiers

### ***/ADPUNIT=unit-number***

Unit number of a device on the MASSBUS adapter. The unit number for a disk drive is the number of the plug on the drive. For magnetic tape drives, the unit number corresponds to the tape controller's number.

### ***/CSR=csr-addr***

Specifies the UNIBUS address of the first addressable location on the controller (usually the status register) for the device. This qualifier must be specified for UNIBUS devices. For devices on multiple device boards (for example, the DMF32), the address must be the CSR address specified in the output of the CONFIGURE command. To specify the address in octal or hexadecimal, precede the address with %O or %X, respectively.

### ***/CSR\_OFFSET=value***

For devices on multiple device boards, specifies the offset from the CSR address of the multiple device board to the CSR address for the specific device being connected. To specify the address in octal or hexadecimal, precede the address with %O or %X, respectively.

### ***/DRIVERNAME=driver***

Specifies the name of the driver as recorded in the prolog table. If the driver has not been loaded, the system acts as if the driver name is also the name of an executable image (file type of EXE) in the SYS\$LOADABLE\_IMAGES directory and loads the driver. The driver name defaults to the first two characters of the device name concatenated with "DRIVER" (for example, LPDRIVER).

### ***/MAXUNITS=max-unit-cnt***

Specifies the maximum number of units the controller can support (that is, the number of UCB slots in the IDB). The default is the number specified in the prolog table of the driver, or 8 if the number is not specified in the prolog table.

### ***/NUMVEC=vector-cnt***

Specifies the number of interrupt vectors for the device. By default, the vector count is 1.

### ***/SYSIDHIGH=value***

Specifies the high-order 16 bits of the 48-bit system identification number and must be 0. To specify the value in octal or hexadecimal, precede the value with %O or %X, respectively.

### ***/SYSIDLOW=value***

Specifies the low-order 32 bits of the 48-bit system identification number. The value must be identical to the DECnet-VAX node number. To specify the value in octal or hexadecimal, precede the value with %O or %X, respectively.



***/VECTOR=vector-addr***

Specifies the UNIBUS address of the interrupt vector for the device or the lowest vector, if there is more than one. This qualifier must be specified for UNIBUS devices. For devices on multiple device boards (for example, the DMF32), the address must be the interrupt vector address for the multiple device board specified in the output of the CONFIGURE command. To specify the address in octal or hexadecimal, precede the address with %O or %X, respectively.

***/VECTOR\_OFFSET=value***

For devices on multiple device boards, specifies the offset from the interrupt vector address of the multiple device board to the interrupt vector address for the specific device being connected. To specify the address in octal or hexadecimal, precede the address with %O or %X, respectively.

**example**

```
SYSGEN> CONNECT LPA0/ADAPTER=3/CSR=%O777514 -  
SYSGEN> /DRIVERNAME=LP2DRIVER/VECTOR=%O200
```

The command in this example connects the device named LPA0 to the driver named LP2DRIVER and loads the driver if it is not already loaded.

---

**CONNECT/NOADAPTER**

Connects a software device and loads its driver if it is not already loaded.

Use of the CONNECT/NOADAPTER command requires the CMKRNL privilege.

**format**

**CONNECT/NOADAPTER** *device*

**parameter**

***device***

Specifies the name of the software device to be connected.

**qualifier**

***/DRIVERNAME=driver***

Specifies the name of the driver as recorded in the prolog table. If the driver has not been loaded, the system acts as if the driver name is also the name of an executable image (file type of EXE) in the SYS\$LOADABLE\_IMAGES directory and loads the driver. The default is the first two characters of the device name concatenated with "DRIVER" (for example, LPDRIVER). The driver prolog table must specify ADAPTER=NULL for this command to work.

## SGN-8   SYSGEN CONNECT/NOADAPTER

### example

```
SYSGEN> CONNECT NET/NOADAPTER/DRIVER=NETDRIVER
```

The command in this example connects the device NET to the driver NETDRIVER and loads the driver if it is not already loaded.

---

## CONNECT CONSOLE

The CONNECT CONSOLE command connects the console block storage devices and loads the driver. The console block storage device driver is CSDRIVER.EXE.

Use of the CONNECT CONSOLE command requires the CMKRNL privilege.

### format

**CONNECT CONSOLE**

### qualifiers

**/NI**

Enables a port for a console connected through the NI.

**/REMOTE**

Enables a remote diagnostic port for a second console or terminal connected to a VAX 8600 or VAX 8650 system.

**/USER**

Enables a port for a system user terminal connected to a VAX 8800 system.

---

## CREATE

Creates or extends a file that can be used as a page, swap, or dump file.

### format

**CREATE** *file-spec*

### parameter

***file-spec***

Specifies the name of the page, swap, or dump file. The default file type is SYS. Primary page and swap files have the names SYS\$SYSTEM:PAGEFILE.SYS and SYS\$SYSTEM:SWAPFILE.SYS.

The dump file name is SYS\$SYSTEM:SYSDUMP.DMP. When you create a new SYSDUMP.DMP file, you must explicitly specify the file type DMP.

## qualifiers

### ***/[NO]CONTIGUOUS***

Controls whether **CREATE** creates a contiguous file. The default is **/NOCONTIGUOUS**, which implies a contiguous-best-try file. If **/NOCONTIGUOUS** is specified, the following logic is used:

1. If the file does not exist, **SYSGEN** creates it.
2. If the file does exist, and the size specified by the **/SIZE** qualifier is smaller than the current size, **SYSGEN** creates a new file of the new size.
3. If the file does exist, and the size specified by the **/SIZE** qualifier is larger than the current size, **SYSGEN** extends the current file to the new size.

The **/CONTIGUOUS** qualifier forces the creation of a new file and guarantees that the file will be contiguous.

### ***/SIZE=block-count***

Specifies the number of blocks to be allocated to the file when the operation is complete. The current limit for the page file size is **%xFFFFFF**

## example

```
SYSGEN> CREATE SYS$SYSTEM:PAGEFILE/SIZE=95000/CONTIGUOUS
%SYSGEN-I-CREATED, SYS$SYSROOT:[SYSEXE]PAGEFILE.SYS;2 created
```

The command in this example creates a contiguous page file of 95,000 blocks.

## DEINSTALL

Deinstalls a page or swap file. Any file installed with the **SYSGEN INSTALL** command can be deinstalled.

Use of the **DEINSTALL** command requires the **CMKRNL** privilege.

## format

**DEINSTALL** *file-spec*

## example

```
SYSGEN> DEINSTALL DRA1:[SYSEXE]PAGEFILE.SYS /PAGEFILE
```

The command in this example deinstalls the system page file.

## SGN-10   SYSGEN

### DISABLE CHECKS

---

### DISABLE CHECKS

Inhibits range checks on parameter values specified in SET commands.

#### format

**DISABLE CHECKS**

#### example

```
SYSGEN> SET WSMAX 20
%SYSGEN-W-SETMIN, Value set to minimum for parameter WSMAX
SYSGEN> DISABLE CHECKS
SYSGEN> SET WSMAX 20
```

In this example, the initial attempt to set WSMAX below the minimum fails because range checks are enabled. However, once the user disables range checks, the SET WSMAX command succeeds.

---

### ENABLE CHECKS

Ensures that range checks are in effect.

Initially, range checks are enabled. Use ENABLE CHECKS only after you enter a DISABLE CHECKS command.

#### format

**ENABLE CHECKS**

#### example

```
SYSGEN> DISABLE CHECKS
SYSGEN> SET WSMAX 20
SYSGEN> ENABLE CHECKS
SYSGEN> SET WSMAX 30
%SYSGEN-W-SETMIN, Value set to minimum for parameter WSMAX
```

This example illustrates the use of the ENABLE CHECKS command to reenables parameter value checks.

---

### EXIT

Returns you to command level. You can also return to command level by pressing CTRL/Z.

#### format

**EXIT**

## HELP

Lists and explains the SYSGEN commands.

### format

**HELP** [*command-name*]

### parameter

#### ***command-name***

Specifies the name of a SYSGEN command or the keyword PARAMETERS. The command HELP PARAMETERS displays a list of all parameters and prompts for a parameter name.

### example

```
SYSGEN> HELP AUTOCONFIGURE
```

```
AUTOCONFIGURE
```

```
Automatically configures the device driver database. It locates each device unit physically attached to the system, loads the appropriate driver, creates the appropriate data structures, and connects the driver to the device's interrupt.
```

```
Format
```

```
AUTOCONFIGURE ALL
```

```
AUTOCONFIGURE adapter-spec
```

```
CMKRNL privilege required
```

```
Additional information available:
```

```
adapter-spec      ALL      qualifiers  
/SELECT    /EXCLUDE  /LOG
```

```
AUTOCONFIGURE Suptopic?
```

The HELP command in this example displays information about the AUTOCONFIGURE command.

---

## INSTALL

Activates a secondary page or swap file. The new page or swap file is effective until system shutdown.

Use of the INSTALL command requires the CMKRNL privilege.

# SGN-12 SYSGEN INSTALL

## format

**INSTALL** *file-spec*

## parameter

### *file-spec*

Specifies the name of the secondary page or swap file created with the SYSGEN command CREATE. The default file type is SYS.

## qualifiers

### **/PAGEFILE**

Specifies that the file is to be installed as an additional page file. All processes created after the page file is installed use the page file with the most available free space; processes created before the additional page file is installed continue to use the page file to which they are assigned.

### **/SWAPFILE**

Specifies that the file is to be installed as an additional swap file. This swap file augments the swap file installed during the bootstrap process.

## example

```
SYSGEN> INSTALL SYS$SYSTEM:PAGEFILE.SYS/PAGEFILE
```

The command in this example installs a secondary page file.

---

## LOAD

Loads an I/O driver.

Use of the LOAD command requires the CMKRNL privilege.

## format

**LOAD** *file-spec*

## parameter

### *file-spec*

Specifies the file specification of the driver image to be loaded. The default file type is EXE.

If the entire file specification is the same as that of a driver already loaded, no load takes place. If only the file name is the same as that of a driver that is already loaded (but the file specification is different), the specified driver replaces the existing driver.

## example

```
SYSGEN> LOAD SYS$SYSTEM:RTTDRIVER
```

The command in this example loads the standard driver for a remote terminal.

---

## MSCP

Loads and starts the MSCP server.

This method of loading the MSCP server has been superseded for VMS Version 5.0 by the SYSGEN parameter `MSCP_LOAD`. To load the MSCP server, set the `MSCP_LOAD` parameter to 1. Define the disks to be served with the `MSCP_SERVE_ALL` parameter.

## format

**MSCP**

---

## RELOAD

Replaces a loaded device driver with a new version.

Use of the RELOAD command requires the CMKRNL privilege.

## format

**RELOAD** *file-spec*

## parameter

*file-spec*

The file specification of the new driver image. The default file type is EXE. The specified image is loaded and replaces any existing driver with the same file specification.

## example

```
SYSGEN> RELOAD SYS$SYSTEM:RTTDRIVER
```

The command in this example reloads the remote terminal driver.

## SGN-14 SYSGEN SET/OUTPUT

---

### SET/OUTPUT

Establishes a file to be used for output during the session. By default the output is written to SYS\$OUTPUT, but you can use the SET/OUTPUT command to designate a disk file.

At any time you can direct the output back to SYS\$OUTPUT by using the SET/OUTPUT=SYS\$OUTPUT command.

#### format

SET/OUTPUT[=] *file-spec*

#### parameter

##### *file-spec*

The name of the output file. The default file type is LIS. The equal sign (=) is optional.

#### example

```
SYSGEN> SET/OUTPUT=PARAMS.LIS
SYSGEN> SHOW/ALL
SYSGEN> SHOW/SPECIAL
SYSGEN> EXIT
```

In this example, output is directed to the file PARAMS.LIS to capture a complete list of all the system parameters (including the SPECIAL parameters reserved for Digital use) and their values.

---

### SET parameter-name

Assigns a value to a system parameter in the SYSGEN work area.

This command does not modify parameter files, the current system parameter file on disk, or the active system; for information on performing these modifications, see the WRITE command.

#### format

SET *parameter-name* *value*

#### parameters

##### *parameter-name*

Specifies the name of a system parameter. If you enter a period (.), it is interpreted as a request for the system parameter specified in the last SET or SHOW command. See the description of the SHOW [parameter] command for an example of the use of the period in place of a parameter name.



You can display the system parameters and request information on them with the SYSGEN command HELP PARAMETERS.

***value***

Usually specifies an integer or the keyword DEFAULT. Integer values must be within the defined minimum and maximum values for the parameter unless the SYSGEN command DISABLE CHECKS was specified.

The keyword DEFAULT specifies the default value for the parameter. You can display the maximum, minimum, and default values for any parameter with the SYSGEN command SHOW [parameter].

**example**

```
SYSGEN> SET PFCDEFAULT 20
```

The command in this example assigns a value of 20 to the PFCDEFAULT parameter.

---

**SET/STARTUP**

Names the site-independent startup command procedure to be associated with a parameter file for subsequent bootstrap operations.

**format**

**SET/STARTUP** *file-spec*

**parameter**

***file-spec***

The file specification of a startup command procedure on the system disk (maximum of 31 characters). The initial site-independent startup command procedure (as named in the software distribution kit) is SYS\$SYSTEM:STARTUP.COM.

**example**

```
SYSGEN> SET/STARTUP SYS$SYSTEM:XSTARTUP.COM
```

The command in this example assigns SYS\$SYSTEM:XSTARTUP.COM as the current site-independent startup command procedure.

## SHARE

Connects a processor to a multiport memory unit already initialized by this or another processor. The number and name of the specified multiport memory unit must be those of an initialized unit, or an error condition results.

Use of the SHARE command requires the CMKRNL privilege.

### format

**SHARE**   *MPMn MPM-name*

### parameters

#### ***MPMn***

Specifies the number on the front panel of the multiport memory unit being connected.

#### ***MPM-name***

The name of the multiport memory unit as specified in a previous SHARE/INITIALIZE command.

### example

```
SYSGEN> SHARE MPM1 SHR_MEM_1
```

The command in this example connects a multiport memory unit. Since no qualifiers are specified, defaults apply to all the parameters.

The unit with a 1 on the front panel must be initialized with the name SHR\_MEM\_1 for the command to work.

---

## SHARE/INITIALIZE

Initializes a multiport memory unit and connects it to the processor on which SYSGEN is running.

Use of the SHARE/INITIALIZE command requires the CMKRNL privilege.

### format

**SHARE/INITIALIZE**   *MPMn MPM-name*

### parameters

#### ***MPMn***

Specifies the number on the front panel of the multiport memory unit being connected.

***MPM-name***

Specifies the name by which the multiport memory unit is to be known to systems using it. The MPM-name is a 1 through 15 alphanumeric character string that may contain dollar signs (\$) and underscores (\_).

**qualifiers**

***/CEFCLUSTERS=cef***

Specifies the total number of common event flag clusters permitted in the multiport memory unit. The cef value is an integer with a default of 32.

***/GBLSECTIONS=gbl***

Specifies the total number of global sections permitted in the multiport memory unit. The gbl value is an integer with a default of 32.

***/MAILBOXES=mail***

Specifies the total number of mailboxes permitted in the multiport memory unit. The mail value is an integer with a default of 32.

***/MAXCEFCLUSTERS=max-cef***

Specifies the maximum number of common event flag clusters that the processor can create in the multiport memory unit. The default is no limit.

***/MAXGBLSECTIONS=max-gbl***

Specifies the maximum number of global sections that the processor can create in the multiport memory unit. The default is no limit.

***/MAXMAILBOXES=max-mail***

Specifies the maximum number of mailboxes the processor can create in the multiport memory unit. The default is no limit.

***/POOLBCOUNT=block-cnt***

Specifies the number of blocks allocated to the multiport memory unit's dynamic pool. The block-cnt value is an integer with a default of 128.

***/POOLBSIZE=block-size***

Specifies the size of each block in the dynamic pool. The block-size value is an integer with a default of 128 bytes.

***/PRQCOUNT=prq-cnt***

Specifies the number of interprocessor request blocks (PRQs) allocated. The prq-cnt value is an integer with a default of 64.

**example**

```
SYSGEN> SHARE MPM1 SHR_MEM_1/INITIALIZE -  
SYSGEN> /GBLSECTIONS=128/MAILBOXES=64/CEFCLUSTERS=0
```

The command in this example initializes a multiport memory unit with defaults on all but the gbl, mail, and cef parameters. In this example, assume that the number of the multiport memory unit as it appears on the front panel is 1, and the unit name is SHR\_MEM\_1.

## SGN-18 SYSGEN SHOW/ADAPTER

---

### SHOW/ADAPTER

Lists all the nexus numbers and generic names on the adapter.

Use of the SHOW/ADAPTER command requires the CMEXEC privilege.

#### format

**SHOW/ADAPTER**

#### example

SYSGEN> SHOW/ADAPTER

The following is a sample display produced by the SHOW/ADAPTER command:

CPU Type: 11/780

Nexus	Generic Name or Description
1	16K memory, non-interleaved
3	UB0
8	MB0
9	MB1

---

### SHOW/CONFIGURATION

Displays information on the device configuration.

Use of the SHOW/CONFIGURATION command requires the CMEXEC privilege.

#### format

**SHOW/CONFIGURATION**

#### qualifiers

**/ADAPTER=*nexus***

Specifies the number of MASSBUS or UNIBUS adapters to be displayed. The nexus value can be expressed as an integer or with one of the generic names listed by the SYSGEN command SHOW/ADAPTER.

**/COMMAND\_FILE**

Specifies that SYSGEN formats all the device data into CONNECT/ADAPTER=*adapter-spec* commands and writes the commands in an output file you specify. In this way, you can completely reconfigure a system for UNIBUS devices without the use of the SYSGEN command AUTOCONFIGURE.

***/OUTPUT=file-spec***

Specifies the file specification of an optional output file. If you specify the /OUTPUT qualifier but omit the file type, the default is LIS. However, if you specify /COMMAND\_FILE and /OUTPUT qualifiers together, the default file type for the output file is COM.

**example**

SYSGEN> SHOW/CONFIGURATION

The command in this example displays the current system I/O database. The following illustrates a typical display produced by this command:

System CSR and Vectors on 15-JUN-1989 13:49:26.84

Name: DRA	Units: 3	Nexus:4	(MBA)						
Name: DBA	Units: 1	Nexus:4	(MBA)						
Name: DBB	Units: 2	Nexus:5	(MBA)						
Name: DRB	Units: 1	Nexus:5	(MBA)						
Name: MTA	Units: 2	Nexus:5	(MBA)						
Name: DMA	Units: 2	Nexus:8	(UBA)	CSR: 777440	Vector1: 210	Vector2: 000			
Name: LPA	Units: 1	Nexus:8	(UBA)	CSR: 777514	Vector1: 200	Vector2: 000			
Name: DYA	Units: 2	Nexus:8	(UBA)	CSR: 777170	Vector1: 264	Vector2: 000			
Name: XMA	Units: 1	Nexus:8	(UBA)	CSR: 760070	Vector1: 300	Vector2: 304			
Name: XMB	Units: 1	Nexus:8	(UBA)	CSR: 760100	Vector1: 310	Vector2: 314			
Name: XMC	Units: 1	Nexus:8	(UBA)	CSR: 760110	Vector1: 320	Vector2: 324			
Name: TTA	Units: 8	Nexus:8	(UBA)	CSR: 760130	Vector1: 330	Vector2: 334			
Name: TTB	Units: 8	Nexus:8	(UBA)	CSR: 760140	Vector1: 340	Vector2: 344			
Name: TTC	Units: 8	Nexus:8	(UBA)	CSR: 760150	Vector1: 350	Vector2: 354			
Name: TTD	Units: 8	Nexus:8	(UBA)	CSR: 760160	Vector1: 360	Vector2: 364			
Name: TTE	Units: 8	Nexus:8	(UBA)	CSR: 760170	Vector1: 370	Vector2: 374			
Name: TTF	Units: 8	Nexus:8	(UBA)	CSR: 760200	Vector1: 400	Vector2: 404			

---

**SHOW/DEVICE=device-driver**

Displays full information on device drivers loaded into the system, the devices connected to them, and their I/O databases. All addresses are in hexadecimal and are virtual.

Use of the SHOW/DEVICE=device-driver command requires the CMEXEC privilege.

**format**

**SHOW/DEVICE=device-driver**

**SGN-20    SYSGEN**  
**SHOW/DEVICE=device-driver**

**example**

SYSGEN> SHOW/DEVICE=DBDRIVER

The command in this example displays the following information about the DBDRIVER:

<u>Driver</u>	<u>Start</u>	<u>End</u>	<u>Dev</u>	<u>DDB</u>	<u>CRB</u>	<u>IDB</u>	<u>Unit</u>	<u>UCB</u>
DBDRIVER	80082390	80082A7E						
			DBA	80000848	800988C0	80098920		
							0	8000087C
							1	8008A4F0
							2	8008A590
							5	8008A630
							7	8008A6D00

---

**SHOW/DRIVER=device-driver**

Displays the starting and ending address of the specified device driver loaded into the system. If you omit the driver name, SHOW/DRIVER displays the starting and ending address of all device drivers loaded into the system. All addresses are in hexadecimal and are virtual.

Use of the SHOW/DRIVER command requires the CMEXEC privilege.

**format**

**SHOW/DRIVER=device-driver**

**example**

SYSGEN> SHOW/DRIVER

The command in this example displays the starting and ending addresses of all drivers, as follows:

<u>Driver</u>	<u>Start</u>	<u>End</u>
RTTDRIVER	800C1060	800C1960
NETDRIVER	800BAFD0	800BD4B0
TMDRIVER	800B3950	800B4BF0
DRDRIVER	800B2950	800B3290
DDDRIVER	800B1740	800B2060
DLDRIVER	800B0D10	800B15A0
DMDRIVER	800B0070	800B0990
LCDRIVER	800AFC50	800AFFB0
YCDRIVER	800AED20	800AF3E0
XGDRIVER	800AC3F0	800AE9E0
XDDRIVER	800AA5A0	800AC380
DZDRIVER	800A4F30	800A59B0
XMDRIVER	800A3E10	800A4A50
DYDRIVER	800A3300	800A3C30
LPDRIVER	800A2E90	800A3300
DBDRIVER	800DE7A0	800DEFB7
TTDRIVER	800DC770	800DE79B
OPERATOR	80001650	80001F8B
NLDRIVER	80001626	80001D20
MBDRIVER	800015FC	80001CBE

## SHOW [parameter]

Displays the values of system parameters in the SYSGEN work area, plus the default, minimum, and maximum values of the parameters and their units of measure.

### format

**SHOW** [*parameter-name*]

### parameter

#### *parameter-name*

Specifies the name of a system parameter. If you enter a period (.), it is interpreted as a request for the system parameter specified in the last SET parameter-name or SHOW [parameter] command.

### qualifiers

#### **/ACP**

Specifies that all ACP parameter values are displayed.

#### **/ALL**

Specifies that all parameter values other than SPECIAL parameter values are displayed.

#### **/CLUSTER**

Specifies that all CLUSTER parameter values are displayed.

#### **/DYNAMIC**

Specifies that all DYNAMIC parameter values are displayed.

#### **/GEN**

Specifies that all GEN parameter values are displayed.

#### **/HEX**

Specifies that the values of parameters be displayed in hexadecimal representation. Specify the /HEX system parameter name or the parameter type. If you specify the /HEX qualifier with the /NAMES qualifier, /HEX is ignored.

#### **/JOB**

Specifies that all JOB parameter values are displayed.

#### **/LGI**

Specifies that all LGI parameter values are displayed.

#### **/MAJOR**

Specifies that all MAJOR parameter values are displayed.

#### **/MULTIPROCESSING**

Specifies that all MULTIPROCESSING parameters are displayed.

## SGN-22 SYSGEN

### SHOW [parameter]

#### **/NAMES**

Specifies that the names of all parameters are displayed.

#### **/PQL**

Specifies that all PQL parameter values are displayed.

#### **/RMS**

Specifies that all VAX RMS parameter values are displayed.

#### **/SCS**

Specifies that all SCS parameter values are displayed.

#### **/SPECIAL**

Specifies that all parameter values reserved for Digital use are displayed.

#### **/SYS**

Specifies that all SYS parameter values are displayed.

#### **/TTY**

Specifies that all terminal parameter values are displayed.

### example

```
SYSGEN> SHOW GBLSECTIONS
GBLSECTIONS      100      40      20      -1 Sections
SYSGEN> SET . 110
SYSGEN> SHOW .
GBLSECTIONS      110      40      20      -1 Sections
```

In this example, the user first displays the values of the GBLSECTIONS parameter and then refers to the parameter with a period to set its current value to 110. The next SHOW command also uses the period notation to obtain confirmation that the change occurred.

---

### SHOW/STARTUP

Displays the name of the current site-independent startup command procedure.

#### format

**SHOW/STARTUP**

#### example

```
SYSGEN> SHOW/STARTUP
Startup command file = SYS$SYSTEM:STARTUP.COM
```

The command in this example displays the name of the site-independent startup command procedure.



## SHOW/UNIBUS

Displays the addresses in UNIBUS I/O space that can be addressed.

Use of the SHOW/UNIBUS command requires the CMKRNL privilege.

### format

**SHOW/UNIBUS**

### qualifier

**/ADAPTER[=*nexus*]**

Specifies that the address of the specified UNIBUS adapter is to be displayed. The nexus value specifies the number of the UNIBUS. It can be expressed as an integer or as one of the names listed by the SYSGEN command SHOW/ADAPTER. If you do not specify a particular adapter, every UNIBUS is displayed.

### example

```
SYSGEN> SHOW/UNIBUS/ADAPTER=4
```

The command in this example displays the available addresses for nexus 4, as follows:

```
**UNIBUS map for nexus #4 on 30-JUN-1989 14:19:38.00 **
Address 760070 (8001F838) responds with value 9B6E (hex)
Address 760072 (8001F83A) responds with value 0340 (hex)
Address 760074 (8001F83C) responds with value 403C (hex)
Address 760076 (8001F83E) responds with value 0240 (hex)
Address 760100 (8001F840) responds with value 8000 (hex)
Address 760102 (8001F842) responds with value 0340 (hex)
Address 760104 (8001F844) responds with value 7DAC (hex)
Address 760106 (8001F846) responds with value 000A (hex)
Address 760110 (8001F848) responds with value 8000 (hex)
Address 760112 (8001F84A) responds with value 0340 (hex)
Address 760114 (8001F84C) responds with value AD5C (hex)
Address 760116 (8001F84E) responds with value 000A (hex)

Address 760130 (8001F858) responds with value 9B6E (hex)
Address 760132 (8001F85A) responds with value 030D (hex)
Address 760134 (8001F85C) responds with value FF00 (hex)
Address 760136 (8001F85E) responds with value CECE (hex)
Address 760140 (8001F860) responds with value 4060 (hex)
Address 760142 (8001F862) responds with value 0761 (hex)
Address 760144 (8001F864) responds with value FF00 (hex)
```

.  
.
  
.

## **TERMINAL/ECHO**

Modifies the CTRL/C, CTRL/O, CTRL/Y, and CTRL/Z echo strings on a systemwide basis.

### **format**

**TERMINAL/ECHO**

---

## **USE**

Initializes the SYSGEN work area with system parameter values and the name of the site-independent startup command procedure. You specify the source for both the parameter values and the procedure name. They can be retrieved from a parameter file, the current system parameter file on disk, the active system in memory, or the default list.

Existing values in the SYSGEN work area are overwritten.

### **format**

**USE** *file-spec*

### **parameter**

#### ***file-spec***

The file specification of a system parameter file from which data is to be retrieved. The parameter file is either SYS\$SYSTEM:AUTOGEN.PAR or the name of a parameter file you created with the SYSGEN command WRITE. The default file type is PAR.

### **example**

```
SYSGEN> USE DEFAULT
```

The command in this example initializes the SYSGEN work area with parameter values that should allow VMS to boot on any standard configuration. The initial values of the SYSGEN work area when the utility is invoked are the active values.

## WRITE

Writes the system parameter values and the name of the site-independent startup command procedure from the SYSGEN work area to either a parameter file, the current system parameter file on disk, or the active system in memory. (Only the dynamic parameter values are written to the active system.)

Use of the WRITE ACTIVE command requires the CMKRNL privilege.  
Use of the WRITE CURRENT command requires the SYSPRV privilege.

### format

WRITE *file-spec*

### parameter

*file-spec*

The file specification of a new parameter file to be created. The default file type is PAR.

### example

```
SYSGEN> WRITE CURRENT
```

The command in this example modifies the current system parameter file on disk (SYS\$SYSTEM:VAXVMSSYS.PAR).

## **Supplemental SYSGEN Information**

This section contains the following information:

- The SYSGEN device table
- Tables of the VMS system parameters

### **The SYSGEN Device Table**

Table SGN-2 lists the characteristics of all Digital devices. This table indicates the following information for each device type:

- Device name
- Device controller name
- Interrupt vector
- Number of interrupt vectors per controller
- Vector alignment factor
- Address of the first device register for each controller recognized by SYSGEN (the first register is usually, but not always, the CSR)
- Number of registers per controller
- Device driver name
- Indication of whether the driver is or is not supported

Devices not listed in the SYSGEN device table include:

- Non-Digital-supplied devices with fixed CSR and vector addresses. These devices have no effect on autoconfiguration. Customer-built devices should be assigned CSR and vector addresses beyond the floating address space reserved for Digital-supplied devices.
- Those Digital-supplied, floating-vector devices that the AUTOCONFIGURE command does not recognize. Use the CONNECT command to attach these devices to the system.

SYSGEN    SGN-27

**Supplemental SYSGEN Information**

**Table SGN-2: SYSGEN Device Table**

<b>Device Name</b>	<b>Controller Name</b>	<b>Vector</b>	<b>Number of Vectors</b>	<b>Vector Alignment</b>	<b>CSR /Rank</b>	<b>Register Alignment</b>	<b>Driver Name</b>	<b>Support</b>
CR	CR11	230	1	—	777160	—	CRDRIVER	Yes
DM	RK611	210	1	—	777440	—	DMDRIVER	Yes
LP	LP11	200 170 174 270 274	—	—	777514 764004 764014 764024 764034	—	LPDRIVER	Yes
DL	RL11	160	1	—	774400	—	DLDRIVER	Yes
MS	TS11	224	1	—	772520	—	TSDRIVER	Yes
DY	RX211	264	1	—	777170	—	DYDRIVER	Yes
DQ	RB730	250	1	—	775606	—	DQDRIVER	Yes
PU	UDA	154	1	—	772150	—	PUDRIVER	Yes
PT	TU81	260	1	—	774500	—	PUDRIVER	Yes
XE	UNA	120	1	—	774510	—	XEDRIVER	Yes
XQ	QNA	120	1	—	774440	—	XQDRIVER	Yes
OM	DC11	Float	2	8	774000 774010 774020 774030 . . . 32 units maxi- mum	—	OMDRIVER	No
DD	TU58	Float	2	8	776500 776510 776520 776530 . . . 16 units maxi- mum	—	DDRIVER	Yes

(continued on next page)

**SGN-28 SYSGEN**  
**Supplemental SYSGEN Information**

**Table SGN-2 (Cont.): SYSGEN Device Table**

<b>Device Name</b>	<b>Controller Name</b>	<b>Vector</b>	<b>Number of Vectors</b>	<b>Vector Alignment</b>	<b>CSR /Rank</b>	<b>Register Alignment</b>	<b>Driver Name</b>	<b>Support</b>
OB	DN11	Float	1	4	775200 775210 775220 775230 . . 16 units maximum	—	OBDRIVER	No
YM	DM11B	Float	1	4	770500 770510 770520 770530 . . 16 units maximum	—	YMDRIVER	No
OA	DR11C	Float	2	8	767600 767570 767560 767550 . . 16 units maximum	—	OADRIVER	No
PR	PR611	Float	1	8	772600 772604 772610 772614 . . 8 units maximum	—	PRDRIVER	No

(continued on next page)

SYSGEN    SGN-29  
**Supplemental SYSGEN Information**

**Table SGN-2 (Cont.): SYSGEN Device Table**

Device Name	Controller Name	Vector	Number of Vectors	Vector Alignment	CSR /Rank	Register Alignment	Driver Name	Support
PP	PP611	Float	1	8	772700 772704 772710 772714 . . 8 units maximum	—	PPDRIVER	No
OC	DT11	Float	2	8	777420 777422 777424 777426 . . 8 units maximum	—	OCDRIVER	No
OD	DX11	Float	2	8	776200 776240	—	ODDRIVER	No
YL	DL11C	Float	2	8	775610 775620 775630 775640 . . 31 units maximum	—	YLDIVER	No
YJ	DJ11	Float	2	8	Float	8	YJDRIVER	No
YH	DH11	Float	2	8	Float	16	YHDRIVER	No
OE	GT40	Float	4	8	772000 772010	—	OEDRIVER	No
LS	LPS11	Float	6	8	770400	—	LSDRIVER	No
OR	DQ11	Float	2	8	Float	8	ORDRIVER	No
OF	KW11W	Float	2	8	772400	—	OFDRIVER	No
XU	DU11	Float	2	8	Float	8	XUDRIVER	No
XW	DUP11	Float	2	8	Float	8	OODRIVER	No

(continued on next page)

**SGN-30 SYSGEN**  
**Supplemental SYSGEN Information**

**Table SGN-2 (Cont.): SYSGEN Device Table**

<b>Device Name</b>	<b>Controller Name</b>	<b>Vector</b>	<b>Number of Vectors</b>	<b>Vector Alignment</b>	<b>CSR /Rank</b>	<b>Register Alignment</b>	<b>Driver Name</b>	<b>Support</b>
XV	DV11	Float	3	8	775000 775040 775100 775140	—	XVDRIVER	No
OG	LK11	Float	2	8	Float	8	OGDRIVER	No
XM	DMC11	Float	2	8	Float	8	XMDRIVER	Yes
TTA	DZ11	Float	2	8	Float	8	DZDRIVER	Yes
XK	KMC11	Float	2	8	Float	8	XKDRIVER	No
OH	LPP11	Float	2	8	Float	8	OHDRIVER	No
OI	VMV21	Float	2	8	Float	8	OIDRIVER	No
OJ	VMV31	Float	2	8	Float	16	OJDRIVER	No
OK	DWR70	Float	2	8	Float	8	OKDRIVER	No
DL	RL11	Float	1	4	Float	8	DLDRIVER	Yes
MS	TS11	Float	1	4	772524 772530 772534	—	TSDRIVER	Yes
LA	LPA11	Float	2	8	770460	—	LADRIVER	Yes
LA	LPA11	Float	2	8	Float	16	LADRIVER	Yes
OL	KW11C	Float	2	8	Float	8	OLDRIVER	No
RSV	RSV	Float	1	8	Float	8	RSVDRIVER	No
DY	RX211	Float	1	4	Float	8	DYDRIVER	Yes
XA	DR11W	Float	1	4	Float	8	XADRIVER	Yes
XB	DR11B	124	—	—	772410	—	XBDRIVER	No
XB	DR11B	Float	1	4	772430	—	XBDRIVER	No
XB	DR11B	Float	1	4	Float	8	XBDRIVER	No
XD	DMP11	Float	2	8	Float	8	XDDRIVER	Yes
ON	DPV11	Float	2	8	Float	8	ONDRIVER	No
IS	ISB11	Float	2	8	Float	8	ISDRIVER	No
XD	DMV11	Float	2	8	Float	16	XDDRIVER	No
XE	UNA	Float	1	4	Float	8	XEDRIVER	No
XQ	QNA	Float	1	4	774460	—	XQDRIVER	Yes
PU	UDA	Float	1	4	Float	4	PUDRIVER	Yes

(continued on next page)



SYSGEN    SGN-31  
Supplemental SYSGEN Information

**Table SGN-2 (Cont.): SYSGEN Device Table**

Device Name	Controller Name	Vector	Number of Vectors	Vector Alignment	CSR /Rank	Register Alignment	Driver Name	Support
XS	KMS11	Float	3	8	Float	16	XSDRIVER	No
XP	PCL11	Float	2	8	764200 764240 764300 764340	—	XPDRIVER	No
VB	VS100	Float	1	4	Float	16	VBDRIVER	No
PT	TU81	Float	1	4	Float	4	PUDRIVER	Yes
OQ	KMV11	Float	2	8	Float	16	OQDRIVER	No
UK	KCT32	Float	2	8	764400 764440 764500 764540	—	UKDRIVER	No
IX	IEQ11	Float	2	8	764100	—	IXDRIVER	No
TX	DHV11	Float	2	8	Float	16	YFDRIVER	Yes
DT	TC11	214	1	—	777340	—	DTDRIVER	No
VC	VCB01	Float	2	1	777200	—	VCDRIVER	Yes
VC	VCB01	Float	2	1	Float	64	VCDRIVER	Yes
OT	LNV11	Float	1	4	776200	—	OTDRIVER	No
LD	LNV21	Float	1	4	Float	16	LDDRIVER	No
ZQ	QTA	Float	1	4	772570	—	ZQDRIVER	No
ZQ	QTA	Float	1	4	Float	8	ZQDRIVER	No
SJ	DSV11	Float	1	4	Float	8	SJDRIVER	No
OU	ADV11C	Float	2	8	Float	8	OUDRIVER	No
OV	AAV11C	Float	0	8	770440	—	OVDRIVER	No
OV	AAV11C	Float	0	8	Float	8	OVDRIVER	No
AX	AXV11C	140	2	—	776400	—	AXDRIVER	No
AX	AXV11C	Float	2	8	Float	8	AXDRIVER	No
KZ	KWV11C	Float	2	8	770420	—	KZDRIVER	No
KZ	KWV11C	Float	2	8	Float	4	KZDRIVER	No
AZ	ADV11D	Float	2	8	776410	—	AZDRIVER	No
AZ	ADV11D	Float	2	8	Float	4	AZDRIVER	No
AY	AAV11D	Float	2	8	776420	—	AYDRIVER	No

(continued on next page)

**SGN-32 SYSGEN**  
**Supplemental SYSGEN Information**

**Table SGN-2 (Cont.): SYSGEN Device Table**

<b>Device Name</b>	<b>Controller Name</b>	<b>Vector</b>	<b>Number of Vectors</b>	<b>Vector Alignment</b>	<b>CSR /Rank</b>	<b>Register Alignment</b>	<b>Driver Name</b>	<b>Support</b>
AY	AAV11D	Float	2	8	Float	4	AYDRIVER	No
VA	VC802	Float	3	16	777400 777402 777404 777406 . . 8 units maximum	—	VADRIVER	Yes
DN	DRV11J	Float	16	4	764160 764140 764120	—	DNDRIVER	No
HX	DRQ3B	Float	2	8	Float	16	HXDRIVER	No
VQ	VSV24	Float	1	4	Float	8	VQDRIVER	No
VV	VSV21	Float	1	4	Float	8	VVDRIVER	No
BQ	IBQ01	Float	1	4	Float	8	BQDRIVER	No
UT	MIRA	Float	2	8	Float	8	UTDRIVER	No
IX	IEQ11	Float	2	8	Float	16	IXDRIVER	No
AW	ADQ32	Float	2	8	Float	32	AWDRIVER	No
VX	DTC04	Float	2	8	Float	2	VXDRIVER	No

**Parameter Categories**

This section describes the functions of the VMS system parameters. The system parameters fall into eleven general categories:

- **ACP**—Parameters associated with file system caches and Files-11 ancillary control processes (ACPs)
- **CLUSTER**—Parameters that affect VAXcluster operation
- **JOB**—Job control parameters
- **LGI**—Login security parameters
- **MULTIPROCESSING**—Parameters associated with symmetric multiprocessing (SMP)
- **PQL**—Parameters associated with process creation limits and quotas
- **RMS**—Parameters associated with VAX RMS

**Supplemental SYSGEN Information**

- SCS—Parameters that control System Communication Services (SCS) and port driver operation. The parameters that affect SCS operation have the prefix SCS. The parameters that affect the CI780/CI750 port driver have the prefix PA
- SPECIAL—Special parameters. These parameters should be used only by Digital personnel
- SYS—Parameters that affect overall system operation
- TTY—Parameters associated with terminal behavior

There are also four parameters that can be user-defined: USERD1, USERD2, USER3, and USER4. USERD1 and USERD2 are dynamic.

Parameters may have one or more of the following attributes:

- DYNAMIC—Active values can be modified
- GEN—Affect the creation and initialization of data structures at bootstrap time
- MAJOR—Most likely to require modification

**NOTE:** Each parameter has associated default, minimum, and maximum values that define the scope of allowable values. To determine these values, invoke SYSGEN and issue a SHOW [parameter-name] command (with appropriate qualifiers). For example, to display the values for WSMAX, you can specify SHOW WSMAX; to display the values for the TTY parameters, you can specify SHOW/TTY. You can also display parameters grouped by attributes. To display DYNAMIC parameters, for example, specify SHOW/DYNAMIC.

Following is a list of system parameters grouped according to category. An asterisk indicates that a parameter is dynamic. Refer to the online SYSGEN Help for a description of each parameter.

---

**ACP Parameters**

---

ACP_BASEPRIO*	ACP_DATACHECK*	ACP_DINDXCACHE*
ACP_DIRCACHE*	DJTQUOTA*	ACP_EXTCACHE*
ACP_EXTLIMIT*	ACP_FIDCACHE*	ACP_HDRCACHE*
ACP_MAPCACHE*	ACP_MAXREAD*	MJTQUOTA

**SGN-34**    **SYSGEN**  
**Supplemental SYSGEN Information**

---

**ACP Parameters**

---

ACP_MULTIPLE*	ACP_QUOCACHE*	ACP_REBLDSYSD
ACP_SHARE*	ACP_SWAPFLGS*	ACP_SYSACC*
ACP_WINDOW*	ACP_WRITEBACK*	ACP_WORKSET*
ACP_XQP_RES*		

---

---

**CLUSTER Parameters**

---

ALLOCLASS*	DISK_QUORUM*	EXPECTED_VOTES
LOCKDIRWT	MSCP_BUFFER	MSCP_CREDITS
MSCP_LOAD	MSCP_SERVE_ALL	NISCS_CONV_BOOT
NISCS_LOAD_PEA0	NISCS_PORT_SERV	RECNXINTERVAL*
QDSKINTERVAL	QDKSVOTES	VAXCLUSTER
VOTES		

---

---

**JOB Parameters**

---

BJOBLIM*	DEFPRI*	DEFQUEPRI*
IJOBLIM*	MAXQUEPRI*	NJOBLIM*
RJOBLIM*		

---

---

**LGI Parameters**

---

LGI_BRK_DISUSER*	LGI_BRK_LIM*	LGI_BRK_TERM*
LGI_BRK_TMO*	LGI_HID_TIM*	LGI_RETRY_LIM*
LGI_RETRY_TMO*		

---

---

**MULTIPROCESSING Parameters**

---

MULTIPROCESSING	SMP_CPUS	SMP_LNGSPINWAIT
SMP_SANITY	SMP_SPINWAIT	

---

---

**PQL Parameters**

---

PQL_DASTLM*	PQL_DBIOLM*	PQL_DBYTLM*
PQL_DCPULM*	PQL_DDIOLM*	PQL_DENQLM*
PQL_DFILLM*	PQL_DPGFLQUOTA*	PQL_DPRCLM*
PQL_DTQELM*	PQL_DWSDEFAULT	PQL_DWSEXTENT*
PQL_DWSQUOTA*	PQL_MASTLM*	PQL_MBIOLM*
PQL_MBYTLM*	PQL_MCPULM*	PQL_MDIOLM*
PQL_MENQLM*	PQL_MFILLM*	PQL_MPGFLQUOTA*
PQL_MPRCLM*	PQL_MTQELM*	PQL_MWSDEFAULT
PQL_MWSEXTENT*	PQL_MWSQUOTA*	

---



---

**RMS Parameters**

---

RMS_DFMBBC*	RMS_DFMBFSDK*	RMS_DFMBFSMT*
RMS_DFMBFSUR*	RMS_DFMBFREL*	RMS_DFMBFIDX*
RMS_DFMBFHSH*	RMS_DFNBC*	RMS_PROLOGUE*
RMS_EXTEND_SIZE*	RMS_FILEPROT	RMS_GBLBUFQUO*

---



---

**SCS Parameters**

---

SCSBUFFCNT	SCSCONNCNT	SCSRESPCNT
SCSMAXDG	SCSMAXMSG	SCSFLOWCUSH*
SCSSYSTEMID	SCSSYSTEMIDH	SCSNODE
PRCPOLINTERVAL*	PASTIMOUT*	PASTDGBUF
PANUMPOLL*	PAMAXPORT*	PAPOLLINTERVAL*
PAPOOLINTERVAL*	PASANITY*	PANOPOLL*
UDABURSTRATE		

---



---

**Special Parameters**

---

CHANNELCNT	CONCEAL_DEVICES	DLCKEXTRASTK
EXUSRSTK	IMGIOCNT	IOTA
LOCKRETRY	LPRMIN	MPW_PRIO
NOAUTOCONFIG		NOCLUSTER

**SGN-36    SYSGEN**  
**Supplemental SYSGEN Information**

---

**Special Parameters**

---

PAGTBLPFC	PFRATS	PHYSICALPAGES
PIXSCAN	PSEUDOLOA	QBUS_MULT_INTR
RESALLOC	S0_PAGING	SMP_TICK_CNT
SRPMIN	SSINHIBIT	SWP_PPIO
SWPALLOCINC	SWPFAIL	SWPRATE
SYSPFC	TBSKIPWSL	VMS
WRITABLESYS		

---



---

**SYS Parameters**

---

AWSMIN*	AWSTIME*	BALSETCNT
BORROWLIM*	BUGCHECKFATAL*	BUGREBOOT*
CLISYMTBL*	CRDENABLE	DEADLOCK_WAIT*
DEFMBXBUFQUO*	DEFMBXMXMSG*	
DEFPRI*	DISMOUMSG*	DORMANTWAIT*
DUMPBUG	DUMPSTYLE	ERRORLOGBUFFERS
EXTRACPU*	FREEGOAL	FREELIM
GBLPAGES	GBLPAGFIL	GBLSECTIONS
GROWLIM*	INTSTKPAGES	IRPCOUNT
IRPCOUNTV	LAMAPREGS	LNMPHASHTBL
LNMSHASHTBL	LOCKIDTBL	LOCKIDTBL_MAX*
LONGWAIT*	LRPCOUNT	LRPCOUNTV
LRPSIZE	MAXBUF*	MAXPROCESSCNT
MAXSYSGROUP*	MINWSCNT	MOUNTMSG*
MPW_HILIMIT	MPW_IOLIMIT	MPW_LOLIMIT
MPW_LOWAITLIMIT*	MPW_THRESH*	MPW_WAITLIMIT*
MPW_WRTCLUSTER	MVTIMEOUT*	NPAGEDYN
NPAGEVIR	PAGEDYN	PAGFILCNT
PFCDEFAULT*	PFRATH*	PFRATL*
PROCSECTCNT	QUANTUM*	REALTIME_SPTS
RESHASHTBL	SAVEDUMP	SETTIME
SPTREQ	SRPCOUNT	SRPCOUNTV

Supplemental SYSGEN Information

---

**SYS Parameters**

---

STARTUP_P1-8	SWPFILCNT	SWPOUTPGCNT*
SYSMWCNT	TAPE_MVTIMEOUT*	TIMEPROMPTWAIT
UAFALTERNATE	VIRTUALPAGECNT	WINDOW_SYSTEM
WSDEC*	WSINC*	WSMAX
XFMAXRATE*		

---



---

**TTY Parameters**

---

TTY_ALTYPAMD	TTY_ALTALARM	TTY_AUTOCHAR
TTY_BUF	TTY_CLASSNAME	TTY_DEFCHAR
TTY_DEFCHAR2	TTY_DIALTYPE	TTY_DMASIZE*
TTY_OWNER	TTY_PARITY	TTY_PROT
TTY_RSPEED	TTY_SCANDELTA	TTY_SILOTIME
TTY_SPEED	TTY_TIMEOUT	TTY_TYPAHDSZ

---





---

## **SYSMAN Utility**

The System Management Utility (SYSMAN) centralizes system management so that you can manage nodes or clusters from one location.

### **format**

**RUN SYS\$SYSTEM:SYSMAN**

### **parameters**

None.

### **usage summary**

To invoke SYSMAN, enter the following command at the DCL prompt:

```
$ RUN SYS$SYSTEM:SYSMAN
```

The utility displays with the following prompt:

```
SYSMAN>
```

You can then enter SYSMAN commands at the SYSMAN> prompt. These commands follow the standard rules of DCL syntax.

As an alternative, you can enter the RSX command MCR, which expands to RUN SYS\$SYSTEM:

```
$ MCR SYSMAN
```

With the MCR command, you can invoke SYSMAN and supply a command in one command string. With any SYSMAN command (except SET ENVIRONMENT), SYSMAN executes the command string and exits. After executing a SET ENVIRONMENT command, the utility returns the SYSMAN> prompt.

To exit from SYSMAN and return to the DCL command level, enter the EXIT command at the SYSMAN> prompt or press CTRL/Z.

**NOTE:** To use SYSMAN, you must have the OPER privilege on the local node and authorization for the OPER or SETPRV privilege on any remote node in the management environment.

You must also have the privileges required by individual commands, as documented in the Command Section. To determine which privileges are required for DCL commands or for system management utilities, refer to the *VMS DCL Dictionary* or the appropriate utility reference.

**NOTE:** SYSMAN has the following restrictions:

- You cannot run SYSMAN from a batch job in any environment that requires a password.
- Some DCL commands, such as SET CLUSTER/QUORUM, MOUNT/CLUSTER, and some forms of the REPLY command, operate clusterwide by design, and should not be run in a SYSMAN environment defined as a cluster.

## SYSMAN Commands

This section describes the following SYSMAN commands and provides examples of their use.

---

### ALF ADD

Adds a new record to the ALF database.

Requires **READ (R)** and **WRITE (W)** access to the **SYSALF** database (**SYS\$SYSTEM:SYSALF.DAT**).

#### format

**ALF ADD** *device user*

#### parameters

##### ***device***

Specifies the terminal name or port name that you want to assign to a user name. The parameter **device** must be a terminal name if you do not specify qualifiers on the command line.

##### ***user***

Specifies the user name of the account that you want to assign to a particular terminal or port.

#### qualifiers

##### ***/TERMINAL***

Causes SYSMAN to treat **device** as a terminal name. This is the default behavior.

##### ***/PORT***

Causes SYSMAN to treat **device** as a port name. If the port name contains a special characters, such as a slash (/), or if it contains lowercase letters that you want to preserve, you must enclose the port name within quotation marks (" ").

##### ***/PROXY***

Causes SYSMAN to treat **device** as a port name. SYSMAN also checks that *device* is in the **NODE::USERNAME** format.

##### ***/LOG***

Causes SYSMAN to echo the device name and user name added to the ALF database.

## SM-4 SYSMAN ALF ADD

### description

You can use the SYSMAN ALF ADD command to associate a terminal or port with a particular user name. This will enable certain users to log in to certain terminals without specifying a user name.

The SYSMAN ALF ADD command adds a new record to the ALF database.

### example

```
SYSMAN> ALF ADD TTA3 JBERGERON  
SYSMAN> ALF ADD "MN34C3/LC-1-2" FMARTIN /PORT
```

The first command assigns terminal TTA3 to user FMARTIN. The second command assigns port MN34C3/LC-1-2 to user FMARTIN.

---

## ALF REMOVE

Removes one or more records from the ALF database.

**Requires READ (R) and WRITE (W) access to the SYSALF database (SYS\$SYSTEM:SYSALF.DAT).**

### format

**ALF REMOVE** [*device*]

### parameter

**[*device*]**

Specifies the terminal name or port name whose record you want to remove from ALF. You can use wildcard characters in the terminal name or port name.

**NOTE:** When you specify *device* to remove a record from the ALF database, be sure to use the correct format. Include special characters such as underscores ( `_` ) and colons ( `:` ). Enter the ALF SHOW command to display the device name format.

### qualifiers

**/USERNAME=user**

Allows you to remove a record in ALF by specifying a user name rather than a terminal name or port name. You can use wildcard characters with the /USERNAME qualifier.

**/CONFIRM**

Causes SYSMAN to display a message asking you to verify that you want to remove the record.

### **/LOG**

Causes SYSMAN to echo the device name and user name removed from the ALF database.

## **description**

The SYSMAN ALF REMOVE command removes one or more records from the ALF database.

## **example**

```
SYSMAN> ALF REMOVE _TTA3:  
SYSMAN> ALF REMOVE /USERNAME=SMITHSON
```

The first command removes the record for terminal TTA3. The second command removes all records assigned to user name SMITHSON

---

## **ALF SHOW**

Displays one or more records in the ALF database.

**Requires READ (R) and WRITE (W) access to the SYSALF database (SYS\$SYSTEM:SYSALF.DAT).**

## **format**

**ALF SHOW** [*device*]

## **parameter**

**[*device*]**

Specifies the terminal name or port name whose record you want to display. You can use wildcard characters in the terminal name or port name.

## **qualifiers**

**/USERNAME=*user***

Allows you to display the records held by the specified user name.

**/OUTPUT[=*file-spec*]**

Allows you to direct the output of the command to a file. If you do not include a file specification with this qualifier, SYSMAN writes the output to the file SYSMAN.LIS in your default directory.

## **description**

The SYSMAN ALF SHOW command displays one or more records in the ALF database.

## example

```
SYSMAN> ALF SHOW TTA* /USERNAME=MANESS /OUTPUT=ALF.TXT
```

The command in this example selects the records for all terminals named TTAx that are assigned to user MANESS and directs a listing of the records to the file ALF.TXT

---

## CONFIGURATION SET CLUSTER\_AUTHORIZATION

Modifies security data in a local area cluster.

Requires SYSPRV privilege.

### format

```
CONFIGURATION  
SET CLUSTER_AUTHORIZATION
```

### parameters

None.

### qualifiers

***/GROUP\_NUMBER=[n]***

Specifies the cluster group number that is recorded in SYS\$SYSTEM:CLUSTER\_AUTHORIZE.DAT. A group number uniquely identifies each local area cluster configuration on a single Ethernet. This number must be in the range from 1 to 4095 or 61440 to 65535.

***/PASSWORD=password***

Specifies a password for cluster access. A password consists of 1 to 31 characters, including alphanumeric characters, the dollar sign, and underscore. A password provides a second level of validation to ensure the integrity of individual clusters on the same Ethernet that accidentally use identical group numbers. A password also prevents an intruder who discovers the group number from joining the cluster.

### description

The CONFIGURATION SET CLUSTER\_AUTHORIZATION command modifies the group number and password of a local area cluster, as recorded in SYS\$SYSTEM:CLUSTER\_AUTHORIZE.DAT. If your configuration has multiple system disks, SYSMAN automatically updates each copy of CLUSTER\_AUTHORIZE.DAT, provided the environment is defined as a cluster (SET ENVIRONMENT/CLUSTER). For more information about CLUSTER\_AUTHORIZE.DAT, see the *VMS VAXcluster Manual*.

## CONFIGURATION SET CLUSTER\_AUTHORIZATION

**CAUTION:** If you change either the group number or the password, you must reboot the entire cluster.

The file CLUSTER\_AUTHORIZE.DAT is initialized during execution of CLUSTER\_CONFIG.COM and maintained through the SYSMAN Utility. Under normal conditions, you do not need to alter records in the CLUSTER\_AUTHORIZE.DAT file interactively. However, if you suspect a security breach, use the CONFIGURATION commands in SYSMAN to make the change.

### example

```
SYSMAN> SET ENVIRONMENT/CLUSTER/NODE=ASCONA
SYSMAN> SET PROFILE /PRIVILEGE=SYSPRV
SYSMAN> CONFIGURATION SET CLUSTER_AUTHORIZATION/PASSWORD=GILLIAN
Enter cluster group number [4027]: [RET]
%SYSMAN-I-GRPNOCHG, Group number not changed
SYSMAN-I-CAFREBOOT, cluster authorization file updated.
The entire cluster should be rebooted.
```

The CONFIGURATION SET CLUSTER\_AUTHORIZATION command in this example sequence modifies the cluster password. Note that the environment is defined to be a cluster, and the SYSPRV privilege is established before entering the CONFIGURATION SET CLUSTER\_AUTHORIZATION command.

## CONFIGURATION SET TIME

Modifies the current system time.

**Requires LOG\_IO privilege, and, in a cluster environment, SYSLCK privilege.**

### format

**CONFIGURATION SET TIME[=*time*]**

### parameters

None.

### description

The CONFIGURATION SET TIME command allows you to reset the system time. Specify a time value using the following format:

[dd-mmm-yyyy[:]] [hh:mm:ss.cc]

See the *VMS DCL Concepts Manual* for a discussion of acceptable time formats.

## SM-8 SYSMAN CONFIGURATION SET TIME

In an environment of individual nodes, SYSMAN sets the time to the specified value on each node. Without a time specification, SYSMAN sets the time according to the time-of-year clock on each node.

In a cluster environment, SYSMAN sets the time to the specified value on each node. However, if you do not specify a value, SYSMAN uses the time-of-year clock. In a local cluster, SYSMAN reads the clock on the node from which you are executing SYSMAN and assigns this value to all nodes in the cluster. In a remote cluster, SYSMAN reads the clock on the target node in the cluster and assigns that value to all nodes. Note that the time-of-year clock is optional for some processors; see the *VAX Hardware Handbook* for further information.

SYSMAN uses special processing in a cluster environment to ensure that all processors in the cluster are set to the same time. Because of communication and processing delays, it is not possible to synchronize clocks exactly. However, the variation is typically less than a few hundredths of a second. If SYSMAN cannot set the time to within one half second of the specified time, you receive a warning message that names the node that failed to respond quickly enough.

As a result of slight inaccuracies in each processor clock, times on various members of a cluster tend to drift apart. The following procedure synchronizes system times in a cluster environment:

```
$ SYNCH_CLOCKS:
$ RUN SYS$SYSTEM:SYSMAN
    SET ENVIRONMENT/CLUSTER
    CONFIGURATION SET TIME
    EXIT
$ WAIT 6:00:00
$ GOTO SYNCH_CLOCKS
```

The procedure sets the time on all cluster nodes to the value obtained from the local time-of-year clock, waits 6 hours, then resets the time for the cluster.

### example

```
SYSMAN> SET ENVIRONMENT/NODE=(ASCONA, LUGANO, LUCERN)
SYSMAN> SET PROFILE /PRIVILEGE=LOG_IO
SYSMAN> CONFIGURATION SET TIME=12:38:00
```

The CONFIGURATION SET command in this example sequence modifies the system time on nodes ASCONA, LUGANO, and LUCERN.



## CONFIGURATION SHOW CLUSTER\_AUTHORIZATION

---

**CONFIGURATION SHOW CLUSTER\_AUTHORIZATION**

Displays the group number and multicast address of a local area cluster.

**Requires SYSPRV privilege.**

**format**

```
CONFIGURATION
SHOW CLUSTER_AUTHORIZATION
```

**parameters**

None.

**qualifiers**

*/OUTPUT[=file-spec]*

Redirects output from SYS\$OUTPUT to the specified file. If no file specification is provided, SYSMAN writes the output to SYSMAN.LIS in the current directory.

**description**

The CONFIGURATION SHOW CLUSTER\_AUTHORIZATION command displays the group number and multicast address of a local area cluster, as recorded in SYS\$SYSTEM:CLUSTER\_AUTHORIZE.DAT during the CLUSTER\_CONFIG dialog. In a cluster or multinode environment, SYSMAN displays the group number of the first node and then displays the names of any nodes in the cluster whose group numbers, passwords, or both, are different. This command also displays the multicast address of the cluster.

**example**

```
SYSMAN> SET ENVIRONMENT/CLUSTER/NODE=ZENITH
.
.
.
SYSMAN> SET PROFILE /PRIVILEGE=SYSPRV
SYSMAN> CONFIGURATION SHOW CLUSTER_AUTHORIZATION
Node ZENITH: Cluster group number 65240
Multicast address: AB-00-04-01-F2-FF
```

The CONFIGURATION SHOW CLUSTER\_AUTHORIZATION command in this example displays the group number and multicast address of node ZENITH. Because the group number and password on other nodes in the cluster are identical, no further information is displayed.

**SM-10 SYSMAN  
CONFIGURATION SHOW TIME**

---

**CONFIGURATION SHOW TIME**

Displays the current date and system time to the hundredths of a second.

**format**

**CONFIGURATION SHOW TIME**

**parameters**

None.

**qualifiers**

***/OUTPUT[=file-spec]***

Redirects output from SYS\$OUTPUT to the specified file. If no file specification is provided, SYSMAN writes the output to SYSMAN.LIS in the current directory.

**example**

```
SYSMAN> SET ENVIRONMENT/CLUSTER/NODE=ZENITH
.
.
.
SYSMAN> CONFIGURATION SHOW TIME
System time on node ZENITH: 19-APR-1990 13:32:19.45
System time on node HOSTA: 19-APR-1990 13:32:27.79
System time on node KEBBI: 19-APR-1990 13:32:58.66
```

The CONFIGURATION SHOW TIME command in this example displays the system time for all nodes in the cluster.

---

**DISKQUOTA ADD**

Adds an entry to a disk quota file and initializes the usage count to zero.

**Requires WRITE (W) access to the quota file.**

**format**

**DISKQUOTA ADD *uic***

**parameter**

***uic***

Specifies the user identification code (UIC) for which the quota entry is added. You can specify the UIC in numeric or alphanumeric format. For complete information on UIC specification, refer to the *VMS DCL Concepts Manual*.

You can also add quota entries for rights identifiers. These are rights granted a user with the AUTHORIZE Utility. Rights identifiers use an ID format rather than a UIC format. See the *VMS System Services Reference Manual* for a complete description.

When working in nonlocal environments, be careful that the alphanumeric UIC or rights identifiers that you use are valid for the environment.

## qualifiers

### ***/DEVICE=device-spec***

Specifies the location of the quota file. SYSMAN validates the device specification. You can specify a logical name for *device-spec*. If you do, the logical name is translated in the target environment.

Without a device specification, SYSMAN uses the default disk on the target node. Unless you have set a default device with the SET PROFILE command, the default disk is the current device on the local node or the login default on another node, depending on the established environment.

### ***/OVERDRAFT=value***

Specifies a positive integer that provides an overdraft value for the specified UIC. If omitted, the overdraft value defaults to the overdraft value in the entry for [0,0].

### ***/PERMQUOTA=value***

Specifies a positive integer that provides the quota for the specified UIC. If omitted, the permanent quota defaults to the value of the quota in the entry for [0,0].

## description

The DISKQUOTA ADD command appends individual entries to a quota file on the specified disk. Note that the quota file must already exist and be enabled.

Unless you specify the permanent quota and overdraft values, the utility applies the default values from the UIC entry [0,0]. You adjust UIC [0,0] with the DISKQUOTA MODIFY command.

## example

```
SYSMAN> SET ENVIRONMENT/NODE=(ZURICH,ASCONA) ❶
%SYSMAN-I-ENV, Current command environment:
  Individual nodes: ZURICH,ASCONA
  Username ALEXIS will be used on nonlocal nodes.
SYSMAN> SET PROFILE /PRIVILEGE=SYSPRV ❷
SYSMAN> DISKQUOTA ADD [MKT,MORSE] /DEVICE=WORK1 /PERMQUOTA=200 -
_SYSMAN> /OVERDRAFT=50 ❸
SYSMAN> DISKQUOTA ADD PAYROLL /DEVICE=WORK1 /PERMQUOTA=1000 ❹
```

## SM-12 SYSMAN DISKQUOTA ADD

- ① Defines the management environment to be nodes ZURICH and ASCONA.
- ② Adds SYSPRV privilege to the user's current privileges in order to write to the quota file.
- ③ Adds UIC [MKT,MORSE] to the quota file on the device named WORK1 on both nodes ZURICH and ASCONA, setting the permanent quota to 200 disk blocks and the overdraft limit to 50 disk blocks, for an absolute limit of 250 blocks.
- ④ Adds an entry for the rights identifier PAYROLL. Any user holding the PAYROLL identifier can use this disk space.

---

## DISKQUOTA CREATE

Creates and enables a quota file for a disk volume that does not currently contain one.

**Requires WRITE (W) access to the volume's master file directory (MFD), plus one of the following: SYSPRV privilege, a system UIC, or ownership of the volume.**

### format

DISKQUOTA CREATE

### parameters

None.

### qualifiers

***/DEVICE=device-spec***

Specifies the disk volume on which to create a quota file. SYSMAN validates the device specification. A logical name may be specified for device-spec. If so, it is translated in the target environment.

Without a device specification, SYSMAN uses the default disk on the target node. Unless you have set a default device with the SET PROFILE command, the default disk is the current device on the local node or the login default on another node, depending on the established environment.

## description

The DISKQUOTA CREATE command creates a quota file for a volume that does not currently have one.

Only one quota file, [000000]QUOTA.SYS, can be present on any volume or volume set. As soon as you create a quota file, establish default values for quotas and overdrafts by adjusting UIC [0,0] with the DISKQUOTA MODIFY command. When a disk has existing files, use the DISKQUOTA REBUILD command to have SYSMAN update the quota file to contain current usage values.

**NOTE:** Digital recommends that you do not create and enable a quota file on the system disk.

## example

```
SYSMAN> SHOW ENVIRONMENT
%SYSMAN-I-ENV, Current command environment:
      Node ATHENS of local cluster
      Username ALEXIS      will be used on nonlocal nodes

SYSMAN> DO SHOW DEVICES
.
.
.
SYSMAN> DISKQUOTA CREATE /DEVICE=DJA31:
SYSMAN> DISKQUOTA MODIFY /DEVICE=DJA31: [0,0] /PERMQUOTA=10000 -
_SYSMAN> /OVERDRAFT=100
```

The commands in this example sequence display the characteristics of the current management environment and verify the device name. Then they create a quota file on the disk DJA31 and set up default quota values.

## DISKQUOTA DELETE

Deletes an entry from a quota file.

**Requires write access to the quota file.**

### format

**DISKQUOTA DELETE** *uic*

### parameter

#### *uic*

Specifies the user identification code (UIC). You can specify the UIC in numeric or alphanumeric format. For complete information on UIC specification, refer to the *VMS DCL Concepts Manual*.

## SM-14 SYSMAN DISKQUOTA DELETE

You can also specify quota entries for rights identifiers. These are rights granted a user with the AUTHORIZE Utility. Rights identifiers use an ID format rather than a UIC format. See the *VMS System Services Reference Manual* for a complete description.

When working in nonlocal environments, be careful that the alphanumeric UIC or rights identifiers that you use are valid for the environment.

### qualifiers

#### ***/DEVICE=device-spec***

Specifies the disk volume containing the quota file. SYSMAN validates the device specification and translates any logical name in the target environment before deleting the UIC entry.

Without a device specification, SYSMAN uses the default disk on the target node. Unless you have set a default device with the SET PROFILE command, the default disk is the current device on the local node or the login default on another node, depending on the established environment.

### description

The DISKQUOTA DELETE command eliminates the specified UIC from the quota file on the named device.

If the usage count for the UIC is not zero, the utility issues a warning message before it removes the UIC. Files remain on disk, and the user can still log on; however, any attempt to create files will fail.

The UIC [0,0] entry cannot be removed.

### example

```
SYSMAN> SET ENVIRONMENT/NODE=MARS
SYSMAN> SHOW PROFILE
%SYSMAN-I-DEFDIR, Default directory on node MARS -- WORK2:[CASEY]
%SYSMAN-I-DEFPRIV, Process privileges on node MARS --
    TMPMGX
    OPER
    NETMBX
    SYSPRV
SYSMAN> DISKQUOTA DELETE /DEVICE=DUA45: [TTD,DAVIS]
```

The command in this example deletes UIC [TTD,DAVIS] from the quota file for disk DUA45, which is located on node MARS.

## DISKQUOTA DISABLE

Suspends the maintenance and enforcement of disk quotas on a volume.

**Requires SYSPRV privilege, a system UIC, or ownership of the volume.**

### format

**DISKQUOTA DISABLE**

### parameters

None.

### qualifiers

***/DEVICE=device-spec***

Specifies a disk volume on which to disable a quota file. SYSMAN validates the device specification. A logical name may be specified for device-spec. If so, it is translated in the target environment.

Without a device specification, SYSMAN uses the default disk on the target node. Unless you have set a default device with the SET PROFILE command, the default disk is the current device on the local node or the login default on another node, depending on the established environment.

### description

The DISKQUOTA DISABLE command suspends quota operations on a volume. To permanently disable quotas on a device, disable the quotas with the DISKQUOTA DISABLE command and delete the file QUOTA.SYS. Otherwise, the system implicitly enables quotas when the disk is mounted, leaving invalid quota information.

If you enable the quota file later, enter the DISKQUOTA REBUILD command to update UIC entries and usage counts.

### example

```
SYSMAN> SET ENVIRONMENT/NODE=AMANDA  
SYSMAN> DISKQUOTA DISABLE /DEVICE=DJA1:
```

The command in this example suspends quota enforcement on disk DJA1, located on node AMANDA.

## DISKQUOTA ENABLE

Resumes quota enforcement on a disk volume.

**Requires SYSPRV privilege, a system UIC, or ownership of the volume.**

### format

DISKQUOTA ENABLE

### parameters

None.

### qualifiers

***/DEVICE=device-spec***

Specifies a disk volume on which to enable the quota file. SYSMAN validates the device specification. A logical name may be specified for device-spec. If so, it is translated in the target environment.

Without a device specification, SYSMAN uses the default disk on the target node. Unless you have set a default device with the SET PROFILE command, the default disk is the current device on the local node or the login default on another node, depending on the established environment.

### description

The DISKQUOTA ENABLE command reinstates the enforcement of quotas on a volume that had been suspended with the DISKQUOTA DISABLE command. Whenever you enable quotas on a volume, use the DISKQUOTA REBUILD command to update UIC entries and usage counts.

### example

```
SYSMAN> SET ENVIRONMENT/NODE=BAKER  
SYSMAN> SET PROFILE/DEFAULT=DJA12:[ALEXIS.MGR]  
SYSMAN> DISKQUOTA ENABLE  
SYSMAN> DISKQUOTA REBUILD
```

The command in this example resumes quota enforcement on the default disk DJA12, which is located on node BAKER. The DISKQUOTA REBUILD command updates the quota file, correcting quotas and adding any new entries.



## DISKQUOTA MODIFY

Changes an entry in a quota file or adjusts default values for quotas and overdrafts. If a new quota limit is less than the current usage count, the utility issues a warning message before it implements the new quota.

**Requires WRITE (W) access to the quota file.**

### format

**DISKQUOTA MODIFY** *uic*

### parameter

#### *uic*

Specifies the user identification code (UIC). You can specify the UIC in numeric or alphanumeric format. For complete information on UIC specification, refer to the *VMS DCL Dictionary*.

You can also specify quota entries for rights identifiers. These are rights granted a user with the AUTHORIZE Utility. Rights identifiers use an ID format rather than a UIC format. See the *VMS System Services Reference Manual* for a complete description.

When working in nonlocal environments, make sure that the alphanumeric UIC or rights identifiers that you use are valid for the environment.

### qualifiers

#### ***/DEVICE=device-spec***

Specifies the disk volume that contains the quota file. SYSMAN validates the device specification. A logical name may be specified for device-spec. If so, it is translated in the target environment.

Without a device specification, SYSMAN uses the default disk on the target node. Unless you have set a default device with the SET PROFILE command, the default disk is the current device on the local node or the login default on another node, depending on the established environment.

#### ***/OVERDRAFT=value***

Specifies a positive integer that provides an overdraft value for the specified UIC. If you omit a value, the overdraft value defaults to the overdraft value in the entry for [0,0].

#### ***/PERMQUOTA=value***

Specifies a positive integer that provides the quota for the specified UIC. If you omit a value, the permanent quota defaults to the value of the quota in the entry for [0,0].

## SM-18 SYSMAN DISKQUOTA MODIFY

### description

The DISKQUOTA MODIFY command changes values in a quota file for the disk named in the device specification. If you establish a quota limit that is less than the current usage count, a user can still log in and out, but cannot create files.

After creating a quota file, use the DISKQUOTA MODIFY command to set default values for quotas and overdrafts. UIC [0,0] sets the default permanent quota and overdraft values for a quota file, so you must change the entry [0,0] to values appropriate for your installation. Unless you specify quota and overdraft values when adding a file entry, the utility applies these defaults to UIC entries.

### example

```
SYSMAN> SET ENVIRONMENT/NODE=SIREN
SYSMAN> DISKQUOTA MODIFY /DEVICE=DUA12: [0,0] /PERMQUOTA=3000 -
_SYSMAN> /OVERDRAFT=300
```

The command in this example edits the entry for UIC [0,0] in the quota file on DUA12, which is located on node SIREN.

```
SYSMAN> DISKQUOTA MODIFY /DEVICE=SYS$DISK1 [TTD,DAVIS] -
_SYSMAN> /PERMQUOTA=900
```

The command in this example sets the permanent quota for UIC [TTD,DAVIS] to 900 blocks, while making no change to the overdraft limit. SYSMAN modifies the quota file that is located on disk SYS\$DISK1 in the current environment.

---

## DISKQUOTA REBUILD

Updates a quota file, adding new UICs and correcting usage counts for each user on the volume.

**Requires WRITE (W) access to the quota file, plus one of the following: SYSPRV privilege, a system UIC, or ownership of the volume.**

### format

**DISKQUOTA REBUILD**

### parameters

None.

## qualifiers

### */DEVICE=device-spec*

Specifies the disk volume that contains the quota file. SYSMAN validates the device specification and translates any logical name in the target environment before rebuilding the file.

Without a device specification, SYSMAN uses the default disk on the target node. Unless you have set a default device with the SET PROFILE command, the default disk is the current device on the local node or the login default on another node, depending on the established environment.

## description

The DISKQUOTA REBUILD command reads the disk, and updates usage counts for all existing entries and adds new entries. It sets quota and overdraft values to the defaults set in UIC [0,0] if the entry did not previously exist. While the REBUILD command is executing, file activity on the volume is frozen. No files can be created, deleted, extended, or truncated.

Use the DISKQUOTA REBUILD command in the following circumstances:

- After creating a quota file on a volume with existing files.
- When the quota file has been enabled after a period of being disabled. The command corrects the usage counts and adds any new UICs.

## example

```
SYSMAN> SET ENVIRONMENT /NODE=WEST  
SYSMAN> SET PROFILE /PRIVILEGE=SYSPRV  
SYSMAN> DISKQUOTA ENABLE /DEVICE=DUA226:  
SYSMAN> DISKQUOTA REBUILD /DEVICE=DUA226:
```

The command in this example enables the quota file and reconstructs the usage counts for all entries on disk DUA226, which is located on node WEST.

---

## DISKQUOTA SHOW

Displays quotas, overdrafts, and usage counts.

**Requires no additional privileges to display your own quota, overdraft, and usage count, but otherwise requires READ (R) access to the quota file.**

## SM-20 SYSMAN DISKQUOTA SHOW

### format

DISKQUOTA SHOW *uic*

### parameter

#### *uic*

Specifies the user identification code (UIC). You can specify the UIC in numeric or alphanumeric format. For complete information on UIC specification, refer to the *VMS DCL Concepts Manual*.

You can also specify quota entries for rights identifiers. These are rights granted a user with the AUTHORIZE Utility. Rights identifiers use an ID format rather than a UIC format. See the *VMS System Services Reference Manual* for a complete description.

You can use an asterisk wildcard character (\*) to specify the quota entry as follows:

Command	Description
DISKQUOTA SHOW [TTD,CJ]	Show user CJ in group TTD
DISKQUOTA SHOW [TTD,*]	Show all users in group TTD
DISKQUOTA SHOW *	Show all entries

### qualifiers

#### ***/DEVICE=device-spec***

Specifies the disk volume containing the quota file. DISKQUOTA validates device specification and translates any logical name in the target environment before displaying UIC entries.

Without a device specification, SYSMAN uses the default disk on the target node. Unless you have set a default device with the SET PROFILE command, the default disk is the current device on the local node or the login default on another node, depending on the established environment.

#### ***/OUTPUT[=file-spec]***

Directs output to the specified file. Without a file specification, /OUTPUT defaults to SYSMAN.LIS in the current directory on the local node where you are running SYSMAN.

### example

```
SYSMAN> DISKQUOTA SHOW [ACCT,*]
```

The command in this example displays quotas, overdrafts, and usage counts for all users in group ACCT on the default disk.

---

## DO

Executes a DCL command or DCL command procedure on all nodes in the current environment.

**Requires the privileges of the DCL command being executed.**

### format

**DO** *[command-line]*

### parameters

***[command-line]***

Specifies a command string that SYSMAN passes to the DCL for execution. For complete information on DCL command syntax, refer to the *VMS DCL Dictionary*.

### qualifiers

***/OUTPUT[=file-spec]***

Records output from the command in the specified file, which is located on the node from which you are executing SYSMAN. Position the qualifier immediately after the DO command. The default file specification is SYSMAN.LIS in the current device and directory. SYSMAN prefaces output with the message “%SYSMAN-I-OUTPUT Output From Node xxxxxx.”

### description

The DO command executes the accompanying DCL command or DCL command procedure on all nodes in the current environment. Each DO command executes as an independent process, so there is no process context retained between DO commands. For this reason, you must express all DCL commands in a single command string, and you cannot run a program that expects input.

In a cluster environment, SYSMAN executes the commands sequentially on all nodes in the cluster. Each command executes completely before SYSMAN sends it to the next node in the environment. Any node that is unable to execute the command returns an error message. The utility displays an error message if the timeout period expires before the node responds. Some DCL commands, such as MOUNT/CLUSTER, operate clusterwide by design. For these commands to execute successfully in SYSMAN, define the environment to be a single node within the cluster.

Use the RSX command MCR to run programs located in SYS\$SYSTEM. The MCR command allows you to run a program and supply a command in a single command string.

## SM-22 SYSMAN DO

### example

```
SYSMAN> SET ENVIRONMENT/CLUSTER/NODE=NONAME  
SYSMAN> DO/OUTPUT SHOW SYSTEM/BATCH
```

The first command in this example defines the management environment to be the cluster where NONAME is a member. The second command executes a DCL command on each node in the cluster. Output goes to the file SYSMAN.LIS rather than to the terminal.

```
SYSMAN> SET PROFILE /PRIVILEGES=(CMKRNL, SYSPRV) /DEFAULT=SYS$SYSTEM  
SYSMAN> DO INSTALL ADD /OPEN/SHARED WRKD$: [MAIN] STATSHR  
SYSMAN> DO MCR AUTHORIZE ADD JONES/PASSWORD=COLUMBINE/DEVICE=WORK1 -  
_SYSMAN> /DIRECTORY=[JONES]
```

The first command in this example adds CMKRNL and SYSPRV privileges to the current privileges because they are required by the INSTALL and the AUTHORIZE utility. The next command installs the file STATSHR. The last command sets up an account for user JONES, specifying a password as well as a default device and directory.

The MCR command in the last line of the example allows you to invoke the Authorize Utility from SYS\$SYSTEM and add a record to the UAF in one command string.

```
SYSMAN> SET ENVIRONMENT/NODE=LONDON  
SYSMAN> SET PROFILE /DEFAULT=[CJ.PROGRAMS] /PRIVILEGES=NOSYSPRIV  
SYSMAN> DO/OUTPUT @PROCESS_INFO
```

The commands in this example define the environment to be a single node and adjust the current privileges and directory. The DO command executes the command procedure PROCESS\_INFO.COM, located in directory [CJ.PROGRAMS] and writes any output to SYSMAN.LIS in the directory from which SYSMAN is running.

---

### EXIT

Terminates the SYSMAN session and returns control to the DCL command level. Any profile changes, established on the local node with the command SET PROFILE, are restored to their values at the time SYSMAN was invoked. You can also press CTRL/Z to exit at any time.

### format

EXIT

### parameters

None.

## qualifiers

None.

---

## HELP

Provides online help information for using the SYSMAN commands, parameters, and qualifiers. Press CTRL/Z to exit.

## format

**HELP** *[keyword...]*

## parameter

***[keyword]***

Specifies the command, parameter, or qualifier for which help information is to be displayed. If you omit the keyword, the HELP command displays a list of available help topics and prompts you for a particular keyword.

## qualifiers

None.

## example

SYSMAN> HELP DO

The command in this example displays help information about the SYSMAN command, DO.

---

## LICENSE LOAD

Activates licenses registered in the LICENSE database.

**Requires CMKRNL, SYSNAM, and SYSPRV privileges.**

## format

**LICENSE LOAD** *product*

## parameter

***product***

Specifies the name of the product whose license you want to activate.

## SM-24 SYSMAN LICENSE LOAD

### qualifiers

#### ***/DATABASE=file-spec***

Allows you to specify the location of the LICENSE database. The default file specification is SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB. You do not need to use the /DATABASE qualifier if you use the default LICENSE database name and location.

#### ***/PRODUCER=string***

Allows you to specify the name of the company that owns the product for which you have a license. Use this qualifier only if the product is from a company other than Digital.

### description

You can use the LICENSE LOAD command to activate licenses on multiple and nonlocal systems in the system management environment. The SYSMAN LICENSE commands are a subset of the License Management Facility (LMF) commands. For more information about the LMF, see the *VMS License Management Utility Manual*.

### example

```
SYSMAN> LICENSE LOAD FORTRAN
```

This command activates the license for VAX FORTRAN. Because the license is for a Digital product, the command does have the /PRODUCER qualifier.

---

## LICENSE UNLOAD

Deactivates licenses registered in the LICENSE database.

Requires CMKRNL, SYSNAM, and SYSPRV privileges.

### format

```
LICENSE UNLOAD [product]
```

### parameter

#### ***[product]***

Specifies the name of the product whose license you want to deactivate. If you enter the LICENSE UNLOAD command without specifying a product name, the system deactivates all available registered licenses.



## qualifiers

### ***/PRODUCER=string***

Allows you to specify the name of the company that owns the product for which you have a license. Use this qualifier only if the product is from a company other than Digital.

## description

The LICENSE UNLOAD command can be used to deactivate licenses on multiple and non local systems in the system management environment. The SYSMAN LICENSE commands are a subset of the License Management Facility (LMF) commands. For more information about the LMF, see the *VMS License Management Utility Manual*.

## example

```
SYSMAN> LICENSE UNLOAD FORTRAN
```

This command deactivates the license for VAX FORTRAN. Because the license is for a Digital product, the command does have the */PRODUCER* qualifier.

---

## PARAMETERS DISABLE CHECKS

Bypasses validation of parameter values. SYSMAN parameter validation consists of ensuring that the parameters fall within the defined minimum and maximum values specified in the PARAMETERS SET command.

## format

**PARAMETERS DISABLE CHECKS**

## parameters

None.

## qualifiers

None.

## description

The PARAMETERS DISABLE CHECKS command allows you to override minimum and maximum values established for system parameters. Parameter checks are *enabled* by default. If you attempt to set parameter values outside the allowable limits when checks are enabled, the operating system issues an error message. By disabling checks you can set parameter values regardless of the minimum and maximum limits.

## SM-26 SYSMAN PARAMETERS DISABLE CHECKS

**NOTE:** Range checks are enabled by default because Digital suggests that systems operate within these minimum and maximum values. Setting parameters outside these limits can result in system failures or hangs.

### example

```
SYSMAN> SET ENVIRONMENT/CLUSTER
SYSMAN> SET PROFILE/DEFAULT=SYS$SYSTEM/PRIVILEGES=CMEEXEC
SYSMAN> PARAMETERS SET MAXPROCESSCNT 10
%SMI-E-OUTRANGE, parameter is out of range
SYSMAN> PARAMETERS DISABLE CHECKS
SYSMAN> PARAMETERS SET MAXPROCESSCNT 10
```

In this example, the initial attempt to set MAXPROCESSCNT below the minimum fails because range checks are enabled. However, once range checks are disabled, the PARAMETERS SET MAXPROCESSCNT command succeeds.

---

## PARAMETERS ENABLE CHECKS

Validates all parameter values to ensure they fall within the defined minimum and maximum values.

Because range checks are enabled by default, use PARAMETERS ENABLE CHECKS after entering a PARAMETERS DISABLE CHECKS command.

### format

#### PARAMETERS ENABLE CHECKS

### parameters

None.

### qualifiers

None.

### example

```
SYSMAN> PARAMETERS DISABLE CHECKS
SYSMAN> PARAMETERS SET WSMAX 20
SYSMAN> PARAMETERS ENABLE CHECKS
SYSMAN> PARAMETERS SET WSMAX 30
%SMI-E-OUTRANGE, parameter is out of range
SYSMAN> PARAMETERS SHOW WSMAX
Parameter Name      Current  Default  Minimum  Maximum Unit  Dynamic
WSMAX                2000    1024     60       6400 pages
```

The PARAMETERS ENABLE CHECKS command in this example illustrates that when range checking is disabled, the system accepts a working set value (WSMAX) of 20. However, once range checking

is enabled with the PARAMETERS ENABLE CHECKS command, the system does not accept a WSMAX below the minimum, which is 60.

---

## PARAMETERS SET

Changes the value of a specific parameter in the work area.

The PARAMETERS SET command does not modify parameter files, the current system parameter file on disk, or the active system. For information on performing these modifications, see the PARAMETERS WRITE command.

### format

```
PARAMETERS SET  parameter-name  
                 value  
                 /STARTUP file-spec
```

### parameters

#### *parameter-name*

Specifies the name of the parameter to modify. Instead of a name, you can enter a period (.) to change the value of the most recently displayed or the most recently modified parameter. See the PARAMETERS SHOW command for an example of using the period in place of a parameter name.

For a list of system parameters and further information on them, use the command HELP PARAMETERS.

#### *value*

Specifies the new value for the parameter. Enclose values for ASCII parameters in quotation marks if they contain embedded spaces or other special characters.

Typically the value is an integer or the keyword DEFAULT. The keyword DEFAULT sets the parameter to its default value. The PARAMETERS SHOW command displays the defined minimum, maximum, and default values for the parameter, which are required unless range checking is disabled with the command PARAMETERS DISABLE CHECKS.

### qualifiers

#### */STARTUP file-spec*

Sets the name of the site-independent startup procedure to the given file specification. A file specification has a maximum length of 31 characters. The initial startup command procedure is SYS\$SYSTEM:STARTUP.COM.

## SM-28 SYSMAN PARAMETERS SET

### example

```
SYSMAN> PARAMETERS SET PFCDEFAULT 20
```

The PARAMETERS SET command in this example assigns a value of 20 to the PFCDEFAULT parameter.

```
SYSMAN> PARAMETERS SET GBLSECTIONS DEFAULT
```

The PARAMETERS SET command in this example assigns the default value (40) to the GBLSECTIONS parameter.

```
SYSMAN> PARAMETERS SET/STARTUP SYS$SYSTEM:XSTARTUP.COM
```

The command in this example assigns SYS\$SYSTEM:XSTARTUP.COM as the current site-independent startup command procedure.

---

## PARAMETERS SHOW

Displays the value of a parameter or a group of parameters in the work area. In addition, the command shows the minimum, maximum, and default values of a parameter and its unit of measure.

### format

```
PARAMETERS SHOW [parameter-name]
```

### parameter

#### *parameter-name*

Specifies the name of a parameter or a period (.). A period is interpreted as a request for the parameter specified in the last PARAMETERS SET or PARAMETERS SHOW command. The parameter name can be abbreviated, but the abbreviation must be unique because SYSMAN selects the first parameter that matches.

### qualifiers

#### **/ACP**

Displays all Files-11 ACP parameters.

#### **/ALL**

Displays the values of all active parameters.

#### **/CLUSTER**

Displays all parameters specific to clusters.

#### **/DYNAMIC**

Displays all parameters that would be in effect immediately after you enter a PARAMETERS WRITE ACTIVE command.

#### **/GEN**

Displays all general parameters.

***/HEX***

Displays numeric parameters in hexadecimal rather than decimal radix. Specify the */HEX* system parameter name or the parameter type. If you specify the */HEX* qualifier with the */NAMES* qualifier, */HEX* is ignored.

***/JOB***

Displays all Job Controller parameters.

***/LGI***

Displays all LOGIN security control parameters.

***/MAJOR***

Displays the most important parameters.

***/MULTIPROCESSING***

Displays parameters specific to multiprocessing.

***/NAMES***

Displays only parameter names. You can combine other qualifiers with this one.

***/OUTPUT***

Directs output to the specified file rather than SYS\$OUTPUT. Without a file specification, the output goes to SYSMAN.LIS in the current directory.

***/PQL***

Displays the parameters for all default process quotas.

***/RMS***

Displays all parameters specific to VMS Record Management Services (VMS RMS).

***/SCS***

Displays all parameters specific to VAXcluster System Communication Services.

***/SPECIAL***

Displays all special control parameters.

***/STARTUP***

Displays the name of the site-independent startup procedure.

***/SYS***

Displays all active system parameters.

***/TTY***

Displays all parameters for terminal drivers.

**SM-30    SYSMAN  
PARAMETERS SHOW**

**description**

Parameters are displayed in decimal unless the /HEX qualifier is specified. Note that ASCII values are always displayed in ASCII.

Abbreviations for parameter names must be unique because the first parameter matching the abbreviation is selected for display. No ambiguity checks are made. For example, a specification of PARAMETERS SHOW GBL displays the GBLSECTIONS parameter. To display the GBLPAGFIL parameter, you must specify PARAMETERS SHOW GBLPAGF to avoid further ambiguity with the GBLPAGES parameter.

You can use a period (.) to indicate that you want to work with the system parameter that was specified in the last PARAMETERS SET or PARAMETERS SHOW command.

**example**

```
SYSMAN> PARAMETERS SHOW GBLSECTIONS
Parameter Name    Current    Default    Minimum    Maximum Unit    Dynamic
GBLSECTIONS        100        40        20        -1 Sections
SYSMAN> PARAMETERS SET . 110
SYSMAN> PARAMETERS SHOW .
Parameter Name    Current    Default    Minimum    Maximum Unit    Dynamic
GBLSECTIONS        110        40        20        -1 Sections
```

In this example, the user first displays the values of the GBLSECTIONS parameter and then refers to the parameter with a period to set its current value to 110. The next PARAMETERS SHOW command also uses the period notation to obtain confirmation that the change occurred.

```
SYSMAN> PARAMETERS SHOW/ACP
```

The PARAMETERS SHOW command in this example produces the following output:

**SYSMAN    SM-31**  
**PARAMETERS SHOW**

Parameters in use: Active

Parameter Name	Current	Default	Minimum	Maximum	Unit	Dynamic
ACP_MULTIPLE	0	1	0	1	Boolean	D
ACP_SHARE	1	1	0	1	Boolean	
ACP_MAPCACHE	52	8	1	-1	Pages	D
ACP_HDRCACHE	138	128	2	-1	Pages	D
ACP_DIRCACHE	138	80	2	-1	Pages	D
ACP_DINDXCACHE	37	25	2	-1	Pages	D
ACP_WORKSET	0	0	0	-1	Pages	D
ACP_FIDCACHE	64	64	0	-1	File-Ids	D
ACP_EXTCACHE	64	64	0	-1	Extents	D
ACP_EXTLIMIT	300	300	0	1000	Percent/10	D
ACP_QUOCACHE	130	64	0	-1	Users	D
ACP_SYSACC	4	8	0	-1	Directories	D
ACP_MAXREAD	32	32	1	64	Blocks	D
ACP_WINDOW	7	7	1	-1	Pointers	D
ACP_WRITEBACK	1	1	0	1	Boolean	D
ACP_DATACHECK	2	2	0	3	Bit-mask	D
ACP_BASEPRIO	8	8	4	31	Priority	D
ACP_SWAPFLGS	14	15	0	15	Bit-mask	D
ACP_XQP_RES	1	1	0	1	Boolean	
ACP_REBLDSYS	0	1	0	1	Boolean	

SYSMAN> PARAMETERS SHOW/ACP/HEX

**The PARAMETERS SHOW command in this example produces a hexadecimal display of the values of the ACP system parameters.**

Parameters in use: Active

Parameter Name	Current	Default	Minimum	Maximum	Unit	Dynamic
ACP_MULTIPLE	00000000	00000001	00000000	00000001	Boolean	D
ACP_SHARE	00000001	00000001	00000000	00000001	Boolean	
ACP_MAPCACHE	00000034	00000008	00000001	FFFFFFFF	Pages	D
ACP_HDRCACHE	0000008A	00000080	00000002	FFFFFFFF	Pages	D
ACP_DIRCACHE	0000008A	00000050	00000002	FFFFFFFF	Pages	D
ACP_DNDXCACHE	00000025	00000019	00000002	FFFFFFFF	Pages	D
ACP_WORKSET	00000000	00000000	00000000	FFFFFFFF	Pages	D
ACP_FIDCACHE	00000040	00000040	00000000	FFFFFFFF	File-Ids	D
ACP_EXTCACHE	00000040	00000040	00000000	FFFFFFFF	Extents	D
ACP_EXTLIMIT	0000012C	0000012C	00000000	000003E8	Percent/10	D
ACP_QUOCACHE	00000082	00000040	00000000	FFFFFFFF	Users	D
ACP_SYSACC	00000004	00000008	00000000	FFFFFFFF	Directories	D
ACP_MAXREAD	00000020	00000020	00000001	00000040	Blocks	D
ACP_WINDOW	00000007	00000007	00000001	FFFFFFFF	Pointers	D
ACP_WRITEBACK	00000001	00000001	00000000	00000001	Boolean	D
ACP_DATACHECK	00000002	00000002	00000000	00000003	Bit-mask	D
ACP_BASEPRIO	00000008	00000008	00000004	0000001F	Priority	D
ACP_SWAPFLGS	0000000E	0000000F	00000000	0000000F	Bit-mask	D
ACP_XQP_RES	00000001	00000001	00000000	00000001	Boolean	
ACP_REBLDSYS	00000000	00000001	00000000	00000001	Boolean	

SYSMAN> PARAMETERS SHOW/STARTUP

Startup command file = SYS\$SYSTEM:STARTUP.COM

**The PARAMETERS SHOW command in this example displays the name of the site-independent startup command procedure.**

## PARAMETERS USE

Reads a set of system parameters into the work area for inspection or manipulation.

### format

**PARAMETERS USE**    *source*

### parameter

#### ***source***

The source of a system parameter file for data to be read into the work area. The source can be any of the following:

#### **ACTIVE**

Read parameters from the currently running system. When the work area is empty, this is the default. Use of the ACTIVE parameter requires the CMEXEC privilege.

#### **CURRENT**

Read parameters from the disk image of the currently running system. Use of the CURRENT parameter requires write access to SYS\$SYSTEM:VAXVMSSYS.PAR and the CMEXEC privilege.

#### **DEFAULT**

Read a parameter set containing the default values for all parameters.

#### ***file-spec***

Read parameters from a previously created system parameter file. The default file type is PAR. Use of the parameter requires write access to the file.

### qualifiers

None.

### description

The PARAMETERS USE command initializes the work area to use the values of a new parameter file, the current system parameter file, or the default values, if the active values do not provide a suitable base for subsequent operations.

### example

```
SYSMAN> PARAMETERS USE DEFAULT
```

The PARAMETERS USE command in this example initializes the work area with parameter values that should allow the VMS operating system to boot on any standard configuration. The initial values of the work area when the utility is invoked are the active values.



## PARAMETERS WRITE

Writes the contents of the work area to the specified destination.

### format

**PARAMETERS WRITE** *destination*

### parameter

#### *destination*

The destination of a new parameter file can be any of the following:

#### **ACTIVE**

Write the parameter set to the currently running system. Use of the ACTIVE parameter requires the CMKRNL privilege.

#### **CURRENT**

Write the parameter set to the disk image of the currently running system. The disk image is the current system parameter file on disk. Use of the CURRENT parameter requires write access to SYS\$SYSTEM:VAXVMSSYS.PAR and the CMEEXEC privilege.

#### *file-spec*

Create the given file and write the parameter set to it. The default file type is PAR. Use of the parameter requires write access to the file.

### qualifiers

None.

### description

The PARAMETERS WRITE command writes the system parameter values and the name of the site-independent startup command procedure from the work area to your choice of a parameter file, the current system parameter file on disk, or the active system in memory. (Only the dynamic parameter values are written to the active system.)

Both the PARAMETERS WRITE ACTIVE and PARAMETERS WRITE CURRENT commands send a message to OPCOM to record the event.

### example

```
SYSMAN> PARAMETERS WRITE SYS$SYSTEM:SPECIAL
```

The command in this example creates a new parameter specification file.

```
SYSMAN> PARAMETERS WRITE CURRENT
```

The command in this example modifies the current system parameter file on disk (SYS\$SYSTEM:VAXVMSSYS.PAR).

## SET ENVIRONMENT

Defines the node(s) or cluster to which subsequent commands apply.

**Requires OPER or SETPRV privilege on all nodes in the target environment.**

### format

**SET ENVIRONMENT**

### parameters

None.

### qualifiers

#### **/CLUSTER**

Directs SYSMAN to apply subsequent commands to all nodes in the cluster. By default, the management environment is the local cluster. Specify a nonlocal cluster by naming one cluster member with the /NODE qualifier.

#### **/NODE=(node1,node2,...)**

Specifies that SYSMAN execute subsequent commands on the given nodes. If accompanied by the /CLUSTER qualifier, the environment becomes the cluster where the given node is a member. A node name can be a system name or a cluster alias. You can also specify a logical name for the /NODE qualifier. However, before you can use logical names to define the command environment, you must set up the logical name table SYSMAN\$NODE\_TABLE.

#### **/USERNAME=username**

Specifies that this user name should be used for access control purposes on another node. SYSMAN uses the current user name if none is supplied. The utility prompts for a password whenever a new user name is specified.

### description

The SET ENVIRONMENT command defines the target node(s) or cluster for subsequent commands. When invoked, the system management environment is the local node where you are running SYSMAN. You can change the environment to any other node(s) in the cluster, the entire cluster, or any node(s) or cluster available through DECnet.

Designate a cluster environment with the /CLUSTER qualifier. When specifying a nonlocal cluster, also include the /NODE qualifier to identify the cluster.

You can display the current environment with the command **SHOW ENVIRONMENT**. To adjust privileges and defaults for the current environment, use the **SET PROFILE** command.

An environment exists until you exit from **SYSMAN** or establish another command context with the **SET ENVIRONMENT** command.

### example

```
SYSMAN> SET ENVIRONMENT/CLUSTER
%SYSMAN-I-ENV, Current command environment:
Clusterwide on local cluster
Username ALEXIS    will be used on nonlocal nodes
```

The commands in this example define the command environment as the local cluster. **SYSMAN** confirms the new environment.

```
SYSMAN> SET ENVIRONMENT/NODE=CLACK/CLUSTER
Remote Password:
%SYSMAN-I-ENV, Current command environment:
Clusterwide on remote node CLACK
Username ALEXIS    will be used on nonlocal nodes
```

The command in this example establishes a management environment on the cluster where node **CLACK** is a member. **SYSMAN** prompts for a password because it is a nonlocal environment.

```
SYSMAN> SET ENVIRONMENT/NODE=(LESETH,JOSHUA,TORIN)
%SYSMAN-I-ENV, Current command environment:
Individual nodes: LESETH,JOSHUA,TORIN
Username ALEXIS    will be used on nonlocal nodes
```

The command in this example defines the management environment to be three individual nodes.

```
$ CREATE/NAME_TABLE/PARENT=LNMS$SYSTEM_DIRECTORY SYSMAN$NODE_TABLE
$ DEFINE LAVCS SYS1,SYS2,SYS3,SYS4/TABLE=SYSMAN$NODE_TABLE
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT/NODE=(LAVCS)
%SYSMAN-I-ENV, Current command environment SYS1,SYS2,SYS3,SYS4)
Username ALEXIS    will be used on nonlocal nodes
```

The commands in this example set up the logical name table **SYSMAN\$NODE\_TABLE**, define a logical name (**LAVCS**), and use the logical name to define the command environment.

## SET PROFILE

Temporarily modifies a user's current privileges and default device and directory.

### format

**SET PROFILE**

### parameters

None.

### qualifiers

***/DEFAULT=device:[directory]***

Specifies the default disk device and directory name that the system should use in this environment to locate and catalog files.

***/PRIVILEGES=(priv1,priv2...)***

Specifies the privileges to add to the current privileges. Any enhanced privileges must be authorized.

### description

You need to consider the privilege requirements of commands that you will enter in an environment. The SET PROFILE command modifies process attributes for the current management environment. SYSMAN can add or delete current privileges, if they are authorized. It can also set a new default device and directory. Other attributes of your process remain constant. The profile is in effect until you change it, reset the environment, or exit from the utility.

### example

```
SYSMAN> SET PROFILE/DEFAULT=WORK1:[ALEXIS]
```

The command in this example changes the default device and directory in the user account to directory ALEXIS on device WORK1.

```
SYSMAN> SET PROFILE/PRIVILEGES=(SYSPRV,CMKRNL)
```

The command in this example make the authorized privileges, SYSPRV and CMKRNL, part of the current privileges. The privileges remain in effect until the environment changes, you enter another SET PROFILE command, or you exit.

## SET TIMEOUT

Establishes the amount of time SYSMAN waits for a node to respond. Once the time limit expires, SYSMAN proceeds to execute the command on the next node in the environment.

### format

**SET TIMEOUT** *time*

### parameter

#### *time*

Specifies a delta time value, which has the following format:

[dddd-] [hh:mm:ss.cc].

This is the amount of time that SYSMAN waits for a node to respond. By default, there is no timeout period, so SYSMAN waits indefinitely. See the *VMS DCL Concepts Manual* for a description of delta time values.

### qualifiers

None.

### example

```
SYSMAN> SET TIMEOUT 00:00:30
SYSMAN> CONFIGURATION SHOW TIME
System time on node ASCONA: 19-APR-1990 14:22:33
%SYSMAN-I-NODEERR, error returned from node LUGANO
%SMI-E-TIMEOUT, remote operation has timed out
System time on node JOSHUA: 19-APR-1990 14:23:15
```

The command in this example establishes a timeout period of 30 seconds. Because node LUGANO did not respond within 30 seconds, SYSMAN displays an error message and proceeds to execute the command on the next node in the environment.

## SHOW ENVIRONMENT

Displays the target nodes or cluster where SYSMAN is executing commands.

### format

**SHOW ENVIRONMENT**

# SM-38 SYSMAN

## SHOW ENVIRONMENT

### parameters

None.

### qualifiers

None.

### description

The **SHOW ENVIRONMENT** command displays the current management environment. It can be the local cluster, local or remote nodes, or a nonlocal cluster. **SYSMAN** indicates if the environment is limited to individual nodes or if it is clusterwide. It also shows the current user name.

The environment exists until you exit from **SYSMAN** or enter another **SET ENVIRONMENT** command.

### example

```
SYSMAN> SHOW ENVIRONMENT
%SYSMAN-I-ENV, Current command environment:
Clusterwide on local cluster
Username ALEXIS will be used on nonlocal nodes
```

The command in this example shows the current environment is the local cluster. User name **ALEXIS** will be used on other nodes in the cluster.

```
SYSMAN> SHOW ENVIRONMENT
%SYSMAN-I-ENV, Current command environment:
Clusterwide on remote cluster CLACK
Username ALEXIS will be used on nonlocal nodes
```

The command in this example shows that the command environment is a nonlocal cluster where node **CLACK** is a member.

```
SYSMAN> SHOW ENVIRONMENT
%SYSMAN-I-ENV, Current command environment:
Individual nodes: TURIN, JOSHUA
At least one node is not in local cluster
Username ALEXIS will be used on nonlocal nodes
```

The command in this example shows that the command environment consists of 2 nodes.

---

## SHOW PROFILE

Displays the current privileges and the default device and directory being used in the current environment.

### format

**SHOW PROFILE**

### parameters

None.

### qualifiers

**/DEFAULT=device:[directory]**

Specifies the default disk device and directory name that the system uses in this environment to locate and catalog files.

**/PRIVILEGES=(priv1,priv2...)**

Specifies privileges in effect for the current environment. Any enhanced privileges must be authorized.

### description

The SHOW PROFILE command displays the current privileges and the default device and directory that is being used in the current environment. You can modify these attributes with the SET PROFILE command.

These values remain in effect until you change environments or enter another SET PROFILE command.

### example

```
SYSMAN> SHOW PROFILE
%SYSMAN-I-DEFDIR, Default directory on node BAKER -- WORK1:[BERGERON]
%SYSMAN-I-DEFPRIV, Process privileges on node BAKER --
TMPMGX
OPER
NETMBX
SYSPRV
```

The command in this example shows the default device and directory as well as current privileges.

---

## SHOW TIMEOUT

Displays the amount of time SYSMAN waits for a node to respond. By default, there is no timeout period.

### format

**SHOW TIMEOUT**

### parameter

None.

### qualifiers

None.

### example

```
SYSMAN> SHOW TIMEOUT
%SYSMAN-I-TIMEVAL, timeout value is 00:00:04.00
```

The SHOW TIMEOUT command in this example displays the current timeout value, which is 4 seconds.

---

## STARTUP ADD

Adds a component to the startup database.

**Requires READ (R) and WRITE (W) access to the startup database.**

### format

**STARTUP ADD**    *FILE file-spec*

### parameter

#### **FILE**

Directs SYSMAN to add a component to the startup database. SYSMAN modifies STARTUP\$STARTUP\_LAYERED by default.

#### **file-spec**

Specifies which file to add to the startup database. Each component of the startup database must have a file type of COM or EXE and reside in SYS\$STARTUP.

### qualifiers

#### **/[NO]CONFIRM**

Controls whether SYSMAN displays the file specification of each file before adding it to the startup database and requests you to confirm the addition. If you specify /CONFIRM, you must respond to the prompt with



a Y (Yes) or a T (True) and press RETURN before the file is added. If you enter anything else, such as N or NO, the requested file is not added. The default is `/[NO]CONFIRM`.

***/[NO]LOG***

Controls whether the STARTUP ADD command displays the file specification of each file after it has been added.

***/MODE=mode***

Specifies the mode of execution for the file.

***/NODE=(node1,node2,...,nodex)***

Names the nodes within the cluster that run the file during startup. By default, a startup file executes on all nodes in the cluster.

***/PARAMETER=(P1:arg1,P2:arg2,...,P8:arg8)***

Specifies the parameters that are to be passed to the file during startup. Parameters that are omitted receive the default parameters defined by the system parameter `STARTUP_Pn`.

***/PHASE=phase-name***

Indicates the phase within system startup when the file is to be executed. Valid phases include `LPBEGIN`, `LPMAIN`, `LPBETA`, and `END`. `LPMAIN` is the default.

**description**

The STARTUP ADD command adds a component to the startup database. Startup components are the command procedures or executable files that perform actual startup work. Files from the startup database are used to start the VMS operating system, site-specific programs, and layered products. `STARTUP$STARTUP_VMS` and `STARTUP$STARTUP_LAYERED` list the components of the startup database.

Because a cluster shares one copy of the startup database, the SYSMAN environment can be defined as clustered or as a single node within the cluster.

**example**

```
SYSMAN> STARTUP ADD FILE /MODE=DIRECT /PHASE=LPMAIN -
_SYSMAN> FOR$LPMAIN_043_STARTUP.COM
```

The STARTUP ADD command in this example adds a record to the startup database that starts FORTRAN Version 4.3.

## STARTUP DISABLE

Prevents a file in the startup database from executing.

Requires **READ (R)** and **WRITE (W)** access to the startup database.

### format

**STARTUP DISABLE** *FILE file-spec*

### parameter

#### **FILE**

Directs SYSMAN to disable a component of the startup database. SYSMAN modifies STARTUP\$STARTUP\_LAYERED by default.

#### **file-spec**

Specifies the name of a component in the startup database. The startup file must reside in SYS\$STARTUP and have a file type of COM or EXE. The wildcard characters % and \* are permitted.

### qualifiers

#### **/[NO]CONFIRM**

Controls whether the STARTUP DISABLE command displays the file specification of each file before disabling it in the startup database and requests you to confirm that the file should be disabled. If you specify /CONFIRM, you must respond to the prompt with a Y (Yes) or a T (True) and press RETURN before the file is disabled. If you enter anything else, such as N or NO, the requested file is not disabled. The default is /[NO]CONFIRM.

#### **/[NO]LOG**

Controls whether the STARTUP DISABLE command displays the file specification of each file after it has been disabled.

#### **/NODE=(node1,node2,...,nodex)**

Identifies nodes within the cluster that do not run the file during startup. By default, the startup file is disabled on all nodes in the cluster.

#### **/PHASE=phase-name**

Indicates the phase of system startup in which the specified file normally executes. Valid phases include LPBEGIN, LPMAIN, LPBETA, and END. LPMAIN is the default.

### description

The STARTUP DISABLE command prevents a file in the startup database from executing. The command edits a record in the startup database, temporarily disabling the file.

## example

```
SYSMAN> STARTUP DISABLE FILE /NODE=BELA FOR$LPMAIN_043_STARTUP.COM
```

The command in this example modifies the startup database so that FORTRAN will not be installed on node BELA.

---

## STARTUP ENABLE

Allows a previously disabled file in the startup database to execute during system startup.

Requires **READ (R)** and **WRITE (W)** access to the startup database.

### format

**STARTUP ENABLE** *FILE file-spec*

### parameter

#### **FILE**

Directs SYSMAN to enable a component of the startup database. SYSMAN modifies STARTUP\$STARTUP\_LAYERED by default.

#### **file-spec**

Specifies the name of the startup file that you are enabling. Wildcard characters are accepted.

### qualifiers

#### **/[NO]CONFIRM**

Controls whether the **STARTUP ENABLE** command displays the file specification of each file before enabling it in the startup database and requests you to confirm that the file should be enabled. If you specify **/CONFIRM**, you must respond to the prompt with a Y (Yes) or a T (True) and press RETURN before the file is enabled. If you enter anything else, such as N or NO, the requested file is not enabled. The default is **/[NO]CONFIRM**.

#### **/[NO]LOG**

Controls whether the **STARTUP ENABLE** command displays the file specification of each file after it has been enabled.

#### **/NODE=(node1,node2,...,nodex)**

Names nodes within the cluster where the file should be enabled. By default, the startup file is enabled on all nodes.

#### **/PHASE=phase-name**

Indicates the phase within system startup when the specified file is to be enabled. Valid phases include LPBEGIN, LPMAIN, LPBETA, and END. LPMAIN is the default.

## SM-44 SYSMAN STARTUP ENABLE

### description

The **STARTUP ENABLE** command permits a file that was previously disabled to execute during system startup.

### example

```
SYSMAN> STARTUP ENABLE FILE /NODE=ZURICH FOR$LPMAIN_043_STARTUP.COM
```

The command in this example modifies the startup database. Node ZURICH will have FORTRAN Version 4.3 installed at startup.

---

## STARTUP MODIFY

Changes information associated with a startup file in the startup database.

**Requires READ (R) and WRITE (W) access to the startup database.**

### format

**STARTUP MODIFY** *FILE file-spec*

### parameter

#### **FILE**

Directs SYSMAN to modify a record in the startup database. SYSMAN modifies `STARTUP$STARTUP_LAYERED` by default.

#### **file-spec**

Selects a startup file for modification. Wildcard characters are accepted.

### qualifiers

#### **/[NO]CONFIRM**

Controls whether the **STARTUP MODIFY** command displays the file specification of each file before modifying its startup characteristics in the startup data file and requests you to confirm that the file characteristics should be modified. If you specify **/CONFIRM**, you must respond to the prompt with a Y (Yes) or a T (True) and press RETURN before the file is modified. If you enter anything else, such as N or NO, the requested file is not modified. The default is **/[NO]CONFIRM**.

#### **/[NO]LOG**

Controls whether the **STARTUP MODIFY** command displays the file specification of each file after its startup characteristics have been modified.

#### **/MODE=mode**

Changes the mode of execution for a startup file.

***/NAME=file-spec***

Changes the name of the startup file. The file must reside in SYS\$STARTUP.

***/PARAMETER=(P1:arg1,P2:arg2,...,P8:arg8)***

Changes the parameters that are to be passed to the file during startup. Parameters that are omitted receive the default parameters defined by the system parameter STARTUP\_Pn.

***/PHASE=phase-name***

Selects startup files for modification based on the phase in which they run. Valid phases include LPBEGIN, LPMAIN, LPBETA, and END. LPMAIN is the default.

## description

The STARTUP MODIFY command edits startup information associated with components in the startup database. For example, the command can rename a file or change the parameters that are passed to a file during startup. You can select a group of files for modification based on the phase in which they run.

## example

```
SYSMAN> STARTUP MODIFY FILE FOR$LPMAIN_043_STARTUP.COM -  
_SYSMAN> /PARAM=(P3:TRUE,P4:FALSE) /CONFIRM
```

The command in this example changes two startup parameters for the command procedure FOR\$LPMAIN\_043\_STARTUP.COM.

---

## STARTUP REMOVE

Deletes a record in the startup database, so the specified startup file no longer executes during system startup.

**Requires READ (R) and WRITE (W) access to the startup database.**

## format

**STARTUP REMOVE**    *FILE file-spec*

## parameter

***FILE***

Directs SYSMAN to remove a component from the startup database. SYSMAN modifies STARTUP\$STARTUP\_LAYERED by default.

***file-spec***

Specifies the name of the file to remove from the startup database. Wildcard characters are accepted.

SM-46    **SYSMAN**  
**STARTUP REMOVE**

**qualifiers**

***/[NO]CONFIRM***

Controls whether the **STARTUP REMOVE** command displays the file specification of each file before deleting its record in the startup database and requests you to confirm that the file should be deleted. If you specify **/CONFIRM**, you must respond to the prompt with a Y (Yes) or a T (True) and press RETURN before the file is removed. If you enter anything else, such as N or NO, the requested file is not removed. The default is **/[NO]CONFIRM**.

***/[NO]LOG***

Controls whether **SYSMAN** displays the file specification of each file after it has been removed.

***/PHASE=phase-name***

Indicates the phase of system startup from which the file should be removed. Valid phases include LPBEGIN, LPMAIN, LPBETA, and END.

**example**

```
SYSMAN> STARTUP REMOVE FILE FOR$LPMAIN_043_STARTUP.COM /LOG
```

The command in this example takes the file **FOR\$LPMAIN\_043\_STARTUP.COM** out of the startup database.

---

**STARTUP SET DATABASE**

Establishes the current startup database.

**format**

**STARTUP SET DATABASE**    *database*

**parameter**

***database***

Specifies the name of the target database, which is **STARTUP\$STARTUP\_LAYERED** by default. The second database, **STARTUP\$STARTUP\_VMS** is available for viewing; however, Digital recommends that you do not modify it.

**qualifiers**

None.

**example**

```

SYSMAN> STARTUP SET DATABASE STARTUP$STARTUP_LAYERED
%SYSMAN-I-NEWCOMPFIL, current component file is now STARTUP$STARTUP_LAYERED
SYSMAN> STARTUP SHOW FILE
%SYSMAN-I-COMPFIL, contents of component database on node LUCERN
Phase      Mode      File
-----
LPBEGIN    DIRECT    VMS$LPBEGIN_050_STARTUP.COM
LPMAIN     DIRECT    FOR$LPMAIN_043_STARTUP.COM

```

The commands in this example establish the layered products database as the default, so it can be displayed.

**STARTUP SHOW**

Displays the name of the current startup database or its components.

**format**

```

STARTUP SHOW    DATABASE
                  FILE

```

**parameter**

***DATABASE***

Directs SYSMAN to display the name of the current startup database. There are two startup databases: STARTUP\$STARTUP\_LAYERED and STARTUP\$STARTUP\_VMS. Digital recommends that you do not modify the STARTUP\$STARTUP\_VMS database.

***FILE***

Displays the contents of the current startup database. The display includes the file name, phase, and mode of execution for each component in the database.

**qualifiers**

***/FULL***

Displays full information about each component in the database. In addition to the phase, file name, and mode of execution for each startup component, SYSMAN displays the node(s) on which the file executes and the parameters passed to the file. Relevant with the FILE parameter.

***/NODE***

Displays the nodes within the cluster on which the file executes. By default, a startup file executes on all nodes in an environment. Relevant with the FILE parameter.

## SM-48 SYSMAN STARTUP SHOW

### ***/OUTPUT=file-spec***

Redirects command output from SYS\$OUTPUT to the file named with the qualifier. Without a file-spec, SYSMAN writes the output to SYSMAN.LIS in the current directory.

### ***/PARAMETERS***

Lists the parameters with which the startup file executes. Parameters that are not specified receive the defaults defined by the system parameter STARTUP\_Pn. Relevant with the FILE parameter.

### ***/PHASE=phase-name***

Displays components that execute in a specific phase of system startup. Valid phases include LPBEGIN, LPMAIN, LPBETA, and END. LPMAIN is the default. Relevant with the FILE parameter.

### **example**

```
SYSMAN> STARTUP SET DATABASE STARTUP$STARTUP_VMS
SYSMAN> STARTUP SHOW FILE
%SYSMAN-I-COMPFIL, contents of component database on node LUCERN
Phase      Mode      File
-----
BASEENVIRON  DIRECT  VMS$BASEENVIRON_050_LIB.COM
BASEENVIRON  CALLED  VMS$BASEENVIRON_050_SMISERVER.COM
BASEENVIRON  DIRECT  VMS$BASEENVIRON_050_VMS.COM
.
.
.
```

The commands in this example display the contents of the VMS startup database.



---

## Terminal Fallback Utility

The VMS Terminal Fallback Utility (TFU) is the user interface to the VMS Terminal Fallback Facility (TFF). This facility provides character conversion for terminals and can perform character compose emulation on input from a terminal.

Use the Terminal Fallback Utility to set up the system to use TFF character conversion tables, and to set, change, and display TFF terminal-related parameters.

### format

**RUN SYS\$SYSTEM:TFU**

### usage summary

To use the Terminal Fallback Utility (TFU), enter the following command in response to the DCL prompt:

```
$ RUN SYS$SYSTEM:TFU
```

The utility responds with the following prompt:

```
VAX/VMS Terminal Fallback Facility (TFF)
TFU>
```

After you invoke TFU, you can enter any of the TFU commands. These commands follow the standard rules for DCL commands.

To exit from TFU, enter the **EXIT** command at the TFU prompt:

```
TFU> EXIT
```

You can also exit from TFU by pressing **CTRL/Z**.

## TFU-2 TFU DIRECTORY

### TFU Commands

This section describes the Terminal Fallback Utility (TFU) commands.

---

#### DIRECTORY

Provides a directory of a TFF library file. You can specify selective, brief, or full directory listings.

If you specify a library name, that library becomes the current work library.

#### format

**DIRECTORY** [*library-name*]

#### parameter

##### *library-name*

Indicates the name of the library for which a directory listing is requested. If you have already established a work library, **library-name** is optional.

#### qualifiers

##### **/ALL**

Lists all tables in the target library.

##### **/COMPOSE**

Lists only compose sequence tables. You cannot use /COMPOSE simultaneously with /ALL or /FALLBACK.

##### **/FALLBACK**

Lists only fallback tables. This is the default for the DIRECTORY command. You cannot use /FALLBACK simultaneously with /ALL or /COMPOSE.

##### **/FULL**

Displays more detailed table information. By default, only one line of information is displayed about each table you select.

**example**

```
TFU> DIRECTORY
Directory of TFF library SYS$COMMON:[SYSEXE]TFF$MASTER.DAT;1
Name                   Type Base Description
-----
ASCII                   Fbk  MCS  MCS for ASCII (US)
ASCII_OVST              Fbk  MCS  MCS for hardcopy ASCII terminal (overstrike)
BRITISH                 Fbk  MCS  MCS for British NRC (BS 4730 [ISO 646 variant])
CANADIAN                Fbk  MCS  MCS for French-Canadian NRC (CSA Z243.4-1985)
.
.
.
SWEDISH_D47             Fbk  MCS  MCS for Swedish NRC (old type D47)
SWEDISH_E47             Fbk  MCS  MCS for Swedish NRC (SEN 85 02 00 - E47)
SWISS_VT102PY           Fbk  MCS  MCS for Swiss VT102PY
TURKISH                 Fbk  MCS  MCS for Turkish NRC (partial ISO 6937/2)
VT100_MCS               Fbk  MCS  MCS for VT100s with DEC-Supp in ROM#1
YUGOSLAVIAN             Fbk  MCS  MCS for Yugoslavian NRC (JUS I B1.002)
A total of 28 tables
TFU>
```

This example shows how to produce a brief directory listing of all the fallback tables in the current work library.

**EXIT**

Terminates the TFU session and returns you to the DCL command level. You can also type QUIT or press CTRL/Z or CTRL/C to exit from TFU.

**format**

**EXIT**

**HELP**

Allows you to obtain online information about the Terminal Fallback Utility.

**format**

**HELP** *[topic]*

**parameter**

***topic***

Indicates a topic about which you want information.

## TFU-4 TFU HELP

### example

TFU> HELP \*

This command provides information about all of the TFU commands. To obtain information about the individual commands or topics, enter HELP followed by the desired topic.

---

## LOAD TABLE

Loads a table from the current work library into nonpaged dynamic memory pool. Before you use this command, the fallback driver, FBDRIVER, must be loaded into memory by means of the System Generation Utility (SYSGEN) or SYS\$MANAGER:TFF\$STARTUP.COM. A table must be loaded into nonpaged dynamic memory pool before it can be used.

The following tables are always present and cannot be loaded or unloaded:

- ASCII—Fallback
- LATIN\_1—Compose sequence validation

### format

**LOAD TABLE** *table-name*

### parameter

*table-name*

Indicates the name of the table to be loaded.

### example

TFU> LOAD TABLE HEBREW\_VT100  
TFU>

This example shows how to load table HEBREW\_VT100 into nonpaged dynamic memory pool from the current work library.

---

## QUIT

Terminates the TFU session and returns you to the DCL command level. You can also type EXIT or press CTRL/Z or CTRL/C to exit from TFU.

**format**

QUIT

---

**SET DEFAULT\_TABLE**

Establishes a default table for the system. Before you specify a table as the system default, you must load the table into nonpaged dynamic memory pool using the LOAD command. The SET DEFAULT\_TABLE command reads the table type (fallback or compose) from the specified table's header and makes the target table the default for its type.

Before you enable any defaults, the following defaults apply:

- ASCII—Fallback
- LATIN\_1—Compose validation

**format****SET DEFAULT\_TABLE** *table-name***parameter*****table-name***

Indicates the name of the table to be the default table.

**example**

```
TFU> SET DEFAULT_TABLE HEBREW_VT100
TFU> SHOW DEFAULT_TABLE
System default TFF tables are:
  HEBREW_VT100                (fallback)
  LATIN_1                     (compose sequence validation)
TFU>
```

The command in this example establishes HEBREW\_VT100 as the default fallback table for the system. The table HEBREW\_VT100 must be loaded before you enter this command.

## SET LIBRARY

Allows you to declare a work library. Note that some commands implicitly declare a work library. If the library is located, it becomes the new work library.

### format

**SET LIBRARY** *library-name*

### parameter

#### *library-name*

Indicates the name of the library to be made the current library. You must specify a library with the SET LIBRARY command.

### example

```
TFU> SET LIBRARY SYS$SYSTEM:TFF$MASTER.DAT
TFU> LOAD HEBREW_VT100
TFU>
```

In this example, the first command sets the library to be `SYS$SYSTEM:TFF$MASTER.DAT` which is the default file name and location. This command directs TFF to use character conversion tables located in that file. The second command loads the table `HEBREW_VT100` into nonpaged dynamic memory pool.

---

## SET TERMINAL/FALLBACK

Enables or modifies TFF terminal parameters. The `/FALLBACK` qualifier is required, but you can place it before or after the *terminal-name* parameter.

`SET TERMINAL/NOFALLBACK` takes no options and is equivalent to `SET TERMINAL/FALLBACK=TABLE:NONE`.

### format

**SET TERMINAL/FALLBACK** [= (*option,...*)] [*terminal-name*]  
**SET TERMINAL/NOFALLBACK** [*terminal-name*]

### parameters

#### *terminal-name*

Indicates the target terminal for the set operation. If not specified, your own terminal is used. Note that you can use TFF only from local terminals; you cannot use terminal fallback on a remote terminal (RTAx),<sup>1</sup> the fallback terminal device (FBA0), a Packet Switch Interface (PSI)

---

<sup>1</sup> You can use TFF locally and then use the DCL command SET HOST to access a remote system.

## SET TERMINAL/FALLBACK

terminal (NVA0), a disconnected virtual terminal, or a terminal set for dynamic switching (DYN SWITCH) with DECnet.

**option**

Modifies the terminal parameters. If you specify more than one, enclose them in parentheses, and separate each with commas. You can use the following options with the FALLBACK=option qualifier:

Option	Definition
ACCEPT NOACCEPT	Enables input of 8-bit characters if the terminal is capable of generating 8-bit characters. The default is 7-bit character generation. 7-bit terminals, such as VT1xx and LA1xx, should have this feature turned off whereas VT2xx terminals may have it on (depending on the active table). The NOACCEPT option causes TFF to clear the eighth bit.
AUTOCOMPOSE NOAUTOCOMPOSE	Enables or disables all auto-compose keys available for the fallback table associated with a terminal. The AUTOCOMPOSE and NOAUTOCOMPOSE options override any keys specified with the ENABLE and DISABLE options.
DISABLE=(value[,...])	Disables one or more active auto-compose keys. Keys are chosen from the list of keys available for the fallback table associated with a terminal. The <b>value</b> argument is a list of the decimal values of the keys to disable. If you specify more than one value, separate the values with commas and place them in parentheses. SHOW TERMINAL/FALLBACK lists the currently active keys and their decimal values.
ENABLE=(value[,...])	Enables one or more auto-compose keys. Choose keys from the list of keys available for the fallback table associated with the specified terminal. The <b>value</b> argument is a list of the decimal values of the keys to enable. If you specify more than one value, separate the values with commas and place them in parentheses. SHOW TERMINAL/FALLBACK lists the currently active keys and their decimal values.

TFU-8 TFU  
SET TERMINAL/FALLBACK

Option	Definition
GX_DEFAULT:gx-name	<p>Defines as the default character set the name of a character set, previously defined and stored in Read Only Memory (ROM) of the specified terminal. For example, VT100LD specifies the line drawing alternate character set available on VT100 terminals, and DECSUPP specifies Digital's supplemental character set.</p> <p>These options are available for a variety of incompatible terminals. For example, the ASCII option applies to a special class of older Digital terminals that do not have an ASCII ROM that allows display of the full ASCII character set. These terminals have only the NRC set of characters.</p> <p>Currently you can specify any of the following character sets for the default: ASCII, CANADA, CANADA_2, DECSUPP, FINLAND, FINLAND_2, FRANCE, GERMANY, ITALY, JIS, NETHERLAND, NORDAN, NORWAY, NORWAY_2, SPAIN, SPECIAL1, SPECIAL2, SPECIAL3, SWEDEN, SWEDEN_2, SWISS, TCS, UK, or VT100LD.</p> <p>For more information about available default and alternate ROM-based character sets, see the documentation for your specific terminal.</p>
SIGNAL NOSIGNAL	<p>Enables the output of a BELL character to sound a terminal bell when an invalid compose sequence is entered. This is the default. You can disable this feature for applications that split escape sequences (for output) into two or more QIOs, because the BELL character may destroy such a sequence.</p>
SOFT NOSOFT_COMPOSE	<p>Enables software emulated compose, using the terminal's compose sequence validation table. You can enter compose sequences either by pressing CTRL/K followed by the sequence, or by pressing an auto-compose key followed by the second character of the sequence.</p>
SUSPEND NOSUSPEND	<p>Suspends or resumes TFF intervention. In command procedures that perform data transfers over the terminal line, use the SUSPEND option to avoid having to remember which TFF parameters are to be reset. The SUSPEND option suspends TFF intervention until you specify NOSUSPEND.</p>



TFU    TFU-9

## SET TERMINAL/FALLBACK

---

Option	Definition
TABLE:table-name	<p>Indicates the name of the fallback table to enable. If you omit the <b>table-name</b> option and the terminal does not yet have fallback enabled, then the system default is used. Otherwise, no change is made to the terminal's table. Specify <b>NONE</b> for the table to disable fallback for the target terminal. This is equivalent to <b>SET TERMINAL/NOFALLBACK</b>.</p> <p>Before you can enable it, the target table must be present in nonpaged dynamic memory pool. Use the <b>SHOW TABLES</b> command for information about what tables are available.</p>
TERMINAL:terminal_type	<p>Specifies the terminal type, as seen by TFF. The terminal type controls part of the escape sequence parsing done by TFF. Thus, you should set this to the correct value. Use one of the following values: <b>VT100</b>, <b>VT102</b>, <b>VT200</b>, or <b>AL_ARABI</b>. <b>VT102</b> also includes the terminals that are named <b>VT100xy</b>, for example, <b>VT100WF</b>.</p>

---

### example

```
TFU> SET TERMINAL/FALLBACK=(ACCEPT, NOSIGNAL)
TFU>
```

The command in this example enables fallback using system defaults, if they are not already enabled. The option **ACCEPT** enables input of 8-bit characters; **NOSIGNAL** disables the terminal bell that sounds when invalid compose sequences are entered.

---

### SHOW DEFAULT\_TABLE

Displays the default fallback tables for your system.

#### format

**SHOW DEFAULT\_TABLE**

#### example

```
TFU> SHOW DEFAULT_TABLE
System default TFF tables are:
  CANADIAN                (fallback)
  LATIN_1                  (compose sequence validation)
TFU>
```

The command in this example displays the default fallback and compose tables as they were established before the command was entered. In this example, the table **CANADIAN** is the default fallback table, and the table **LATIN\_1** is the default compose sequence validation table.

# TFU-10 TFU

## SHOW LIBRARY

---

### SHOW LIBRARY

Provides information about the current work library.

#### format

**SHOW LIBRARY**

#### example

```
TFU> SHOW LIBRARY
%TFF-I-READIS, Current input library is SYSSCOMMON:[SYSEXE]TFF$MASTER.DAT;1
TFU>
```

The command in this example lists the current work library. In this case, the default library TFF\$MASTER.DAT is listed.

---

### SHOW STATISTICS

Displays memory and other statistical information related to TFF.

#### format

**SHOW STATISTICS**

#### example

```
TFU> SHOW STATISTICS
TFF system statistics:
Memory (bytes) -
  Fixed memory:
    FBDRIVER                               5608
  Loaded tables:
    Compose tables (0)                      0
    Fallback tables (2)                    2288
  Memory allocated by fallback terminals (0):
    FBKs                                     0
    Replaced vectors                        0
  Total memory used (bytes):                7896
Misc -
  Total tables loaded since boot: 2
System default TFF tables are:
  CANADIAN                                 (fallback)
  LATIN_1                                  (compose sequence validation)
TFU>
```

The command in this example displays information about TFF use. From this display you can see that two fallback tables have been loaded (in addition to the default table), no new compose tables have been loaded, and no fallback terminals have memory allocated to them. Other information is also displayed.

## SHOW TABLES

Displays information about all loaded TFF conversion tables.

### format

**SHOW TABLES**

### example

```
TFU> SHOW TABLES
The following TFF tables are currently loaded
Name                               Type Base Crefc Trefc
-----
ASCII                               Fbk  MCS   *   0 0
LATIN_1                             Cmp  MCS   *   0 0
HEBREW_VT100                         Fbk  Hebr   0 0
CANADIAN                             Fbk  MCS    0 0
%TFF-W-NOMORETAB, No more tables in wildcard scan
```

This example shows how to display a line of information about the tables currently loaded into nonpaged dynamic memory pool.

## SHOW TERMINAL /FALLBACK

Displays TFF statistics about a specific terminal. The **/FALLBACK** qualifier is required, but you can place it before or after the **terminal-name** parameter.

### format

**SHOW TERMINAL/FALLBACK** [*terminal-name*]

### parameter

#### *terminal-name*

Indicates the target terminal for the show operation. If excluded, your own terminal is used. Note that you can use TFF only from local terminals; you cannot use terminal fallback on a remote terminal (RTAx), the fallback terminal device (FBA0), a Packet Switch Interface (PSI) terminal (NVA0), a disconnected virtual terminal, or a terminal set for dynamic switching (DYNSWITCH) with DECnet.

### qualifiers

#### **/ESCAPE\_STATE**

Displays information about escape sequence parsing and triggering Read Only Memories (ROMs). Use this information to debug your application.

#### **/FLAGS**

Displays which TFF terminal flags (options) you can set from the terminal, and displays any internal TFF flags.

**TFU-12 TFU**  
**SHOW TERMINAL /FALLBACK**

***/FULL***

Displays full information about the terminal. You cannot use this qualifier with */ESCAPE\_STATE* or */STATISTICS*.

***/STATISTICS***

Displays statistics about the specified terminal.

***/TABLES***

Displays the names of tables assigned to the specified terminal, including auto-compose keys for the fallback table.

**example**

```
TFU> SHOW TERMINAL/FALLBACK/FULL TXB0:  
TFF status for physical terminal _TXB0:
```

```
Active tables:
```

```
ASCII          (FALLBACK)  
LATIN_1        (compose sequence validation)
```

```
Autocompose-keys (Parenthesized values are character's decimal value):
```

```
None
```

```
Settable flags:
```

```
Nosuspend, Noaccept_8bit, Soft_compose, Signal, NoGR_terminal
```

```
Internal state flags:
```

```
None
```

```
Rom(s) that will trigger TFF I/O conversion:
```

```
ASCII
```

```
Escape sequence parsing states:
```

```
Input_state: Off (0), Output_state: Off (0)
```

```
Terminal graphic registers for the next character (setup = VT00):
```

```
G0 = ASCII, G1 = ASCII
```

```
Output mapping:
```

```
GL = G0 (maps 7-bit; 8th bit is truncated)
```

```
Output formatter expansion:
```

```
Received: 4579 Transmitted: 4579 Expansion rate: +0.0%
```

```
Replaced vector sizes (bytes):
```

```
Port vector: 99, Class vector: 139
```

```
TFU>
```

This example shows how to produce a full display of TFF information for terminal TXB0.

## UNLOAD TABLE

Unloads a table from nonpaged dynamic memory pool, releasing all memory used by the specified table. You can only unload tables that are not currently referenced by users and that are not the system default table. You must log out or enter SET TERMINAL/NOFALLBACK from your terminal to release a table for unloading. Note that you cannot unload the ASCII and LATIN\_1 tables.

### format

**UNLOAD TABLE** *table-name*

### parameter

***table-name***

Indicates the name of the table to be unloaded.

### example

```
TFU> UNLOAD TABLE HEBREW_VT100  
TFU>
```

The command in this example unloads table HEBREW\_VT100 from nonpaged dynamic memory pool.



# Index

## A

- Access
  - denying with identifier ACE, 11-22
  - network object, 6-29
  - protecting network against unauthorized, 6-27
- Access control list entry
  - See ACE
- Access controls
  - for circuits, 6-29
  - for nodes, 6-29
  - for system, 6-29
- Access control string
  - definition, 6-27
  - invalid, 6-37
  - secondary passwords with, 11-8
  - using to protect file, 6-27
- Access type
  - and security audit, 11-25
- Account
  - automatic login, 4-8
  - default DECnet-VAX, 6-30
  - disabling, 4-13
  - disguising identity, 11-25
  - proxy, 6-28
  - setting up to use project identifiers, 11-23
- ACCOUNTING.DAT
  - See Accounting log file
- Accounting log file
  - events recorded, 10-12
  - overview, 10-12
  - producing reports from, 10-14
  - records and fields in, 10-14
- Accounting Utility (ACCOUNTING)
  - 10-12
  - as network troubleshooting aid, 6-41
- ACE (access control list entry), 11-18
  - identifier, 11-19
  - positioning considerations, 11-19, 11-22
  - syntax of, 11-19 to 11-22
  - types of, 11-19
- ACL (access control list), 11-15 to 11-22
  - creation and maintenance of, 11-16
  - maintaining current, 11-22
  - usage considerations, 11-22
- ACL Editor, 11-16
- ACTIVE reserved word
  - plural form of component name, 6-32
- ADD/IDENTIFIER command, 11-23
- ADJACENT reserved word
  - plural form of component name, 6-32
- Alarm
  - security applications, 11-24
- ALFMaint procedure, 4-8
- Alias node identifier, 6-4
  - in a VAXcluster environment, 6-13
- Alias node name, 6-4
- ALL parameter
  - with NCP SET command, 6-26
- Alternate system root
  - specifying for software installations, 3-6
  - VMSINSTAL.COM option, 3-10
  - restriction, 3-10
- ANALYZE/ERROR\_LOG command, 10-3
- Answer file (for software installation), 3-7
- Area
  - number, 6-9

## Index-2

AST limit

See ASTLM

ASTLM

for software installation, 3-2

value for efficient backups, 8-11

Asynchronous connection

See also Dynamic asynchronous connection

configuration, 6-15

DDCMP, 6-4

dynamic DDCMP, 6-15

static, 6-15

static DDCMP, 6-15

troubleshooting problems, 6-42

Asynchronous DDCMP driver, 6-16

Audit analysis, 11-29

Auditing

applications, 11-25

as security feature, 11-25

AUTHORIZE

See Authorize Utility

Authorize Utility (AUTHORIZE)

adding a user account, 4-6

checking UAF quotas for software installation, 3-3

creating a captive account, 4-7

creating an automatic login account, 4-8

creating and modifying records in NETUAF.DAT, 4-1

creating and modifying records in rights database file, 4-1

creating and modifying records in SYSUAF.DAT, 4-1

default values for new users, 4-3

deleting a user record, 4-11

disabling a user account, 4-13

for network proxy database management, 6-28

listing user records, 4-10

modifying a user account, 4-10

overview, 4-1

proxy account, 6-28

restricting login hours with, 9-5

sample user record, 4-2

Authorize Utility (AUTHORIZE) (cont'd.)

setting process quotas for efficient backups, 8-10

Autodial protocol, 6-22

AUTOGEN

performance tuning, 9-7

Automatic configuration

of DECnet-VAX network, 6-5, 6-6

Automatic login account, 4-8

Automatic switching

of terminal line, 6-22

## B

Backup

command line, 8-4

command procedures for, 8-6

interactive backup, 8-9

nightly image backup, 8-6

nightly incremental backup, 8-8

definition, 8-1

image

definition, 8-2

procedure for making, 8-4

restoring files from, 8-14

sample command procedure for, 8-6

incremental

definition, 8-2

procedure for making, 8-5

restoring files from, 8-16

sample command procedure for, 8-8

initializing a tape, 8-5

input specifier, 8-4

interactive

sample command procedure for, 8-9

open files during a backup, 8-3

output specifier, 8-4

preparing a system for, 8-10

restoring files

See also Restoring files

overview, 8-13

save set

definition, 8-3



- Backup
    - save set (cont'd.)
      - listing contents of, 8-19
      - standalone Backup, 8-20
      - tape label processing, 8-22
      - volume label
        - processing procedure, 8-22
        - volume label, definition, 8-5
  - BATCH identifier, 11-17
  - Batch job, 5-1
    - changing priority of, 5-8
    - deleting, 5-5
    - holding, 5-6
    - monitoring, 5-4
    - releasing, 5-6
    - requeueing, 5-6, 5-7
    - retaining, 5-6
    - submitting at system startup, 2-9
    - submitting in startup command file, 2-9
  - Batch queue
    - See also Queue
    - definition, 2-6
  - Binary output, 11-28, 11-31
  - Binary qualifier, 11-31
  - BIOLM
    - for software installation, 3-2
    - value for efficient backups, 8-11
  - Booting
    - definition, 2-2
  - Boot node
    - See Boot server
  - Boot server, 7-2
    - definition, 7-1
    - establishing a node as, 7-5
  - Break-in
    - auditing, 11-25
    - counteraction through dual password, 11-7
    - detection and evasion, 11-12
  - Break-in database, 11-15
  - Buffered byte count quota limit
    - See BYTLM
  - Buffered I/O limit
    - See BIOLM
  - BYTLM
    - for software installation, 3-2
    - value for efficient backups, 8-11
- ## C
- Cable
    - null modem, 6-15
  - Captive account
    - creating, 4-7
    - definition, 4-6
  - CI-only clusters, 7-2
  - Circuit
    - access control, 6-29
    - detecting failure, 6-31
    - determining status, 6-31
    - displaying counter information, 6-31
    - logging failures, 6-33
  - Cluster
    - executing commands on each node, 7-7
  - Clusters
    - See also Local area cluster
    - adding satellite nodes, 7-5
    - alias node identifier, 6-13
    - alias node name, 6-4
    - benefits of, 7-3
    - boot server, 7-2
    - CI-only, 7-2
    - communication mechanisms, 7-6
    - establishing a boot server, 7-5
    - installing the VMS operating system on, 7-3
    - local area, 7-2
    - node address, 6-4, 6-9
    - node name, 6-9
    - preparing a system for, 7-3
    - print and batch queues on, 7-3
    - satellite nodes, 7-2
    - shared resources, 7-3
    - types of, 7-1
    - using CLUSTER\_CONFIG.COM, 7-4
    - using SYSMAN in, 7-6
  - CLUSTER\_CONFIG.COM command
    - procedure, 7-4
    - adding a satellite node with, 7-5

## Index-4

CLUSTER\_CONFIG.COM command  
    procedure (cont'd.)  
    establishing a boot server with, 7-5

Collection points  
    for network events, 6-34

Command procedure  
    site-specific startup  
        See SYSTARTUP\_V5.COM  
        command procedure

Command procedures  
    configuring a cluster  
        See CLUSTER\_CONFIG.COM  
        command procedure

    for backup  
        See Backup, command procedures  
        for

    for configuring a DECnet-VAX network  
        See NETCONFIG.COM command  
        procedure

    for defining systemwide logical names  
        See SYLOGICALS.COM command  
        procedure

    for installing software products  
        See VMSINSTALL.COM command  
        procedure

    shutting down the system  
        See SHUTDOWN.COM command  
        procedure

    site-specific startup  
        See SYSTARTUP\_V5.COM  
        command procedure

    system login command  
        See SYLOGIN.COM command  
        procedure

Communications line  
    definition, 6-2

Configuration  
    automatic network, 6-5, 6-6  
    DECnet-VAX node, 6-5  
    detecting changes in network, 6-31  
    manual network, 6-6

Configuration database, 6-38  
    DECnet-VAX, 6-5, 6-26  
    permanent, 6-26  
    tailoring with NCP, 6-25

Configuration database (cont'd.)  
    volatile, 6-26

Connection  
    asynchronous DDCMP, 6-4  
    count of requests for, 6-32  
    Ethernet, 6-4  
    synchronous DDCMP, 6-4

Counters  
    frequency of logging, 6-33  
    network use of, 6-32  
    resetting to zero, 6-33

Counter timer  
    expiration of, 6-33  
    setting, 6-33

Crash dump  
    system dump analyzer, 2-8

## D

Database  
    creating (volatile node), 6-14  
    DECnet-VAX configuration, 6-5, 6-25,  
        6-38  
    default object, 6-5  
    memory-resident (volatile), 6-27  
    node, 6-4  
    permanent network, 6-6  
    permanent proxy, 6-28  
    rights, 11-16  
    volatile network, 6-6, 6-21

Data link  
    problems, 6-39

Data packet transmission  
    and circuit counters, 6-32

DDCMP (Digital Data Communications  
    Message Protocol)  
    asynchronous communication, 6-15  
    asynchronous driver, 6-16, 6-20  
    dynamic connection, 6-15  
    static connection, 6-15

DECnet Test Sender/DECnet Test  
    Receiver Utility, 6-30  
    as a network monitoring tool, 6-31

DECnet-VAX  
    at system startup, 2-8  
    automatic configuration, 6-6

## DECnet-VAX (cont'd.)

- cluster connections, 7-6
  - configuration database, 6-5, 6-26
  - default account, 6-37
  - default account (nonprivileged), 6-30
  - defining node names, 6-13
  - dynamic asynchronous connection,
    - 6-15, 6-21, 6-22, 6-42
  - end node PAK (DVNETEND), 6-5
  - error messages and meanings, 6-36
  - event class, 6-34
  - event logger, 6-25, 6-33
    - displaying information, 6-31
  - event type, 6-34
  - full function PAK (DVNETRTG), 6-5
  - INBOUND parameter, 6-21
  - installing dynamic asynchronous connection, 6-20
  - installing static asynchronous connection, 6-15
  - license, 6-4, 6-5
  - manual configuration, 6-6
  - node address, 6-8
  - node name, 6-8
  - object, 6-25
  - PAK, 6-4, 6-5
  - procedure to begin using, 6-1
  - receive password, 6-18, 6-21
  - registering the PAK, 6-10
  - restarting, 6-27
  - restarting after system shutdown,
    - 6-25
  - shutting down, 6-25
  - shutting down for software installation,
    - 3-2
  - static asynchronous connection, 6-15
  - transmit password, 6-18, 6-21
  - using with DECwindows, 6-24
- DECnet-VAX account
- default, 6-37
- DECwindows
- using DECnet-VAX with, 6-24
- DEFAULT ACCESS parameter
- for NCP commands, 6-29
- DEFAULT account
- user authorization file entry, 4-3

## Default ownership

- management, 11-24
- DELETE/ENTRY command, 5-5
- DELETE/INTRUSION command, 11-15
- Deleting user account, 4-11
- See also Disabling user account
- DEUNA
- Ethernet UNA device, 6-26
- Device
- DEUNA, 6-26
  - Ethernet UNA, 6-26
  - site-specific startup, 2-6
  - status report, 10-7
- Dialup
- retries, controlling, 11-11
- DIALUP identifier, 11-17
- Dialup line
- connection security, 6-18, 6-21, 6-29
  - using for dynamic asynchronous connection, 6-20
  - using for static asynchronous connection, 6-15, 6-16, 6-18, 6-19
- Digital Data Communications Message Protocol
- See DDCMP
- Digital Network Architecture
- See DNA
- DIOLM
- for software installation, 3-2
  - value for efficient backups, 8-11
- Direct I/O limit
- See DIOLM
- Directory
- restoring from a backup copy, 8-18
- Disabling network event logging, 6-36
- Disabling user account, 4-13
- See also Deleting user account
- DISFORCE\_PWD\_CHANGE flag, 11-9
- Disk I/O
- reducing to improve performance, 9-10
- Disk quota
- example, 11-23
- Disk space
- usage and charging, 11-23

## Index-6

- Disk volume
  - mounting public, 2-5
- DNA (Digital Network Architecture)
  - layered design and troubleshooting, 6-38
- DNA layers
  - as basis for troubleshooting network, 6-38
- Driver
  - See Asynchronous DDCMP driver
- DTS/DTR
  - See DECnet Test Sender/DECnet Test Receiver Utility
- Dual passwords
  - advantages and disadvantages, 11-7
  - and maximum security, 11-5
- Dump file
  - See System dump file
- DVNETEND
  - end node DECnet-VAX PAK, 6-5
- DVNETRTG
  - full function DECnet-VAX PAK, 6-5
- Dynamic asynchronous connection, 6-15
  - automatic switching of terminal line, 6-22
  - manual terminal line switching, 6-22
  - procedure for establishing, 6-20
  - reasons for failure, 6-42
  - receive password, 6-21
  - security, 6-21
  - switching of terminal line, 6-20
  - terminating link, 6-23
  - transmit password, 6-21
- DYN SWITCH image, 6-20
- E**
- Editor
  - See ACL Editor
- Emulator
  - See Terminal emulator
- End node, 6-4, 6-9
- ENQLM
  - for software installation, 3-2
- Enqueue quota limit
  - See ENQLM
- Environmental factors in security, 11-4
- ERRFMT process, 10-3
- ERRLOG.SYS
  - See Error log file
- Error log file
  - maintaining, 10-4
  - overview, 10-3
  - printing, 10-5
- Error Log Utility, 10-3
- Error Log Utility, overview, 10-3
- Error message
  - during network operations, 6-36
- Error report, 10-4
- Error statistics
  - displaying with NCP commands, 6-32
- Ethernet
  - address of satellite nodes, 7-5
  - configurator, as network monitoring tool, 6-31
  - devices, 6-26
- Evasive action
  - duration, 11-14
  - invoked as counteraction for break-in, 11-12
- Event (network)
  - class, 6-34
  - message format, 6-35
  - type, 6-34
- Event logging
  - DECnet-VAX, 6-31, 6-33
  - disabling, 6-36
  - enabling, 6-25
  - network, 6-6
- Execution queue
  - definition, 5-1
- Expiration
  - of password, 11-5
- F**
- Failure, login
  - See Login failure

**FIELD** account  
 modifying after system installation,  
 4-5  
 user authorization file entry, 4-4

**File**  
 backing up  
   See Backup  
 open during backup, 8-3  
 restoring  
   See also Restoring files  
   individual file, 8-19

**File browser**, 11-26

**File log option**

VMSINSTAL.COM, 3-7, 3-9

**File protection violation**

auditing, 11-26

**FILLM**

for software installation, 3-2  
 value for efficient backups, 8-11

**Full backup**

See Backup, image

## G

**General identifier**, 11-16, 11-17  
 reasons for using, 11-22

**Generic queue**  
 definition, 5-1

**GET** save set option

VMSINSTAL.COM, 3-8

**GRANT/IDENTIFIER** command, 11-23

**Group**

overlapping user, 11-16

## H

**Hardware problem**

during system startup, 2-23

**Hexadecimal UIC identifier**, 11-17

**High-water marking**

disabling to improve system  
 performance, 9-9

## I

**Identifier**

alias node, 6-4  
 general, 11-16, 11-17  
 system-defined, 11-17, 11-18  
 types, 11-16

**Identifier ACE**, 11-19

example of, 11-20  
 specifying access in, 11-21  
 specifying identifiers in, 11-19  
 specifying options with, 11-20

**Image backup**

See Backup, image

**INBOUND** parameter

for node type specification, 6-21

**Incremental backup**

See Backup, incremental

**Input specifier**

for BACKUP command, 8-4

**INSTALL** command, 2-7

**INTERACTIVE** identifier, 11-17

## J

**JBCSYSQUE.DAT**

See Queue, manager

See Queue file

## K

**Known image**

assigning attribute to, 9-6

installing, 2-7

reasons for installing, 9-6

site-specific startup, 2-7

**KNOWN** reserved word

plural form of component name, 6-26,  
 6-32

## Index-8

### L

#### LAT (Local Area Transport)

See also Terminal servers

definition, 2-10

in site-specific startup file, 2-10

supporting user terminals with, 2-10

#### LAVC

See Local area clusters

#### Levels of security

defined, 11-2

#### LGI parameters, 11-12

See also System Generation Utility  
(SYSGEN)

#### LIBDECOMP.COM command procedure, 9-9

#### License

DECnet-VAX, 6-4, 6-5

#### Limits

for DEFAULT account, 4-7

#### Line

displaying counter information, 6-31

#### Link

See also Logical link, Data link  
terminating dynamic asynchronous,  
6-23

#### Local area cluster

adding satellite nodes, 7-5

boot server, 7-2

components, 7-2

establishing a boot server, 7-5

installing the VMS operating system  
on, 7-3

satellite nodes, 7-2

#### Local Area Transport

See LAT

#### Local circuit

defining at network startup, 6-25

#### LOCAL identifier, 11-17

#### Local node, 6-5

defining at network startup, 6-25

displaying counter, 6-31

#### Logging console

default, 6-34

#### Logging file

of network events, 6-34

#### Logging sink

definition, 6-34

#### Logical link

troubleshooting problems, 6-40

#### Logical name

assigning systemwide, 2-19

#### Login

alternate command procedure, 2-20

controlling number of dialup attempts,  
11-11

#### failure

counting for break-in detection,  
11-13

flags, 11-9

logging in to a new system, 2-2

#### remote

and system password, 11-6

system login command procedure,  
2-12

sample, 2-13

type as system identifier, 11-17

#### LTLOAD.COM command procedure, 2-10

### M

#### Magnetic tape

assigning volume labels to, 8-22

automatic unloading by Backup, 8-22

characters allowed in volume labels,  
8-22

initializing (for backup), 8-5

tape label processing by Backup, 8-22

#### Manual network configuration, 6-6

#### Manual switching of terminal line, 6-22

#### Menu

for system management tasks, 2-3

#### Message

See also Error message

network-related error (explanations),  
6-36

operator log file, 10-6

operator reply, 10-9

user request, 10-9

#### MGRMENU.COM procedure, 2-3

#### Mixed-interconnect cluster

definition, 7-2

#### Modem, 6-16, 6-20

**Modem (cont'd.)**

- autodial, 6-22
- null cable, 6-15

**MODIFY/SYSTEM\_PASSWORD**

- command, 11-7

**MONITOR**

- See Monitor Utility

**MONITOR.COM** command procedure, 9-3**Monitoring a network**

- See Network monitoring tools

**Monitor Utility (MONITOR)**, 9-2

- creating a summary file, 9-3
- creating clusterwide summary reports, 9-3

- starting as a detached process, 9-3

**MONSUM.COM** command procedure, 9-3**MOUNT** command, 2-5**Mounting volumes**

- and security audit, 11-25

**MOUNTMSG/DISMOUMSG** parameters

- effect on product installation, 3-2

**N****NCP (Network Control Program)**

- as a network monitoring tool, 6-30
- counters, 6-33
- display types, 6-31
- plural forms of component names, 6-26
- tailoring the configuration database, 6-25
- using to control proxy login, 6-28
- using to define nodes, 6-13
- using to display network information, 6-31

**NCP commands**

- ALL parameter with SET, 6-26
- CLEAR, 6-6, 6-26
- DEFINE, 6-6, 6-26
- DEFINE LOGGING, 6-34
- DEFINE NODE, 6-13
- effect of invalid parameter value, 6-36
- enabling logging, 6-34

**NCP commands (cont'd.)**

- LIST, 6-26, 6-31
- LIST NODE, 6-27
- PURGE, 6-6, 6-26
- PURGE LOGGING, 6-36
- PURGE NODE, 6-27
- SET, 6-6, 6-26
- SET KNOWN NODES, 6-13
- SET LOGGING, 6-34
- SET OBJECT, 6-29
- SHOW, 6-26, 6-31
- SHOW COUNTER, 6-32
- SHOW LOGGING, 6-35
- SHOW NODE, 6-27
- ZERO COUNTERS, 6-33

**NETCONFIG.COM** command procedure

- automatic establishment of logging, 6-34

- network configuration, 6-5, 6-6

- purpose, 6-5

- sample dialogue, 6-10

- to establish default nonprivileged DECnet account and directory, 6-30

**NETCONFIG\_UPDATE.COM** command procedure, 6-5**NETPROXY.DAT**

- See Network user authorization file

**NETSERVER.LOG** file, 6-36

- as troubleshooting aid, 6-41, 6-42

**Network**

- access control string, 11-8

**component**

- displaying information, 6-32
- name, 6-32

- component name, 6-26

- configuration, 6-5

- counters, 6-31

- resetting to zero, 6-33

- database, 6-4, 6-13, 6-14, 6-21

- definition, 6-1

- deleting nodes, 6-27

- determining configuration changes, 6-31

- displaying information about, 6-31

- displaying logging activity, 6-35

- displaying nodes, 6-27

## Index-10

### Network (cont'd.)

- error message explanations, 6-36
  - establishing volatile network database, 6-26
  - event logging, 6-6
    - disabling, 6-36
  - INBOUND parameter, 6-21
  - modifying access methods, 6-5
  - monitoring tools, 6-30
  - object
    - See Object, network
  - object MAIL and proxy access, 6-29
  - on a workstation using DECwindows, 6-24
  - operations in a cluster, 7-6
  - preparing system for joining, 6-3
  - purging nodes, 6-27
  - restarting, 6-25
  - security, 6-18
  - shutting down, 6-25
  - site-specific startup, 2-8
  - starting automatically from VMS
    - system boot, 6-25
  - starting manually, 6-25
  - startup command procedure
    - STARTNET.COM, 6-25
  - startup values, 6-31
- Network Control Program
- See NCP
- NETWORK identifier, 11-17
- Network manager
- assigning node names, 6-14
- Network operator
- designated by OPCOM, 6-34
  - enabling terminal as, 6-34
- Network user authorization file
- definition, 4-1
  - normal protection, 11-11
  - permanent proxy database, 6-28
  - setting protection for, 4-2
- Node
- See also Boot server
  - access control, 6-29
  - address, 6-8
  - configuring for DECnet-VAX, 6-5
  - database, 6-4, 6-13

### Node (cont'd.)

- definition, 6-2
  - determining status, 6-31
  - executor, 6-5
  - local, 6-5, 6-25
  - reconfiguration, 6-5, 6-6
  - remote, 6-13
  - type, 6-21
- Node address, 6-4, 6-8
- Node database
- permanent, 6-13
  - volatile, 6-13
- Node name
- VAXcluster alias, 6-4
- Node number, 6-8
- Nonprivileged DECnet-VAX default account, 6-30
- Null modem cable, 6-15

## O

### Object, network

- defining at network startup, 6-25
  - MAIL, 6-29
  - modifying proxy access, 6-29
  - PHONE, 6-38
- OPCCRASH program, 2-23
- OPCOM (Operator Communication Manager), 10-6
- defining network operator, 6-34
  - event message format, 6-35
  - restarting, 10-11, 10-12
- OPCOM message
- during product installation, 3-2
- Open file quota
- See FILLM
- Operator
- terminal
    - enabling and disabling, 10-8
- OPERATOR.LOG
- See Operator log file
- Operator Communication Facility
- See OPCOM
- Operator Communication Manager
- See OPCOM



Operator console  
 as OPCOM terminal, 6-34

Operator log file, 10-6  
 device status message, 10-7  
 example, 10-6  
 initialization message, 10-6  
 maintaining, 10-10  
 message, 10-6  
 printing, 10-11  
 purging, 2-9  
 recording changes to system  
 parameters, 10-10  
 security alarm messages, 10-10

OPTIONS keyword  
 in VMSINSTAL.COM, 3-5

Orderly system shutdown  
 See System shutdown

Output specifier  
 for BACKUP command, 8-4

Ownership  
 managing defaults, 11-24

## P

Packets  
 monitoring network events for lost,  
 6-33

PAK (Product authorization key)  
 DECnet-VAX, 6-4  
 definition, 6-4  
 DVNETEND, 6-5  
 DVNETRTG, 6-5  
 registering DECnet-VAX, 6-10

Password  
 See also Password generator  
 See also Password protection  
 changing, 11-9  
 dual, 11-5  
 expiration  
 how to preexpire, 11-5  
 how to set, 11-8  
 forced change, 11-9  
 forgotten by user, 4-10  
 initial, 11-5  
 length of, 11-9

Password (cont'd.)  
 locked, 11-10  
 management, 11-11  
 primary, 11-5  
 secondary, 11-7

Password generator  
 using to obtain initial password, 11-5  
 when to require, 11-10

Password protection, 11-10  
 avoiding detection, 11-14

Path  
 lost connection, 6-37

Penetration  
 as security problem, 11-2

Performance  
 and automatic password generator,  
 11-10

Performance improvements  
 decompressing system libraries, 9-9  
 disabling high-water marking, 9-9  
 installing frequently used images, 9-9  
 LIBDECOMP.COM command  
 procedure, 9-9  
 reducing system disk I/O, 9-10  
 relinking images, 9-9  
 setting RMS file extend parameters,  
 9-9

Performance management  
 definition, 9-1

Permanent database  
 network, 6-6, 6-13, 6-26  
 proxy, 6-28

PGFLQUO  
 value for efficient backups, 8-11

Phone Utility (PHONE)  
 network operations, 6-38  
 object, 6-38

Physical security, 11-4

Ports  
 publicly accessible, 11-7  
 terminal, 6-22

Printer  
 setting characteristics, 2-6

Print job, 5-1  
 changing priority of, 5-8  
 deleting, 5-5  
 holding, 5-6

## Index-12

### Print job (cont'd.)

- monitoring, 5-4
- releasing, 5-6
- requeueing, 5-6, 5-7
- retaining, 5-6

### Print queue

- See also Queue
- definition, 2-6

### Prober

- how to catch, 11-12, 11-26

### Probing

- as security problem, 11-2

### Procedure

- See Command procedure

### Process quotas

- recommended values for backups,  
8-12

### Process rights list, 11-18

### Product authorization key

- See PAK

### Product list

- VMSINSTAL.COM parameter syntax,  
3-4

### Project account, 11-23

### Protection

- of remote files, 6-27

### Protocol

- autodial, 6-22

### Proxy

- database, 6-28

### Proxy account, 6-28

- definition, 6-3

### Proxy parameters

- for NCP commands, 6-28

### Public volume

- mounting, 2-5

### PURGE LOGGING command, 6-36

## Q

### Queue

- changing attributes of, 5-3
- changing priority of job in, 5-8
- command  
SET QUEUE, 5-3

### Queue

#### command (cont'd.)

- SHOW ENTRY, 5-4
- SHOW QUEUE, 5-3
- START/QUEUE, 5-3
- STOP/QUEUE, 5-3

#### commands

- DELETE/ENTRY, 5-5

#### definition, 5-1

#### deleting a job from, 5-5

#### execution, 5-1

#### generic, 5-1

#### in a VAXcluster environment, 7-3

#### initializing, 2-6, 5-2

#### manager

- restarting, 5-3

- starting, 5-2

- stopping, 5-3

#### print and batch queues, definition, 2-6

#### queue file, 5-2

#### resuming execution of, 5-3

#### retaining a job in, 5-6

#### showing status of, 5-3

#### starting, 5-2

### Queue file

- definition, 5-2

### Queue manager

- See Queue, manager

## R

### Receive password

- in network operations, 6-18

### Reconfiguring node, 6-6

### Release notes option

- VMSINSTAL.COM, 3-9

### REMOTE identifier, 11-17

### Remote login

- and system password, 11-6

### Remote node

- address, 6-4

- copying database, 6-13

- displaying counter information, 6-31

- name of, 6-4

- REPLY/ENABLE=NETWORK command
    - enabling network operator terminal, 6-34
  - REPLY/ENABLE command, 10-8
  - REPLY/LOG command, 10-6
  - REQUEST command, 10-9
  - Resource attribute, 11-23
  - Restarting
    - DECnet-VAX, 6-25, 6-27
  - Restoring files
    - complete directory structure, 8-18
    - for an entire disk, 8-14
    - from an image backup, 8-14
    - from an incremental backup, 8-16
    - individual files, 8-19
    - overview, 8-13
  - Retries
    - controlling number for dialups, 11-11
  - Rights database file, 11-16
    - definition, 4-1
    - setting protection for, 4-2
  - Rights list
    - See Rights database file
  - RIGHTSLIST.DAT
    - See Rights database file
  - Router, 6-4, 6-9
  - Routing
    - problems, 6-40
  - Routing path
    - tracing, 6-40
- S**
- Satellite node, 7-2
    - adding, 7-5
    - obtaining Ethernet address of, 7-5
  - Save set
    - definition, 8-3
    - listing contents of, 8-19
  - Secondary password, 11-7
  - Security
    - alarm messages, 10-10
    - at network circuit level, 6-29
    - at network node level, 6-29
    - at network system level, 6-29
  - Security (cont'd.)
    - auditing, 11-25
    - for dynamic asynchronous connection, 6-21
    - for static asynchronous connection, 6-18
    - levels of, 11-3
    - passwords, 11-5
    - security alarm, 11-24
  - Security alarm application, 11-24
  - Security archive audit log file, 11-29
  - Security auditing, 11-25
  - Security operator
    - See operator
  - SECURITY privilege, 11-6
  - SET ACL command, 11-16
  - SET AUDIT command
    - suggested auditing applications, 11-25
  - SET DIRECTORY/ACL command
    - example, 11-23
  - SET HOST/DTE command
    - using over the network, 6-22
  - SET HOST command
    - and network security, 6-27
  - SET KNOWN NODES command, 6-13
  - SET LOGINS/INTERACTIVE command, 9-5
  - SET PASSWORD/GENERATE command, 11-10
  - SET PASSWORD/SYSTEM/GENERATE command, 11-6
  - SET PASSWORD/SYSTEM command, 11-6
  - SET PRINTER command, 2-6
  - SET PROTECTION command
    - and network file security, 6-27
  - SET QUEUE command, 5-3
  - SET TERMINAL/SYSPWD command, 11-6
  - SET TERMINAL command, 2-6, 6-16
    - using over the network, 6-20
  - Shared resource
    - definition, 7-3
  - SHOW ACL command, 11-16
  - SHOW ENTRY command, 5-4
  - SHOW INTRUSION command, 11-15

## Index-14

- SHOW QUEUE command, 5-3
- Shutdown
  - See System shutdown
- SHUTDOWN\$INFORM\_NODES logical name, 2-28
- SHUTDOWN.COM command procedure, 2-23
  - shutting down the system with, 2-24
- Site-specific startup, 2-4
  - announcements, 2-11
  - command procedure, 2-4
  - installing known images, 2-7
  - setting up queues, 2-6
  - setting up spooled devices, 2-6
- Software installation
  - preparing for, 3-1
- Software problem
  - during system startup, 2-23
- Source parameter
  - for VMSINSTAL.COM, 3-5
- Spooled device
  - definition, 2-7
- Standalone Backup, 8-20
- START/QUEUE command, 5-3
- STARTNET.COM command procedure, 6-10
- STARTUP.COM command procedure, 2-15
- Startup command procedure
  - site-specific, 2-4
- Static asynchronous connection
  - installing, 6-15
  - local intermittent, 6-19
  - procedure for establishing, 6-15
  - reasons for failure, 6-42
  - receive password, 6-18
  - security, 6-18
  - switching of terminal line, 6-19
  - transmit password, 6-18
  - turning back on, 6-19
  - turning on and off line and circuit, 6-19
- Statistics
  - network performance and error, 6-32
- STOP/QUEUE command, 5-3
- SUBMON.COM command procedure, 9-3
- Switching of terminal line
  - automatic, 6-22
  - manual, 6-22
- SYCONFIG.COM command procedure, 2-16
- SYLOGICALS.COM command procedure, 2-16
  - defining logical names with, 2-19
- SYLOGIN.COM command procedure, 2-12, 2-16
  - sample, 2-13
- SYPAGSWPFILES.COM command procedure, 2-16
- SYS\$ANNOUNCE
  - defining, 2-11
- SYS\$STARTUP
  - definition, 2-17
- SYS\$WELCOME
  - defining, 2-11
- SYSDUMP.DMP
  - See System dump file
- SYSGEN
  - See System Generation Utility
- SYSHUTDWN.COM command procedure, 2-23
- SYSMAN
  - See System Management Utility
- SYSTARTUP\_V5.COM
  - initializing a queue with, 5-2
- SYSTARTUP\_V5.COM command procedure, 2-4, 2-16
  - defining SYS\$ANNOUNCE and SYS\$WELCOME, 2-11
  - initializing and starting queues, 2-6
  - installing known images, 2-7
  - limiting interactive users, 2-10
  - mounting public disks with, 2-5
  - purging the operator's log file, 2-9
  - setting device characteristics with, 2-6
  - starting DECnet, 2-8
  - starting LAT with, 2-10
  - starting System Dump Analyzer, 2-8
  - submitting batch jobs, 2-9
  - systemwide announcements, 2-11

**System**

- files, moving to improve performance, 9-10
- libraries, decompressing, 9-9
- non-VMS, asynchronous connection to a VMS system, 6-15
- SYSTEM** account
  - modifying after system installation, 4-5
  - required limits for software installation, 3-2
  - setting process quotas for efficient backups, 8-10
  - user authorization file entry, 4-4
- System-defined identifier, 11-17, 11-18
- System disk
  - backing up for software installations, 3-2
  - procedure for building and copying, 2-14
- System Dump Analyzer (SDA)
  - site-specific startup, 2-8
- System dump file, 10-2
- System failure
  - system dump analyzer, 2-8
  - VMSINSTAL.COM response to, 3-10
- System Generation Utility (SYSGEN)
  - LGI\_BRK\_DISUSER parameter, 11-14
  - LGI\_BRK\_LIM parameter, 11-12
  - LGI\_BRK\_TERM parameter, 11-13
  - LGI\_BRK\_TMO parameter, 11-13
  - LGI\_HID\_TIM parameter, 11-14
  - LGI\_RETRY\_LIM parameter, 11-12
  - LGI\_RETRY\_TMO parameter, 11-12
  - operator log messages, 10-10
- System login command procedure, 2-12
  - sample, 2-13
- System Management Utility (SYSMAN)
  - privileges needed to use, 7-6
  - setting a clusterwide environment, 7-6
  - using in a VAXcluster environment, 7-6
  - using on remote nodes in a cluster, 7-7
- System password, 11-6, 11-7
  - disadvantages, 11-7
  - guidelines, 11-6
  - minimum length requirement, 11-10
  - recommended change frequency, 11-9
  - where stored, 11-7
- System process
  - OPCOM, 10-6
- System shutdown
  - after software installation, 3-6
  - DECnet-VAX, 6-25
  - emergency procedure, 2-29
  - emergency shutdown, 2-23
  - normal procedure, 2-24
  - normal procedure (example), 2-27
  - notification, 2-28
  - procedures for, 2-23
  - SHUTDOWN\$INFORM\_NODES
    - logical name, 2-28
  - site-specific procedure, 2-23
  - using OPCCRASH, 2-23, 2-29
  - using SHUTDOWN.COM, 2-24
- System startup
  - bypassing startup and login procedures, 2-22
  - emergency, after changing system parameters, 2-22
  - emergency procedures, 2-20
  - hardware problems during, 2-23
  - procedure, 2-15
  - software problems during, 2-23
- SYSTEST account
  - modifying after system installation, 4-5
  - user authorization file entry, 4-4
- SYSUAF.DAT
  - See UAF (user authorization file)

**T**

- Tampering with system file
  - how to detect, 11-26
- Tape label processing (Backup), 8-22
  - disabling, 8-22
- Telephone line
  - dialup, 6-15

## Index-16

### TELL prefix

for NCP command SHOW, 6-32

### Template files, 2-19

### Terminal

automatic line switching, 6-22

controlling access through system

password, 11-6

manual line switching, 6-22

operator, 10-8

port, 6-22

setting characteristics, 2-6

site-specific startup, 2-6

virtual, 6-20

### Terminal emulator, 6-21

### Terminal line

asynchronous DECnet, 6-15

automatic switching of, 6-22

manual switching of, 6-22

### Terminal servers, 11-6

considerations for break-in detection,  
11-13

### Terminating

dynamic asynchronous link, 6-23

### Timeouts

count of network, 6-32

### Tracing routing path

with NCP command prefix TELL,  
6-40

### Traffic

count of user data, 6-32

### Transmit password

in network operations, 6-18

### TTY\_DEFCHAR2 parameter, 11-6

### Tuning

definition, 9-7

evaluating success of, 9-8

predicting when required, 9-8

## U

### UAF (user authorization file)

adding a user account, 4-6

bypassing, 2-20

checking quotas for software  
installation, 3-3

### UAF (user authorization file) (cont'd.)

creating a captive account, 4-7

DEFAULT account, 4-3

default values for new users, 4-3

deleting a user record, 4-11

disabling a user account, 4-13

FIELD account, 4-4

listing records in, 4-10

maintenance, 4-5

modifying

after system installation, 4-5

and security audit, 11-25

user record, 4-10

normal protection, 11-11

overview, 4-1

proxy account, 6-28

setting protection for, 4-2

SYSTEM account, 4-4

SYSTEST account, 4-4

### UIC identifier, 11-16, 11-17

### User account

adding a new account, 4-6

changing quotas or privileges, 4-10

default values for, 4-3

deleting, 4-11

disabling, 4-13

listing records of, 4-10

modifying, 4-10

### User authorization file

See UAF

### User irresponsibility

as security problem, 11-1

### User name

as identifier, 11-17

### User penetration

as security problem, 11-2

### User probing

as security problem, 11-2

### Users

limiting number of interactive, 2-10

## V

### VAXcluster

See Clusters

## Verification

- of user identity, 11-7

- Virtual terminal, 6-20

- VMS\$LAYERED.DAT file, 2-18

- VMS\$VMS.DAT file, 2-17

- VMSINSTAL.COM command procedure

- alternate system root option, 3-10

- restriction, 3-10

- answer file, 3-7

- command line syntax, 3-4

- destination for installing product, 3-6

- file log option, 3-7, 3-9

- GET save set option, 3-8

- using to store product save set, 3-8

- getting help in, 3-3

- manual recovery from system failure, 3-10

- options

- file log, 3-7

- GET save set, 3-8

- keyword, 3-5

- list of, 3-5, 3-7

- release notes, 3-9

- overview, 3-1

- parameters

- product list, 3-4

- source, 3-5

- preparing to use, 3-1

- product save-set format, 3-8

- starting, 3-3

- system failure

- conditions, 3-10

- recovery, 3-10

- system shutdown following, 3-6

- VMSKITBLD command procedure, 2-14

- VMS operating system

- asynchronous connection to non-VMS

- system, 6-15

- installing as a new installation, 2-1

- installing as an upgrade, 2-1

- VMSPHASES.DAT file, 2-17

- Volatile database

- network, 6-6, 6-13, 6-14, 6-21, 6-26

- Volume header record

- on magnetic tape, 8-22

## Volume label

- assigning to magnetic tape, 8-22

- definition, 8-5

## W

- Wildcard character

- in DECnet event types, 6-34

- Work load

- importance of knowing, 9-2

- managing, 9-4

- WSEXTENT

- value for efficient backups, 8-11

- WSQUOTA

- value for efficient backups, 8-11





# 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

<b>Your Location</b>	<b>Call</b>	<b>Contact</b>
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 <sup>1</sup>	_____	SDC Order Processing - WMO/E15 <i>or</i> Software Distribution Center Digital Equipment Corporation Westminster, Massachusetts 01473

---

<sup>1</sup>For internal orders, you must submit an Internal Software Order Form (EN-01740-07).



# Reader's Comments

VMS System  
Manager's Manual  
AA-LA00B-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.

<b>I rate this manual's:</b>	<b>Excellent</b>	<b>Good</b>	<b>Fair</b>	<b>Poor</b>
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

**igital**™



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 Publications—Spit Brook  
ZK01-3/J35  
110 SPIT BROOK ROAD  
NASHUA, NH 03062-9987



Do Not Tear - Fold Here

Cut Along Dotted Line