

---

# HP OpenVMS System Management Utilities Reference Manual: A–L

Order Number: AA–PV5PJ–TK

**January 2005**

This document describes reference information for System Management utilities used with the OpenVMS Alpha and I64 operating systems.

**Revision/Update Information:** This manual supersedes the *HP OpenVMS System Management Utilities Reference Manual: A–L*, OpenVMS Version 7.3-2.

**Software Version:** OpenVMS I64 Version 8.2  
OpenVMS Alpha Version 8.2

**Hewlett-Packard Company**  
**Palo Alto, California**

---

© Copyright 2005 Hewlett-Packard Development Company, L.P.

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Printed in the US

ZK6258

The HP OpenVMS documentation set is available on CD-ROM.

This document was prepared using DECdocument, Version 3.3-1b.

---

# Contents

<b>Preface</b> .....	xv
<b>1 Access Control List Editor</b>	
1.1 ACL Editor Description .....	1-3
1.2 ACL Editor Usage Summary .....	1-3
1.3 ACE Formats .....	1-5
Alarm ACE .....	1-6
Audit ACE .....	1-8
Creator ACE .....	1-10
Default Protection ACE .....	1-12
Identifier ACE .....	1-14
Subsystem ACE .....	1-16
1.4 ACL Editor Qualifiers .....	1-18
/CLASS .....	1-19
/JOURNAL .....	1-20
/MODE .....	1-21
/OBJECT_TYPE .....	1-22
/RECOVER .....	1-23
<b>2 Accounting Utility</b>	
2.1 ACCOUNTING Description .....	2-3
2.2 ACCOUNTING Usage Summary .....	2-3
2.3 ACCOUNTING Qualifiers .....	2-3
/ACCOUNT .....	2-5
/ADDRESS .....	2-6
/BEFORE .....	2-7
/BINARY .....	2-8
/BRIEF .....	2-9
/ENTRY .....	2-11
/FULL .....	2-12
/IDENT .....	2-14
/IMAGE .....	2-15
/JOB .....	2-16
/LOG .....	2-17
/NODE .....	2-18
/OUTPUT (Alpha and I64) .....	2-19
/OWNER .....	2-20
/PRIORITY .....	2-21

/PROCESS	2-22
/QUEUE	2-23
/REJECTED	2-24
/REMOTE_ID	2-25
/REPORT	2-26
/SINCE	2-28
/SORT	2-29
/STATUS	2-32
/SUMMARY	2-33
/TERMINAL	2-35
/TITLE	2-36
/TYPE	2-37
/UIC	2-39
/USER	2-40
/WIDE	2-41

### 3 Analyze/Disk\_Structure Utility

3.1	ANALYZE/DISK_STRUCTURE Description	3-3
3.1.1	Disk Error Reporting and Repair	3-4
3.1.2	Detecting Shadow Set Errors	3-6
3.2	ANALYZE/DISK_STRUCTURE Usage Summary	3-6
3.3	ANALYZE/DISK_STRUCTURE Qualifiers	3-7
	/CONFIRM	3-9
	/HOMEBLOCKS	3-10
	/LIST	3-11
	/LOCK_VOLUME (Alpha and I64)	3-12
	/OUTPUT	3-13
	/READ_CHECK	3-14
	/RECORD_ATTRIBUTES	3-15
	/REPAIR	3-16
	/SHADOW	3-17
	/STATISTICS	3-19
	/USAGE[=filespec]	3-21

### 4 Audit Analysis Utility

4.1	ANALYZE/AUDIT Description	4-3
4.2	ANALYZE/AUDIT Usage Summary	4-3
4.3	ANALYZE/AUDIT Qualifiers	4-4
	/BEFORE	4-5
	/BINARY	4-6
	/BRIEF	4-7
	/EVENT_TYPE	4-8
	/FULL	4-10
	/IGNORE	4-11
	/INTERACTIVE	4-12
	/OUTPUT	4-13
	/PAUSE	4-14

	/SELECT .....	4-15
	/SINCE .....	4-25
	/SUMMARY .....	4-26
4.4	ANALYZE/AUDIT Commands .....	4-28
	CONTINUE .....	4-29
	DISPLAY .....	4-30
	EXIT .....	4-32
	HELP .....	4-33
	LIST .....	4-34
	NEXT FILE .....	4-35
	NEXT RECORD .....	4-36
	POSITION .....	4-37
	SHOW .....	4-38

## 5 Authorize Utility

5.1	AUTHORIZE Description .....	5-3
5.2	AUTHORIZE Usage Summary .....	5-4
5.3	AUTHORIZE Commands .....	5-5
	ADD .....	5-7
	ADD/IDENTIFIER .....	5-20
	ADD/PROXY .....	5-22
	COPY .....	5-24
	CREATE/PROXY .....	5-36
	CREATE/RIGHTS .....	5-37
	DEFAULT .....	5-38
	EXIT .....	5-50
	GRANT/IDENTIFIER .....	5-51
	HELP .....	5-53
	LIST .....	5-55
	LIST/IDENTIFIER .....	5-57
	LIST/PROXY .....	5-59
	LIST/RIGHTS .....	5-60
	MODIFY .....	5-61
	MODIFY/IDENTIFIER .....	5-73
	MODIFY/PROXY .....	5-75
	MODIFY/SYSTEM_PASSWORD .....	5-77
	REMOVE .....	5-78
	REMOVE/IDENTIFIER .....	5-79
	REMOVE/PROXY .....	5-80
	RENAME .....	5-81
	RENAME/IDENTIFIER .....	5-84
	REVOKE/IDENTIFIER .....	5-85
	SHOW .....	5-86
	SHOW/IDENTIFIER .....	5-91
	SHOW/PROXY .....	5-93
	SHOW/RIGHTS .....	5-95

## 6 AUTOGEN Command Procedure

6.1	AUTOGEN Description .....	6-3
6.1.1	NEWPARAMS.DAT .....	6-3
6.1.1.1	How NEWPARAMS.DAT Works .....	6-4
6.1.1.2	What Goes into NEWPARAMS.DAT .....	6-5
6.1.1.2.1	Product Name .....	6-5
6.1.1.2.2	Parameter Assignment .....	6-5
6.1.1.2.3	How to Remove Assignments from CLU\$PARAMS.DAT .....	6-6
6.1.1.3	What CLU\$PARAMS.DAT Looks Like .....	6-6
6.2	AUTOGEN Usage Summary .....	6-7
6.3	Feedback .....	6-7
6.4	Phases .....	6-8
6.4.1	SAVPARAMS .....	6-8
6.4.2	GETDATA .....	6-9
6.4.3	GENPARAMS .....	6-9
6.4.4	TESTFILES .....	6-9
6.4.5	GENFILES .....	6-10
6.4.6	SETPARAMS .....	6-10
6.4.7	SHUTDOWN .....	6-10
6.4.8	REBOOT .....	6-10
6.4.9	HELP .....	6-11
6.5	Execution Modes .....	6-11
6.6	Files Used by AUTOGEN .....	6-11
6.7	AUTOGEN Usage Summary .....	6-12

## 7 Backup Utility

7.1	BACKUP Description .....	7-3
7.2	BACKUP Command Line Format .....	7-5
7.3	BACKUP Input and Output Specifiers .....	7-5
7.3.1	Input and Output Specifier Element Lists .....	7-6
7.3.2	BACKUP Qualifiers .....	7-7
7.3.3	Using Wildcard Characters with BACKUP .....	7-8
7.4	BACKUP Usage Summary .....	7-10
7.5	BACKUP Qualifiers .....	7-11
	/ALIAS .....	7-15
	/ASSIST .....	7-16
	/BACKUP .....	7-17
	/BEFORE .....	7-18
	/BLOCK_SIZE .....	7-19
	/BRIEF .....	7-20
	/BUFFER_COUNT .....	7-21
	/BY_OWNER .....	7-22
	/BY_OWNER .....	7-23
	/BY_OWNER .....	7-24
	/COMMENT .....	7-25
	/COMPARE .....	7-26
	/CONFIRM .....	7-28
	/CONVERT .....	7-29
	/CRC .....	7-30
	/CRC .....	7-31

/CREATED .....	7-32
/DELETE .....	7-33
/DENSITY .....	7-34
/EXACT_ORDER .....	7-36
/EXCLUDE .....	7-39
/EXPIRED .....	7-40
/FAST .....	7-41
/FILES_SELECTED .....	7-42
/FULL .....	7-43
/GROUP_SIZE .....	7-45
/HEADER_ONLY .....	7-46
/IGNORE=option .....	7-48
/IMAGE .....	7-51
/INCREMENTAL .....	7-53
/INITIALIZE .....	7-55
/INPUT_FILES .....	7-57
/INTERCHANGE .....	7-58
/JOURNAL .....	7-59
/LABEL .....	7-62
/LIST .....	7-64
/LOG .....	7-66
/MEDIA_FORMAT=[NO]COMPACTION .....	7-67
/MODIFIED .....	7-68
/NEW_VERSION .....	7-69
/NOINCREMENTAL .....	7-70
/OVERLAY .....	7-71
/OWNER_UIC .....	7-72
/PHYSICAL .....	7-73
/PROTECTION .....	7-74
/RECORD .....	7-76
/RELEASE_TAPE .....	7-77
/REPLACE .....	7-78
/REWIND .....	7-79
/REWIND .....	7-80
/SAVE_SET .....	7-83
/SAVE_SET .....	7-84
/SELECT .....	7-85
/SINCE .....	7-86
/TAPE_EXPIRATION .....	7-88
/TRUNCATE .....	7-89
/VERIFY .....	7-90
/VOLUME .....	7-91
7.6 BACKUP Examples .....	7-92

## 8 Crash Log Utility Extractor (CLUE) (VAX Only)

8.1	CLUE Description (VAX Only) . . . . .	8-3
8.2	CLUE Usage Summary (VAX Only) . . . . .	8-3
8.3	CLUE Commands (VAX Only) . . . . .	8-3
	DELETE (VAX Only) . . . . .	8-5
	DIRECTORY (VAX Only) . . . . .	8-6
	EXIT (VAX Only) . . . . .	8-8
	EXTRACT (VAX Only) . . . . .	8-9
	HELP (VAX Only) . . . . .	8-10
	SHOW (VAX Only) . . . . .	8-11

## 9 EFI Utilities for OpenVMS

9.1	EFI Utilities Description . . . . .	9-3
	VMS_BCFG . . . . .	9-4
	VMS_SET . . . . .	9-6
	VMS_SHOW . . . . .	9-7

## 10 Error Log Viewer Utility (ELV)

10.1	ELV Description . . . . .	10-3
10.2	ELV Usage Summary . . . . .	10-4
10.3	Understanding Categories of Events . . . . .	10-5
10.4	ELV Commands . . . . .	10-5
	CONVERT . . . . .	10-6
	DUMP . . . . .	10-11
	EXIT . . . . .	10-16
	HELP . . . . .	10-17
	TRANSLATE . . . . .	10-18
	WRITE . . . . .	10-25
10.5	ELV Sample Reports . . . . .	10-30
10.5.1	/ONE_LINE Example . . . . .	10-30
10.5.2	/BRIEF Example . . . . .	10-31
10.5.3	No Qualifiers Example . . . . .	10-31
10.5.4	/FULL Example . . . . .	10-32
10.5.5	/TERSE Example . . . . .	10-32

## 11 Install Utility

11.1	INSTALL Description . . . . .	11-3
11.2	INSTALL Usage Summary . . . . .	11-3
11.3	INSTALL Qualifiers . . . . .	11-3
	/TRANSLATE . . . . .	11-4
11.4	INSTALL Commands . . . . .	11-5
	ADD . . . . .	11-6
	CREATE . . . . .	11-7
	DELETE . . . . .	11-12
	EXIT . . . . .	11-13
	HELP . . . . .	11-14
	LIST . . . . .	11-15



PURGE .....	11-21
REMOVE .....	11-22
REPLACE .....	11-23

## 12 LAN Control Program (LANCP) Utility

12.1	LANCP Description .....	12-3
12.2	LANCP Usage Summary .....	12-3
12.3	LANCP Commands .....	12-4
	@ (Execute Procedure) .....	12-6
	CLEAR DEVICE .....	12-8
	CLEAR DLL .....	12-9
	CLEAR NODE .....	12-10
	CONNECT NODE .....	12-11
	CONVERT DEVICE_DATABASE .....	12-13
	CONVERT NODE_DATABASE .....	12-14
	DEFINE DEVICE .....	12-15
	DEFINE NODE .....	12-16
	EXIT .....	12-18
	HELP .....	12-19
	LIST DEVICE .....	12-20
	LIST NODE .....	12-22
	PURGE DEVICE .....	12-24
	PURGE NODE .....	12-25
	SET ACP .....	12-26
	SET DEVICE .....	12-27
	SET NODE .....	12-37
	SHOW ACP .....	12-41
	SHOW CONFIGURATION .....	12-42
	SHOW DEVICE .....	12-43
	SHOW DLL or MOPDLL .....	12-47
	SHOW LOG .....	12-48
	SHOW NODE .....	12-50
	SPAWN .....	12-52
	TRIGGER NODE .....	12-53
	UPDATE DEVICE .....	12-54

## 13 LAT Control Program (LATCP) Utility

13.1	LATCP Description .....	13-3
13.2	LATCP Usage Summary .....	13-3
13.3	LATCP Commands .....	13-4
	ATTACH .....	13-5
	CREATE LINK .....	13-7
	CREATE PORT .....	13-10
	CREATE SERVICE .....	13-15
	DEFINE/KEY .....	13-19
	DELETE LINK .....	13-22

DELETE PORT .....	13-23
DELETE QUEUE_ENTRY .....	13-24
DELETE SERVICE .....	13-25
EXIT .....	13-26
HELP .....	13-27
RECALL .....	13-28
REFRESH .....	13-29
SCROLL .....	13-30
SET LINK .....	13-31
SET NODE .....	13-33
SET PORT .....	13-43
SET SERVICE .....	13-47
SHOW LINK .....	13-50
SHOW NODE .....	13-55
SHOW PORT .....	13-61
SHOW QUEUE_ENTRY .....	13-64
SHOW SERVICE .....	13-66
SPAWN .....	13-69
ZERO COUNTERS .....	13-70

## 14 Log Manager Control Program (LMCP) Utility

14.1	LMCP Description .....	14-3
14.2	LMCP Usage Summary .....	14-3
14.3	LMCP Commands .....	14-3
	CLOSE LOG .....	14-4
	CONVERT LOG .....	14-5
	CREATE LOG .....	14-7
	DUMP .....	14-9
	EXIT .....	14-14
	HELP .....	14-15
	REPAIR .....	14-16
	SHOW LOG .....	14-19

## A ACL Editor Keypad Editing Commands

A.1	ACL Editor Keypad Commands .....	A-1
A.2	Additional ACL Editing Keys and Key Sequences .....	A-4
A.3	ACL Editing Keys on the Supplemental Keypad (LK201-Series Keyboards) .....	A-5

## B Customizing the ACL Editor

B.1	Modifying Variables in the ACL Section File .....	B-1
B.2	Using the ACL Editor CALL_USER Routine .....	B-3

## C Accounting Information for Programmers

C.1	Format of an Accounting File Record . . . . .	C-1
C.1.1	Types of Accounting Record . . . . .	C-3
C.1.2	Format of an Information Packet . . . . .	C-4
C.1.2.1	General Format . . . . .	C-4
C.1.2.2	File Name Packet (ACR\$K_FILENAME) . . . . .	C-6
C.1.2.3	Identification Packet (ACR\$K_ID) . . . . .	C-6
C.1.2.4	Image Name Packet (ACR\$K_IMAGENAME) . . . . .	C-8
C.1.2.5	Print Resource Packet (ACR\$K_PRINT) . . . . .	C-9
C.1.2.6	Resource Packet (ACR\$K_RESOURCE) . . . . .	C-10
C.1.2.7	User Data Packet (ACR\$K_USER_DATA) . . . . .	C-11

## D ANALYZE/DISK\_STRUCTURE—Stage Checks

D.1	Stage 1 . . . . .	D-1
D.2	Stage 2 . . . . .	D-1
D.3	Stage 3 . . . . .	D-2
D.4	Stage 4 . . . . .	D-2
D.5	Stage 5 . . . . .	D-3
D.6	Stage 6 . . . . .	D-3
D.7	Stage 7 . . . . .	D-3
D.8	Stage 8 . . . . .	D-4
D.9	Annotated Example . . . . .	D-4

## E ANALYZE/DISK\_STRUCTURE—Usage File

## F Security Audit Message Format

F.1	Audit Header Packet . . . . .	F-2
F.2	Audit Data Packets . . . . .	F-6

## G Valid Combinations of BACKUP Qualifiers

## Index

## Examples

14-1	Sample Transaction Log . . . . .	14-10
------	----------------------------------	-------

## Figures

A-1	Keypad for an LK201-Series Keyboard . . . . .	A-2
C-1	Format of an Accounting Record . . . . .	C-2
C-2	Format of an Information Packet . . . . .	C-5
C-3	Format of a File Name Packet . . . . .	C-6
C-4	Format of an Identification Packet . . . . .	C-7
C-5	Format of an Image Name Packet . . . . .	C-8
C-6	Format of a Print Resource Packet . . . . .	C-9
C-7	Format of a Resource Packet . . . . .	C-10
C-8	Format of a User Data Packet . . . . .	C-12

F-1	Format of a Security Audit Message . . . . .	F-1
F-2	Audit Header Packet Format . . . . .	F-2
F-3	Audit Data Packet Format . . . . .	F-7
G-1	Command Qualifiers Used in Save Operations . . . . .	G-2
G-2	Input File-Selection Qualifiers Used in Save Operations . . . . .	G-3
G-3	Output Save-Set Qualifiers Used in Save Operations . . . . .	G-4
G-4	Command Qualifiers Used in Restore Operations . . . . .	G-5
G-5	Input Save-Set Qualifiers Used in Restore Operations . . . . .	G-6
G-6	Output File Qualifiers Used in Restore Operations . . . . .	G-6
G-7	Command Qualifiers Used in Copy Operations . . . . .	G-7
G-8	Input File-Selection Qualifiers Used in Copy Operations . . . . .	G-8
G-9	Output File Qualifiers Used in Copy Operations . . . . .	G-9
G-10	Command Qualifiers Used in Compare Operations . . . . .	G-10
G-11	Input File-Selection Qualifiers Used in Compare Operations . . . . .	G-11
G-12	Input Save-Set Qualifiers Used in Compare Operations . . . . .	G-11

## Tables

1	Manual Structure . . . . .	xv
5-1	Screen Control Keys . . . . .	5-87
6-1	AUTOGEN Phases . . . . .	6-8
6-2	AUTOGEN Execution Modes . . . . .	6-11
6-3	Files Used by AUTOGEN . . . . .	6-11
7-1	BACKUP Input and Output by Operation Type . . . . .	7-6
7-2	BACKUP Qualifier Summary . . . . .	7-11
7-3	Save Operation Quick Reference . . . . .	7-92
7-4	Restore Operation Quick Reference . . . . .	7-93
7-5	Copy Operation Quick Reference . . . . .	7-95
7-6	Compare Operation Quick Reference . . . . .	7-95
7-7	List Operation Quick Reference . . . . .	7-95
10-1	Report Formats . . . . .	10-19
12-1	Default Functional Address Mapping for Token Ring Devices . . . . .	12-33
A-1	ACL Editor Keypad Commands . . . . .	A-2
A-2	Additional ACL Editing Keys and Key Sequences . . . . .	A-4
A-3	ACL Editing Keys on the Supplemental Keypad . . . . .	A-5
B-1	ACL Section File Variables . . . . .	B-1
B-2	CALL_USER Function Codes . . . . .	B-3
C-1	Summary of Accounting System Services . . . . .	C-1
C-2	Fields in an Accounting Record Header . . . . .	C-2
C-3	ACR\$W_TYPE Fields in an Accounting Record Header . . . . .	C-2
C-4	Types of Accounting Record . . . . .	C-3
C-5	Fields in an Information Packet Header . . . . .	C-5
C-6	ACR\$W_TYPE Fields in an Information Packet Header . . . . .	C-5
C-7	Data Fields in a File Name Packet . . . . .	C-6
C-8	Data Fields in an Identification Packet . . . . .	C-7
C-9	Data Field in an Image Name Packet . . . . .	C-9
C-10	Data Fields in a Print Resource Packet . . . . .	C-10

C-11	Data Fields in a Resource Packet . . . . .	C-11
C-12	Data Field in a User Data Packet . . . . .	C-12
D-1	Stage 3 Maps . . . . .	D-2
E-1	Identification Record Format (Length USG\$K_IDENT_LEN) . . . . .	E-1
E-2	File Record Format (Length USG\$K_FILE_LEN) . . . . .	E-2
F-1	Description of the Audit Header Fields . . . . .	F-2
F-2	Description of Audit Event Types and Subtypes . . . . .	F-3
F-3	Description of Audit Event Flags . . . . .	F-6
F-4	Description of the Audit Data Packet . . . . .	F-7
F-5	Types of Data in Audit Packets . . . . .	F-7



---

# Preface

The *HP OpenVMS System Management Utilities Reference Manual: A–L* contains reference information about the utilities that are used to manage the OpenVMS Alpha and I64 operating systems. This manual describes each system management utility and provides examples for frequently used commands and qualifiers. In addition to system management utilities, a description and usage summary of the AUTOGEN command procedure is presented in this reference manual.

Refer to the *HP OpenVMS System Management Utilities Reference Manual: M–Z* for information on the system management utilities not discussed in this book.

All commands follow the standard rules of grammar as specified in the *HP OpenVMS DCL Dictionary*.

For information on how to use these system management utilities and AUTOGEN, please refer to the *HP OpenVMS System Manager's Manual*.

## Intended Audience

This manual is intended for system managers and users of the system management utilities for the OpenVMS Alpha and I64 operating systems.

## Document Structure

This manual has 14 parts, arranged alphabetically. Each part, except the section on the AUTOGEN command procedure, contains reference information for a system management utility. Table 1 shows the structure.

**Table 1 Manual Structure**

Part	Utility
1	Access Control List Editor (ACL editor)
2	Accounting (ACCOUNTING)
3	Analyze/Disk_Structure (ANALYZE/DISK_STRUCTURE)
4	Audit Analysis (ANALYZE/AUDIT)
5	Authorize (AUTHORIZE)
6	AUTOGEN Command Procedure
7	Backup (BACKUP)
8	Crash Log Utility Extractor (CLUE)
9	Error Log Viewer (ELV)

(continued on next page)

**Table 1 (Cont.) Manual Structure**

Part	Utility
10	Install (INSTALL)
11	LAN Control Program (LANCP)
12	LAT Control Program (LATCP)
13	Log Manager Control Program (LMCP)

## Related Documents

For more information on the system management utilities, refer to the following documents:

- *HP OpenVMS System Management Utilities Reference Manual: M-Z*
- *DEC Text Processing Utility Reference Manual*
- *HP OpenVMS Guide to System Security*
- *HP OpenVMS DCL Dictionary*
- *HP OpenVMS System Manager's Manual*
- *HP OpenVMS Programming Concepts Manual*
- *OpenVMS Record Management Services Reference Manual*
- *HP OpenVMS System Services Reference Manual*
- *OpenVMS User's Manual*
- *OpenVMS VAX Device Support Manual* (archived)
- *OpenVMS VAX System Dump Analyzer Utility Manual*
- *HP Volume Shadowing for OpenVMS*

For additional information about HP OpenVMS products and services, visit the following World Wide Web address:

<http://www.hp.com/go/openvms>

## Reader's Comments

HP welcomes your comments on this manual. Please send comments to either of the following addresses:

Internet	<b>openvmsdoc@hp.com</b>
Postal Mail	Hewlett-Packard Company OSSG Documentation Group, ZKO3-4/U08 110 Spit Brook Rd. Nashua, NH 03062-2698

## How To Order Additional Documentation

For information about how to order additional documentation, visit the following World Wide Web address:

<http://www.hp.com/go/openvms/doc/order>



## Conventions

VMScluster systems are now referred to as OpenVMS Cluster systems. Unless otherwise specified, references to OpenVMS Cluster systems or clusters in this document are synonymous with VMScluster systems.

The contents of the display examples for some utility commands described in this manual may differ slightly from the actual output provided by these commands on your system. However, when the behavior of a command differs significantly between OpenVMS VAX, Alpha, and I64, that behavior is described in text and rendered, as appropriate, in separate examples.

In this manual, every use of DECwindows and DECwindows Motif refers to DECwindows Motif for OpenVMS software.

The following conventions are also used in this manual:

Ctrl/ <i>x</i>	A sequence such as Ctrl/ <i>x</i> indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
PF1 <i>x</i>	A sequence such as PF1 <i>x</i> indicates that you must first press and release the key labeled PF1 and then press and release another key or a pointing device button.
<span style="border: 1px solid black; padding: 2px;">Return</span>	<p>In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)</p> <p>In the HTML version of this document, this convention appears as brackets, rather than a box.</p>
...	<p>A horizontal ellipsis in examples indicates one of the following possibilities:</p> <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 command format descriptions, parentheses indicate that you must enclose the options in parentheses if you choose more than one.
[ ]	In command format descriptions, brackets indicate optional choices. You can choose one or more items or no items. Do not type the brackets on the command line. However, you must include the brackets in the syntax for OpenVMS directory specifications and for a substring specification in an assignment statement.
[   ]	In command format descriptions, vertical bars separate choices within brackets or braces. Within brackets, the choices are options; within braces, at least one choice is required. Do not type the vertical bars on the command line.

{ }	In command format descriptions, braces indicate required choices; you must choose at least one of the items listed. Do not type the braces on the command line.
<b>bold text</b>	This typeface represents the introduction of a new term. It also represents the name of an argument, an attribute, or a reason.
<i>italic text</i>	Italic text indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error <i>number</i> ), in command lines ( <i>PRODUCER=name</i> ), and in command parameters in text (where <i>dd</i> represents the predefined code for the device type).
UPPERCASE TEXT	Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
Monospace type	Monospace type indicates code examples and interactive screen displays.  In the C programming language, monospace type in text identifies the following elements: keywords, the names of independently compiled external functions and files, syntax summaries, and references to variables or identifiers introduced in an example.
-	A hyphen at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line.
numbers	All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radices—binary, octal, or hexadecimal—are explicitly indicated.

---

**Access Control List Editor**



## 1.1 ACL Editor Description

The access control list editor (ACL editor) is a screen-oriented editor used to create and maintain access control lists (ACLs). An ACL is a collection of access control entries (ACEs) that grant or deny access for specific users or groups of users of an object. (For a description of the entry and display format for ACEs, see Section 1.3.) ACLs enable you to control access more closely than you can by using the default user identification code (UIC) based protection.

The system does not limit the number of ACEs that an ACL can contain or the number of characters in an ACE. However, long ACLs increase the amount of time necessary to gain access to an object. In practice, memory constraints can limit the size of an ACL.

The order of ACEs in an ACL is important. ACEs granting or denying access to an object for specific users must appear before ACEs identifying broader classes of users. For example, to grant user SMITH read access to a system object and to deny all other interactive users all types of access to the object, place the ACE for user SMITH before the ACE identifying all interactive users on the system.

You can place ACLs on the following object classes:

- Capability
- Common event flag cluster
- Device
- File
- Group global section
- Logical name table
- Queue
- Resource domain
- Security class
- System global section
- Volume

## 1.2 ACL Editor Usage Summary

The access control list editor (ACL editor) creates or modifies an access control list (ACL) for a specified object.

### Format

EDIT/ACL object-spec

#### Parameter

##### **object-spec**

Specifies the object whose access control list is to be created or edited. If an access control list does not exist, it is created.

You can specify an object from any of the following object classes:

- Capability
- Common event flag cluster
- Device
- File
- Group global section
- Logical name table
- Queue
- Resource domain

## ACL

### 1.2 ACL Editor Usage Summary

Security class  
System global section  
Volume

The default object class is a file. A file must be a disk file on a Files-11 On-Disk Structure Level 2 or 5 formatted volume. For any object other than a file, you must specify the object class with the /CLASS qualifier.

Note that the ACL editor does not provide a default file type for files. To prevent the ACL editor from using a null file type, specify the file type on the command line. If the object is a directory, specify the .DIR file type.

Do not include wildcard characters in the object specification.

### Description

You can invoke the ACL editor to create or modify an ACL for an object that you own, have control access to, or can gain access to by a privilege such as BYPASS, GRPPRV, or SYSPRV. To invoke the ACL editor, enter the DCL command EDIT/ACL. In the command line, specify the name of the object whose ACL you want to edit. For example, to create an ACL for the file INVENTORY.DAT, enter the following command:

```
$ EDIT/ACL INVENTORY.DAT
```

You can use either the EDIT/ACL command or the SET SECURITY/EDIT command to invoke the ACL editor. For more information about the SET SECURITY command, refer to the *HP OpenVMS DCL Dictionary* and the *HP OpenVMS Guide to System Security*.

By default, the ACL editor creates and modifies ACLs for files. To create an ACL for an object other than a file (for example, to create an ACL for a queue), you must specify the object class when you invoke the ACL editor. For example, the following command invokes the ACL editor to create an ACL for the disk DAPR:

```
$ EDIT/ACL/CLASS=DEVICE DAPR
```

If an ACL for the object you specify already exists, the ACL editor displays the ACL. You can then use keypad editing commands to add, replace, or delete one or more ACEs in the ACL (see Section A.1). To exit from a completed editing session, press Ctrl/Z. To end an editing session without incorporating any of your edits, press the GOLD key (PF1) and then press Ctrl/Z.

For a description of keypad editing commands supplied by the ACL editor, see Appendix A. For information about how to modify the ACL editor by modifying ACL section files, see Appendix B.

---

#### Note

---

In addition to invoking the ACL editor directly or by entering commands at the DCL prompt (\$), you can modify an ACL by using the callable interface to the ACL editor (the ACLEDIT\$EDIT routine). For information about how to use the ACLEDIT\$EDIT routine, refer to the *OpenVMS Utility Routines Manual*.

---

## 1.3 ACE Formats

This section describes the entry and display format for the following access control entries (ACEs):

- Alarm ACE for security auditing of an object
- Audit ACE for security auditing of an object
- Creator ACE to set the ownership access for new files created in a directory
- Default Protection ACE to set a default protection code through a directory structure
- Identifier ACE for object access control
- Subsystem ACE for protected subsystem access control

The *HP OpenVMS Guide to System Security* describes how to use each of these ACEs. You can also use other types of ACEs. For example, applications can use an Application ACE to store application-specific information associated with a file. For a description of the internal format used to store an ACE, refer to the *HP OpenVMS Programming Concepts Manual*.

## **Alarm ACE**

Specifies the access criteria that cause an alarm message to be sent to all security operator terminals.

ACL alarms are enabled by default; however, alarms are not written to the system security audit log file. If you have existing files or resources protected by Alarm ACEs and you want messages to be recorded in the log file, replace the Alarm ACEs with Audit ACEs.

## **Format**

(ALARM=SECURITY [,OPTIONS=attributes], ACCESS=access-type[+access-type...])

## **Parameters**

### **options**

Specify any of the following attributes:

Default	Indicates that an ACE is to be included in the ACL of any files created within a directory. When the entry is propagated, the Default attribute is removed from the ACE of the created file. This attribute is valid for directory files only.
Hidden	Indicates that this ACE should be changed only by the application that adds it. Although the Hidden attribute is valid for any ACE type, its intended use is to hide Application ACEs. To delete or modify a hidden ACE, you must use the SET SECURITY command.  Users need the SECURITY privilege to display a hidden ACE with the DCL commands SHOW SECURITY or DIRECTORY/SECURITY. SECURITY privilege is also required to modify or delete a hidden ACE with the DCL command SET SECURITY. The ACL editor displays the ACE only to show its relative position within the ACL, not to facilitate editing of the ACE. To create a hidden ACE, an application can invoke the \$SET_SECURITY system service.



- Protected Protects the ACE against casual deletion. Protected ACEs can be deleted only in the following ways:
- By using the ACL editor
  - By specifying the ACE explicitly when deleting it  
Use the command SET SECURITY/ACL=(ace)/DELETE to specify and delete an ACE.
  - By deleting all ACEs, both protected and unprotected  
Use the command SET SECURITY/ACL/DELETE=ALL to delete all ACEs.

The following commands do not delete protected ACEs:

```
SET SECURITY/ACL/DELETE
SET SECURITY/LIKE
SET SECURITY/DEFAULT
```

- Nopropagate Indicates that the ACE cannot be copied by operations that usually propagate ACEs. For example, the ACE cannot be copied by the SET SECURITY/LIKE or SET SECURITY/DEFAULT commands.

- None Indicates that no attributes apply to an entry. Although you can create an ACL entry with OPTIONS=None, the attribute is not displayed. Whenever you specify additional attributes with the None attribute, the other attributes take precedence. The None attribute is equivalent to omitting the field.

#### **access**

Specify any access that is valid for the object class. refer to the *HP OpenVMS Guide to System Security* for a listing of valid access types. For an Alarm ACE to have any effect, you must include the keywords SUCCESS, FAILURE, or both with the access types. For example, if the auditing criterion is a failure to obtain write access to an object, specify the following Alarm ACE:

```
(ALARM=SECURITY, ACCESS=WRITE+FAILURE)
```

## **Audit ACE**

Specifies the access criteria that cause an audit message to be written to the system security audit log file. A message is recorded by default. A message is recorded only if ACL audits are enabled with the DCL command SET AUDIT/AUDIT/ENABLE=ACL.

### **Format**

(AUDIT=SECURITY [,OPTIONS=attributes], ACCESS=access-type[+access-type...])

### **Parameters**

#### **options**

Specify one of the following attributes:

- |           |   |
|-----------|---|
| Default   | Indicates that an ACE is to be included in the ACL of any files created within a directory. When the entry is propagated, the Default attribute is removed from the ACE of the created file. This attribute is valid for directory files only.  |
| Hidden    | Indicates that this ACE should be changed only by the application that adds it. Although the Hidden attribute is valid for any ACE type, its intended use is to hide Application ACEs. To delete or modify a hidden ACE, you must use the SET SECURITY command.<br><br>Users need the SECURITY privilege to display a hidden ACE with the DCL commands SHOW SECURITY or DIRECTORY/SECURITY. SECURITY privilege is also required to modify or delete a hidden ACE with the DCL command SET SECURITY. The ACL editor displays the ACE only to show its relative position within the ACL, not to facilitate editing of the ACE. To create a hidden ACE, an application can invoke the \$SET_SECURITY system service. |
| Protected | Protects the ACE against casual deletion. Protected ACEs can be deleted only in the following ways: <ul style="list-style-type: none"><li>• By using the ACL editor</li><li>• By specifying the ACE explicitly when deleting it<br/>Use the command SET SECURITY/ACL=(ace)/DELETE to specify and delete an ACE.</li><li>• By deleting all ACEs, both protected and unprotected<br/>Use the command SET SECURITY/ACL/DELETE=ALL to delete all ACEs.</li></ul>  |

The following commands do not delete protected ACEs:

```
SET SECURITY/ACL/DELETE
SET SECURITY/LIKE
SET SECURITY/DEFAULT
```

Nopropagate	Indicates that the ACE cannot be copied by operations that usually propagate ACEs. For example, the ACE cannot be copied by the SET SECURITY/LIKE or SET SECURITY/DEFAULT commands.
None	Indicates that no attributes apply to an entry. Although you can create an ACL entry with OPTIONS=None, the attribute is not displayed. Whenever you specify additional attributes with the None attribute, the other attributes take precedence. The None attribute is equivalent to omitting the field.

**access**

Specify any access that is valid for the object class. Refer to the *HP OpenVMS Guide to System Security* for a listing of valid access types. For an Audit ACE to have any effect, you must include the keywords SUCCESS, FAILURE, or both with the access types. For example, if the auditing criterion is a failure to obtain write access to an object, specify the following Audit ACE:

```
(AUDIT=SECURITY,ACCESS=WRITE+FAILURE)
```

## ACL Creator ACE

---

### Creator ACE

Adds an extra ACE to the ACL for a file created within the directory to which you assign the Creator ACE. The Creator ACE applies only when the following conditions exist:

- The file being created is not owned by the user identification code (UIC) of the process creating the file.
- The process creating the file does not have system privileges.

For example, both of these conditions exist when a process holding a general identifier with the Resource attribute creates a file in a directory owned by that identifier. In this situation, the system adds an extra ACE at the top of the new file's ACL. If a Creator ACE exists in the ACL for the parent directory, the system propagates the access specified in the Creator ACE to the new ACE. If a directory lacks a Creator ACE, the system assigns an extra ACE with a combination of control access and ownership access. A Creator ACE with ACCESS=None suppresses the addition of the extra ACE.

The Creator ACE applies to directory files only.

Refer to the *HP OpenVMS Guide to System Security* for more information.

### Format

```
(CREATOR  
[,OPTIONS=attribute[+attribute...]],ACCESS=access-type[+access-type...])
```

### Parameters

#### options

Specify any of the following attributes:

Protected	Protects the ACE against casual deletion. Protected ACEs can be deleted only in the following ways: <ul style="list-style-type: none"><li>• By using the ACL editor</li><li>• By specifying the ACE explicitly when deleting it Use the command SET SECURITY/ACL=(ace)/DELETE to specify and delete an ACE.</li><li>• By deleting all ACEs, both protected and unprotected Use the command SET SECURITY/ACL/DELETE=ALL to delete all ACEs.</li></ul>
-----------	--

The following commands do not delete protected ACEs:

```
SET SECURITY/ACL/DELETE  
SET SECURITY/LIKE  
SET SECURITY/DEFAULT
```

Nopropagate	Indicates that the ACE cannot be copied by operations that usually propagate ACEs. For example, the ACE cannot be copied by the SET SECURITY/LIKE or SET SECURITY/DEFAULT commands.
-------------	---

None

Indicates that no attributes apply to an entry. Although you can create an ACL entry with `OPTIONS=None`, the attribute is not displayed. Whenever you specify additional attributes with the None attribute, the other attributes take precedence. The None attribute is equivalent to omitting the field.

**access**

Specify access types that are valid for files (read, write, execute, delete, and control).

## ACL

### Default Protection ACE

---

#### Default Protection ACE

Defines a UIC-based protection to be propagated to new files throughout a directory tree. The protection code in the ACE is assigned to new files created in the directory. The Default Protection ACE applies to directory files only. Although the system propagates the Default Protection ACE to new subdirectories, the protection code is not assigned to the subdirectories. Instead, the subdirectories receive a modified copy of the parent directory's protection code in which delete access is not granted.

An example of a Default Protection ACE is as follows:

```
(DEFAULT_PROTECTION,S:RWED,O:RWED,G,W)
```

The ACE grants read, write, execute, and delete access to users in the system (S) and owner (O) categories but no access to users in the group and world categories. For more information, refer to the *HP OpenVMS Guide to System Security*.

#### Format

```
(DEFAULT_PROTECTION[,OPTIONS=attribute[+attribute...]],access)
```

#### Parameters

##### options

Specify any of the following attributes:

Hidden	Indicates that this ACE should be changed only by the application that adds it. Although the Hidden attribute is valid for any ACE type, its intended use is to hide Application ACEs. To delete or modify a hidden ACE, you must use the SET SECURITY command.  Users need the SECURITY privilege to display a hidden ACE with the DCL commands SHOW SECURITY or DIRECTORY/SECURITY. SECURITY privilege is also required to modify or delete a hidden ACE with the DCL command SET SECURITY. The ACL editor displays the ACE only to show its relative position within the ACL, not to facilitate editing of the ACE. To create a hidden ACE, an application can invoke the \$SET_SECURITY system service.
--------	---

Protected	<p>Protects the ACE against casual deletion. Protected ACEs can be deleted only in the following ways:</p> <ul style="list-style-type: none"><li>• By using the ACL editor</li><li>• By specifying the ACE explicitly when deleting it Use the command SET SECURITY/ACL=(ace)/DELETE to specify and delete an ACE.</li><li>• By deleting all ACEs, both protected and unprotected Use the command SET SECURITY/ACL/DELETE=ALL to delete all ACEs.</li></ul> <p>The following commands do not delete protected ACEs:</p> <pre>SET SECURITY/ACL/DELETE SET SECURITY/LIKE SET SECURITY/DEFAULT</pre>
Nopropagate	<p>Indicates that the ACE cannot be copied by operations that usually propagate ACEs. For example, the ACE cannot be copied by the SET SECURITY/LIKE or SET SECURITY/DEFAULT commands.</p>
None	<p>Indicates that no attributes apply to an entry. Although you can create an ACL entry with OPTIONS=None, the attribute is not displayed. Whenever you specify additional attributes with the None attribute, the other attributes take precedence. The None attribute is equivalent to omitting the field.</p>

**access**

Specify access in the format of a UIC-based protection code, which is as follows:

[category: list of access allowed (, category: list of access allowed,...)]

- User categories include system (S), owner (O), group (G), and world (W). Refer to the *HP OpenVMS Guide to System Security* for a definition of these categories. Access types for files include read (R), write (W), execute (E), and delete (D). The access type is assigned to each ownership category and is separated from its access types with a colon (:).
- A null access list means no access, so when you omit an access type for a user category, that category of user is denied that type of access. To deny all access to a user category, specify the user category without any access types. Omit the colon after the user category when you deny access to a category of users.
- When you omit a user category from a protection code, the current access allowed that category of user is set to no access.

## ACL Identifier ACE

---

### Identifier ACE

Controls the type of access allowed to a particular user or group of users. An example of an Identifier ACE is as follows:

```
(IDENTIFIER=SALES,ACCESS=READ+WRITE)
```

A system manager can use the Authorize utility (AUTHORIZE) to grant the SALES identifier to a specific group of users. Read and write access to the file INVENTORY.DAT is then granted to users who hold the SALES identifier.

For more information, refer to the *HP OpenVMS Guide to System Security*.

### Format

```
(IDENTIFIER=identifier[+identifier...] [,OPTIONS=attributes[+attributes...]]  
,ACCESS=access-type[+access-type...])
```

### Parameters

#### identifier

Specifies a user or groups of users whose access to an object is defined in the ACE. A system manager creates or removes identifiers and assigns users to hold these identifiers.

Types of identifiers are as follows:

UIC	Identifiers in alphanumeric format that are based on the user identification codes (UICs) and that uniquely identify each user on the system. Users with accounts on the system automatically receive a UIC identifier, for example, [GROUP1,JONES] or [JONES]. Thus, each UIC identifier specifies a particular user.
General	Identifiers defined by the security administrator in the rights list to identify groups of users on the system. A general identifier is an alphanumeric string of 1 to 31 characters, containing at least one alphabetic character. It can include the letters A to Z, dollar signs (\$), underscores (_), and the numbers 0 to 9, for example, 92SALES\$, ACCOUNT_3, or PUBLISHING.
Environmental	Identifiers describing different types of users based on their initial entry into the system. Environmental identifiers are also called system-defined identifiers. Environmental identifiers correspond directly to the login classes described in the <i>HP OpenVMS Guide to System Security</i> . They include batch, network, interactive, local, dialup, and remote.

For more information, refer to the *HP OpenVMS Guide to System Security*.

#### options

Specify any of the following attributes:



Default	<p>Indicates that an ACE is to be included in the ACL of any files created within a directory. When the entry is propagated, the Default attribute is removed from the ACE of the created file. This attribute is valid for directory files only.</p> <p>Note that an Identifier ACE with the Default attribute has no effect on access.</p>
Hidden	<p>Indicates that this ACE should be changed only by the application that adds it. Although the Hidden attribute is valid for any ACE type, its intended use is to hide Application ACEs. To delete or modify a hidden ACE, you must use the SET SECURITY command.</p> <p>Users need the SECURITY privilege to display a hidden ACE with the DCL commands SHOW SECURITY or DIRECTORY/SECURITY. SECURITY privilege is also required to modify or delete a hidden ACE with the DCL command SET SECURITY. The ACL editor displays the ACE only to show its relative position within the ACL, not to facilitate editing of the ACE. To create a hidden ACE, an application can invoke the \$SET_SECURITY system service.</p>
Protected	<p>Protects the ACE against casual deletion. Protected ACEs can be deleted only in the following ways:</p> <ul style="list-style-type: none"> <li>• By using the ACL editor</li> <li>• By specifying the ACE explicitly when deleting it Use the command SET SECURITY/ACL=(ace)/DELETE to specify and delete an ACE.</li> <li>• By deleting all ACEs, both protected and unprotected Use the command SET SECURITY/ACL/DELETE=ALL to delete all ACEs.</li> </ul> <p>The following commands do not delete protected ACEs:</p> <pre style="margin-left: 40px;">SET SECURITY/ACL/DELETE SET SECURITY/LIKE SET SECURITY/DEFAULT</pre>
Nopropagate	<p>Indicates that the ACE cannot be copied by operations that usually propagate ACEs. For example, the ACE cannot be copied by the SET SECURITY/LIKE or SET SECURITY/DEFAULT commands.</p>
None	<p>Indicates that no attributes apply to an entry. Although you can create an ACL entry with OPTIONS=None, the attribute is not displayed. Whenever you specify additional attributes with the None attribute, the other attributes take precedence. The None attribute is equivalent to omitting the field.</p>

**access**

Specify access types that are valid for the object class. Refer to the *HP OpenVMS Guide to System Security* for a listing of valid access types.

# ACL

## Subsystem ACE

---

### Subsystem ACE

Grants additional identifiers to a process while it is running the image to which the Subsystem ACE applies. Users with execute access to the image can access objects that are in the protected subsystem, such as data files and printers, but only when they run the subsystem images. The Subsystem ACE applies to executable images only.

An example of a Subsystem ACE is as follows:

```
(SUBSYSTEM, IDENTIFIER=ACCOUNTING)
```

### Format

```
(SUBSYSTEM,[OPTIONS=attribute[+attribute...],]IDENTIFIER=identifier  
[,ATTRIBUTES=attribute[+attribute...]] [,IDENTIFIER=identifier  
[,ATTRIBUTES=attribute[+attribute...],...])
```

### Parameters

#### options

Specify any of the following attributes:

Protected	Protects the ACE against casual deletion. Protected ACEs can be deleted only in the following ways: <ul style="list-style-type: none"><li>• By using the ACL editor</li><li>• By specifying the ACE explicitly when deleting it Use the command SET SECURITY/ACL=(ace)/DELETE to specify and delete an ACE.</li><li>• By deleting all ACEs, both protected and unprotected Use the command SET SECURITY/ACL/DELETE=ALL to delete all ACEs.</li></ul>
-----------	--

The following commands do not delete protected ACEs:

```
SET SECURITY/ACL/DELETE  
SET SECURITY/LIKE  
SET SECURITY/DEFAULT
```

Nopropagate	Indicates that the ACE cannot be copied by operations that usually propagate ACEs. For example, the ACE cannot be copied by the SET SECURITY/LIKE or SET SECURITY/DEFAULT commands.
None	Indicates that no attributes apply to an entry. Although you can create an ACL entry with OPTIONS=None, the attribute is not displayed. Whenever you specify additional attributes with the None attribute, the other attributes take precedence. The None attribute is equivalent to omitting the field.

#### identifier

A general identifier specifying the users or groups of users who are allowed or denied access to an object. It is an alphanumeric string of 1 through 31 characters, containing at least one alphabetic character. It can include the letters

A to Z, dollar signs (\$), underscores (\_), and the numbers 0 to 9. For more information, refer to the *HP OpenVMS Guide to System Security*.

A Subsystem ACE can have multiple pairs of identifiers, with special attributes assigned to the identifiers. A subsystem might require several identifiers to work properly. For example:

```
(SUBSYSTEM, IDENTIFIER=MAIL_SUBSYSTEM, ATTRIBUTE=NONE, IDENTIFIER=BLDG5, ATTRIBUTE=NONE)
```

**attribute**

The identifier characteristics you specify when you add identifiers to the rights list or grant identifiers to users. You can specify the following attribute:

Resource	Allows holders of the identifier to charge disk space to the identifier. Used only for file objects.
----------	--

## Access Control List Editor

### 1.4 ACL Editor Qualifiers

#### 1.4 ACL Editor Qualifiers

When you invoke the ACL editor, you can include qualifiers on the command line that identify the object class and the editing mode (prompt or noprompt). You can also use qualifiers to name a journaling file or to recover an ACL editing session. This section describes the qualifiers listed in the following table:

Qualifier	Description
/CLASS	Specifies the class of object whose ACL is being edited
/JOURNAL	Controls whether a journal file is created for the editing session
/MODE	Specifies the use of prompting during the editing session
/OBJECT_TYPE	Superseded by the /CLASS qualifier
/RECOVER	Restores an ACL from a journal file at the beginning of an editing session

All of the qualifiers described in this section also apply to the SET SECURITY/EDIT command. You can substitute the SET SECURITY/EDIT command wherever the EDIT/ACL command is shown; the syntax is the same for both commands.

---

**/CLASS**

Specifies the class of the object whose ACL is being edited. Unless the object is a file, you must specify the object class.

**Format**

`/CLASS =object-class`

**Description**

To edit the ACL for an object other than a file, specify the object class with the `/CLASS` qualifier. Specify one of the following classes:

<code>CAPABILITY</code>	A system capability, such as the ability to process vector instructions. Currently, the only defined object name for the <code>CAPABILITY</code> class is <code>VECTOR</code> , which governs the ability of a subject to access a vector processor on the system. Note that you must supply the capability name as the object name parameter.
<code>COMMON_EVENT_CLUSTER</code>	A common event flag cluster.
<code>DEVICE</code>	A device, such as a disk or tape drive.
<code>FILE</code>	A file or a directory file. This is the default.
<code>GROUP_GLOBAL_SECTION</code>	A group global section.
<code>LOGICAL_NAME_TABLE</code>	A logical name table.
<code>QUEUE</code>	A batch queue or a device (printer, server, or terminal) queue.
<code>RESOURCE_DOMAIN</code>	A resource domain.
<code>SECURITY_CLASS</code>	A security class.
<code>SYSTEM_GLOBAL_SECTION</code>	A system global section.
<code>VOLUME</code>	A disk or tape volume.

**Examples**

1. `$ EDIT/ACL/CLASS=DEVICE WORK1`

The command in this example specifies that the object `WORK1` is a device.

2. `$ EDIT/ACL/CLASS=QUEUE FAST_BATCH`

The command in this example creates an ACL for the queue `FAST_BATCH`. Note that if you create an ACL for a generic queue, you must create identical ACLs for all execution queues to which jobs can be directed.

## /JOURNAL

---

### /JOURNAL

Controls whether a journal file is created for the editing session.

#### Format

```
/JOURNAL [=file-spec]
```

```
/NOJOURNAL
```

#### Description

By default, the ACL editor keeps a journal file containing a copy of modifications made during an editing session. The journal file is given the name of the object and a .TJL file type. If you specify a different name for the file, do not include any wildcard characters.

To prevent the ACL editor from creating a journal file, specify `/NOJOURNAL`.

If your editing session ends abnormally, you can recover the changes made during the aborted session by invoking the ACL editor with the `/RECOVER` qualifier.

#### Examples

1. `$ EDIT/ACL/JOURNAL=COMMONACL.SAV MECH1117.DAT`

With this command, you create a journal file named `COMMONACL.SAV`. The file contains a copy of the ACL and the editing commands used to create the ACL for the file `MECH1117.DAT`.

If the editing session is interrupted, you can recover your edits by specifying the name `COMMONACL.SAV` with the `/RECOVER` qualifier.

2. `$ EDIT/ACL/CLASS=RESOURCE/JOURNAL=ZERO_RESOURCE.TJL [0]`

If you edit an ACL for the resource domain `[0]`, the ACL editor attempts to create the file `[0].TJL` on the default device and fails. To create an ACL for the resource `[0]`, you must specify a different name for the journal file (as shown in this example) or suppress the creation of a journal file with the `/NOJOURNAL` qualifier.

---

**/MODE**

Specifies the use of prompting during the editing session.

**Format**

`/MODE =option`

**Description**

By default, the ACL editor prompts you for each ACE and provides values for some of the fields within an ACE (`/MODE=PROMPT`). To disable prompting, specify `/MODE=NOPROMPT` on the command line.

**Examples**

1. `$ EDIT/ACL/MODE=NOPROMPT WEATHER.TBL.DAT`

With this command, you initiate an ACL editing session to create an ACL for the file `WEATHER.TBL.DAT`. The `/MODE=NOPROMPT` qualifier specifies that no assistance is required in entering the ACL entries.

**/OBJECT\_TYPE**

---

**/OBJECT\_TYPE**

The /OBJECT\_TYPE qualifier is superseded by the /CLASS qualifier.



---

**/RECOVER**

Restores an ACL from a journal file at the beginning of an editing session.

**Format**

`/RECOVER [=file-spec]`

`/NORECOVER`

**Description**

The `/RECOVER` qualifier specifies that the ACL editor must restore the ACL from a journal file. The ACL editor restores the ACL to the state it was in when the last ACL editing session ended abnormally.

By default the journal file is given the name of the object and a `.TJL` file type. If you specify a more meaningful name for the journal file when you invoke the ACL editor (by using `/JOURNAL`), specify that file name with the `/RECOVER` qualifier.

**Example**

```
$ EDIT/ACL/JOURNAL=SAVEACL MYFILE.DAT

.
.
.
User creates ACL until system crashes
.
.
.
$ EDIT/ACL/JOURNAL=SAVEACL/RECOVER=SAVEACL MYFILE.DAT
.
.
.
ACL is restored and user proceeds with editing until done
.
.
.
^Z
$
```

The first command in this example starts the ACL editing session and specifies that the ACL editor must save the journal file `SAVEACL.TJL` if the session ends abnormally. The session proceeds until it is aborted by a system crash.

The next command restores the lost session with the journal file `SAVEACL.TJL`. To end the session, press `Ctrl/Z`. The ACL editor saves the edits and deletes the journal file.







## 2.1 ACCOUNTING Description

The Accounting utility (ACCOUNTING) produces reports of system resource use.

You can use ACCOUNTING qualifiers to:

- Produce a number of report formats
- Choose how the reports are organized
- Choose on which resources you want reports

You can use the reports to learn more about how the system is used and how it performs.

## 2.2 ACCOUNTING Usage Summary

Produces reports of resource use.

### Format

ACCOUNTING [filespec[,...]]

#### Parameter

##### filespec[,...]

Specifies the accounting files you want to process.

Each file specification can include the percent (%) and asterisk (\*) wildcard characters. If it does not include the device or directory, your current default device or directory is used. If it does not include the file name or file type, the values ACCOUNTNG and DAT are used respectively.

If you do not specify a file, the command processes the file SYS\$MANAGER:ACCOUNTNG.DAT.

### Description

Use this DCL command to run the Accounting utility:

```
$ ACCOUNTING [filespec[,...]]
```

You are returned to DCL level when the command has finished processing the specified accounting files.

By default, the command directs its output to the current SYS\$OUTPUT device. If you want to direct the output to a file, use the /OUTPUT qualifier.

Requires READ access to the accounting files you specify, and to the directories containing them.

## 2.3 ACCOUNTING Qualifiers

This section describes and provides examples of each ACCOUNTING qualifier. The following table summarizes the ACCOUNTING qualifiers:

Qualifier	Description
/ACCOUNT	Selects or rejects records for the specified account names
/ADDRESS	Selects or rejects records for DECnet for OpenVMS requests made by the specified nodes

## ACCOUNTING

### 2.3 ACCOUNTING Qualifiers

Qualifier	Description
/BEFORE	Selects all records time-stamped before the specified time
/BINARY	Copies the selected records to a new file in binary format
/BRIEF	Produces a brief report of the selected records
/ENTRY	Selects or rejects records for print and batch jobs with the specified queue entry numbers
/FULL	Produces a full report of the selected records
/IDENT	Selects or rejects records for the specified processes
/IMAGE	Selects or rejects records for the specified images
/JOB	Selects or rejects records for print and batch jobs with the specified job names
/LOG	Outputs informational messages
/NODE	Selects or rejects records for DECnet for OpenVMS requests made by the specified nodes
/OUTPUT	Specifies the output file (Alpha and I64)
/OWNER	Selects or rejects records for subprocesses created by the specified processes
/PRIORITY	Selects or rejects records for the specified priority
/PROCESS	Selects or rejects records for the specified types of process
/QUEUE	Selects or rejects records for print or batch jobs executed by the specified queues
/REJECTED	Copies the rejected records to a new file
/REMOTE_ID	Selects or rejects records for DECnet for OpenVMS requests made by the specified remote IDs
/REPORT	Specifies the resources that you want to summarize in a summary report
/SINCE	Selects all records time-stamped at or after the specified time
/SORT	Sorts the selected records
/STATUS	Selects or rejects records with the specified final exit status codes
/SUMMARY	Produces a summary report of the selected records
/TERMINAL	Selects or rejects records for interactive sessions at the specified terminals
/TITLE	Specifies the title shown on the first line of a summary report
/TYPE	Selects or rejects the specified types of record
/UIC	Selects or rejects records for the specified UICs
/USER	Selects or rejects records for the specified user names
/WIDE	Changes the width of Buffered I/O and Direct I/O fields in a report from 8 to 10 characters

---

## **/ACCOUNT**

Selects or rejects records for the specified account names.

### **Format**

`/ACCOUNT=[-]account[,...]`

### **Description**

The `/ACCOUNT` qualifier uses the value of the account field to select records for processing. This field is present in all records except file backward link and file forward link records.

The `/ACCOUNT` qualifier selects only records that have the specified values in the account field. If you precede the values with a minus sign, it selects all records *except* those with the specified values.

The following table shows the values stored in the account field of login failure and system initialization records:

<b>Value</b>	<b>Description</b>
<code>&lt;batch&gt;</code>	Batch job login failure
<code>&lt;det&gt;</code>	Detached process login failure
<code>&lt;login&gt;</code>	Interactive login failure
<code>&lt;net&gt;</code>	Network login failure
<code>&lt;start&gt;</code>	System startup

Note that when you specify these account field values as qualifier values, you must enclose them in quotes. Like all DCL commands, the `ACCOUNTING` command converts strings to uppercase unless they are enclosed in quotes.

### **Examples**

1. `$ ACCOUNTING /ACCOUNT=(SALES, QA)`

This example processes the file `SYS$MANAGER:ACCOUNTNG.DAT`. It produces a brief report of all records for the account names `SALES` and `QA`.

2. `$ ACCOUNTING /ACCOUNT=(-SALES, QA) /FULL`

This example processes the file `SYS$MANAGER:ACCOUNTNG.DAT`. It produces a full report of all records *except* for the account names `SALES` and `QA`.

## ACCOUNTING /ADDRESS

---

### /ADDRESS

Selects or rejects records for DECnet for OpenVMS requests made by the specified nodes.

### Format

```
/ADDRESS=[-]node_address[,...]
```

### Description

The /ADDRESS qualifier uses the value of the remote node address field to select records for processing. This field is present in all records except file backward link and file forward link records. For records that contain information about DECnet for OpenVMS requests, it contains the address of the node that made the request.

The /ADDRESS qualifier selects only records with the specified values in the remote node address field. If you precede the values with a minus sign, it selects all records *except* those with the specified values.

See also the /NODE and /REMOTE\_ID qualifiers, which select or reject records for DECnet for OpenVMS requests made by specified node names and remote IDs respectively.

### Example

```
$ ACCOUNTING /ADDRESS=19656
```

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records for DECnet for OpenVMS requests made by the node with the address 19656. (The decimal equivalent of this address is 19.200.)



## **/BEFORE**

Selects all records time-stamped before the specified time.

### **Format**

`/BEFORE[=time]`

### **Description**

All records in an accounting file are time-stamped with the time the record was logged in the file.

The `/BEFORE` qualifier selects only the records time-stamped before the specified time. You can specify an absolute time, a delta time, or a combination of the two. If you omit the time, 00:00 hours on the current day is used.

See also the `/SINCE` qualifier, which selects records time-stamped at or after a specified time.

### **Example**

```
$ ACCOUNTING /SINCE=1-NOV-2002 /BEFORE=1-DEC-2002
```

This example produces a brief report of all records time-stamped in the file `SYS$MANAGER:ACCOUNTNG.DAT` during November 2002.

## ACCOUNTING /BINARY

---

### /BINARY

Copies the selected records to a new file in binary format.

### Format

/BINARY

### Description

The /BINARY qualifier specifies that records are output in binary format to the file specified by the /OUTPUT qualifier. (/OUTPUT is Alpha- and I64-only, however.) Use the Accounting utility to process this file later.

See also the /BRIEF, /FULL, and /SUMMARY qualifiers, which process the selected records to produce a report.

You cannot use the /BINARY qualifier with the /BRIEF, /FULL, or /SUMMARY qualifiers.

### Examples

1. \$ ACCOUNTING /USER=SMITH /BINARY /OUTPUT=MYDISK:[ACCOUNTING]MYACC.DAT

This example creates the file MYDISK:[ACCOUNTING]MYACC.DAT. It processes the file SYS\$MANAGER:ACCOUNTNG.DAT, copying all records for the user SMITH to the new file in binary format.

2. \$ ACCOUNTING /TYPE=PRINT -  
\_ \$ /BINARY /OUTPUT=PRINT\_INFO.DAT /REJECTED=NOT\_PRINT\_INFO.DAT

This example creates two files in the default directory, PRINT\_INFO.DAT and NOT\_PRINT\_INFO.DAT. It processes the file SYS\$MANAGER:ACCOUNTNG.DAT, copying print records to PRINT\_INFO.DAT and other records to NOT\_PRINT\_INFO.DAT. These records are in binary format.

---

**/BRIEF**

Produces a brief report of the selected records.

**Format**

/BRIEF (default)

**Description**

The /BRIEF qualifier is the default. It produces a brief report of the selected records. The report is directed to the current SYS\$OUTPUT device, unless you use the /OUTPUT qualifier to write it to a file. (Note that /OUTPUT is Alpha- and I64-only.)

Each line of a brief report corresponds to a record in the accounting file. It does not show resources used, but gives the information shown in the following table about each record in the accounting file:

Column	Description
Date/Time	When the record was logged in the accounting file.
Type	The type of the record.
Subtype	For records of type IMAGE, this is the name of the image (the file name portion of its file specification). For records of type PROCESS, it is the type of the process (BATCH, DETACHED, INTERACTIVE, NETWORK, or SUBPROCESS).
User name	The user name. For login failures where the user did not give a valid user name, this is shown as <login>.
ID	The process identifier (PID). For print jobs, this is the PID of the process that submitted the job.
Source	The terminal associated with an interactive process or, for DECnet for OpenVMS requests, the name of the node that issued the request.
Status	The final exit status code, expressed as a hexadecimal value.

To translate the final exit status code into the equivalent message text, use the F\$MESSAGE lexical function, and precede the status code with %X, as in this example:

```
$ MESSAGE = F$MESSAGE(%X00000001)
$ SHOW SYMBOL MESSAGE
  MESSAGE = "%SYSTEM-S-NORMAL, normal successful completion"
```

See also the /BINARY qualifier, which copies the selected records to a file, and the /FULL and /SUMMARY qualifiers, which produce full and summary reports of the selected records.

You cannot use the /BRIEF qualifier with the /BINARY, /FULL, or /SUMMARY qualifiers.

# ACCOUNTING /BRIEF

## Example

\$ ACCOUNTING

This example produces a brief report of all records in the file  
SYS\$MANAGER:ACCOUNTNG.DAT.

This is an example of the report that is produced:

Date / Time	Type	Subtype	Username	ID	Source	Status
7-JAN-2002 17:20:08	FILE	BL		00000000		00000000
7-JAN-2002 17:22:05	PROCESS	DETACHED	JONES	516000E1		02DBA002
7-JAN-2002 17:22:10	PROCESS	INTERACTIVE	JONES	516000DD	TWA10:	00000001
7-JAN-2002 17:22:16	PROCESS	INTERACTIVE	JONES	51600104	TWA11:	0001C0F4
7-JAN-2002 17:22:20	PROCESS	DETACHED	JONES	51600103		12DB821C
8-JAN-2002 01:06:36	PROCESS	SUBPROCESS	SYSTEM	51600106		10000001
8-JAN-2002 03:09:59	PROCESS	BATCH	SYSTEM	5160010F		10030001
8-JAN-2002 09:13:15	LOGFAIL			51600105		00D3803C
8-JAN-2002 09:14:40	IMAGE	LOGINOUT	JONES	51600110		00000000
8-JAN-2002 09:28:57	PROCESS	SUBPROCESS	SMITH	51600119		10000001
8-JAN-2002 09:50:18	PROCESS	SUBPROCESS	SMITH	5160011A		00000001

---

## /ENTRY

Selects or rejects records for print and batch jobs with the specified queue entry numbers.

### Format

/ENTRY=( [-]entry\_number[,...])

### Description

The /ENTRY qualifier uses the value of the queue entry number field to select records for processing. This field is present in all records except file backward link and file forward link records. For records that contain information about print or batch jobs, it contains the unique entry number assigned to the job in the print or batch queue.

The /ENTRY qualifier selects only records that have the specified values in the queue entry number field. If you precede the values with a minus sign, it selects all records *except* those with the specified values.

See also the /JOB and /QUEUE qualifiers, which select or reject records for print and batch jobs with specified job and queue names.

### Examples

1. \$ ACCOUNTING /ENTRY=(211,212,213)

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records for print or batch jobs with a queue entry number of 211, 212, or 213.

2. \$ ACCOUNTING /ENTRY=(-25,50)

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records except those for print or batch jobs with a queue entry number of 25 or 50.

## ACCOUNTING /FULL

---

### /FULL

Produces a full report of the selected records.

### Format

/FULL

### Description

The /FULL qualifier produces a full report of the selected records. The report is directed to the current SYS\$OUTPUT device, unless you use the /OUTPUT qualifier to write it to a file. (Note that /OUTPUT is Alpha- and I64-only.)

Full reports display one screen of information for each selected record. The information displayed, and the way that it is laid out, depend on the type of the record and the data it contains.

The first line shows the event that caused the record to be logged in the accounting file. For example, for a record that was logged when an interactive process terminated, the first line shows INTERACTIVE Process Termination.

For subprocesses, the Owner ID field shows the process identifier (PID) of the parent process.

For records that contain information about DECnet for OpenVMS requests, the three Remote fields identify the remote user and remote node.

The Processor time field shows the total CPU time used. This includes any vector CPU time used. The Vector CPU time field is shown only if vector CPU time has been used.

Vector CPU time is the time that the process was scheduled on a vector-present CPU while that process was a vector consumer. Note that:

- When a process is a vector consumer, it accrues vector CPU time when it is scheduled, even if it does not issue any vector instructions.
- Processes that are scalar consumers or marginal vector consumers do not accrue vector CPU time, even when they are scheduled on vector-present CPUs.

The privilege is shown as two hexadecimal numbers that represent the first and last 32 bits of the 64-bit privilege mask. To translate the privilege bit mask into privileges, refer to the definitions of the symbols that begin PRV\$\_ in the \$PRVDEF macro in the STARLET library. For example, the \$PRVDEF macro defines the PRV\$\_READALL symbol to equate to 35. This means that READALL privilege is represented by bit 35 set in the privilege mask.

If you are processing only one file and you are displaying it on your screen, when you do not want to look at any more records, press Ctrl/Z to return to the DCL prompt.

See also the /BINARY qualifier, which copies the selected records to a file, and the /BRIEF and /SUMMARY qualifiers, which produce brief and summary reports of the selected records.

You cannot use the /FULL qualifier with the /BINARY, /BRIEF, or /SUMMARY qualifiers.

## Examples

1. \$ ACCOUNTING /FULL

This example displays a full report of all the records in the file SYS\$MANAGER:ACCOUNTNG.DAT. This example screen shows a record that was logged when an interactive process terminated. The interactive process was created when the user JONES at the node HQ222 entered a SET HOST command to connect to the local node.

INTERACTIVE Process Termination

```
-----
Username:      FISH          UIC:           [DOC,FISH]
Account:       DOC           Finish time:    23-JAN-2002 15:21:23.83
Process ID:    20A0029B      Start time:     23-JAN-2002 15:19:08.28
Owner ID:      0             Elapsed time:   0 00:02:15.55
Terminal name: RTA2:        Processor time: 0 00:00:04.14
Remote node addr: 63576     Priority:       4
Remote node name: HQ222    Privilege <31-00>: 00108000
Remote ID:     JONES        Privilege <63-32>: 00000000
Queue entry:   0             Final status code: 00000001
Queue name:
Job name:
Final status text: %SYSTEM-S-NORMAL, normal successful completion

Page faults:   2043          Direct IO:      159
Page fault reads: 68        Buffered IO:    228
Peak working set: 852       Volumes mounted: 0
Peak page file: 5512        Images executed: 10
                                   Vector CPU time: 0 00:00:0.54

Press RETURN for Next Record >
```

2. \$ ACCOUNTING /FULL /OUTPUT=MYACC

This example creates the output file MYACC.LIS in the default directory. It processes the file SYS\$MANAGER:ACCOUNTNG.DAT, writing a full report of all records to the new output file.

## ACCOUNTING /IDENT

---

### /IDENT

Selects or rejects records for the specified processes.

### Format

/IDENT=(-)pid[,...]

### Description

The /IDENT qualifier uses the value of the process identifier (PID) field to select records for processing. This field is present in all records except file backward link and file forward link records. For print job records, it contains the PID of the process that submitted the job.

The /IDENT qualifier selects only records that have the specified values in the PID field. If you precede the values with a minus sign, it selects all records *except* those with the specified values.

See also the /OWNER qualifier, which selects or rejects records for subprocesses created by specified processes.

### Examples

1. \$ ACCOUNTING /IDENT=(25634,045A6B)

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records for processes with a PID of 25634 or 045A6B.

2. \$ ACCOUNTING /IDENT=(-25634,045A6B)

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records except those for processes with a PID of 25634 or 045A6B.



---

## /IMAGE

Selects or rejects records for the specified images.

### Format

/IMAGE=([-]image\_name[,...])

### Description

The /IMAGE qualifier uses the value of the image name field to select records for processing. This field is present only in records of type IMAGE, and contains the name of the image.

Note that the system does not track records of type IMAGE by default. To enable the tracking of IMAGE records, use the SET ACCOUNTING command.

The /IMAGE qualifier selects only records that have the specified values in the image name field. If you precede the values with a minus sign, it selects all records *except* those with the specified values.

Each image name is a string that gives the file name portion of the image file specification. Do not include the device, directory, or file type.

### Examples

1. \$ ACCOUNTING /IMAGE=DIRECTORY

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records for the DIRECTORY.EXE image.

2. \$ ACCOUNTING /IMAGE=-DIRECTORY

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records except those for the DIRECTORY.EXE image.

## ACCOUNTING /JOB

---

### /JOB

Selects or rejects records for print and batch jobs with the specified job names.

### Format

/JOB=(-]job\_name[,...])

### Description

The /JOB qualifier uses the value of the job name field to select records for processing. This field is present in all records except file backward link and file forward link records. For records that contain information about print and batch jobs, it contains the name of the job.

The /JOB qualifier selects only records that have the specified values in the job name field. If you precede the values with a minus sign, it selects all records *except* those with the specified values.

See also the /QUEUE and /ENTRY qualifiers, which select or reject records for print and batch jobs with specified queue names and queue entry numbers.

### Examples

1. \$ ACCOUNTING /JOB=(MYJOB1,MYJOB2)

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records for print or batch jobs named MYJOB1 or MYJOB2.

2. \$ ACCOUNTING /JOB=(-MYJOB1,MYJOB2) /FULL

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a full report of all records except those for print or batch jobs named MYJOB1 or MYJOB2.

---

## /LOG

Outputs informational messages.

### Format

/LOG

### Description

The /LOG qualifier outputs these informational messages to the current SYS\$OUTPUT device:

- For each file processed, the name of the file and the number of records selected and rejected from that file
- If you use the /SORT qualifier, the total number of records merged in the sort (this is the total number of records selected from all the files that were processed)
- If you process more than one file, the total number of files that were processed, and the total number of records selected and rejected

### Example

```
$ ACCOUNTING MYFILE1.DAT,MYFILE2.DAT /TYPE=PRINT /SORT=USER /OUTPUT=OUTFILE
```

```
%ACC-I-INPUT, SYS$SYSROOT:[SYSMGR]MYFILE1.DAT;7, 297 selected, 16460 rejected  
%ACC-I-INPUT, SYS$SYSROOT:[SYSMGR]MYFILE2.DAT;13,302 selected, 16388 rejected  
%ACC-I-MERGE, 599 records to be merged  
%ACC-I-TOTAL, 599 selected, 32848 rejected, 2 input files
```

This example processes two accounting files. It writes a brief report of all the records for print jobs, sorted in user name order, to an output file and displays informational messages that tell you which files were processed and how many records were selected and rejected.

## ACCOUNTING /NODE

---

### /NODE

Selects or rejects records for DECnet for OpenVMS requests made by the specified nodes.

### Format

/NODE=([-]node\_name[,...])

### Description

The /NODE qualifier uses the value of the remote node name field to select records for processing. This field is present in all records except file backward link and file forward link records. For records that contain information about DECnet for OpenVMS requests, it contains the name of the node that made the request.

The /NODE qualifier selects only records that have the specified values in the remote node name field. If you precede the values with a minus sign, it selects all records *except* those with the specified values.

Do not include the double colon (::) after the name of the node.

See also the /ADDRESS and /REMOTE\_ID qualifiers, which select or reject records for DECnet for OpenVMS requests made by specified node addresses and remote IDs respectively.

### Examples

1. \$ ACCOUNTING /NODE=HQ291 /FULL

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a full report of all records for DECnet for OpenVMS requests made by the node HQ291.

2. \$ ACCOUNTING /NODE=(-HQ222,HQ223)

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records except those for DECnet for OpenVMS requests made by the nodes HQ222 or HQ223.

## **/OUTPUT (Alpha and I64)**

Specifies the output file.

Requires read and write access to the directory in which the output file is created.

### **Format**

`/OUTPUT[=filespec]`

### **Description**

The `/OUTPUT` qualifier creates the specified output file and writes the report or copies the selected records to that file.

If you omit the `/OUTPUT` qualifier, or you use the `/OUTPUT` qualifier and omit the file specification, the report or selected records are output to the current `SYS$OUTPUT` device.

If the file specification does not include the device or directory name, your current default device or directory is used. If you omit the file name, the file name of the first input file is used (the first file listed in the parameter to the `ACCOUNTING` command). If you omit the file type, the default file type is `.LIS` if you are producing reports, and `.DAT` if you are copying records.

### **Examples**

1. `$ ACCOUNTING MYFILE1.DAT,MYFILE2.DAT /SORT=USER /BINARY /OUTPUT=.NEW`

This example creates the output file `MYFILE1.NEW` in the default directory. It processes two accounting files, `MYFILE1.DAT` and `MYFILE2.DAT`, sorting their records in user name order, then copies these records to the new output file.

2. `$ ACCOUNTING MYFILE1.NEW /FULL  
/OUTPUT=MYDISK:[ACCOUNTING]STAT`

This example creates the output file `MYDISK:[ACCOUNTING]STAT.LIS`, and writes a full report of all the records in `MYFILE1.NEW` to the new output file.

## ACCOUNTING /OWNER

---

### /OWNER

Selects or rejects records for subprocesses created by the specified processes.

### Format

```
/OWNER=[-]owner_pid[,...]
```

### Description

The /OWNER qualifier uses the value of the process owner field to select records for processing. This field is present in all records except file backward link and file forward link records. For a subprocess, this field contains the process identifier (PID) of the process that created it.

The /OWNER qualifier selects only records that have the specified values in the process owner field. If you precede the values with a minus sign, it selects all records *except* those with the specified values.

See also the /IDENT qualifier, which selects or rejects records for specified processes.

### Example

```
$ ACCOUNTING /OWNER=(25634,045A6B)
```

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records for subprocesses created by processes with a PID of 25634 or 045A6B.

## **/PRIORITY**

Selects or rejects records for the specified priority.

### **Format**

`/PRIORITY=([-]priority[,...])`

### **Description**

The `/PRIORITY` qualifier uses the value of the priority field to select records for processing. This field is present in all records except file backward link and file forward link records. For print and batch job records, this field contains the priority of the job in the print or batch queue. For other records, it contains the base process priority.

The `/PRIORITY` qualifier selects only records that have the specified values in the priority field. If you precede the values with a minus sign, it selects all records *except* those with the specified values.

### **Example**

```
$ ACCOUNTING /PRIORITY=3
```

This example processes the file `SYS$MANAGER:ACCOUNTNG.DAT`. It produces a brief report of all records for processes with a base priority of 3 and for print and batch jobs with a queue priority of 3.

# ACCOUNTING /PROCESS

---

## /PROCESS

Selects or rejects records for the specified types of process.

### Format

/PROCESS=[(-)]process\_type[,...]

### Keyword

#### **process\_type[,...]**

Specifies which types of process you want to select or reject. The following table shows the keywords available:

Keyword	Type of Process
BATCH	Batch process
DETACHED	Detached process
INTERACTIVE	Interactive process
NETWORK	Network process
SUBPROCESS	Subprocess of any of the other process types

### Description

The /PROCESS qualifier uses the value of the process type field to select records for processing. This field is present only in records of type IMAGE and type PROCESS. For records of type IMAGE, this field contains the type of the process in which the image was executed.

The /PROCESS qualifier selects only records that match the specified types of process. If you precede the list with a minus sign, it selects all records *except* those for the specified types of process.

See also the /TYPE qualifier, which selects or rejects specified types of record.

### Example

```
$ ACCOUNTING /TYPE=IMAGE /PROCESS=INTERACTIVE /FULL
```

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a full report of the resources used by images running in interactive processes.



---

## /QUEUE

Selects or rejects records for print or batch jobs executed by the specified queues.

### Format

/QUEUE=[(-)queue\_name[,...]]

### Description

The /QUEUE qualifier uses the value of the queue name field to select records for processing. This field is present in all records except file backward link and file forward link records. For records that contain information about print or batch jobs, it contains the name of the queue that executed the job.

The /QUEUE qualifier selects only records that have the specified values in the queue name field. If you precede the values with a minus sign, it selects all records *except* those with the specified values.

See also the /JOB and /ENTRY qualifiers.

### Example

```
$ ACCOUNTING /QUEUE=SYS$MYNODE_BATCH
```

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records for jobs executed by the SYS\$MYNODE\_BATCH queue.

## ACCOUNTING /REJECTED

---

### /REJECTED

Copies the rejected records to a new file.

Requires read and write access to the directory in which the specified file is created.

### Format

/REJECTED=filespec

### Description

The /REJECTED qualifier creates the specified file, then copies the records that do not match your selection criteria to this file in binary format. Use the Accounting utility to process this file later.

If the file specification does not include the device or directory name, your current default device or directory is used. If you omit the file name, the file name of the first input file is used (the first file listed in the parameter to the ACCOUNTING command). If you omit the file type, .REJ is used.

### Example

```
$ ACCOUNTING /TYPE=PRINT /BINARY /OUTPUT=PRINT_INFO.DAT -  
_$_ /REJECTED=NOT_PRINT_INFO.DAT
```

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It creates two files, PRINT\_INFO.DAT and NOT\_PRINT\_INFO.DAT, in the default directory. It copies print job records to PRINT\_INFO.DAT and all other records to NOT\_PRINT\_INFO.DAT.

---

## /REMOTE\_ID

Selects or rejects records for DECnet for OpenVMS requests made by the specified remote IDs.

### Format

```
/REMOTE_ID=([-]remote_id[,...])
```

### Description

The /REMOTE\_ID qualifier uses the value of the remote ID field to select records for processing. This field is present in all records except file backward link and file forward link records. For records that contain information about DECnet for OpenVMS requests, this field contains a string that identifies the user who made the request. If the remote process was on an OpenVMS node, this is the user name of the user at the remote node.

The /REMOTE\_ID qualifier selects only records that have the specified values in the remote ID field. If you precede the values with a minus sign, it selects all records *except* those with the specified values.

See also the /NODE and /ADDRESS qualifiers, which select or reject records for DECnet for OpenVMS requests made by nodes with specified names and addresses respectively.

### Example

```
$ ACCOUNTING /NODE=HQ223 /REMOTE_ID=SMITH /FULL
```

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a full report of all the records for DECnet for OpenVMS requests made by user SMITH at the node HQ223.

# ACCOUNTING /REPORT

---

## /REPORT

Specifies the resources that you want to summarize in a summary report.

### Format

/REPORT[=(resource[,...])]

### Keyword

#### **resource[,...]**

Specifies the resources that you want to summarize in the report. The following table shows the keywords available:

<b>Keyword</b>	<b>Description</b>	<b>How Summarized</b>
BUFFERED_IO <sup>2</sup>	Number of buffered I/Os	Total
DIRECT_IO <sup>2</sup>	Number of direct I/Os	Total
ELAPSED <sup>1, 2</sup>	Elapsed time	Total
EXECUTION <sup>2</sup>	Number of images run by the process	Total
FAULTS <sup>2</sup>	Number of hard and soft page faults	Total
GETS <sup>1</sup>	Number of GETs from the file that was printed	Total
PAGE_FILE <sup>2</sup>	Page file usage	Maximum
PAGE_READS <sup>2</sup>	Number of hard page faults	Total
PAGES <sup>1</sup>	Number of pages printed	Total
PROCESSOR <sup>2</sup>	Total CPU time used	Total
QIOS <sup>1</sup>	Number of QIOs to the printer	Total
RECORDS	Number of accounting file records processed	Total
VECTOR_PROCESSOR <sup>2</sup>	Vector CPU time used (see the description of the /FULL qualifier for further details)	Total
VOLUMES <sup>2</sup>	Number of volumes mounted	Total
WORKING_SET <sup>2</sup>	Working set size	Maximum

---

<sup>1</sup>Present in records of type PRINT

<sup>2</sup>Present in records of type IMAGE, LOGFAIL, PROCESS, and SYSINIT

---

The RECORDS keyword is the default if you omit either the keywords or the /REPORT qualifier. It gives the total number of records for each summary key value.

## **Description**

The /REPORT qualifier specifies the resources that you want to summarize in a summary report. The resources are summarized, either as totals or maximum values, for each summary key value specified by the /SUMMARY qualifier.

When a record is processed that does not contain the specified resource field, a default value of 0 is used. For example, if you use the PAGES keyword to summarize the total pages printed, the value of 0 is used for each record that is not of type PRINT.

Note that the resource usage data stored in records of type IMAGE is a subset of the data stored in records of type PROCESS. For example, the CPU time stored in a record of type PROCESS includes the CPU time used by the images executed by that process. To make sure that you do not count the same resource data twice when you are summarizing process resources by totals, use the /TYPE qualifier to exclude records of type IMAGE.

You cannot use the /REPORT qualifier without the /SUMMARY qualifier.

## **Examples**

1. \$ ACCOUNTING /SUMMARY=IMAGE /REPORT=(RECORDS,PROCESSOR)

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a summary report that shows for each image the number of times it was executed and the total CPU time consumed.

2. \$ ACCOUNTING /TYPE=-IMAGE /SUMMARY=USER /REPORT=EXECUTION

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a summary report that shows the total number of images executed by each user. Notice the use of the /TYPE qualifier to exclude records of type IMAGE to avoid double counting.

## ACCOUNTING /SINCE

---

### /SINCE

Selects all records time-stamped at or after the specified time.

### Format

/SINCE[=time]

### Description

All records in an accounting file are time-stamped with the time the record was logged in the file.

The /SINCE qualifier selects only the records time-stamped on or after the specified time. You can specify an absolute time, delta time, or a combination of the two. If you omit the time, 00:00 hours on the current day is used.

See also the /BEFORE qualifier, which selects records time-stamped before a specified time.

### Example

```
$ ACCOUNTING /SINCE=5-JAN-2002
```

This example produces a brief report of all records time-stamped at or after 5-JAN-2002 in the file SYS\$MANAGER:ACCOUNTNG.DAT.

---

**/SORT**

Sorts the selected records.

**Format**

/SORT=[(-)sort\_field[,...]]

**Keyword**

**sort\_field[,...]**

Specifies the sort key.

The following table shows the keywords available. You can specify up to ten sort fields.

Keyword	Sorts on This Field
ACCOUNT	Account
ADDRESS	Address of the node that made the DECnet for OpenVMS request
BUFFERED_IO	Number of buffered I/Os
DIRECT_IO	Number of direct I/Os
ELAPSED	Elapsed time
ENTRY	Print or batch job queue entry number
EXECUTION	Number of images run by the process
FAULTS	Number of hard and soft page faults
FINISHED	Time record was logged in the accounting file
GETS	Number of GETs from the file that was printed
IDENT	Process identifier (PID)
IMAGE	Image name (sorts only on file name portion of the image file specification)
JOB	Name of print or batch job
NODE	Name of the node that made the DECnet for OpenVMS request
OWNER	PID of parent process
PAGE_FILE	Peak page file usage
PAGE_READS	Number of hard page faults
PAGES	Number of pages printed
PRIORITY	Base process priority, or print or batch queue priority
PROCESS	Type of process
PROCESSOR	Total CPU time used
QIOS	Number of QIOS to the printer
QUEUE	Name of print or batch queue
QUEUED	Time print job was queued
STARTED	Start time

## ACCOUNTING /SORT

Keyword	Sorts on This Field
STATUS	Final exit status code
TERMINAL	Terminal name
TYPE	Type of record
UIC	User identification code
USER	User name at local node
VECTOR_PROCESSOR	Vector CPU time (see the description of the /FULL qualifier for further details)
VOLUMES	Number of volumes mounted
WORKING_SET	Peak working set size

For each keyword, see the description of the corresponding Accounting utility qualifier or the table in the /TYPE qualifier section for details of the types of record in which the corresponding field is present.

### Description

The /SORT qualifier merges the selected records from each input file (each file listed in the parameter to the ACCOUNTING command) and sorts them using the specified sort key. The records are sorted according to the value of the first sort field in the list, and when two or more records have the same value in this field, they are sorted by the value of the second sort field in the list, and so on.

The records are sorted in ascending order of the sort field value. If the keyword is preceded by a minus sign, the records are sorted in descending order.

When you use the /SORT qualifier, records are rejected if they do not contain the sort field. For example, /SORT=IMAGE rejects all records that are not of type IMAGE, because the image name field is only present in records of type IMAGE. Similarly, /SORT=PAGES rejects all records except those for print jobs.

You cannot use the /SORT qualifier with the /SUMMARY qualifier.

### Examples

1. `$ ACCOUNTING /TYPE=PRINT /SORT=USER`

This example processes the file `SY$MANAGER:ACCOUNTNG.DAT`. It produces a brief report of all the records for print jobs and displays them in user name order.

The following example shows the report that is produced:

Date / Time	Type	Subtype	Username	ID	Source	Status
14-JAN-2002 09:53:05	PRINT		BROWN	20A00193		00040001
13-JAN-2002 13:36:04	PRINT		BROWN	20A00442		00000001
13-JAN-2002 12:42:37	PRINT		BROWN	20A00442		00000001
13-JAN-2002 14:43:56	PRINT		DECNET_MAIL	20A00456		00000001
14-JAN-2002 19:39:01	PRINT		DECNET_MAIL	20A00265		00000001
14-JAN-2002 20:09:03	PRINT		DECNET_MAIL	20A00127		00000001
14-JAN-2002 20:34:45	PRINT		DECNET_MAIL	20A00127		00000001
14-JAN-2002 11:23:34	PRINT		FISH	20A0032E		00040001
14-JAN-2002 16:43:16	PRINT		JONES	20A00070		00040001
14-JAN-2002 09:30:21	PRINT		SMITH	20A00530		00040001



2. \$ ACCOUNTING MYFILE1.DAT,MYFILE2.DAT /SORT=IMAGE -  
\_ \$ /FULL /REJECTED=NON\_IMAGE.DAT

This example processes two files, MYFILE1.DAT and MYFILE2.DAT, to produce a full report of records of type IMAGE, sorted in image name order. It creates the file NON\_IMAGE.DAT, and copies all records except those of type IMAGE to that file. Notice that no selection qualifiers are used, and so all records are selected for processing. When the records are sorted, records that do not contain an image name are rejected.

## ACCOUNTING /STATUS

---

### /STATUS

Selects or rejects records with the specified final exit status codes.

#### Format

```
/STATUS=[-]status_code[,...]
```

#### Description

The /STATUS qualifier uses the value of the final status code field to select records for processing. This field is present in all records except records of type USER, file backward link records, and file forward link records.

The /STATUS qualifier selects only records that have the specified values in the final status code field. If you precede the values with a minus sign, it selects all records *except* those with the specified values.

See the description of the /BRIEF qualifier for details of how to convert a final exit status code to the equivalent message text.

#### Example

```
$ ACCOUNTING /STATUS=10D38064
```

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records with a final exit status code of 10D38064.

**/SUMMARY**

Produces a summary report of the selected records.

**Format**

/SUMMARY[=(summary\_item[,...])]

**Keyword**

**summary\_item[,...]**

Specifies the summary key. The following table lists keywords:

<b>Keyword</b>	<b>Description</b>
ACCOUNT	Account
DATE	Date
DAY	Day of month (1–31)
HOUR	Hour of day (0–23)
IMAGE	Image name (file name portion of image file specification)
JOB	Name of print or batch job
MONTH	Month of year (1–12)
NODE	Name of the node that issued the DECnet for OpenVMS request
PROCESS	Process type
QUEUE	Print or batch job queue name
TERMINAL	Terminal name
TYPE	Record type
UIC	User identification code
USER	User name
WEEKDAY	Day of week (0=Sunday, 1=Monday, and so on)
YEAR	Year

If you omit these keywords, the user name is used as the summary key.

**Description**

The /SUMMARY qualifier produces a summary report of the selected records. The report is directed to the current SYS\$OUTPUT device, unless you use the /OUTPUT qualifier to write it to a file.

Summary reports give statistical summaries of the resources specified by the /REPORT qualifier for each value of the summary key specified by the /SUMMARY qualifier. If you omit the /REPORT qualifier, the summary report gives the total number of records processed for each summary key value.

The first line of the summary report shows the time span of the data processed (when the first and last records processed were logged in the input files), with a title in the middle. You can use the /TITLE qualifier to specify your own title.

## ACCOUNTING /SUMMARY

The next few lines of the report are column headings. There is one column for each summary\_item, then one column for each resource specified by the /REPORT qualifier. The columns are laid out in the same left-to-right sequence as the equivalent keywords in the /SUMMARY and /REPORT qualifiers.

The rest of the report uses one line for each summary key value. It gives a summary of the resources associated with that summary key value. The data is sorted in ascending order of the summary key value.

See also the /BINARY qualifier, which copies the selected records to a file, and the /BRIEF and /FULL qualifiers, which produce brief and full reports of the selected records.

You cannot use the /SUMMARY qualifier with the /BINARY, /BRIEF, or /FULL qualifiers.

### Examples

1. \$ ACCOUNTING /TYPE=PRINT /SUMMARY=USER /REPORT=(PAGES,RECORDS)

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It processes all the print job records and produces a summary report that shows, for each user, the total number of pages printed and the number of records that were added together to produce this total. This is an example of the report that is produced:

From: 12-JAN-2002 15:55 VAX/VMS Accounting Report To: 15-JAN-2002 15:17

Username	Pages Printed	Total Records
BROWN	115	2
CROW	3	1
CUTHBERT	20	4
FOSTER	46	1
SMITH	50	3
WHITE	50	7

2. \$ ACCOUNTING /SUMMARY=IMAGE /REPORT=(PROCESSOR,RECORDS)

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a summary report that shows the total CPU time used by each image. This is an example of the report that is produced:

From: 12-JAN-2002 15:55 VAX/VMS Accounting Report To: 15-JAN-2002 15:17

Image name	Processor Time	Total Records
	0 00:09:09.83	51
ACC	0 00:01:36.72	99
AUTHORIZE	0 00:00:04.17	8
CDU	0 00:00:33.25	21
COPY	0 00:00:05.97	30
DELETE	0 00:00:02.79	12
DIRECTORY	0 00:00:09.67	38
DUMP	0 00:00:04.51	3
EDT	0 00:00:05.85	7
LOGINOUT	0 00:04:03.48	75
NETSERVER	0 00:00:00.63	23
SHOW	0 00:00:04.80	22

---

## /TERMINAL

Selects or rejects records for interactive sessions at the specified terminals.

### Format

/TERMINAL=[-]terminal\_name[,...]

### Description

The /TERMINAL qualifier uses the value of the terminal name field to select records for processing. This field is present in all records except file backward link and file forward link records. For records that contain information about interactive sessions, this field contains the name of the terminal associated with the session.

The /TERMINAL qualifier selects only records that have the specified values in the terminal name field. If you precede the values with a minus sign, it selects all records *except* those with the specified values.

Give the terminal name as the standard device name and include the colon (:).

### Example

```
$ ACCOUNTING /TERMINAL=TTB3:
```

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records for interactive sessions at the terminal TTB3.

## ACCOUNTING /TITLE

---

### /TITLE

Specifies the title shown on the first line of a summary report.

### Format

/TITLE=title

### Description

The /TITLE qualifier specifies the title shown in the center of the first line of summary reports. The title is truncated if it is too long. For reports displayed on your screen, the title is truncated if it is longer than (W-56) characters, where W is the width (in characters) of your screen.

### Example

```
$ ACCOUNTING /SUMMARY=IMAGE /TITLE="June Accounting Report"
```

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a summary report that shows the number of times each image was executed. The title "June Accounting Report" appears at the top of the report.

**/TYPE**

Selects or rejects the specified types of record.

**Format**

/TYPE=(**[-**]record\_type[,...])

**Keyword**

**record\_type[,...]**

Specifies the types of record that you want to select or reject. The following table shows the keywords available:

<b>Keyword</b>	<b>Type of Record</b>	<b>Description of Record</b>
FILE	FILE_BL	File backward link. This is the first record in the accounting file. It is logged when the file is created, and contains the name of the previous accounting file.
	FILE_FL	File forward link. This is the last record in the file. It is logged when the file is closed, and contains the name of the next accounting file.
IMAGE	IMAGE	Image termination. It contains details of the resources used by the image.
LOGFAIL	LOGFAIL	Failed attempt to log in. It contains details of the resources used by the login attempt.
PRINT	PRINT	Print job termination. It contains details of the resources used by the print job.
PROCESS	PROCESS	Process termination. It contains details of the resources used by the process. Note that this includes the resources used by the images executed by that process.
SYSINIT	SYSINIT	System booted. It contains details of resources used to boot the system.
UNKNOWN		Record not recognized as one of the other types in this table.
USER	USER	Record logged by a program calling the \$SNDJBC system service to send an accounting message.

**Description**

All records in an accounting file contain a type field that contains the type of the record.

The /TYPE qualifier selects the specified types of record. If you precede the list with a minus sign, it selects all records *except* those specified.

See also the /PROCESS qualifier, which selects or rejects records for particular types of process.

## ACCOUNTING /TYPE

### Examples

1. \$ ACCOUNTING /TYPE=PRINT

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records for print jobs.

2. \$ ACCOUNTING /TYPE=-PRINT

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records except those for print jobs.



---

**/UIC**

Selects or rejects records for the specified UICs.

**Format**

/UIC=(-)uic[,...]

**Description**

The /UIC qualifier uses the value of the UIC field to select records for processing. This field is present in all records except file backward link and file forward link records. It contains the value [SYSTEM] for login failure records where the user did not give a valid user name.

The /UIC qualifier selects only records that have the specified values in the UIC field. If you precede the values with a minus sign, it selects all records *except* those with the specified values.

You can specify the UIC in numeric or alphanumeric format, and can use the asterisk (\*) wildcard character.

**Example**

```
$ ACCOUNTING /UIC=( [360,*] , [ADMIN,COTTON] )
```

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records for users in group number 360 or users whose UIC is [ADMIN,COTTON].

## ACCOUNTING /USER

---

### /USER

Selects or rejects records for the specified user names.

### Format

/USER=([-]user name[,...])

### Description

The /USER qualifier uses the value of the user name field to select records for processing. This field is present in all records except file backward link and file forward link records. It contains the value <login> for login failure records where the user did not give a valid user name.

The /USER qualifier selects only records that have the specified values in the user name field. If you precede the values with a minus sign, it selects all records *except* those with the specified values.

### Examples

1. \$ ACCOUNTING /USER=SMITH

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all records for the user SMITH.

2. \$ ACCOUNTING /USER=(-SMITH,JONES)

This example processes the file SYS\$MANAGER:ACCOUNTNG.DAT. It produces a brief report of all the records except for those of the users SMITH and JONES.

---

**/WIDE**

Changes the width of Buffered I/O and Direct I/O fields in a report from 8 to 10 characters.

**Format**

/WIDE

**Description**

The /WIDE qualifier corrects a problem that users have had with ACCOUNTING reports: the Buffered I/O and Direct I/O fields were too small and displayed asterisks (\*) when numbers exceeded 8 characters.

The /WIDE qualifier changes the widths of the Buffered I/O and Direct I/O fields in reports to 10 characters.

**Example**

```
$ ACCOUNTING /PROC=BATCH /TYP=PROC -
  /REPORT=(RECORDS,PROCESSOR,DIRECT_IO,BUFFER) -
  /SUMM=MONTH /SIN=1-JAN /WIDE
```

MM	TOTAL RECORDS	PROCESSOR TIME	DIRECT I/O	BUFFERED I/O
01	2043	19 06:52:40.97	532675222	551986091
02	1767	9 00:14:34.00	183290432	420000532

Without the /WIDE qualifier, the Direct I/O or Buffered I/O fields print \*\*\*\*\* if the field overflows. With the /WIDE qualifier, these fields print correctly.



---

**Analyze/Disk\_Structure Utility**



### 3.1 ANALYZE/DISK\_STRUCTURE Description

You can use the Analyze/Disk\_Structure utility (ANALYZE/DISK\_STRUCTURE) in two important ways:

- Use the utility on a regular basis to check disks for inconsistencies and errors, and to recover lost files.
- Use the utility with the /SHADOW qualifier to examine the entire contents of a shadow set or a specified range of blocks in a shadow set.

These two uses are explained in the following sections.

#### Checking Disks

ANALYZE/DISK\_STRUCTURE detects problems on On-Disk Structure (ODS) Levels 1, 2, and 5 Files–11 disks; hardware errors, system errors, or user errors can cause these problems. By using ANALYZE/DISK\_STRUCTURE to identify and delete **lost files** and files marked for deletion, you can reclaim disk space.

ANALYZE/DISK\_STRUCTURE performs the verification of a volume or volume set in eight distinct stages. During these stages, ANALYZE/DISK\_STRUCTURE collects information used in reporting errors or performing repairs. However, ANALYZE/DISK\_STRUCTURE repairs volumes only when you specify the /REPAIR qualifier. For a complete description of each of the eight stages, and an annotated example of an ANALYZE/DISK\_STRUCTURE session, refer to Appendix D.

ANALYZE/DISK\_STRUCTURE allocates virtual memory to hold copies of the index file and storage bitmaps. With larger bitmaps introduced in OpenVMS Version 7.2, the virtual memory requirements increase correspondingly. To use this utility on volumes with large bitmaps, you might need to increase your page file quota. On OpenVMS VAX systems, you might also need to increase the system parameter VIRTUALPAGECNT.

Virtual memory size requirements for the bitmaps are in VAX pages (or Alpha and I64 512-byte pagelets) per block of bitmap. Note that the size of the index file bitmap in blocks is the maximum number of files divided by 4096. For ANALYZE/DISK\_STRUCTURE, this requirement is the sum of the following across the entire volume set:

- 3 times all the storage bitmaps plus the largest bitmap in the volume set
- 117 times the index file bitmaps
- An additional 96 times the index file bitmaps if /USAGE was requested
- Approximately 600 pages additional fixed scratch space

#### Examining Shadow Sets

The ANALYZE/DISK\_STRUCTURE/SHADOW command is especially useful if a shadow set was initialized with the INITIALIZE/SHADOW command but *without* the /ERASE qualifier.

Another use of the ANALYZE/DISK\_STRUCTURE/SHADOW command is if an error is logged on a member device, and you do not know whether the error was caused by a disk error or by some other hardware component such as a disk controller or cable. When you use the ANALYZE/DISK\_STRUCTURE/SHADOW command, every block of every member is read and compared.

Refer to the Section 3.1.2 and to the /SHADOW qualifier documentation for further details.

## ANALYZE/DISK\_STRUCTURE

### 3.1 ANALYZE/DISK\_STRUCTURE Description

#### 3.1.1 Disk Error Reporting and Repair

You can invoke the Analyze/Disk\_Structure utility to operate in any of the following three modes for disks errors:

- Error reporting with no repairs
- Error reporting with repairs
- User-controlled selective repairs

By default, ANALYZE/DISK\_STRUCTURE reports errors, but does not make repairs. For example, use the following command to report all errors on device DBA1:

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

When you issue this command, ANALYZE/DISK\_STRUCTURE runs through eight stages of data collection, then, by default, prints a list of all errors and lost files to your terminal. One type of problem that ANALYZE/DISK\_STRUCTURE locates is an invalid directory backlink; a backlink is a pointer to the directory in which a file resides. If your disk has a file with an invalid directory backlink, ANALYZE/DISK\_STRUCTURE displays the following message and the file specification to which the error applies:

```
%VERIFY-I-BACKLINK, incorrect directory back link [SYSEXE]SYSBOOT.EXE;1
```

To instruct ANALYZE/DISK\_STRUCTURE to repair the errors that it detects, use the /REPAIR qualifier. For example, the following command reports and repairs all errors on the DBA1 device:

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

---

#### Notes

---

HP recommends using a colon (:) after device names in commands.

When you update the storage control block (SCB) within a BITMAP.SYS file, the VERIFY utility forces the volume to perform mount verification if the volume is controlled by host-based shadowing.

---

To select which errors ANALYZE/DISK\_STRUCTURE repairs, use both the /REPAIR and /CONFIRM qualifiers:

```
$ ANALYZE/DISK_STRUCTURE DBA1:/REPAIR/CONFIRM
```

When you issue this command, ANALYZE/DISK\_STRUCTURE displays a description of each error and prompts you for confirmation before making a repair. For example, the previous command might produce the following messages and prompts:

```
%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
```



## ANALYZE/DISK\_STRUCTURE

### 3.1 ANALYZE/DISK\_STRUCTURE Description

Consider running ANALYZE/DISK\_STRUCTURE twice for each volume. First, invoke the utility to report all errors. Evaluate the errors and decide on an appropriate action. Then invoke the utility again with the /REPAIR qualifier to repair all errors, or with the /REPAIR and /CONFIRM qualifiers to repair selected errors.

For message descriptions, use the online Help Message (MSGHLP) utility or refer to the OpenVMS system messages documentation.

#### Recovering Lost Files

A lost file is a file that is not linked to a directory. Under normal circumstances, files do not become lost. However, files occasionally become lost because of disk corruption, hardware problems, or user error. For example, in cleaning up files and directories, you might inadvertently delete directories that still point to files. When you delete a directory file (a file with the file type .DIR) without first deleting its subordinate files, the files referred to by that directory become lost files. Though lost, these files remain on the disk and consume space.

When you run ANALYZE/DISK\_STRUCTURE specifying the /REPAIR qualifier, the utility places lost files in SYSLOST.DIR.

For example, to report and repair all errors and lost files found on the device DDA0, issue the following command:

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

If it discovers lost files on your disk, ANALYZE/DISK\_STRUCTURE issues messages similar to those that follow:

```
%VERIFY-W-LOSTHEADER, file (16,1,1) [X.X;1
    not found in a directory
%VERIFY-W-LOSTHEADER, file (17,1,1) [Y.Y;1
    not found in a directory
%VERIFY-W-LOSTHEADER, file (18,1,1) [Z.Z;1
    not found in a directory
%VERIFY-W-LOSTHEADER, file (19,1,1) [X.X;2
    not found in a directory
%VERIFY-W-LOSTHEADER, file (20,1,1) [Y.Y;2
    not found in a directory
%VERIFY-W-LOSTHEADER, file (21,1,1) [Z.;1
    not found in a directory
%VERIFY-W-LOSTHEADER, file (22,1,1) [Z.;2
    not found in a directory
%VERIFY-W-LOSTHEADER, file (23,1,1) LOGIN.COM;163
    not found in a directory
%VERIFY-W-LOSTHEADER, file (24,1,1) MANYACL.COM;1
    not found in a directory
```

All lost files in this example are automatically moved to SYSLOST.DIR.

#### Erasing Old Home Blocks

When you initialize a volume, the initialize operation might not erase old home blocks. These are blocks that were created by previous initialize operations. If a volume that has old home blocks is damaged, you may not be able to recover the volume without erasing the blocks.

You can erase old home blocks manually by using the /HOMEBLOCKS qualifier on the ANALYZE/DISK\_STRUCTURE command as follows:

```
$ ANALYZE/DISK_STRUCTURE/REPAIR/HOMEBLOCKS
```

Note that this operation can take up to 30 minutes to complete.

## ANALYZE/DISK\_STRUCTURE

### 3.1 ANALYZE/DISK\_STRUCTURE Description

#### ANALYZE/DISK\_STRUCTURE Output

By default, the Analyze/Disk\_Structure utility directs all output to your terminal. If you prefer, you can use the /LIST qualifier to generate a file containing the following information for each file on the disk:

- File identification (FID)
- File name
- Owner
- Errors associated with the file

To generate a disk usage accounting file, use the /USAGE qualifier. The first record of the file, called the identification record, contains a summary of disk and volume characteristics. The identification record is followed by a series of summary records; one summary record is created for each file on the disk. A summary record contains the owner, size, and name of the file.

For more information about the disk usage accounting file, see Appendix E.

#### 3.1.2 Detecting Shadow Set Errors

When you enter the ANALYZE/DISK\_STRUCTURE/SHADOW command, the system checks for shadow set discrepancies—to ensure that every block on the disk is identical. A **discrepancy** is a block that should be the same on all members but is not. For example, when a WRITE is executed, it might not be written to all of the members when the ANALYZE/DISK/SHADOW processes it.

If a discrepancy is found, a clusterwide WRITE lock is taken on the shadow set, and the questionable blocks are reread. Then either one of two actions occurs:

- If a discrepancy is still present on the second read, the system displays the file name on the screen. The system also dumps the data block containing the discrepancy to the screen or to a file if you specify the /OUTPUT qualifier.
- If no discrepancy is found on the second read, the system considers the error to be a transient one (for example, if a WRITE to that disk block was in progress). The system then logs the transient error in the summary displayed on the user's terminal. However, verification that all members contained the same information is considered a success—in other words, the data on the disk is actually the same, although for a brief period it was not.

### 3.2 ANALYZE/DISK\_STRUCTURE Usage Summary

The Analyze/Disk\_Structure utility checks the readability and validity of Files-11 Structure Levels 1, 2, and 5 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]
```

## ANALYZE/DISK\_STRUCTURE

### 3.2 ANALYZE/DISK\_STRUCTURE Usage Summary

#### 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 about the Mount utility, refer to the MOUNT documentation in this manual.

### Usage Summary

Use the following command to invoke the utility:

```
ANALYZE/DISK_STRUCTURE device-name: /qualifiers
```

To terminate an ANALYZE/DISK\_STRUCTURE session, press Ctrl/C or Ctrl/Y while the utility executes. You cannot resume a session 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. You must also have write access to INDEXF.SYS to force the flushing of the caches for this file. You must also have write access to BITMAP.SYS for the same reason: to force the flushing of the caches for this file. (You need write access to QUOTA.SYS only if the volume is running disk quotas.)

For a complete explanation of file access, refer to the *HP OpenVMS Guide to 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.

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, HP recommends that you run ANALYZE/DISK\_STRUCTURE when the disk is in a quiescent state.

### 3.3 ANALYZE/DISK\_STRUCTURE Qualifiers

This section describes and provides examples of each ANALYZE/DISK\_STRUCTURE qualifier. The following table summarizes the qualifiers:

Qualifier	Description
/CONFIRM	Determines whether ANALYZE/DISK_STRUCTURE prompts you to confirm each repair
/HOMEBLOCKS	Erases damaged home blocks on an initialized volume
/LIST[=filespec]	Determines whether ANALYZE/DISK_STRUCTURE produces a listing of the index file
/LOCK_VOLUME	(Alpha and I64) Prevents updates to a volume while you are analyzing it

## ANALYZE/DISK\_STRUCTURE

### 3.3 ANALYZE/DISK\_STRUCTURE Qualifiers

Qualifier	Description
/OUTPUT[=filespec]	Specifies the output file to which ANALYZE/DISK_STRUCTURE writes the disk structure errors
/READ_CHECK	Determines whether ANALYZE/DISK_STRUCTURE performs a read check of all allocated blocks on the specified disk
/RECORD_ATTRIBUTES	Determines whether ANALYZE/DISK_STRUCTURE repairs files containing erroneous settings in the record attributes section of their associated file attribute block (FAT)
/REPAIR	Determines whether ANALYZE/DISK_STRUCTURE repairs errors that are detected in the file structure of the specified device
/SHADOW	Causes the entire contents of a shadow set or a specified range of blocks in a shadow set to be checked for discrepancies.
/STATISTICS	Produces statistical information about the volume under verification and creates a file, STATS.DAT, which contains per-volume statistics
/USAGE[=filespec]	Specifies that a disk usage accounting file should be produced, in addition to the other specified functions of ANALYZE/DISK_STRUCTURE

## **/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**

```
/CONFIRM  
/NOCONFIRM
```

### **Description**

You can use the /CONFIRM qualifier only with the /REPAIR qualifier. The default is /NOCONFIRM.

### **Example**

```
$ ANALYZE/DISK STRUCTURE DBA0:/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.

## **ANALYZE/DISK\_STRUCTURE /HOMEBLOCKS**

---

### **/HOMEBLOCKS**

Erases home blocks from a volume whose home blocks were not deleted during previous initialization operations.

### **Format**

/HOMEBLOCKS

### **Description**

You can use the /HOMEBLOCKS qualifier only with the /REPAIR qualifier. The operation can take 30 minutes to complete.

### **Example**

```
$ ANALYZE/DISK_STRUCTURE DBA0:/REPAIR/HOMEBLOCKS
```

The command in this example causes the Analyze/Disk\_Structure utility to erase home blocks on DBA0.

---

**/LIST**

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 file identifications (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.

**Example**

```
$ ANALYZE/DISK_STRUCTURE DLA2:/LIST=INDEX
$ TYPE INDEX
Listing of index file on DLA2:
31-OCT-2002 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. Because 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.

## **ANALYZE/DISK\_STRUCTURE /LOCK\_VOLUME (Alpha and I64)**

---

### **/LOCK\_VOLUME (Alpha and I64)**

Prevents updates to a volume while you are analyzing it.

#### **Format**

`/LOCK_VOLUME`

`/NOLOCK_VOLUME`

#### **Description**

`/LOCK_VOLUME` provides a way to prevent file system activity on a volume while you are using the `ANALYZE/DISK_STRUCTURE` utility on that volume. This qualifier operates the same way as `/REPAIR` does: it software write-locks the file structure while the utility is running. (The qualifier does not, however, affect any repairs on the volume.) The default is `/NOLOCK_VOLUME`.

Using this qualifier reduces the number of false error messages that might occur when you run the utility on an active volume. `/LOCK_VOLUME` stops the activity of applications that open, close, or modify files on the target volume for the period the utility is running.

---

#### **Note**

---

Be careful about using this qualifier, especially for volumes that contain active system files such as `SYSUAF` or `RIGHTSLIST`.

---

#### **Example**

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

The command in this example stops file system activity on DBA1: while `ANALYZE/DISK_STRUCTURE` is running.



## **/OUTPUT**

Specifies the output file to which the Analyze/Disk\_Structure utility is to write the disk structure errors.

### **Format**

/OUTPUT[=filespec]

/NOOUTPUT[=filespec]

### **Description**

Specifies the output file for the disk structure errors. If you omit the /OUTPUT file specification, output is directed to SYS\$OUTPUT. If /NOOUTPUT is specified, no disk structure errors are displayed. If the /CONFIRM qualifier is specified, output is forced to SYS\$OUTPUT regardless of whether this qualifier is used.

## **ANALYZE/DISK\_STRUCTURE /READ\_CHECK**

---

### **/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**

/READ\_CHECK

/NOREAD\_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.

---

## **/RECORD\_ATTRIBUTES**

Determines whether the Analyze/Disk\_Structure utility repairs files containing erroneous settings in the record attributes section of their associated file attribute block (FAT).

### **Format**

/RECORD\_ATTRIBUTES

### **Description**

You can use the /RECORD\_ATTRIBUTES qualifier with the /REPAIR qualifier. If attribute repair is enabled during the repair phase, erroneous bits are cleared from a file's record attributes. This action might not correctly set a file's record attributes as it is beyond the scope of this utility to determine their correct values.

HP recommends that system managers not perform an attribute repair; instead, they should notify the owners of the files about the inconsistencies and have the owners reset the files' attributes using the SET FILE/RECORD\_ATTRIBUTES=(record-attributes) command.

### **Example**

```
$ ANALYZE/DISK_SYS$SYSDEVICE:
%ANALDISK-I-BAD_RECATTR, file (2930,1,1) [USER]ATTRIBUTES.DAT;13
file record format: Variable
inconsistent file attributes: Bit 5
%ANALDISK-I-BAD_RECATTR, file (2931,1,1) [USER]ATTRIBUTES.DAT;14
file record format: Variable
inconsistent file attributes: FORTRAN carriage control, Bit 5
%ANALDISK-I-BAD_RECATTR, file (2932,1,1) [USER]ATTRIBUTES.DAT;15
file record format: Variable
inconsistent file attributes: Implied carriage control, Bit 5
%ANALDISK-I-BAD_RECATTR, file (2933,1,1) [USER]ATTRIBUTES.DAT;16
file record format: Variable
inconsistent file attributes: Non-spanned, Bit 5
%ANALDISK-I-BAD_RECATTR, file (2934,1,1) [USER]ATTRIBUTES.DAT;17
file record format: Variable
inconsistent file attributes: FORTRAN carriage control,
Non-spanned, Bit 5
```

## **ANALYZE/DISK\_STRUCTURE /REPAIR**

---

### **/REPAIR**

Determines whether the Analyze/Disk\_Structure utility repairs errors that are detected in the file structure of the specified device.

### **Format**

/REPAIR

/NOREPAIR

### **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.

To effectively scan a disk (/NOREPAIR), you must have read access to all files on the disk. You must also have write access to INDEXF.SYS to force the flushing of the caches for this file. You must also have write access to BITMAP.SYS for the same reason: to force the flushing of the caches for this file. (You need write access to QUOTA.SYS only if the volume is running disk quotas.)

### **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.

---

## **/SHADOW**

Examines the entire contents of a shadow set or a specified range of blocks in a shadow set for discrepancies.

### **Format**

/SHADOW

### **Parameters**

None.

### **Qualifiers**

**/BLOCKS={{(START:n, COUNT:x, END:y,) FILE\_SYSTEM, ALL}}**

Directs the system to compare only the range specified. The options are the following:

START:n	Number of the first block to be analyzed. The default is the first block.
COUNT:x	Number of blocks to be analyzed. You can use this option in combination with or instead of the END option.
END:y	Number of the last block to be analyzed. The default is the last block of the volume.
FILE_SYSTEM	Blocks currently in use by valid files on the disk. This is the default.
ALL	All blocks on the disk.

You can specify START,END,COUNT and either ALL or FILE\_SYSTEM. For example, if you specify /BLOCKS=(START,END,COUNT:100,ALL), the software checks the first 100 blocks on the disk, whether or not the file system is using them.

If you specify /BLOCKS=(START,END,COUNT:100,FILE\_SYSTEM), the software checks only those blocks that valid files on the disk are using.

### **/BRIEF**

Displays only the logical block number (LBN) if the data in a block is found to be different. Without this qualifier, if differences exist for an LBN, the hexadecimal data of that block will be displayed for each member.

### **/IGNORE**

#### **[NO]IGNORE**

Ignore “special” files that are likely to have some blocks with different data. These differences, however, are not unusual and can, therefore, be ignored.

Other special files are the following:

SWAPFILE\*.\*  
PAGEFILE\*.\*  
SYSDUMP.DMP  
SYS\$ERRLOG.DMP

IGNORE is the default.

## ANALYZE/DISK\_STRUCTURE /SHADOW

### **/OUTPUT=filename**

Output the information to the specified file.

### **/STATISTICS**

Display only the file header and footer. The best use of this qualifier is with the /OUTPUT qualifier.

## Description

When you enter the ANALYZE/DISK\_STRUCTURE/SHADOW command, the system checks for shadow set discrepancies. If a discrepancy is found, a clusterwide WRITE lock is taken on the shadow set, and the questionable blocks are reread. Then either one of two actions occurs:

- If a discrepancy is still present on the second read, the system displays the file name on the screen. The system also dumps the data block containing the discrepancy to the screen or to a file if you specify the /OUTPUT qualifier.
- If no discrepancy is found on the second read, the system considers the error to be a transient one (for example, a WRITE to that disk block was in progress).

See Section 3.1.2 for more details.

## Example

```
$ ANALYZE/DISK_STRUCTURE/SHADOW/BRIEF/BLOCKS=COUNT:1000 dsa716:
Starting to check DSA716: at 14-MAY-2002 13:42:52.43
Members of shadow set DSA716: are $252$MDAO: $252$DUA716:
and the number of blocks to be compared is 1000.
Checking LBN #0 (approx 0%)
Checking LBN #127 (approx 12%)
Checking LBN #254 (approx 25 %)
Checking LBN #381 (approx 38%)
Checking LBN #508 (approx 50%)
Checking LBN #635 (approx 63%)
Checking LBN #762 (approx 76%)
Checking LBN #889 (approx 88%)

Run statistics for DSA716: are as follows:
  Finish Time = 14-MAY-2002 13:42:52.73
  ELAPSED TIME = 0 00:00:00.29
  CPU TIME = 0:00:00.02
  BUFFERED I/O COUNT = 10
  DIRECT I/O COUNT = 16
  Failed LBNs = 0
  Transient LBN compare errors = 0
$
```

The command in this example causes ANALYZE/DISK\_STRUCTURE/SHADOW to examine the first 1000 blocks of the DSA716: virtual unit to ensure that the device \$252\$MDAO: and \$252\$DUA716: have identical data in those blocks.

---

**/STATISTICS**

Produces statistical information about the volume under verification and creates a file, STATS.DAT, which contains per-volume statistics.

**Format**

```
/STATISTICS
```

**Description**

The following information is placed in the STATS.DAT file:

- The number of ODS-2 and ODS-5 headers on the volume
- The number of special headers on ODS-5 volumes
- The distribution of file name lengths
- The distribution of extension header chain lengths
- The distribution of header identification area free space
- The distribution of header map area and ACL area free space
- The totals of header space that is in use and header space that is not in use

**Example**

```
$ ANALYZE/DISK_STRUCTURE MDA2000: /STATISTICS
```

The OpenVMS Alpha volume in this example, which is on device MDA2000:, has been converted from ODS-2 to ODS-5 using the SET VOLUME command. The STATS.DAT file created contains the following information:

```
***** Statistics for volume 001 of 001 *****  
Volume is ODS level 5.  
Volume has 00000004 ODS-2 primary headers.  
Volume has 00000003 ODS-5 primary headers.  
Volume has 00000000 ODS-5 -1 segnum headers.  
  
00000001 filenames of length 009 bytes.  
00000002 filenames of length 011 bytes.  
00000001 filenames of length 013 bytes.  
00000002 filenames of length 015 bytes.  
00000001 filenames of length 073 bytes.  
  
00000007 extension header chains of length 00000.  
  
00000001 ODS-2 headers have 071 ident area free bytes.  
00000001 ODS-2 headers have 073 ident area free bytes.  
00000001 ODS-2 headers have 075 ident area free bytes.  
00000001 ODS-2 headers have 077 ident area free bytes.  
  
Total ODS-2 ident area free bytes is 00000296.  
  
00000001 ODS-5 headers have 001 ident area free bytes.  
00000001 ODS-5 headers have 029 ident area free bytes.  
00000001 ODS-5 headers have 033 ident area free bytes.  
  
Total ODS-5 ident area free bytes is 00000063.
```

## **ANALYZE/DISK\_STRUCTURE /STATISTICS**

```
00000001 headers have 277 free bytes in total.  
00000001 headers have 335 free bytes in total.  
00000001 headers have 339 free bytes in total.  
00000001 headers have 377 free bytes in total.  
00000001 headers have 379 free bytes in total.  
00000001 headers have 381 free bytes in total.  
00000001 headers have 383 free bytes in total.  
  
Total header area in bytes is 00003584.  
Total header area free bytes is 00002791.  
Total header area used bytes is 00000793.
```



## **/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;1
```

```
Total of 1 file.
```

The first command in this example causes ANALYZE/DISK\_STRUCTURE to produce a disk usage accounting file. Because 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 files with a file name of usage in the current directory. The OpenVMS Alpha device in this example, MDA2000:, has been converted from ODS-2 to ODS-5 using the SET VOLUME command.



---

**Audit Analysis Utility**



## 4.1 ANALYZE/AUDIT Description

The Audit Analysis utility (ANALYZE/AUDIT) is a system management tool that enables system managers or site security administrators to produce reports from security audit log files.

The OpenVMS operating system automatically audits a limited number of events, such as changes to the authorization database and use of the SET AUDIT command. Depending on your site's requirements, you may want to enable other forms of reporting. However, collecting security audit messages is useful only if you develop and implement a procedure to periodically review the audit log file for suspicious activity. Use ANALYZE/AUDIT to examine the data in security audit log files or security archive files.

The ANALYZE/AUDIT command's different qualifiers allow you to specify the type of information the utility extracts from the security audit log file. The utility can produce an audit report in a variety of formats and direct a report to a file or a terminal.

A description of the format of the auditing messages written to the security auditing file appears in Appendix F.

In a mixed-version cluster, an audit log file contains entries from systems running different versions of the operating system. To analyze the log file, you must invoke the Audit Analysis utility (ANALYZE/AUDIT) from a node running Version 6.1 or later.

For information about how to generate audit messages records and how to use ANALYZE/AUDIT, refer to the *HP OpenVMS Guide to System Security*.

## 4.2 ANALYZE/AUDIT Usage Summary

The Audit Analysis utility (ANALYZE/AUDIT) processes event messages in security audit log files to produce reports of security-related events on the system.

### Format

```
ANALYZE/AUDIT [file-spec[,...]]
```

#### Parameter

##### **file-spec[,...]**

Specifies one or more security audit log files as input to ANALYZE/AUDIT. If you specify more than one file name, separate the names with commas.

If you omit the **file-spec** parameter, the utility searches for the default audit log file SECURITY.AUDIT\$JOURNAL.

The default audit log file is created in the SYS\$COMMON:[SYSMGR] directory. To use the file, specify SYS\$MANAGER on the ANALYZE/AUDIT command line. If you do not specify a directory, the utility searches for the file in the current directory.

You can include wildcard characters, such as the asterisk (\*) or percent sign (%), in the file specification.

The audit log file can be located in any directory. To display the current location, use the DCL command SHOW AUDIT/ALL.

## ANALYZE/AUDIT

### 4.2 ANALYZE/AUDIT Usage Summary

#### Description

Use the DCL command ANALYZE/AUDIT to analyze security audit log files or security archive files. An ANALYZE/AUDIT command line can specify the name of one or more log files, as follows:

```
ANALYZE/AUDIT [file-spec,...]
```

You can also use the ANALYZE/AUDIT command to extract security event messages from security archive files or from binary files (created with previous ANALYZE/AUDIT commands).

Each ANALYZE/AUDIT request runs until the log file is completely processed. You can interrupt the processing to modify the display or to change position in the report if you activate command mode by pressing Ctrl/C. 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 file specification as an argument to the /OUTPUT qualifier. By default, the output is directed to SYS\$OUTPUT.

Use of ANALYZE/AUDIT requires no special privileges other than access to the files specified in the command line.

#### 4.3 ANALYZE/AUDIT Qualifiers

This section describes ANALYZE/AUDIT and provides examples of each qualifier. The following table summarizes the ANALYZE/AUDIT qualifiers:

Qualifier	Description
/BEFORE	Controls whether records dated earlier than the specified time are selected
/BINARY	Controls whether output is a binary file
/BRIEF	Controls whether a brief, single-line record format is used in ASCII displays
/EVENT_TYPE	Selects the classes of events to be extracted from the security log file
/FULL	Controls whether a full format is used in ASCII displays
/IGNORE	Excludes records from the report that match the specified criteria
/INTERACTIVE	Controls whether interactive command mode is enabled when ANALYZE/AUDIT is invoked
/OUTPUT	Specifies where to direct output from ANALYZE/AUDIT
/PAUSE	Specifies the length of time each record is displayed in a full format display
/SELECT	Specifies the criteria for selecting records
/SINCE	Indicates that the utility must operate on records dated with the specified time or after the specified time
/SUMMARY	Specifies that a summary of the selected records be produced after all records are processed

## **/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 *OpenVMS User's Manual*.

### **Description**

By default, all records in the security audit log file may be examined. You must specify **/BEFORE** to exclude records created after a specific point in time.

### **Examples**

1. \$ ANALYZE/AUDIT /BEFORE=25-NOV-2002 -  
\_ \$ SYS\$MANAGER:SECURITY.AUDIT\$JOURNAL

The command in this example selects all records dated earlier than November 25, 2002.

2. \$ ANALYZE/AUDIT /BEFORE=14:00/SINCE=12:00 -  
\_ \$ SYS\$MANAGER:SECURITY.AUDIT\$JOURNAL

The command in this example selects all records generated between noon and 2 P.M. today.

## ANALYZE/AUDIT /BINARY

---

### /BINARY

Controls whether output is a binary file.

### Format

/BINARY

/NOBINARY

### Keywords

None.

### Description

When you use /BINARY, the output file you specify with 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/SINCE=TODAY/OUTPUT=25OCT00.AUDIT -  
_ $ SYS$MANAGER:SECURITY.AUDIT$JOURNAL
```

The command in this example selects all audit records generated today and writes the records in binary format to 25OCT00.AUDIT.



## **/BRIEF**

Controls whether a brief, single-line record format is used in ASCII displays.

### **Format**

/BRIEF (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 audit event record displayed.

The /BINARY, /BRIEF, and /FULL qualifiers cannot be used in combination.

### **Example**

```
$ ANALYZE/AUDIT /OUTPUT=AUDIT.LIS -  
_ $ SYS$MANAGER:SECURITY.AUDIT$JOURNAL
```

The command in this example produces an ASCII file in brief format by default. The report is written to the AUDIT.LIS file.

## ANALYZE/AUDIT /EVENT\_TYPE

---

### /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 utility includes all enabled event classes in the report.

#### 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:

[NO]ACCESS	Access to an object, such as a file
[NO]ALL	All event types
[NO]AUDIT	Use of the SET AUDIT command
[NO]AUTHORIZATION	Change to the authorization database (SYSUAF.DAT, RIGHTSLLIST.DAT, NETPROXY.DAT, or NET\$PROXY.DAT)
[NO]BREAKIN	Break-in detection
[NO]CONNECTION	Establishment of a network connection through the System Management utility (SYSMAN), DECwindows, or interprocess communication (IPC) software or DECnet Phase IV (VAX only)
[NO]CREATE	Creation of an object
[NO]DEACCESS	Completion of access to an object
[NO]DELETE	Deletion of an object
[NO]INSTALL	Modification of the known file list with the Install utility (INSTALL)
[NO]LOGFAIL	Unsuccessful login attempt
[NO]LOGIN	Successful login
[NO]LOGOUT	Successful logout
[NO]MOUNT	Execution of DCL commands MOUNT or DISMOUNT
[NO]NCP	Modification of the DECnet network configuration databases
[NO]NETPROXY	Modification of the network proxy authorization file (NETPROXY.DAT or NET\$PROXY.DAT)
[NO]PRIVILEGE	Privilege auditing
[NO]PROCESS	Use of one or more of the process control system services: \$CREPRC, \$DELPRC, \$SCHDWK, \$CANWAK, \$WAKE, \$SUSPND, \$RESUME, \$GRANTID, \$REVOKID, \$GETJPI, \$FORCEX, \$SETPRI

[NO]RIGHTSDB	Modification of the rights database (RIGHTSLIST.DAT)
[NO]SYSGEN	Modification of system parameters through the System Generation utility (SYSGEN) or AUTOGEN
[NO]SYSUAF	Modification of the system user authorization file (SYSUAF.DAT)
[NO]TIME	Change in system or cluster time

Specifying the negated form of an event class (for example, NOLOGFAIL) excludes the specified event class from the audit report.

## Examples

1. `$ ANALYZE/AUDIT/EVENT_TYPE=LOGFAIL -  
_ $ SYS$MANAGER:SECURITY.AUDIT$JOURNAL`

The command in this example extracts all records of unsuccessful login attempts, which match the LOGFAIL class, and compiles a brief report.

2. `$ ANALYZE/AUDIT/EVENT_TYPE=(NOLOGIN,NOLOGOUT) -  
_ $ SYS$MANAGER:SECURITY.AUDIT$JOURNAL`

The command in this example builds a report in brief format of all audit records except those in the LOGIN and LOGOUT event classes.

## **ANALYZE/AUDIT /FULL**

---

### **/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` (or enter command mode by pressing `Ctrl/C`) 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 -  
_ $ SYS$MANAGER:SECURITY.AUDIT$JOURNAL
```

The command in this example displays the full contents of each selected record.

## **/IGNORE**

Excludes records from the report that match the specified criteria.

### **Format**

/IGNORE=criteria[,...]

### **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 report. When more than one keyword from the list of possible exclusion criteria are specified, records that meet any of these criteria are excluded by default.

### **Examples**

1. \$ ANALYZE/AUDIT/IGNORE=(SYSTEM=NAME=WIPER,USERNAME=MILANT) -  
\_ \$ SYS\$MANAGER:SECURITY.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).

2. \$ ANALYZE/AUDIT/IGNORE=SUBTYPE=(DIALUP,REMOTE)

The command in this example excludes dialup and remote processes.

## ANALYZE/AUDIT /INTERACTIVE

---

### /INTERACTIVE

Controls whether interactive command mode is enabled when ANALYZE/AUDIT is invoked.

#### Format

/INTERACTIVE (default)  
/NOINTERACTIVE

#### Keywords

None.

#### Description

Interactive command mode, which is enabled by default, allows you to interrupt the audit report being displayed on the terminal and to enter commands either to modify the criteria used to select records for the report or to reposition the display.

To interrupt a full or brief audit report, press Ctrl/C and enter commands at the COMMAND> prompt. Once in command mode, the utility displays the current record in full format. Note that the record might not match the selection or exclusion criteria specified in the previous ANALYZE/AUDIT command.

The NEXT RECORD command is the default when you enter command mode. When ANALYZE/AUDIT reaches the end of the log file, it prompts for the next command. To verify the current log file name and your position within the file, press Ctrl/T.

Enter the CONTINUE command to leave interactive command mode and to resume display of the audit report. Enter the EXIT command to terminate the session. See the ANALYZE/AUDIT Commands section for a description of each interactive command.

To disable interactive mode, specify /NOINTERACTIVE. In this mode, the utility displays audit records one at a time and prompts you to advance the display by pressing the Return key.

#### Examples

1. \$ ANALYZE/AUDIT/FULL -  
\_ \$ SYS\$MANAGER:SECURITY.AUDIT\$JOURNAL

The command in this example produces a full format display of the selected records. New records are displayed every 3 seconds. (See the /PAUSE qualifier description to find how to modify the duration of each record display.) Press Ctrl/C to interrupt the display and to enter interactive commands.

2. \$ ANALYZE/AUDIT/FULL/NOINTERACTIVE -  
\_ \$ SYS\$MANAGER:SECURITY.AUDIT\$JOURNAL

The command in this example invokes the utility in noninteractive mode. It displays the first record selected and prompts you to press the Return key to display each additional selected record. Control returns to the DCL command level when all selected records have been displayed.

## **/OUTPUT**

Specifies where to direct output from ANALYZE/AUDIT. If you omit the qualifier, the report is sent 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 utility uses the current device and directory specification. 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 written to the file AUDIT.AUDIT\$JOURNAL.

### **Example**

```
$ ANALYZE/AUDIT /BINARY/OUTPUT=BIN122588.DAT -  
_ $ SYS$MANAGER:SECURITY.AUDIT$JOURNAL
```

The command in this example selects audit records from the system audit log file and writes them to the binary file BIN122588.DAT.

## ANALYZE/AUDIT /PAUSE

---

### /PAUSE

Specifies the length of time each record is displayed in a full-format display.

### 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 record. By default, the utility displays a record for 3 seconds.

### Description

The /PAUSE qualifier can be used only 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$MANAGER:SECURITY.AUDIT$JOURNAL
```

The command in this example displays a selected record in full format every second. You can interrupt the display and enter interactive commands at any time by pressing Ctrl/C. (See the ANALYZE/AUDIT Commands section for more information.)



---

**/SELECT**

Specifies the criteria for selecting records from the audit log file. Refer to the *HP OpenVMS Guide to System Security* for a description of how to generate audit records.

**Format**

/SELECT=criteria[,...]

/NOSELECT

**Keyword****criteria[,...]**

Specifies the criteria for selecting records. For each specified criterion, ANALYZE/AUDIT has two selection requirements:

- The packet corresponding to the criterion must be present in the record.
- One of the specified values must match the value in that packet.

For example, if you specify (USER=(PUTNAM,WU),SYSTEM=DBASE) as the criteria, ANALYZE/AUDIT selects an event record containing the SYSTEM=DBASE packet and a USER packet with either the PUTNAM value or the WU value.

If you omit the /SELECT qualifier, all event records selected through the /EVENT\_TYPE qualifier are extracted from the audit log file and included in the report.

You can specify any of the following criteria:

**ACCESS=(type,...)**

Specifies the type of object access upon which the selection is based. Access is object-specific and includes the following types:

Associate	Execute	Read
Control	Lock	Submit
Create	Logical	Use
Delete	Manage	Write
	Physical	

The *HP OpenVMS Guide to System Security* describes each of these types.

**ACCOUNT=(name,...)**

Specifies the account name upon which selection is based. You can use wildcards, such as an asterisk (\*) or percent sign (%), to represent all or part of the name.

**ALARM\_NAME=(alarm-name,...)**

Specifies the alarm journal name on which selection is based. You can use wildcards to represent all or part of the alarm name.

**ASSOCIATION\_NAME=(IPC-name,...)**

Specifies the name of the interprocess communication (IPC) association.

## **ANALYZE/AUDIT /SELECT**

### **AUDIT\_NAME=(journal-name,...)**

Specifies the audit journal name on which selection is based. You can use wildcards to represent all or part of the audit journal name.

### **COMMAND\_LINE=(command,...)**

Specifies the command line that the user entered.

### **CONNECTION\_IDENTIFICATION=(IPC-name,...)**

Specifies the name for the interprocess communication (IPC) connection.

### **DECNET\_LINK\_IDENTIFICATION=(value,...)**

Specifies the number of the DECnet logical link.

### **DECNET\_OBJECT\_NAME=(object-name,...)**

Specifies the name of the DECnet object.

### **DECNET\_OBJECT\_NUMBER=(value,...)**

Specifies the number of the DECnet object.

### **DEFAULT\_USERNAME=(username,...)**

Specifies the default local user name for incoming network proxy requests.

### **DEVICE\_NAME=(device-name,...)**

Specifies the name of a device in audit records that have a DEVICE\_NAME packet. Note that this does not select the device name when it occurs in other packet types, such as in a file name or in the TARGET\_DEVICE\_NAME packet.

### **DIRECTORY\_ENTRY=(directory,...)**

Specifies the directory entry associated with file system operation.

### **DIRECTORY\_NAME=(directory,...)**

Specifies the name of the directory file.

### **DISMOUNT\_FLAGS=(flag-name,...)**

Identifies the names of the volume dismounting flags to be used in selecting records. Specify one or more of the following flag names: Abort, Cluster, Nounload, and Unit.

### **EVENT\_CLUSTER\_NAME=(event-flag-cluster-name,...)**

Specifies the name of the event flag cluster.

### **FACILITY=(facility-name,...)**

Specifies that only events audited by the named facility be selected. Provide a name or a number but, in either case, the facility has to be defined through the logical AUDSERV\$FACILITY\_NAME as a decimal number; the system uses the number 0.

### **FIELD\_NAME=(field-name,...)**

Specifies the name of the field that was modified. ANALYZE/AUDIT uses the FIELD\_NAME criterion with packets containing the original data and the new data (specified by the NEW\_DATA criterion).

A FIELD\_NAME is a character string that describes the content of the field. A search for "NEW:" in a full audit report will display records that contain the FIELD\_NAME values that can be specified for this option. Examples of FIELD\_NAME values are Account, Default Directory, Flags, and Password Date.

For sensitive information, see SENSITIVE\_FIELD\_NAME.

**FILE\_NAME=(file-name)**

Specifies the name of the file that caused the audit. Describes audit records for the specified file by using a slightly different display format than is provided by the /OBJECT=NAME=object-name keyword.

**FILE\_IDENTIFICATION=(identification-value)**

Specifies the value of the file's identification. To calculate the value, start with the value listed for File ID when you use the FILE\_NAME keyword. For example, the display lists the File ID as:

File ID: (3024,5,0)

Use the following formula to calculate the value:

$(0 * 65536) + 5 * 65536 + 3024 = 330704$

**FLAGS=(flag-name,...)**

Identifies the names of the audit event flags associated with the audited event. These names should be used in selecting records. Specify one or more of the following flags: ACL, Alarm, Audit, Flush, Foreign, Internal, and Mandatory. (For a description of these flags, see Table F-3.)

**HOLDER=keyword(...)**

Specifies the characteristics of the identifier holder to be used when selecting event records. Choose from the following keywords:

NAME=username	Specifies the name of the holder. You can represent all or part of the name with a wildcard.
OWNER=uic	Specifies the user identification code (UIC) of the holder.

**IDENTIFIER=keyword(...)**

Identifies which attributes of an identifier should be used when selecting event records. Choose from the following keywords:

ATTRIBUTES=name	Specifies the name of the particular attribute. Valid attribute names are as follows: Dynamic, Holder_Hidden, Name_Hidden, NoAccess, Resource, and Subsystem.
NAME=identifier	Specifies the original name of the identifier. You can represent all or part of the name with a wildcard.
NEW_NAME=identifier	Specifies the new name of the identifier. You can represent all or part of the name with a wildcard.
NEW_ATTRIBUTES=name	Specifies the name of the new attribute. Valid attribute names are Dynamic, Holder_Hidden, Name_Hidden, NoAccess, Resource, and Subsystem.
VALUE=value	Specifies the original value of the identifier.
NEW_VALUE=value	Specifies the new value of the identifier.

**IDENTIFIERS\_MISSING=(identifier,...)**

Specifies the identifiers missing in a failure to access an object.

## ANALYZE/AUDIT /SELECT

### **IDENTIFIERS\_USED=(identifier,...)**

Specifies the identifiers used to gain access to an object. An event record matches if the specified list is a subset of the identifiers recorded in the event record.

### **IMAGE\_NAME=(image-name,...)**

Identifies the name of the image to be used when selecting event records. You can represent all or part of the image name with a wildcard.

### **INSTALL=keyword(...)**

Specifies that installation event packets are to be considered when selecting event records. Choose from the following keywords:

**FILE=filename** Specifies the name of the installed file. You can represent all or part of the name with a wildcard.

Note that on Alpha systems prior to Version 6.1 and on VAX systems prior to Version 6.0, audit log files record the installed file name within an object name packet. To select the installed file, you must use the expression **OBJECT=(NAME=object-name)** instead of **FILE=filename**.

**FLAGS=flag-name** Specifies the names of the flags, which correspond to qualifiers of the Install utility (INSTALL); for example, OPEN corresponds to /OPEN.

**PRIVILEGES=privilege-name** Specifies the names of the privileges with which the file was installed.

### **LNM\_PARENT\_NAME=(table-name,...)**

Specifies the name of the parent logical name table.

### **LNM\_TABLE\_NAME=(table-name,...)**

Specifies the name of the logical name table.

### **LOCAL=(characteristic,...)**

Specifies the characteristics of the local (proxy) account to be used when selecting event records. The following characteristic is supported:

**USERNAME=username** Specifies the name of the local account. You can represent all or part of the name with a wildcard.

### **LOGICAL\_NAME=(logical-name,...)**

Specifies the logical name of the mounted (or dismounted) volume upon which selection is based. You can represent all or part of the logical name with a wildcard.

### **MAILBOX\_UNIT=(number,...)**

Specifies the number of the mailbox unit.

### **MOUNT\_FLAGS=(flag-name,...)**

Specifies the names of the volume mounting flags upon which selection is based. Possible flag names include the following names:

**CACHE=(NONE,WRITETHROUGH)**

CDROM  
 CLUSTER  
 COMPACTION  
 DATACHECK=(READ,WRITE)  
 DSI  
 FOREIGN  
 GROUP  
 INCLUDE  
 INITIALIZATION=(ALLOCATE,CONTINUATION)  
 MESSAGE  
 NOASSIST  
 NOAUTOMATIC  
 NOCOMPACTION  
 NOCOPY  
 NOHDR3  
 NOJOURNAL  
 NOLABEL  
 NOMOUNT\_VERIFICATION  
 NOQUOTA  
 NOREBUILD  
 NOUNLOAD  
 NOWRITE

OVERRIDE=(options[,...]) {
 

ACCESSIBILITY
EXPIRATION
IDENTIFICATION
LIMITED_SEARCH
LOCK
NO_FORCED_ERROR
OWNER_IDENTIFIER
SECURITY
SETID

 }

POOL  
 QUOTA  
 SHARE  
 SUBSYSTEM  
 SYSTEM  
 TAPE\_DATA\_WRITE  
 XAR

The names NOLABEL and FOREIGN each point to the FOREIGN flag. The reason for this is that the MOUNT/NOLABEL and MOUNT/FOREIGN commands each set the FOREIGN flag. Therefore, if you used MOUNT/NOLABEL, and you use ANALYZE/AUDIT/SELECT/MOUNT\_FLAGS=NOLABEL, the audit record will display the FOREIGN flag.

**NEW\_DATA=(value,...)**

Specifies the value to use after the event occurs. Use this criterion with the FIELD\_NAME criterion.

For sensitive information, see SENSITIVE\_NEW\_DATA.

**NEW\_IMAGE\_NAME=(image-name,...)**

Specifies the name of the image to be activated in the newly created process, as supplied to the \$CREPRC system service.

## ANALYZE/AUDIT /SELECT

### **NEW\_OWNER=(uic,...)**

Specifies the user identification code (UIC) to be assigned to the created process, as supplied to the \$CREPRC system service.

### **OBJECT=keyword(...)**

Specifies which characteristics of an object should be used when selecting event records. Choose any of the following keywords:

**CLASS=class-name** Specifies the general object class as one of the following classes:

- Capability
- Device
- Event\_cluster
- File
- Group\_global\_section
- Logical\_name\_table
- Queue
- Resource\_domain
- Security\_class
- System\_global\_section
- Volume

You must enter the full class name (for example, CLASS=logical\_name\_table) or use wildcard characters to supply a portion of the class name (for example, CLASS=log\*).

**NAME=object-name**

Specifies the name of the object. You can represent all or part of the name with a wildcard. If you do not use a wildcard, specify the full object name (for example, BOSTON\$DUA0:[RWOODS]MEMO.MEM;1).

**OWNER=value**

Specifies the UIC or general identifier of the object.

**TYPE=type**

Specifies the general object class (type of object). The available classes are as follows:

- Capability
- Device
- File
- Group\_global\_section
- Logical\_name\_table
- Queue
- System\_global\_section

The CLASS keyword supersedes the TYPE keyword. However, TYPE is required to select audit records in files created prior to OpenVMS Alpha Version 6.1 and OpenVMS VAX Version 6.0.

### **PARENT=keyword(...)**

Specifies which characteristics of the parent process are used when selecting event records generated by a subprocess. Choose from the following keywords:

**IDENTIFICATION=value** Specifies the process identifier (PID) of the parent process.

**NAME=process-name** Specifies the name of the parent process. You can represent all or part of the name with a wildcard.

**OWNER=value** Specifies the owner (identifier value) of the parent process.

**USERNAME=username** Specifies the user name of the parent process. You can represent all or part of the name with a wildcard.

**PASSWORD=(password,...)**  
Specifies the password used when the system detected a break-in attempt.

**PRIVILEGES\_MISSING=(privilege-name,...)**  
Specifies privileges the caller needed to perform the operation successfully. Specify any of the system privileges, as described in the *HP OpenVMS Guide to System Security*.

**PRIVILEGES\_USED=(privilege-name,...)**  
Specifies the privileges of the process to be used when selecting event records. Specify any of the system privileges, as described in the *HP OpenVMS Guide to System Security*. Also include the STATUS keyword in the selection criteria so the report can demonstrate whether the privilege was involved in a successful or an unsuccessful operation.

**PROCESS=(characteristic,...)**  
Specifies the characteristics of the process to be used when selecting event records. Choose from the following characteristics:

**IDENTIFICATION=value** Specifies the PID of the process.

**NAME=process-name** Specifies the name of the process. You can represent all or part of the name with a wildcard.

**REMOTE=keyword(...)**  
Specifies that some characteristic of the network request is to be used when selecting event records. Choose from the following keywords:

**ASSOCIATION\_NAME=IPC-name** Specifies the interprocess communication (IPC) association name.

**LINK\_IDENTIFICATION=value** Specifies the number of the DECnet logical link.

## ANALYZE/AUDIT /SELECT

IDENTIFICATION=value	Specifies the DECnet node address.
NODENAME=node-name	Specifies the DECnet node name. You can represent all or part of the name with a wildcard.
USERNAME=username	Specifies the remote user name. You can represent all or part of the remote user name with a wildcard.

### **REQUEST\_NUMBER=(value,...)**

Specifies the request number associated with the DCL command REQUEST/REPLY.

### **SECTION\_NAME=(global-section-name,...)**

Specifies the name of the global section.

### **SENSITIVE\_FIELD\_NAME=(field-name,...)**

Specifies the name of the field that was modified. ANALYZE/AUDIT uses the SENSITIVE\_FIELD\_NAME criterion, such as PASSWORD, with packets containing the original data and the new data (specified by the SENSITIVE\_NEW\_DATA criterion).

### **SENSITIVE\_NEW\_DATA=(value,...)**

Specifies the value to use after the event occurs. Use this criterion with the SENSITIVE\_FIELD\_NAME criterion.

### **SNAPSHOT\_BOOTFILE=(filename,...)**

Specifies the name of the file containing a snapshot of the system.

### **SNAPSHOT\_SAVE\_FILENAME=(filename,...)**

Specifies the name of the system snapshot file for a save operation that is in progress.

### **STATUS=(type,...)**

Specifies the type of success status to be used when selecting event records. Choose from the following status types:

SUCCESSFUL	Specifies any success status.
FAILURE	Specifies any failure status.
CODE=(value)	Specifies a specific completion status.

Note that if you specify CODE more than once, only the last value is matched.

### **SUBJECT\_OWNER=(uic,...)**

Specifies the owner (UIC) of the process causing the event.

### **SUBTYPE=(subtype,...)**

Specifies that the criteria be limited to the value or values specified as a subtype.

Refer to Table F-2 for valid subtype values.



**SYSTEM=keyword(...)**

Specifies the characteristics of the system to be used when selecting event records. Choose from the following keywords:

**IDENTIFICATION=value** Specifies the numeric identification of the system.

**NAME=nodename** Specifies the node name of the system.

**SYSTEM\_SERVICE\_NAME=(service-name,...)**

Specifies the name of the system service associated with the event.

**TARGET\_DEVICE\_NAME=(device-name,...)**

Specifies the target device name used by a process control system service.

**TARGET\_PROCESS\_IDENTIFICATION=(value,...)**

Specifies the target process identifier (PID) used by a process control system service.

**TARGET\_PROCESS\_NAME=(process-name,...)**

Specifies the target process name used by a process control system service.

**TARGET\_PROCESS\_OWNER=(uic,...)**

Specifies the target process owner (UIC) used by a process control system service.

**TARGET\_USERNAME=(username,...)**

Specifies the target user name used by a process control system service.

**TERMINAL=(device-name,...)**

Specifies the name of the terminal to be used when selecting event records. You can represent all or part of the terminal name with a wildcard.

**TRANSPORT\_NAME=(transport-name,...)**

Specifies the name of the transport: interprocess communication (IPC) or System Management Integrator (SMI), which handles requests from the System Management utility.

On VAX systems, it also can specify the DECnet transport name (NSP).

**USERNAME=(username,...)**

Specifies the user name to be used when selecting event records. You can represent all or part of the user name with a wildcard.

**VOLUME\_NAME=(volume-name,...)**

Specifies the name of the mounted (or dismounted) volume to be used when selecting event records. You can represent all or part of the volume name with a wildcard.

**VOLUME\_SET\_NAME=(volume-set-name,...)**

Specifies the name of the mounted (or dismounted) volume set to be used when selecting event records. You can represent all or part of the volume set name with a wildcard.

## ANALYZE/AUDIT /SELECT

### Examples

1. `$ ANALYZE/AUDIT /FULL/SELECT=USERNAME=JOHNSON -`  
`_ $ SYS$MANAGER:SECURITY.AUDIT$JOURNAL`

The command in this example selects all records written to the security audit log file that were generated by user JOHNSON.

2. `$ ANALYZE/AUDIT/FULL/SELECT=PRIVILEGES_USED=(SYSPRV,-`  
`_ $ BYPASS) SYS$MANAGER:SECURITY.AUDIT$JOURNAL`

The command in this example selects all records written to the security audit log file that were generated by events through the use of either SYSPRV or BYPASS privilege.

3. `$ ANALYZE/AUDIT/FULL/EVENT=SYSUAF/SELECT= -`  
`_ $ IMAGE="( *[SYS*SYSEXE]SETP0.EXE", *[SYS*SYSEXE]LOGINOUT.EXE ) -`  
`_ $ SYS$MANAGER:SECURITY`

The command in this example selects all records that involve password changes written to the security audit log file.

The following example is a command procedure that you could run at midnight to select all SYSUAF, AUDIT, and BREAKIN events (excluding password changes) and mail the result to the system manager:

```
#! DAILY_AUDIT.COM
$
$ mail_list = "SYSTEM"
$ audsrv$noselect = %X003080A0
$ audit_events = "SYSUAF,BREAKIN,AUDIT"
$
$ analyze /audit /full -
/event=('audit_events') -
/output=audit.tmp -
/ignore=image="( *[SYS*SYSEXE]SETP0.EXE", *[SYS*SYSEXE]LOGINOUT.EXE ) -
sys$manager:SECURITY.AUDIT$JOURNAL
$
$ status = $status
$ if (status.and.%XFFFFFFF) .eq. audsrv$noselect then goto no_records
$ if .not. status then goto error_analyze
$ if f$file("audit.tmp","eof") .eq. 0 then goto no_records
$ mail /subject="'audit_events' listing from '$time()' " -
audit.tmp 'mail_list'
$ goto new_log
$
$ no_records:
$ mail /subject="No interesting security events" nl: 'mail_list'
$
$ new_log:
$ if f$search("audit.tmp") .nes. "" then delete audit.tmp;*
$ set audit /server=new_log
$ rename sys$manager:SECURITY.AUDIT$JOURNAL;-1 -
sys$common:[sysmgr]'f$element(0," ",f$edit(f$time(),"TRIM"))'
$ exit
$
$ error_analyze:
$ mail/subj="Error analyzing auditing information" nl: 'mail_list'
$ exit
```

---

## **/SINCE**

Indicates the utility must operate on records dated with the specified time or after the specified time.

### **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, a delta time, or a combination of the two. Observe the syntax rules for date and time described in the *OpenVMS User's Manual*.

If you specify /SINCE without the time, the utility uses the beginning of the current day.

### **Examples**

1. \$ ANALYZE/AUDIT /SINCE=25-NOV-2002 -  
\_ \$ SYS\$MANAGER:SECURITY.AUDIT\$JOURNAL

The command in this example selects records dated later than November 25, 2002.

2. \$ ANALYZE/AUDIT /SINCE=25-NOV-2002:15:00 -  
\_ \$ SYS\$MANAGER:SECURITY.AUDIT\$JOURNAL

The command in this example selects records written after 3 P.M. on November 25, 2002.

## **ANALYZE/AUDIT /SUMMARY**

---

### **/SUMMARY**

Specifies that a summary of the selected records be produced after all records are processed.

Note that the `/SUMMARY` qualifier code is executed after the Audit Analyzer is finished, that is, after all the records to be analyzed have been collected and processed. When you specify the `/INTERACTIVE` qualifier (which is the default), the Audit Analyzer never reaches the finished state because `/INTERACTIVE` prompts you repeatedly to enter another command (which might result in a new set of records to be analyzed). To use the `/SUMMARY` qualifier, you must also specify `/NOINTERACTIVE`, which ensures that the Audit Analyzer reaches the finished state that allows the `SUMMARY` code to be executed and to display the proper information. In a future version of OpenVMS, the Audit Analyzer will return an error when `/SUMMARY` and `/INTERACTIVE` are specified together.

You can use the `/SUMMARY` qualifier alone or in combination with the `/BRIEF`, the `/BINARY`, or the `/FULL` qualifier.

### **Format**

`/SUMMARY=presentation`

`/NOSUMMARY`

### **Keyword**

#### **presentation**

Specifies the presentation of the summary. If you do not specify a presentation criterion, `ANALYZE/AUDIT` summarizes the number of audits.

You can specify either of the following presentations:

#### **COUNT**

Lists the total number of audit messages for each class of security event that have been extracted from the security audit log file. This is the default.

#### **PLOT**

Displays a plot showing the class of the audit event, the time of day when the audit was generated, and the name of the system where the audit was generated.

### **Examples**

1. `$ ANALYZE/AUDIT/SUMMARY SYS$MANAGER:SECURITY.AUDIT$JOURNAL`

The command in this example generates a summary report of all records processed.

## ANALYZE/AUDIT /SUMMARY

Total records read:	9701	Records selected:	9701
Record buffer size:	1031	Object creates:	1278
Successful logins:	542	Object accesses:	3761
Successful logouts:	531	Object deaccesses:	2901
Login failures:	35	Object deletes:	301
Breakin attempts:	2	Volume (dis)mounts:	50
System UAF changes:	10	System time changes:	0
Rights db changes:	8	Server messages:	0
Netproxy changes:	5	Connections:	0
Audit changes:	7	Process control audits:	0
Installed db changes:	50	Privilege audits:	91
Sysgen changes:	9		
NCP command lines:	120		

2. \$ ANALYZE/AUDIT/FULL/EVENT\_TYPE=(BREAKIN,LOGFAIL)/SUMMARY -  
\_ \$ SYS\$MANAGER:SECURITY.AUDIT\$JOURNAL

The command in this example generates a full format listing of all logged audit messages that match the break-in or log failure event classes. A summary report is included at the end of the listing.

3. \$ ANALYZE/AUDIT/FULL/EVENT\_TYPE=(BREAKIN,LOGFAIL)/SUMMARY=PLOT -  
\_ \$ SYS\$MANAGER:SECURITY.AUDIT\$JOURNAL

This command generates a histogram that you can display on a character-cell terminal.

```

Date                                     Node: ATHENS
|0000|0100|0200|0300|0400|0500|0600|0700|0800|0900|1000|1100|1200|1300|1400|1500|1600|1700|1800|1900|2000|2100|2200|2300|
6JAN   |      |      |129 928 772 757 397 27 56 440 891 148 561 253 321 86 91 117 71 37          |5 21 5
7JAN   | 6   89 16                                     |
11JAN  |                                     |117 184 93 141 68
13JAN  |                                     |        64 630
14JAN  |                                     |108 239 354 659 16 105 161 160 165 147
15JAN  |138 128 14                                     |180 176 63 27 28 15 43
16JAN  | 12 17 25 17 168 182 200 167 156 104 97 212 208 219 154 241 199 211 43 108 88 5 5
17JAN  |109 70 11                                     |
18JAN  |                                     |522 4 15 11 15 9 5
19JAN  |      5 9 17          5 27 46 134 209 466 417 97 223 240 660 789 246 494 273 213 333 232 273
20JAN  |386 701 23                                     |
21JAN  |      37 35 28 32 47 38 70 81 90 115 155

```

ZK-5912A-AI

## **ANALYZE/AUDIT**

### **4.4 ANALYZE/AUDIT Commands**

---

#### **4.4 ANALYZE/AUDIT Commands**

This section describes the interactive commands available with the Audit Analysis utility (ANALYZE/AUDIT). The qualifiers for this section follow the standard rules of DCL grammar.

The utility runs interactively by default; you disable the feature with the /NOINTERACTIVE qualifier to the ANALYZE/AUDIT command. To enter interactive commands, press Ctrl/C at any time during the processing of a full or brief interactive display. At the COMMAND> prompt, you can enter any command listed in this section. Use the CONTINUE command to resume processing of the event records, or use the EXIT command to terminate the session.

---

**CONTINUE**

Resumes processing of event records.

**Format**

CONTINUE

**Parameters**

None.

**Qualifiers**

None.

**Example**

```
COMMAND> DISPLAY/SINCE=25-JAN-2002/SELECT=USERNAME=JOHNSON  
COMMAND> CONTINUE
```

The first command in this example selects only event records generated by user JOHNSON after January 25, 2002. The second command in the example displays a report based on the new selection criteria.

## ANALYZE/AUDIT DISPLAY

---

### DISPLAY

Changes the criteria used to select event records.

### Format

DISPLAY

### Parameters

None.

For a more complete description of any one of the following qualifiers, refer to the description of the qualifier in the preceding ANALYZE/AUDIT Qualifiers section.

### 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 are 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 the /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 are 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.



## Examples

1. `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.

2. `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.

**ANALYZE/AUDIT  
EXIT**

---

**EXIT**

Terminates the session.

**Format**

EXIT

**Parameters**

None.

**Qualifiers**

None.

## **HELP**

Provides online help information for using ANALYZE/AUDIT commands.

### **Format**

HELP [topic]

### **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 DISPLAY command.

## ANALYZE/AUDIT LIST

---

### LIST

Changes the criteria used to select event records. The LIST command is synonymous with the DISPLAY command.

### Format

LIST

### Parameters

None.

### Qualifiers

See the description of the DISPLAY command.

### 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. The command is useful when you supply a wildcard file specification to the ANALYZE/AUDIT command; for example \*.AUDIT\$JOURNAL. If there are no other audit log files to open, the audit analysis session terminates and control returns to DCL.

**Format**

NEXT FILE

**Parameters**

None.

**Qualifiers**

None.

## **ANALYZE/AUDIT NEXT RECORD**

---

### **NEXT RECORD**

Controls whether the next audit record is displayed. The NEXT RECORD command is the default for interactive mode.

This command is synonymous with the POSITION command.

#### **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.

### **Examples**

1. COMMAND> POSITION 100

The command in this example moves the display forward 100 event records.

2. COMMAND> POSITION -100

The command in this example moves the display back 100 event records.

## ANALYZE/AUDIT SHOW

---

### SHOW

Displays information about the 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:

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.







## 5.1 AUTHORIZE Description

The Authorize utility (AUTHORIZE) is a system management tool used to control access to the system and to allocate resources to users.

AUTHORIZE creates new records or modifies existing records in the following files:

- System user authorization file (SYSUAF.DAT)

You can use AUTHORIZE to assign values to various **fields** within each SYSUAF record. The values you assign identify the user and the user's work environment, and control use of system resources.

You can redirect SYSUAF logical access by defining a logical in your local process logical table; for example:

```
$ DEFINE/PROCESS/EXEC SYSUAF DISK$USER:[MYPROCESSTABLE]SYSUAF.DAT
```

You can, if you like, define the SYSUAF logical in user mode.

If you move the SYSUAF.DAT file, be sure the logical name SYSUAF is defined and points to an existing file. If AUTHORIZE is unable to locate the SYSUAF.DAT file, it displays the following error message:

```
%UAF-E-NAOFIL, unable to open SYSUAF.DAT  
-RMS-E-FNF, file not found  
Do you want to create a new file?
```

A response of YES results in creation of a new SYSUAF file containing a SYSTEM record and a DEFAULT record. These records are initialized with the same values set when the system was installed.

- Network proxy authorization file

The default network proxy authorization file is NET\$PROXY.DAT. However, AUTHORIZE maintains the file NETPROXY.DAT for compatibility.

In a mixed-version cluster where systems are running OpenVMS Alpha or a version of OpenVMS VAX earlier than Version 6.1, you must make all proxy modifications on an OpenVMS VAX Version 6.1 or later system.

You can redirect NETPROXY logical access by defining a logical in your local process logical table; for example:

```
$ DEFINE/PROCESS/EXEC NETPROXY DISK$USER:[MYPROCESSTABLE]NETPROXY.DAT
```

- Rights database file (RIGHTSLIST.DAT)

You can redirect RIGHTSLIST logical access by defining a logical in your local process logical table; for example:

```
$ DEFINE/PROCESS/EXEC RIGHTSLIST DISK$USER:[MYPROCESSTABLE]RIGHTSLIST.DAT
```

These files store system authorization information. By default, they are owned by the system (UIC of [SYSTEM]) and are created with the following protection:

```
SYSUAF.DAT      S:RWED, O:RWED, G, W  
NETPROXY.DAT   S:RWED, O:RWED, G, W  
NET$PROXY.DAT  S, O, G, W  
RIGHTSLIST.DAT S:RWED, O:RWED, G, W:
```

To use AUTHORIZE, you must have write access to all three of these files (you must have an account with the user identification code (UIC) of [SYSTEM] or the SYSPRV privilege).

## AUTHORIZE

### 5.1 AUTHORIZE Description

Note that you must have read access to the RIGHTS.LIST.DAT file (or sufficient privileges) to display the rights identifiers held by other users.

Because certain images (such as MAIL and SET) require access to the system user authorization file (UAF) and are normally installed with the SYSPRV privilege, ensure that you always grant system access to SYSUAF.DAT.

When you install a new system, the software distribution kit provides the following records in the system user authorization file in SYS\$SYSTEM:

On VAX systems:

```
DEFAULT
FIELD
SYSTEM
SYSTEST
SYSTEST_CLIG
```

On Alpha and I64 systems:

```
DEFAULT
SYSTEM
```

If the SYSUAF.DAT becomes corrupted or is accidentally deleted, you can use the template file SYSUAF.TEMPLATE in the SYS\$SYSTEM directory to recreate the file, as follows:

```
$ SET DEFAULT SYS$SYSTEM
$ COPY SYSUAF.TEMPLATE SYSUAF.DAT
```

The file SYSUAF.TEMPLATE contains records that are identical to those defined when the system was installed.

To make an emergency backup for the system SYSUAF file, you can create a private copy of SYSUAF.DAT. To affect future logins, copy a private version of SYSUAF.DAT to the appropriate directory, as shown in the following example:

```
$ COPY MYSYSUAF.DAT SYS$COMMON:[SYSEXE]:SYSUAF.DAT-
_ $ /PROTECTION=(S:RWED,O:RWED,G,W)
```

### 5.2 AUTHORIZE Usage Summary

The Authorize utility (AUTHORIZE) is a system management tool that enables you to control access to the system and to allocate resources to users.

#### Format

```
RUN SYS$SYSTEM:AUTHORIZE
```

#### Parameters

None.

#### Description

To invoke AUTHORIZE, set your default device and directory to SYS\$SYSTEM and enter RUN AUTHORIZE at the DCL command prompt.

At the UAF> prompt, you can enter any AUTHORIZE command described in the following section.

To exit from AUTHORIZE, enter the EXIT command at the UAF> prompt or press Ctrl/Z.

### 5.3 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 to specify user names, node names, and UICs.

AUTHORIZE commands fall into the following four categories:

- Commands that allow you to manage user authorization records. By specifying appropriate qualifiers, you can use these commands to act upon individual fields of SYSUAF records. You can identify the user and the user's work environment and control use of system resources.
- Commands that build and maintain the network proxy authorization file (NETPROXY.DAT or NET\$PROXY.DAT).
- Commands that create and maintain the rights database (RIGHTSLIST.DAT).
- Commands that perform general utility functions or modify the system password.

The following table summarizes the AUTHORIZE commands according to these categories:

Command	Description
<b>Managing System Resources and User Accounts with SYSUAF</b>	
ADD	Adds a user record to the SYSUAF and corresponding identifiers to the rights database.
COPY	Creates a new SYSUAF record that duplicates an existing record.
DEFAULT	Modifies the default SYSUAF record.
LIST	Writes reports for selected UAF records to a listing file, SYSUAF.LIS.
MODIFY	Changes values in a SYSUAF user record. Qualifiers not specified in the command remain unchanged.
REMOVE	Deletes a SYSUAF user record and corresponding identifiers in the rights database. The DEFAULT and SYSTEM records cannot be deleted.
RENAME	Changes the user name of the SYSUAF record (and, if specified, the corresponding identifier) while retaining the characteristics of the old record.
SHOW	Displays reports for selected SYSUAF records.

## AUTHORIZE

### 5.3 AUTHORIZE Commands

Command	Description
<b>Managing Network Proxies with NETPROXY.DAT or NET\$PROXY.DAT</b>	
ADD/PROXY	Adds proxy access for the specified user.
CREATE/PROXY	Creates a network proxy authorization file.
LIST/PROXY	Creates a listing file of all proxy accounts and all remote users with proxy access to the accounts.
MODIFY/PROXY	Modifies proxy access for the specified user.
REMOVE/PROXY	Deletes proxy access for the specified user.
SHOW/PROXY	Displays proxy access allowed for the specified user.
<b>Managing Identifiers with RIGHTSLIST.DAT</b>	
ADD/IDENTIFIER	Adds an identifier name to the rights database.
CREATE/RIGHTS	Creates a new rights database file.
GRANT/IDENTIFIER	Grants an identifier name to a UIC identifier.
LIST/IDENTIFIER	Creates a listing file of identifier names and values.
LIST/RIGHTS	Creates a listing file of all identifiers held by the specified user.
MODIFY/IDENTIFIER	Modifies the named identifier in the rights database.
REMOVE/IDENTIFIER	Removes an identifier from the rights database.
RENAME/IDENTIFIER	Renames an identifier in the rights database.
REVOKE/IDENTIFIER	Revokes an identifier name from a UIC identifier.
SHOW/IDENTIFIER	Displays identifier names and values on the current output device.
SHOW/RIGHTS	Displays on the current output device the names of all identifiers held by the specified user.
<b>General Commands</b>	
EXIT	Returns the user to DCL command level.
HELP	Displays HELP text for AUTHORIZE commands.
MODIFY/SYSTEM_PASSWORD	Sets the system password (equivalent to the DCL command SET PASSWORD/SYSTEM).

## ADD

Adds a user record to the SYSUAF and corresponding identifiers to the rights database.

---

**Note**

---

ADD/IDENTIFIER and ADD/PROXY are documented as separate commands.

---

### Format

ADD newusername

### Parameter

#### **newusername**

Specifies the name of the user record to be included in the SYSUAF. The **newusername** parameter is a string of 1 to 12 alphanumeric characters and can contain underscores. Although dollar signs are permitted, they are usually reserved for system names.

Avoid using fully numeric user names (for example, 89560312). A fully numeric user name cannot receive a corresponding identifier because fully numeric identifiers are not permitted.

### Qualifiers

**/ACCESS[=(range[,...])]**

**/NOACCESS[=(range[,...])]**

Specifies hours of access for all modes of access. The syntax for specifying the range is:

**/[NO]ACCESS=([PRIMARY], [n-m], [n], [...],[SECONDARY], [n-m], [n], [...])**

Specify hours as integers from 0 to 23, inclusive. You can specify single hours (n) or 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 *inclusive*; that is, if you grant access during a given hour, access extends to the end of that hour.

By default, a user has full access every day. See the DCL command **SET DAY** in the *HP OpenVMS DCL Dictionary* for information about overriding the defaults for primary and secondary day types.

All the list elements are optional. Unless you specify hours for a day type, access is permitted for the entire day. By specifying an access time, you prevent access at all other times. Adding **NO** to the qualifier denies the user access to the system for the specified period of time. See the following examples.

<b>/ACCESS</b>	Allows unrestricted access
<b>/NOACCESS=SECONDARY</b>	Allows access on primary days only

## AUTHORIZE ADD

**/ACCESS=(9-17)** Allows access from 9 A.M. to 5:59 P.M. on all days

**/NOACCESS=(PRIMARY, 9-17, SECONDARY, 18-8)** Disallows access between 9 A.M. to 5:59 P.M. on primary days but allows access during these hours on secondary days

To specify access hours for specific types of access, see the **/BATCH**, **/DIALUP**, **/INTERACTIVE**, **/LOCAL**, **/NETWORK**, and **/REMOTE** qualifiers.

Refer to *HP OpenVMS Guide to System Security* for information about the effects of login class restrictions.

### **/ACCOUNT=account-name**

Specifies the default name for the account (for example, a billing name or number). The name can be a string of 1 to 8 alphanumeric characters. By default, AUTHORIZE does not assign an account name.

### **/ADD\_IDENTIFIER (default)**

#### **/NOADD\_IDENTIFIER**

Adds a user (user name and account name) to the rights database. The **/NOADD\_IDENTIFIER** does not create a rights list identifier (user name and account name). The AUTHORIZE command **ADD/IDENTIFIER** is quite different: it adds a record to the AUTHORIZE database UAF file.

### **/ALGORITHM=keyword=type [=value]**

Sets the password encryption algorithm for a user. The keyword **VMS** refers to the algorithm used in the operating system version that is running on your system, whereas a customer algorithm is one that is added through the **\$HASH\_PASSWORD** system service by a customer site, by a layered product, or by a third party. The customer algorithm is identified in **\$HASH\_PASSWORD** by an integer in the range of 128 to 255. It must correspond with the number used in the AUTHORIZE command **MODIFY/ALGORITHM**. By default, passwords are encrypted with the VMS algorithm for the current version of the operating system.

<b>Keyword</b>	<b>Function</b>
<b>BOTH</b>	Set the algorithm for primary and secondary passwords.
<b>CURRENT</b>	Set the algorithm for the primary, secondary, both, or no passwords, depending on account status. <b>CURRENT</b> is the default value.
<b>PRIMARY</b>	Set the algorithm for the primary password only.
<b>SECONDARY</b>	Set the algorithm for the secondary password only.

The following table lists password encryption algorithms:

<b>Type</b>	<b>Definition</b>
<b>VMS</b>	The algorithm used in the version of the operating system that is running on your system.
<b>CUSTOMER</b>	A numeric value in the range of 128 to 255 that identifies a customer algorithm.

The following example selects the VMS algorithm for Sontag's primary password:



```
UAF> MODIFY SONTAG/ALGORITHM=PRIMARY=VMS
```

If you select a site-specific algorithm, you must give a value to identify the algorithm, as follows:

```
UAF> MODIFY SONTAG/ALGORITHM=CURRENT=CUSTOMER=128
```

**/ASTLM=value**

Specifies the AST queue limit, which is the total number of asynchronous system trap (AST) operations and scheduled wake-up requests that the user can have queued at one time. The default is 40 on VAX systems and 300 on Alpha and I64 systems.

**/BATCH[=(range[,...])]**

Specifies the hours of access permitted for batch jobs. For a description of the range specification, see the /ACCESS qualifier. By default, a user can submit batch jobs any time.

**/BIOLM=value**

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. The default is 40 on VAX systems and 150 on Alpha and I64 systems.

**/BYTLM=value**

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 can consume at one time. Nonpaged dynamic memory is used for operations such as I/O buffering, mailboxes, and file-access windows. The default is 32768 on VAX systems and 128,000 on Alpha and I64 systems.

**/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 a string of 1 to 31 alphanumeric characters and should be DCL, which is the default. This setting is ignored for network jobs.

**/CLITABLES=filespec**

Specifies user-defined CLI tables for the account. The *filespec* can contain 1 to 31 characters. The default is SYS\$LIBRARY:DCLTABLES. Note that this setting is ignored for network jobs to guarantee that the system-supplied command procedures used to implement network objects function properly.

**/CPUTIME=time**

Specifies the maximum process CPU time for the CPU field of the UAF record. The maximum process CPU time is the maximum amount of CPU time a user's process can take per session. You must specify a delta time value. For a discussion of delta time values, refer to the *OpenVMS User's Manual*. The default is 0, which means an infinite amount of time.

**/DEFPRIVILEGES=([NO]privname[,...])**

Specifies default privileges for the user; that is, those enabled at login time. A NO prefix removes a privilege from the user. By specifying the keyword [NO]ALL with the /DEFPRIVILEGES qualifier, you can disable or enable all user privileges. The default privileges are TMPMBX and NETMBX. *Privname* is the name of the privilege.

## AUTHORIZE ADD

### **/DEVICE=device-name**

Specifies the name of the user's default device at login. The *device-name* is a string of 1 to 31 alphanumeric characters. If you omit the colon from the *device-name* value, AUTHORIZE appends a colon. The default device is SYS\$SYSDISK.

If you specify a logical name as the *device-name* (for example, DISK1: for DUA1:), you must make an entry for the logical name in the LNM\$SYSTEM\_TABLE in executive mode by using the DCL command DEFINE/SYSTEM/EXEC.

### **/DIALUP[(range[,...])]**

Specifies hours of access permitted for dialup logins. For a description of the range specification, see the /ACCESS qualifier. The default is full access.

### **/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 default is 40 on VAX systems and 150 on Alpha and I64 systems.

### **/DIRECTORY=directory-name**

Specifies the default directory name for the DIRECTORY field of the UAF record. The *directory-name* can be 1 to 39 alphanumeric characters. If you do not enclose the directory name in brackets, AUTHORIZE adds the brackets for you. The default directory name is [USER].

### **/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 by the user at one time. The default is 200 on VAX systems and 4000 on Alpha and I64 systems.

### **/EXPIRATION=time (default)**

#### **/NOEXPIRATION**

Specifies the expiration date and time of the account. The /NOEXPIRATION qualifier removes the expiration date on the account. If you do not specify an expiration time when you add a new account, AUTHORIZE copies the expiration time from the DEFAULT account. (The expiration time on the DEFAULT account is "none" by default.)

### **/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. The default is 300 on VAX systems and 128 on Alpha and I64 systems.

### **/FLAGS=(**[NO]**option[,...])**

Specifies login flags for the user. The prefix NO clears the flag. The options are as follows:

AUDIT	Enables or disables mandatory security auditing for a specific user. By default, the system does not audit the activities of specific users (NOAUDIT).
-------	--

AUTOLOGIN	Restricts the user to the automatic login mechanism when logging in to an account. When set, the flag disables login by any terminal that requires entry of a user name and password. The default is to require a user name and password (NOAUTOLOGIN).
CAPTIVE	Prevents the user from changing any defaults at login, for example, /CLI or /LGICMD. It prevents the user from escaping the captive login command procedure specified by the /LGICMD qualifier and gaining access to the DCL command level. Refer to “Guidelines for Captive Command Procedures” in the <i>HP OpenVMS Guide to System Security</i> . The CAPTIVE flag also establishes an environment where Ctrl/Y interrupts are initially turned off; however, command procedures can still turn on Ctrl/Y interrupts with the DCL command SET CONTROL=Y. By default, an account is not captive (NOCAPTIVE).
DEFCLI	Restricts the user to the default command interpreter by prohibiting the use of the /CLI qualifier at login. By default, a user can choose a CLI (NODEFCLI).
DISCTLY	Establishes an environment where Ctrl/Y interrupts are initially turned off and are invalid until a SET CONTROL=Y is encountered. This could happen in SYLOGIN.COM or in a procedure called by SYLOGIN.COM. Once a SET CONTROL=Y is executed (which requires no privilege), a user can enter a Ctrl/Y and reach the DCL prompt (\$). If the intent of DISCTLY is to force execution of the login command files, then SYLOGIN.COM should issue the DCL command SET CONTROL=Y to turn on Ctrl/Y interrupts before exiting. By default, Ctrl/Y is enabled (NODISCTLY).
DISFORCE_PWD_CHANGE	Removes the requirement that a user must change an expired password at login. By default, a person can use an expired password only once (NODISFORCE_PWD_CHANGE) and then is forced to change the password after logging in. If the user does not select a new password, the user is locked out of the system.  To use this feature, set a password expiration date with the /PWDLIFETIME qualifier.
DISIMAGE	Prevents the user from executing RUN and foreign commands. By default, a user can execute RUN and foreign commands (NODISIMAGE).
DISMAIL	Disables mail delivery to the user. By default, mail delivery is enabled (NODISMAIL).
DISNEWMAIL	Suppresses announcements of new mail at login. By default, the system announces new mail (NODISNEWMAIL).
DISPWDDIC	Disables automatic screening of new passwords against a system dictionary. By default, passwords are automatically screened (NODISPWDDIC).

## AUTHORIZE ADD

DISPWDHIS	Disables automatic checking of new passwords against a list of the user's old passwords. By default, the system screens new passwords (NODISPWDHIS).
DISPWDSYNCH	Suppresses synchronization of the external password for this account. See bit 9 in the SECURITY_POLICY system parameter for systemwide password synchronization control.
DISRECONNECT	Disables automatic reconnection to an existing process when a terminal connection has been interrupted. By default, automatic reconnection is enabled (NODISRECONNECT).
DISREPORT	Suppresses reports of the last login time, login failures, and other security reports. By default, login information is displayed (NODISREPORT).
DISUSER	Disables the account so the user cannot log in. For example, the DEFAULT account is disabled. By default, an account is enabled (NODISUSER).
DISWELCOME	Suppresses the welcome message (an informational message displayed during a local login). This message usually indicates the version number of the operating system that is running and the name of the node on which the user is logged in. By default, a system login message appears (NODISWELCOME).
EXTAUTH	Considers user to be authenticated by an external user name and password, not by the SYSUAF user name and password. (The system still uses the SYSUAF record to check a user's login restrictions and quotas and to create the user's process profile.)
GENPWD	Restricts the user to generated passwords. By default, users choose their own passwords (NOGENPWD).
LOCKPWD	Prevents the user from changing the password for the account. By default, users can change their passwords (NOLOCKPWD).
PWD_EXPIRED	Marks a password as expired. The user cannot log in if this flag is set. The LOGINOUT.EXE image sets the flag when both of the following conditions exist: a user logs in with the DISFORCE_PWD_CHANGE flag set, and the user's password expires. A system manager can clear this flag. By default, passwords are not expired after login (NOPWD_EXPIRED).
PWD2_EXPIRED	Marks a secondary password as expired. Users cannot log in if this flag is set. The LOGINOUT.EXE image sets the flag when both of the following conditions exist: a user logs in with the DISFORCE_PWD_CHANGE flag set, and the user's password expires. A system manager can clear this flag. By default, passwords are not set to expire after login (NOPWD2_EXPIRED).

**PWDMIX** Enables case-sensitive and extended-character passwords. After PWDMIX is specified, you can then use mixed-case and extended characters in passwords. Be aware that before the PWDMIX flag is enabled, the system stores passwords in all upper-case. Therefore, until you change passwords, you must enter your pre-PWDMIX passwords in upper-case.

To change the password after PWDMIX is enabled:

- You (the user) can use the DCL command SET PASSWORD, specifying the new mixed-case password (omitting quotation marks).
- You (the system manager) can use the AUTHORIZE command MODIFY/PASSWORD, and enclose the user's new mixed-case password in quotation marks " ".

**RESTRICTED** Prevents the user from changing any defaults at login (for example, by specifying /LGICMD) and prohibits user specification of a CLI with the /CLI qualifier. The RESTRICTED flag establishes an environment where Ctrl/Y interrupts are initially turned off; however, command procedures can still turn on Ctrl/Y interrupts with the DCL command SET CONTROL=Y. Typically, this flag is used to prevent an applications user from having unrestricted access to the CLI. By default, a user can change defaults (NORESTRICTED).

**VMSAUTH** Allows account to use standard (SYSUAF) authentication when the EXTAUTH flag would otherwise require external authentication. This depends on the application. An application specifies the VMS domain of interpretation when calling SYS\$ACM to request standard VMS authentication for a user account that normally uses external authentication.

**/GENERATE\_PASSWORD[=keyword]**

**/NOGENERATE\_PASSWORD (default)**

Invokes the password generator to create user passwords. Generated passwords can consist of 1 to 10 characters. Specify one of the following keywords:

<b>BOTH</b>	Generate primary and secondary passwords.
<b>CURRENT</b>	Do whatever the DEFAULT account does (for example, generate primary, secondary, both, or no passwords). This is the default keyword.
<b>PRIMARY</b>	Generate primary password only.
<b>SECONDARY</b>	Generate secondary password only.

When you modify a password, the new password expires automatically; it is valid only once (unless you specify /NOPWDEXPIRED). On login, users are forced to change their passwords (unless you specify /FLAGS=DISFORCE\_PWD\_CHANGE).

Note that the /GENERATE\_PASSWORD and /PASSWORD qualifiers are mutually exclusive.

## AUTHORIZE ADD

**/INTERACTIVE[ =(range[,...])]**

**/NOINTERACTIVE**

Specifies the hours of access for interactive logins. For a description of the range specification, see the /ACCESS qualifier. By default, there are no access restrictions on interactive logins.

**/JTQUOTA=value**

Specifies the initial byte quota with which the jobwide logical name table is to be created. By default, the value is 4096 on VAX systems and 4096 on Alpha and I64 systems.

**/LGICMD=filespec**

Specifies the name of the default login command file. The file name defaults to the device specified for /DEVICE, the directory specified for /DIRECTORY, a file name of LOGIN, and a file type of .COM. If you select the defaults for all these values, the file name is SYS\$SYSTEM:[USER]LOGIN.COM.

**/LOCAL[=(range[,...])]**

Specifies hours of access for interactive logins from local terminals. For a description of the range specification, see the /ACCESS qualifier. By default, there are no access restrictions on local logins.

**/MAXACCTJOBS=value**

Specifies the maximum number of batch, interactive, and detached processes that can be active at one time for all users of the same account. By default, a user has a maximum of 0, which represents an unlimited number.

**/MAXDETACH=value**

Specifies the maximum number of detached processes with the cited user name that can be active at one time. To prevent the user from creating detached processes, specify the keyword NONE. By default, a user has a value of 0, which represents an unlimited number.

**/MAXJOBS=value**

Specifies the maximum number of processes (interactive, batch, detached, and network) with the cited user name that can be active simultaneously. The first four network jobs are not counted. By default, a user has a maximum value of 0, which represents an unlimited number.

**/NETWORK[=(range[,...])]**

Specifies hours of access for network batch jobs. For a description of how to specify the range, see the /ACCESS qualifier. By default, network logins have no access restrictions.

**/OWNER=owner-name**

Specifies the name of the owner of the account. You can use this name for billing purposes or similar applications. The owner name is 1 to 31 characters. No default owner name exists.

**/PASSWORD=(password1[,password2])**

**/NOPASSWORD**

Specifies up to two passwords for login. Passwords can be from 0 to 32 alphanumeric characters in length. The dollar sign (\$) and underscore (\_) are also permitted.

Uppercase and lowercase characters are equivalent. All lowercase characters are converted to uppercase before the password is encrypted. Avoid using the word *password* as the actual password.

Use the /PASSWORD qualifier as follows:

- To set only the first password and clear the second, specify /PASSWORD=password.
- 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.

When you modify a password, the new password expires automatically; it is valid only once (unless you specify /NOPWDEXPIRED). On login, the user is forced to change the password (unless you specify /FLAGS=DISFORCE\_PWD\_CHANGE).

Note that the /GENERATE\_PASSWORD and /PASSWORD qualifiers are mutually exclusive.

By default, the ADD command assigns the password USER. When you create a new UAF record with the COPY or RENAME command, you must specify a password. Avoid using the word *password* as the actual password.

**/PBYTLM**

This flag is reserved for HP.

**/PGFLQUOTA=value**

Specifies the paging file limit. This is the maximum number of pages that the person's process can use in the system paging file. By default, the value is 32768 pages on VAX systems and 256,000 pagelets on Alpha and I64 systems.

If decompressing libraries, make sure to set PGFLQUOTA to twice the size of the library.

**/PRCLM=value**

Specifies the subprocess creation limit. This is the maximum number of subprocesses that can exist at one time for the specified user's process. By default, the value is 2 on VAX systems and 8 on Alpha and I64 systems.

**/PRIMEDAYS=(**[NO]**day[,...])**

Defines the primary and secondary days of the week for logging in. Specify the days as a list separated by commas, and enclose the list in parentheses. To specify a secondary day, prefix the day with NO (for example, NOFRIDAY). To specify a primary day, omit the NO prefix.

By default, primary days are Monday through Friday and secondary days are Saturday and Sunday. If you omit a day from the list, AUTHORIZE uses the default value. (For example, if you omit Monday from the list, AUTHORIZE defines Monday as a primary day.)

Use the primary and secondary day definitions in conjunction with such qualifiers as /ACCESS, /INTERACTIVE, and /BATCH.

## AUTHORIZE ADD

### **/PRIORITY=value**

Specifies the default base priority. The value is an integer in the range of 0 to 31 on VAX systems and 0 to 63 on Alpha and I64 systems. By default, the value is set to 4 for timesharing users.

### **/PRIVILEGES=(*[NO]*privname[,...])**

Specifies which privileges the user is authorized to hold, although these privileges are not necessarily enabled at login. (The `/DEFPRIVILEGES` qualifier determines which ones are enabled.) A `NO` prefix removes the privilege from the user. The keyword `NOALL` disables all user privileges. Many privileges have varying degrees of power and potential system impact (see the *HP OpenVMS Guide to System Security* for a detailed discussion). By default, a user holds `TMPMBX` and `NETMBX` privileges. *Privname* is the name of the privilege.

### **/PWDEXPIRED (default)**

#### **/NOPWDEXPIRED**

Specifies the password is valid for only one login. A user must change a password immediately after login or be locked out of the system. The system warns users of password expiration. A user can either specify a new password, with the DCL command `SET PASSWORD`, or wait until expiration and be forced to change. By default, a user must change a password when first logging in to an account. The default is applied to the account only when the password is being modified.

### **/PWDLIFETIME=time (default)**

#### **/NOPWDLIFETIME**

Specifies the length of time a password is valid. Specify a delta time value in the form `[ddd-] [hh:mm:ss.cc]`. For example, for a lifetime of 120 days, 0 hours, and 0 seconds, specify `/PWDLIFETIME="120-"`. For a lifetime of 120 days 12 hours, 30 minutes and 30 seconds, specify `/PWDLIFETIME="120-12:30:30"`. If a period longer than the specified time elapses before the user logs in, the system displays a warning message. The password is marked as expired.

To prevent a password from expiring, specify the time as `NONE`. By default, a password expires in 90 days.

### **/PWDMINIMUM=value**

Specifies the minimum password length in characters. Note that this value is enforced only by the DCL command `SET PASSWORD`. It does not prevent you from entering a password shorter than the minimum length when you use `AUTHORIZE` to create or modify an account. By default, a password must have at least 6 characters. The value specified by the `/PWDMINIMUM` qualifier conflicts with the value used by the `/GENERATE_PASSWORD` qualifier or the DCL command `SET PASSWORD/GENERATE`, the operating system chooses the lesser value. The maximum value for generated passwords is 10.

### **/QUEPRIO=value**

Reserved for future use.

### **/REMOTE[=(range[,...])]**

Specifies hours during which access is permitted for interactive logins from network remote terminals (with the DCL command `SET HOST`). For a description of the range specification, see the `/ACCESS` qualifier. By default, remote logins have no access restrictions.



**/SHRFILLM=value**

Specifies the maximum number of shared files that the user can have open at one time. By default, the system assigns a value of 0, which represents an infinite number.

**/TQELM**

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. By default, a user can have 100.

**/UIC=value**

Specifies the user identification code (UIC). The UIC value is a group number in the range from 1 to 37776 (octal) and a member number in the range from 0 to 17776 (octal), which are separated by a comma and enclosed in brackets. HP reserves group 1 and groups 300–377 for its own use.

Each user must have a unique UIC. By default, the UIC value is [200,200].

**/WSDEFAULT=value**

Specifies the default working set limit. This represents the initial limit to the number of physical pages the process can use. (The user can alter the default quantity up to WSQUOTA with the DCL command SET WORKING\_SET.) By default, a user has 256 pages on VAX systems and 4096 pagelets on Alpha and I64 systems.

The value cannot be greater than WSMAX. This quota value replaces smaller values of PQL\_MWSDEFAULT.

**/WSEXTENT=value**

Specifies the working set maximum. This represents the maximum amount of physical memory allowed to the process. The system provides memory to a process beyond its working set quota only when it has excess free pages. The additional memory is recalled by the system if needed.

The value is an integer equal to or greater than WSQUOTA. By default, the value is 1024 pages on VAX systems and 16384 pagelets on Alpha and I64 systems. The value cannot be greater than WSMAX. This quota value replaces smaller values of PQL\_MWSEXTENT.

**/WSQUOTA=value**

Specifies the working set quota. This is the maximum amount of physical memory a user process can lock into its working set. It also represents the maximum amount of swap space that the system reserves for this process and the maximum amount of physical memory that the system allows the process to consume if the systemwide memory demand is significant.

The value cannot be greater than the value of WSMAX and cannot exceed 8,192 pagelets on Alpha and I64 systems. This quota value replaces smaller values of PQL\_MWSQUOTA.

## **Description**

When you do not specify a value for a field, AUTHORIZE uses values from the DEFAULT record (excluding the default password, which is always USER). The DEFAULT account serves as a template for creating user records in the system user authorization file.

## AUTHORIZE ADD

On Alpha and I64 systems, the DEFAULT account is as follows:

```
Username: DEFAULT                      Owner:
Account:                               UIC: [200,200] ([DEFAULT])
CLI: DCL                               Tables: DCLTABLES
Default: [USER]
LGICMD:
Flags: DisUser
Primary days: Mon Tue Wed Thu Fri
Secondary days:                        Sat Sun
No access restrictions
Expiration: (none) Pwdminimum: 6 Login Fails: 0
Pwdlifetime: 90 00:00 Pwdchange: (pre-expired)
Last Login: (none) (interactive), (none) (non-interactive)
Maxjobs: 0 Fillm: 128 Bytlm: 128000
Maxacctjobs: 0 Shrfillm: 0 Pbytlm: 0
Maxdetach: 0 BIoIm: 150 JTquota: 4096
Prclm: 8 DIoIm: 150 WSdef: 4096
Prio: 4 ASTlm: 300 WSquo: 8192
Queprio: 4 TQElm: 100 WSexent: 16384
CPU: (none) Enqlm: 4000 Pgflquo: 256000
Authorized Privileges:
  NETMBX TMPMBX
Default Privileges:
  NETMBX TMPMBX
```

On VAX systems, the DEFAULT account is as follows:

```
Username: DEFAULT                      Owner:
Account:                               UIC: [200,200] ([DEFAULT])
CLI: DCL                               Tables: DCLTABLES
Default: SYS$SYSDVICE:[USER]
LGICMD: LOGIN
Flags: DisUser
Primary days: Mon Tue Wed Thu Fri
Secondary days:                        Sat Sun
No access restrictions
Expiration: (none) Pwdminimum: 6 Login Fails: 0
Pwdlifetime: 90 00:00 Pwdchange: (pre-expired)
Last Login: (none) (interactive) (none) (non-interactive)
Maxjobs: 0 Fillm: 300 Bytlm: 32768
Maxacctjobs: 0 Shrfillm: 0 Pbytlm: 0
Maxdetach: 0 BIoIm: 40 JTquota: 4096
Prclm: 2 DIoIm: 40 WSdef: 256
Prio: 4 ASTlm: 40 WSquo: 512
Queprio: 0 TQElm: 10 WSexent: 1024
CPU: (none) Enqlm: 200 Pgflquo: 32768
Authorized Privileges:
  TMPMBX NETMBX
Default Privileges:
  TMPMBX NETMBX
```

When you add a new account, specify values for fields that you want to be different. Typically, changing the default values for limits priority, privileges, or the command interpreter is not necessary. As a result, you enter only the password, UIC, directory, owner, account, and device.

---

### Note

Limits are also set by system parameters. To be effective, the limits you set through AUTHORIZE must be within the minimum limits determined by the corresponding system parameters (particularly those beginning with the PQL prefix).

---

When you add a record to the UAF, create a directory for the new user. Specify the device name, directory name, and UIC in the UAF record. The following DCL command creates a directory for user ROBIN:

```
$ CREATE/DIRECTORY SYS$USER:[ROBIN] /OWNER_UIC=[ROBIN]
```

---

**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 automatically (unless you specify the /NOADD\_IDENTIFIER qualifier). Similarly, when you specify an account name (other than the user name) that does not yet have an identifier, AUTHORIZE creates a group identifier in the rights database.

---

## Examples

```
1. UAF> ADD ROBIN /PASSWORD=SP0152/UIC=[014,006] -  
  /DEVICE=SYS$USER/DIRECTORY=[ROBIN]/OWNER="JOSEPH ROBIN" /ACCOUNT=INV  
%UAF-I-ADDMSG, user record successfully added  
%UAF-I-RDBADDMSGU, identifier ROBIN value: [000014,000006] added to  
  RIGHTSLLIST.DAT  
%UAF-I-RDBADDMSGU, identifier INV value: [000014,177777] added to  
  RIGHTSLLIST.DAT
```

This example illustrates the typical ADD command and qualifiers. The resulting record from this command appears in the description of the SHOW command.

```
2. UAF> ADD WELCH /PASSWORD=SP0158/UIC=[014,051] -  
  /DEVICE=SYS$USER/DIRECTORY=[WELCH]/OWNER="ROB WELCH"/FLAGS=DISUSER -  
  /ACCOUNT=INV/LGICMD=SECUREIN  
%UAF-I-ADDMSG, user record successfully added  
%UAF-I-RDBADDMSGU, identifier WELCH value: [000014,000051] added to  
  RIGHTSLLIST.DAT  
UAF> MODIFY WELCH/FLAGS=(RESTRICTED,DISNEWMAIL,DISWELCOME, -  
  NODISUSER,EXTAUTH)/NODIALUP=SECONDARY/NETWORK=PRIMARY -  
  /CLITABLES=DCLTABLES/NOACCESS=(PRIMARY, 9-16, SECONDARY, 18-8)  
%UAF-I-MDFYMSG, user records updated
```

The commands in this example add a record for a restricted account. Because of the number of qualifiers required, a MODIFY command is used in conjunction with the ADD command. This helps to minimize the possibility of typing errors.

In the ADD command line, setting the DISUSER flag prevents the user from logging in until all the account parameters are set up. In the MODIFY command line, the DISUSER flag is disabled (by specifying NODISUSER) to allow access to the account. The EXTAUTH flag causes the system to consider the user as authenticated by an external user name and password, not by the SYSUAF user name and password.

The record that results from these commands and an explanation of the restrictions the record imposes appear in the description of the SHOW command.

# AUTHORIZE ADD/IDENTIFIER

---

## ADD/IDENTIFIER

Adds only an identifier to the rights database. It does not add a user account.

### 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 to 31 alphanumeric characters. The name can contain underscores and dollar signs. It must contain at least one nonnumeric character.

### Qualifiers

#### **/ATTRIBUTES=(keyword[,...])**

Specifies attributes to be associated with the new identifier. The following keywords are valid:

DYNAMIC	Allows unprivileged holders of the identifier to remove and to restore the identifier from the process rights list by using the DCL command SET RIGHTS_LIST.
HOLDER_HIDDEN	Prevents people from getting a list of users who hold an identifier, unless they own the identifier themselves.
NAME_HIDDEN	Allows holders of an identifier to have it translated, either from binary to ASCII or from ASCII to binary, but prevents unauthorized users from translating the identifier.
NOACCESS	Makes any access rights of the identifier null and void. If a user is granted an identifier with the No Access attribute, that identifier has no effect on the user's access rights to objects. This attribute is a modifier for an identifier with the Resource or Subsystem attribute.
RESOURCE	Allows holders of an identifier to charge disk space to the identifier. Used only for file objects.
SUBSYSTEM	Allows holders of the identifier to create and maintain protected subsystems by assigning the Subsystem ACE to the application images in the subsystem. Used only for file objects.

By default, none of these attributes is associated with the new identifier.

#### **/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 formats are valid for the *value-specifier*:

- IDENTIFIER:n** 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). The system displays this type of identifier in hexadecimal. To differentiate general identifiers from UIC identifiers, the system adds %X80000000 to the value you specify.
- GID:n** GID is the POSIX group identifier. It is an integer value in the range 0 to 16,777,215 (%XFFFFFF). The system will add %XA400.0000 to the value you specify and then enter this new value into the system RIGHTSLIST as an identifier.
- UIC:uic** A UIC value in standard UIC format consists of a member name and, optionally, a group name enclosed in brackets. For example, [360,031].
- In numeric UICs, the group number is an octal number in the range of 1 to 37776; the member number is an octal number in the range of 0 to 177776. You can omit leading zeros when you are specifying group and member numbers.
- Regardless of the UIC format you use, the system translates a UIC to a 32-bit numeric value.
- Alphanumeric UICs are not allowed.

Typically, system managers add identifiers as UIC values to represent system users; the system applies identifiers in integer format to system resources.

**Examples**

1. UAF> ADD/IDENTIFIER/VALUE=UIC:[300,011] INVENTORY  
%UAF-I-RDBADDMSGU, 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.

2. UAF> ADD/IDENTIFIER/ATTRIBUTES=(RESOURCE) -  
/VALUE=IDENTIFIER:%X80011 PAYROLL  
%UAF-I-RDBADDMSGU, identifier PAYROLL value: %X80080011 added to  
RIGHTSLIST.DAT

This command adds the identifier PAYROLL and marks it as a resource. To differentiate identifiers with integer values from identifiers with UIC values, %X80000000 is added to the specified code.

## AUTHORIZE ADD/PROXY

---

### ADD/PROXY

Adds an entry to the network proxy authorization files, NETPROXY.DAT and NET\$PROXY.DAT, and signals DECnet to update its volatile database. Proxy additions take effect immediately on all nodes in a cluster that share the proxy database.

### Format

```
ADD/PROXY node::remote-user local-user[,...]
```

### Parameters

#### node

Specifies a DECnet node name. If you provide a wildcard character (\*), the specified remote user on all nodes is served by the account defined as **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 systems that are not OpenVMS and that implement DECnet, specifies the UIC of a user at a remote node. You can specify a wildcard character (\*) in the group and member fields of the UIC.

#### local-user

Specifies the user names of 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.

### 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.

### Description

The ADD/PROXY command adds an entry to the network proxy authorization files, NETPROXY.DAT and NET\$PROXY.DAT, and signals DECnet to update its volatile database. Proxy additions take effect immediately on all nodes in a cluster that share the proxy database.

You can grant a remote user access to one default proxy account and up to 15 other local accounts. To access proxy accounts other than the default proxy account, remote users specify the requested account name in an access control string. To change the default proxy account, use the AUTHORIZE command MODIFY/PROXY.

Proxy login is an effective way to avoid specifying (and, possibly, revealing) passwords in command lines. However, you must use caution in granting access to remote users. While logged in to the local system, remote users can apply the full DCL command set (with the exception of SET HOST). A remote user receives the default privileges of the local user and, therefore, becomes the owner of the local user's files when executing any DCL commands.

To avoid potential security compromises, HP recommends that you create proxy accounts on the local node that are less privileged than a user's normal account on the remote node. By adding an extension such as `_N`, you can identify the account as belonging to a remote user, while distinguishing it from a native account with the same name on the local node. For example, the following command creates a `JONES_N` proxy account on the local node that allows the user `JONES` to access the account from the remote node `SAMPLE`:

```
UAF> ADD/PROXY SAMPLE::JONES JONES N/DEFAULT
%UAF-I-NAFADDMSG, record successfully added to NETPROXY.DAT
```

For more information about creating proxy accounts, refer to the *HP OpenVMS Guide to System Security*.

## Examples

1. UAF> ADD/PROXY SAMPLE::WALTER ROBIN/DEFAULT  
%UAF-I-NAFADDMSG, record successfully added to NETPROXY.DAT

Specifies that user `WALTER` on remote node `SAMPLE` has proxy access to user `ROBIN`'s account on local node `AXEL`. Through proxy login, `WALTER` receives the default privileges of user `ROBIN` when he accesses node `AXEL` remotely.

2. UAF> ADD/PROXY MISHA::\* MARCO/DEFAULT, OSCAR  
%UAF-I-NAFADDMSG, record successfully added to NETPROXY.DAT

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.

3. UAF> ADD/PROXY MISHA::MARCO \*/DEFAULT  
%UAF-I-NAFADDMSG, record successfully added to NETPROXY.DAT

Specifies that user `MARCO` on the remote node `MISHA` can use only the `MARCO` account on the local node for remote file access.

4. UAF> ADD/PROXY TAO::MARTIN MARTIN/D,SALES\_READER  
%UAF-I-NAFADDMSG, proxy from TAO:.TWA.RAN:\_MARTIN to MARTIN added  
%UAF-I-NAFADDMSG, proxy from TAO:.TWA.RAN::MARTIN to SALES\_READER added

Adds a proxy from `TAO::MARTIN` to the local accounts `MARTIN` (the default) and `SALES_READER` on a system running DECnet-Plus.

# AUTHORIZE COPY

---

## COPY

Creates a new SYSUAF record that duplicates an existing UAF record.

## Format

COPY oldusername newusername

## Parameters

### oldusername

Name of an existing user record to serve as a template for the new record.

### newusername

Name for the new user record. The user name is a string of 1 to 12 alphanumeric characters.

## Qualifiers

**/ACCESS[=(range[,...])]**

**/NOACCESS[=(range[,...])]**

Specifies hours of access for all modes of access. The syntax for specifying the range is:

**/[NO]ACCESS=([PRIMARY], [n-m], [n], [...],[SECONDARY], [n-m], [n], [...])**

Specify hours as integers from 0 to 23, inclusive. You can specify single hours (n) or 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 *inclusive*; that is, if you grant access during a given hour, access extends to the end of that hour.

By default, a user has full access every day. See the DCL command SET DAY in the *HP OpenVMS DCL Dictionary* for information about overriding the defaults for primary and secondary day types.

All the list elements are optional. Unless you specify hours for a day type, access is permitted for the entire day. By specifying an access time, you prevent access at all other times. Adding NO to the qualifier denies the user access to the system for the specified period of time. See the following examples.

<b>/ACCESS</b>	Allows unrestricted access
<b>/NOACCESS=SECONDARY</b>	Allows access on primary days only
<b>/ACCESS=(9-17)</b>	Allows access from 9 A.M. to 5:59 P.M. on all days
<b>/NOACCESS=(PRIMARY, 9-17, SECONDARY, 18-8)</b>	Disallows access between 9 A.M. to 5:59 P.M. on primary days but allows access during these hours on secondary days

To specify access hours for specific types of access, see the /BATCH, /DIALUP, /INTERACTIVE, /LOCAL, /NETWORK, and /REMOTE qualifiers.

Refer to *HP OpenVMS Guide to System Security* for information about the effects of login class restrictions.



**/ACCOUNT=account-name**

Specifies the default name for the account (for example, a billing name or number). The name can be a string of 1 to 8 alphanumeric characters. By default, AUTHORIZE does not assign an account name.

**/ADD\_IDENTIFIER (default)**

**/NOADD\_IDENTIFIER**

Adds a user (user name and account name) to the rights database. The /NOADD\_IDENTIFIER does not create a rights list identifier (user name and account name). The AUTHORIZE command ADD/IDENTIFIER is quite different: it adds a record to the AUTHORIZE database UAF file.

**/ALGORITHM=keyword=type [=value]**

Sets the password encryption algorithm for a user. The keyword VMS refers to the algorithm used in the operating system version that is running on your system, whereas a customer algorithm is one that is added through the \$HASH\_PASSWORD system service by a customer site, by a layered product, or by a third party. The customer algorithm is identified in \$HASH\_PASSWORD by an integer in the range of 128 to 255. It must correspond with the number used in the AUTHORIZE command MODIFY/ALGORITHM. By default, passwords are encrypted with the VMS algorithm for the current version of the operating system.

Keyword	Function
BOTH	Set the algorithm for primary and secondary passwords.
CURRENT	Set the algorithm for the primary, secondary, both, or no passwords, depending on account status. CURRENT is the default value.
PRIMARY	Set the algorithm for the primary password only.
SECONDARY	Set the algorithm for the secondary password only.

The following table lists password encryption algorithms:

Type	Definition
VMS	The algorithm used in the version of the operating system that is running on your system.
CUSTOMER	A numeric value in the range of 128 to 255 that identifies a customer algorithm.

The following example selects the VMS algorithm for Sontag's primary password:

```
UAF> MODIFY SONTAG/ALGORITHM=PRIMARY=VMS
```

If you select a site-specific algorithm, you must give a value to identify the algorithm, as follows:

```
UAF> MODIFY SONTAG/ALGORITHM=CURRENT=CUSTOMER=128
```

**/ASTLM=value**

Specifies the AST queue limit, which is the total number of asynchronous system trap (AST) operations and scheduled wake-up requests that the user can have queued at one time. The default is 40 on VAX systems and 300 on Alpha and I64 systems.

## AUTHORIZE COPY

### **/BATCH[=(range[,...])]**

Specifies the hours of access permitted for batch jobs. For a description of the range specification, see the /ACCESS qualifier. By default, a user can submit batch jobs any time.

### **/BIOLM=value**

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. The default is 40 on VAX systems and 150 on Alpha and I64 systems.

### **/BYTLM=value**

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 can consume at one time. Nonpaged dynamic memory is used for operations such as I/O buffering, mailboxes, and file-access windows. The default is 32768 on VAX systems and 128,000 on Alpha and I64 systems.

### **/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 a string of 1 to 31 alphanumeric characters and should be DCL, which is the default. This setting is ignored for network jobs.

### **/CLITABLES=filespec**

Specifies user-defined CLI tables for the account. The *filespec* can contain 1 to 31 characters. The default is SYS\$LIBRARY:DCLTABLES. Note that this setting is ignored for network jobs to guarantee that the system-supplied command procedures used to implement network objects function properly.

### **/CPUTIME=time**

Specifies the maximum process CPU time for the CPU field of the UAF record. The maximum process CPU time is the maximum amount of CPU time a user's process can take per session. You must specify a delta time value. For a discussion of delta time values, refer to the *OpenVMS User's Manual*. The default is 0, which means an infinite amount of time.

### **/DEFPRIVILEGES=([NO]privname[,...])**

Specifies default privileges for the user; that is, those enabled at login time. A NO prefix removes a privilege from the user. By specifying the keyword [NO]ALL with the /DEFPRIVILEGES qualifier, you can disable or enable all user privileges. The default privileges are TMPMBX and NETMBX. *Privname* is the name of the privilege.

### **/DEVICE=device-name**

Specifies the name of the user's default device at login. The *device-name* is a string of 1 to 31 alphanumeric characters. If you omit the colon from the *device-name* value, AUTHORIZE appends a colon. The default device is SYS\$SYSDISK.

If you specify a logical name as the *device-name* (for example, DISK1: for DUA1:), you must make an entry for the logical name in the LNM\$SYSTEM\_TABLE in executive mode by using the DCL command DEFINE/SYSTEM/EXEC.

**/DIALUP[=(range[,...])]**

Specifies hours of access permitted for dialup logins. For a description of the range specification, see the /ACCESS qualifier. The default is full access.

**/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 default is 40 on VAX systems and 150 on Alpha and I64 systems.

**/DIRECTORY=directory-name**

Specifies the default directory name for the DIRECTORY field of the UAF record. The *directory-name* can be 1 to 39 alphanumeric characters. If you do not enclose the directory name in brackets, AUTHORIZE adds the brackets for you. The default directory name is [USER].

**/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 by the user at one time. The default is 200 on VAX systems and 4000 on Alpha and I64 systems.

**/EXPIRATION=time (default)**

**/NOEXPIRATION**

Specifies the expiration date and time of the account. The /NOEXPIRATION qualifier removes the expiration date on the account. If you do not specify an expiration time when you add a new account, AUTHORIZE copies the expiration time from the DEFAULT account. (The expiration time on the DEFAULT account is "none" by default.)

**/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. The default is 300 on VAX systems and 128 on Alpha and I64 systems.

**/FLAGS=[(NO)option[,...]]**

Specifies login flags for the user. The prefix NO clears the flag. The options are as follows:

AUDIT	Enables or disables mandatory security auditing for a specific user. By default, the system does not audit the activities of specific users (NOAUDIT).
AUTOLOGIN	Restricts the user to the automatic login mechanism when logging in to an account. When set, the flag disables login by any terminal that requires entry of a user name and password. The default is to require a user name and password (NOAUTOLOGIN).

## AUTHORIZE COPY

CAPTIVE	Prevents the user from changing any defaults at login, for example, /CLI or /LGICMD. It prevents the user from escaping the captive login command procedure specified by the /LGICMD qualifier and gaining access to the DCL command level. Refer to “Guidelines for Captive Command Procedures” in the <i>HP OpenVMS Guide to System Security</i> . The CAPTIVE flag also establishes an environment where Ctrl/Y interrupts are initially turned off; however, command procedures can still turn on Ctrl/Y interrupts with the DCL command SET CONTROL=Y. By default, an account is not captive (NOCAPTIVE).
DEFCLI	Restricts the user to the default command interpreter by prohibiting the use of the /CLI qualifier at login. By default, a user can choose a CLI (NODEFCLI).
DISCTLY	Establishes an environment where Ctrl/Y interrupts are initially turned off and are invalid until a SET CONTROL=Y is encountered. This could happen in SYLOGIN.COM or in a procedure called by SYLOGIN.COM. Once a SET CONTROL=Y is executed (which requires no privilege), a user can enter a Ctrl/Y and reach the DCL prompt (\$). If the intent of DISCTLY is to force execution of the login command files, then SYLOGIN.COM should issue the DCL command SET CONTROL=Y to turn on Ctrl/Y interrupts before exiting. By default, Ctrl/Y is enabled (NODISCTLY).
DISFORCE_PWD_CHANGE	Removes the requirement that a user must change an expired password at login. By default, a person can use an expired password only once (NODISFORCE_PWD_CHANGE) and then is forced to change the password after logging in. If the user does not select a new password, the user is locked out of the system.  To use this feature, set a password expiration date with the /PWDLIFETIME qualifier.
DISIMAGE	Prevents the user from executing RUN and foreign commands. By default, a user can execute RUN and foreign commands (NODISIMAGE).
DISMAIL	Disables mail delivery to the user. By default, mail delivery is enabled (NODISMAIL).
DISNEWMAIL	Suppresses announcements of new mail at login. By default, the system announces new mail (NODISNEWMAIL).
DISPWDDIC	Disables automatic screening of new passwords against a system dictionary. By default, passwords are automatically screened (NODISPWDDIC).
DISPWDHIS	Disables automatic checking of new passwords against a list of the user’s old passwords. By default, the system screens new passwords (NODISPWDHIS).

DISPWDSYNCH	Suppresses synchronization of the external password for this account. See bit 9 in the SECURITY_POLICY system parameter for systemwide password synchronization control.
DISRECONNECT	Disables automatic reconnection to an existing process when a terminal connection has been interrupted. By default, automatic reconnection is enabled (NODISRECONNECT).
DISREPORT	Suppresses reports of the last login time, login failures, and other security reports. By default, login information is displayed (NODISREPORT).
DISUSER	Disables the account so the user cannot log in. For example, the DEFAULT account is disabled. By default, an account is enabled (NODISUSER).
DISWELCOME	Suppresses the welcome message (an informational message displayed during a local login). This message usually indicates the version number of the operating system that is running and the name of the node on which the user is logged in. By default, a system login message appears (NODISWELCOME).
EXTAUTH	Considers user to be authenticated by an external user name and password, not by the SYSUAF user name and password. (The system still uses the SYSUAF record to check a user's login restrictions and quotas and to create the user's process profile.)
GENPWD	Restricts the user to generated passwords. By default, users choose their own passwords (NOGENPWD).
LOCKPWD	Prevents the user from changing the password for the account. By default, users can change their passwords (NOLOCKPWD).
PWD_EXPIRED	Marks a password as expired. The user cannot log in if this flag is set. The LOGINOUT.EXE image sets the flag when both of the following conditions exist: a user logs in with the DISFORCE_PWD_CHANGE flag set, and the user's password expires. A system manager can clear this flag. By default, passwords are not expired after login (NOPWD_EXPIRED).
PWD2_EXPIRED	Marks a secondary password as expired. Users cannot log in if this flag is set. The LOGINOUT.EXE image sets the flag when both of the following conditions exist: a user logs in with the DISFORCE_PWD_CHANGE flag set, and the user's password expires. A system manager can clear this flag. By default, passwords are not set to expire after login (NOPWD2_EXPIRED).

## AUTHORIZE COPY

PWDMIX	<p>Enables case-sensitive and extended-character passwords. After PWDMIX is specified, you can then use mixed-case and extended characters in passwords. Be aware that before the PWDMIX flag is enabled, the system stores passwords in all upper-case. Therefore, until you change passwords, you must enter your pre-PWDMIX passwords in upper-case.</p> <p>To change the password after PWDMIX is enabled:</p> <ul style="list-style-type: none"><li>• You (the user) can use the DCL command SET PASSWORD, specifying the new mixed-case password (omitting quotation marks).</li><li>• You (the system manager) can use the AUTHORIZE command MODIFY/PASSWORD, and enclose the user's new mixed-case password in quotation marks " ".</li></ul>
RESTRICTED	<p>Prevents the user from changing any defaults at login (for example, by specifying /LGICMD) and prohibits user specification of a CLI with the /CLI qualifier. The RESTRICTED flag establishes an environment where Ctrl/Y interrupts are initially turned off; however, command procedures can still turn on Ctrl/Y interrupts with the DCL command SET CONTROL=Y. Typically, this flag is used to prevent an applications user from having unrestricted access to the CLI. By default, a user can change defaults (NORESTRICTED).</p>
VMSAUTH	<p>Allows account to use standard (SYSUAF) authentication when the EXTAUTH flag would otherwise require external authentication. This depends on the application. An application specifies the VMS domain of interpretation when calling SYS\$ACM to request standard VMS authentication for a user account that normally uses external authentication.</p>

### **/GENERATE\_PASSWORD[=keyword]**

### **/NOGENERATE\_PASSWORD (default)**

Invokes the password generator to create user passwords. Generated passwords can consist of 1 to 10 characters. Specify one of the following keywords:

BOTH	Generate primary and secondary passwords.
CURRENT	Do whatever the DEFAULT account does (for example, generate primary, secondary, both, or no passwords). This is the default keyword.
PRIMARY	Generate primary password only.
SECONDARY	Generate secondary password only.

When you modify a password, the new password expires automatically; it is valid only once (unless you specify /NOPWDEXPIRED). On login, users are forced to change their passwords (unless you specify /FLAGS=DISFORCE\_PWD\_CHANGE).

Note that the /GENERATE\_PASSWORD and /PASSWORD qualifiers are mutually exclusive.

**/INTERACTIVE[ =(range[,...])]  
/NOINTERACTIVE**

Specifies the hours of access for interactive logins. For a description of the range specification, see the /ACCESS qualifier. By default, there are no access restrictions on interactive logins.

**/JTQUOTA=value**

Specifies the initial byte quota with which the jobwide logical name table is to be created. By default, the value is 4096 on VAX systems and 4096 on Alpha and I64 systems.

**/LGICMD=filespec**

Specifies the name of the default login command file. The file name defaults to the device specified for /DEVICE, the directory specified for /DIRECTORY, a file name of LOGIN, and a file type of .COM. If you select the defaults for all these values, the file name is SYS\$SYSTEM:[USER]LOGIN.COM.

**/LOCAL[=(range[,...])]**

Specifies hours of access for interactive logins from local terminals. For a description of the range specification, see the /ACCESS qualifier. By default, there are no access restrictions on local logins.

**/MAXACCTJOBS=value**

Specifies the maximum number of batch, interactive, and detached processes that can be active at one time for all users of the same account. By default, a user has a maximum of 0, which represents an unlimited number.

**/MAXDETACH=value**

Specifies the maximum number of detached processes with the cited user name that can be active at one time. To prevent the user from creating detached processes, specify the keyword NONE. By default, a user has a value of 0, which represents an unlimited number.

**/MAXJOBS=value**

Specifies the maximum number of processes (interactive, batch, detached, and network) with the cited user name that can be active simultaneously. The first four network jobs are not counted. By default, a user has a maximum value of 0, which represents an unlimited number.

**/NETWORK[=(range[,...])]**

Specifies hours of access for network batch jobs. For a description of how to specify the range, see the /ACCESS qualifier. By default, network logins have no access restrictions.

**/OWNER=owner-name**

Specifies the name of the owner of the account. You can use this name for billing purposes or similar applications. The owner name is 1 to 31 characters. No default owner name exists.

**/PASSWORD=(password1[,password2])  
/NOPASSWORD**

Specifies up to two passwords for login. Passwords can be from 0 to 32 alphanumeric characters in length. The dollar sign (\$) and underscore (\_) are also permitted.

## AUTHORIZE COPY

Uppercase and lowercase characters are equivalent. All lowercase characters are converted to uppercase before the password is encrypted. Avoid using the word *password* as the actual password.

Use the /PASSWORD qualifier as follows:

- To set only the first password and clear the second, specify /PASSWORD=password.
- 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.

When you modify a password, the new password expires automatically; it is valid only once (unless you specify /NOPWDEXPIRED). On login, the user is forced to change the password (unless you specify /FLAGS=DISFORCE\_PWD\_CHANGE).

Note that the /GENERATE\_PASSWORD and /PASSWORD qualifiers are mutually exclusive.

When you create a new UAF record with the COPY command, you must specify a password.

### **/PBYTLM**

This flag is reserved for HP.

### **/PGFLQUOTA=value**

Specifies the paging file limit. This is the maximum number of pages that the person's process can use in the system paging file. By default, the value is 32768 pages on VAX systems and 256,000 pagelets on Alpha and I64 systems.

If decompressing libraries, make sure to set PGFLQUOTA to twice the size of the library.

### **/PRCLM=value**

Specifies the subprocess creation limit. This is the maximum number of subprocesses that can exist at one time for the specified user's process. By default, the value is 2 on VAX systems and 8 on Alpha and I64 systems.

### **/PRIMEDAYS=(**[NO]**day[,...])**

Defines the primary and secondary days of the week for logging in. Specify the days as a list separated by commas, and enclose the list in parentheses. To specify a secondary day, prefix the day with NO (for example, NOFRIDAY). To specify a primary day, omit the NO prefix.

By default, primary days are Monday through Friday and secondary days are Saturday and Sunday. If you omit a day from the list, AUTHORIZE uses the default value. (For example, if you omit Monday from the list, AUTHORIZE defines Monday as a primary day.)

Use the primary and secondary day definitions in conjunction with such qualifiers as /ACCESS, /INTERACTIVE, and /BATCH.



**/PRIORITY=value**

Specifies the default base priority. The value is an integer in the range of 0 to 31 on VAX systems and 0 to 63 on Alpha and I64 systems. By default, the value is set to 4 for timesharing users.

**/PRIVILEGES=(*[NO]*privname[,...])**

Specifies which privileges the user is authorized to hold, although these privileges are not necessarily enabled at login. (The */DEFPRIVILEGES* qualifier determines which ones are enabled.) A *NO* prefix removes the privilege from the user. The keyword *NOALL* disables all user privileges. Many privileges have varying degrees of power and potential system impact (see the *HP OpenVMS Guide to System Security* for a detailed discussion). By default, a user holds *TMPMBX* and *NETMBX* privileges. *Privname* is the name of the privilege.

**/PWDEXPIRED (default)**

**/NOPWDEXPIRED**

Specifies the password is valid for only one login. A user must change a password immediately after login or be locked out of the system. The system warns users of password expiration. A user can either specify a new password, with the DCL command *SET PASSWORD*, or wait until expiration and be forced to change. By default, a user must change a password when first logging in to an account. The default is applied to the account only when the password is being modified.

**/PWDLIFETIME=time (default)**

**/NOPWDLIFETIME**

Specifies the length of time a password is valid. Specify a delta time value in the form *[ddd-] [hh:mm:ss.cc]*. For example, for a lifetime of 120 days, 0 hours, and 0 seconds, specify */PWDLIFETIME="120-"*. For a lifetime of 120 days 12 hours, 30 minutes and 30 seconds, specify */PWDLIFETIME="120-12:30:30"*. If a period longer than the specified time elapses before the user logs in, the system displays a warning message. The password is marked as expired.

To prevent a password from expiring, specify the time as *NONE*. By default, a password expires in 90 days.

**/PWDMINIMUM=value**

Specifies the minimum password length in characters. Note that this value is enforced only by the DCL command *SET PASSWORD*. It does not prevent you from entering a password shorter than the minimum length when you use *AUTHORIZE* to create or modify an account. By default, a password must have at least 6 characters. The value specified by the */PWDMINIMUM* qualifier conflicts with the value used by the */GENERATE\_PASSWORD* qualifier or the DCL command *SET PASSWORD/GENERATE*, the operating system chooses the lesser value. The maximum value for generated passwords is 10.

**/QUEPRIO=value**

Reserved for future use.

**/REMOTE[=(range[,...])]**

Specifies hours during which access is permitted for interactive logins from network remote terminals (with the DCL command *SET HOST*). For a description of the range specification, see the */ACCESS* qualifier. By default, remote logins have no access restrictions.

## AUTHORIZE COPY

### **/SHRFILLM=value**

Specifies the maximum number of shared files that the user can have open at one time. By default, the system assigns a value of 0, which represents an infinite number.

### **/TQELM**

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. By default, a user can have 100.

### **/UIC=value**

Specifies the user identification code (UIC). The UIC value is a group number in the range from 1 to 37776 (octal) and a member number in the range from 0 to 177776 (octal), which are separated by a comma and enclosed in brackets. HP reserves group 1 and groups 300–377 for its own use.

Each user must have a unique UIC. By default, the UIC value is [200,200].

### **/WSDEFAULT=value**

Specifies the default working set limit. This represents the initial limit to the number of physical pages the process can use. (The user can alter the default quantity up to WSQUOTA with the DCL command SET WORKING\_SET.) By default, a user has 256 pages on VAX systems and 4096 pagelets on Alpha and I64 systems.

The value cannot be greater than WSMAX. This quota value replaces smaller values of PQL\_MWSDEFAULT.

### **/WSEXTENT=value**

Specifies the working set maximum. This represents the maximum amount of physical memory allowed to the process. The system provides memory to a process beyond its working set quota only when it has excess free pages. The additional memory is recalled by the system if needed.

The value is an integer equal to or greater than WSQUOTA. By default, the value is 1024 pages on VAX systems and 16384 pagelets on Alpha and I64 systems. The value cannot be greater than WSMAX. This quota value replaces smaller values of PQL\_MWSEXTENT.

### **/WSQUOTA=value**

Specifies the working set quota. This is the maximum amount of physical memory a user process can lock into its working set. It also represents the maximum amount of swap space that the system reserves for this process and the maximum amount of physical memory that the system allows the process to consume if the systemwide memory demand is significant.

The value cannot be greater than the value of WSMAX and cannot exceed 8,192 pagelets on Alpha and I64 systems. This quota value replaces smaller values of PQL\_MWSQUOTA.

## Description

The COPY command creates a new SYSUAF record that duplicates an existing SYSUAF record. The command requires the /PASSWORD qualifier. If you do not specify additional qualifiers to the COPY command, the fields in the record you create are the same as those in the record being copied.

For example, you could add a record for a new user named Thomas Sparrow that is identical to that of Joseph Robin (but presumably different from the default record), as follows:

```
UAF> COPY ROBIN SPARROW /PASSWORD=SP0152
```

However, to add a record for Thomas Sparrow that differs from Joseph Robin's in the UIC, directory name, password, and owner, specify the following command:

```
UAF> COPY ROBIN SPARROW /UIC=[200,13]/DIRECTORY=[SPARROW] -  
_/PASSWORD=THOMAS/OWNER="THOMAS SPARROW"
```

You can also use the COPY command to create a set of template records to meet the specific needs of various user groups. For example, if you have programmers, administrators, and data entry personnel working on the same system, you can create records such as PROGRAMMER, ADMINISTRATOR, and DATA\_ENTRY, each tailored to the needs of a particular group. To add an account for a new user in one of these groups, copy the appropriate template record and specify a new user name, password, UIC, directory, and owner.

If you omit the /PASSWORD qualifier when you create an account, AUTHORIZE displays the following error message:

```
%UAF-W-DEFPWD, copied or renamed records must receive new password
```

To specify a password for the account, use the MODIFY command with the /PASSWORD qualifier.

## Examples

1. 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 because the UIC value has no change, no identifier is added to RIGHTSLLIST.DAT. AUTHORIZE issues a "duplicate identifier" error message.

2. UAF> COPY ROBIN SPARROW /UIC=[200,13]/DIRECTORY=[SPARROW] -  
\_/PASSWORD=THOMAS/OWNER="THOMAS SPARROW"  
%UAF-I-COPMSG, user record copied  
%UAF-I-RDBADDMSGU, identifier SPARROW value: [000200,000013] added to  
RIGHTSLIST.DAT

The command in this example adds a record for Thomas Sparrow that is the same as Joseph Robin's except for the UIC, directory name, password, and owner. Note that you could use a similar command to copy a template record when adding a record for a new user in a particular user group.

# AUTHORIZE CREATE/PROXY

---

## CREATE/PROXY

Creates and initializes the network proxy authorization files. The primary network proxy authorization file is NET\$PROXY.DAT. The file NETPROXY.DAT is maintained for compatibility.

---

### Note

---

Do not delete NETPROXY.DAT because DECnet Phase IV and many layered products still use it.

---

## Format

CREATE/PROXY

## Parameters

None.

## Qualifiers

None.

## Description

NETPROXY.DAT is created with no records and is assigned the following protection:

(S:RWED,O:RWED,G,W)

NET\$PROXY.DAT is created with no records and is assigned the following protection:

(S:RWED,O,G,W)

If NETPROXY.DAT or NET\$PROXY.DAT already exist, AUTHORIZE reports the following error message:

```
%UAF-W-NAFAEX, NETPROXY.DAT already exists
```

To create a new file, you must either delete or rename the old one.

## Example

```
UAF> CREATE/PROXY  
UAF>
```

The command in this example creates and initializes the network proxy authorization file.

## **CREATE/RIGHTS**

Creates and initializes the rights database, RIGHTSLIST.DAT.

### **Format**

CREATE/RIGHTS

### **Parameters**

None.

### **Qualifiers**

None.

### **Description**

RIGHTSLIST.DAT is created with no records and is assigned the following protection:

(S:RWED,O:RWED,G:R,W:)

Note that the file is created only if the file does not already exist.

### **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. For more information about rights database management, refer to the *HP OpenVMS Guide to System Security*.

# AUTHORIZE DEFAULT

---

## DEFAULT

Modifies the SYSUAF's DEFAULT record.

## Format

DEFAULT

## Parameters

None.

## Qualifiers

**/ACCESS[=(range[,...])]**

**/NOACCESS[=(range[,...])]**

Specifies hours of access for all modes of access. The syntax for specifying the range is:

**/[NO]ACCESS=([PRIMARY], [n-m], [n], [...],[SECONDARY], [n-m], [n], [...])**

Specify hours as integers from 0 to 23, inclusive. You can specify single hours (n) or 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 *inclusive*; that is, if you grant access during a given hour, access extends to the end of that hour.

By default, a user has full access every day. See the DCL command SET DAY in the *HP OpenVMS DCL Dictionary* for information about overriding the defaults for primary and secondary day types.

All the list elements are optional. Unless you specify hours for a day type, access is permitted for the entire day. By specifying an access time, you prevent access at all other times. Adding NO to the qualifier denies the user access to the system for the specified period of time. See the following examples.

<b>/ACCESS</b>	Allows unrestricted access
<b>/NOACCESS=SECONDARY</b>	Allows access on primary days only
<b>/ACCESS=(9-17)</b>	Allows access from 9 A.M. to 5:59 P.M. on all days
<b>/NOACCESS=(PRIMARY, 9-17, SECONDARY, 18-8)</b>	Disallows access between 9 A.M. to 5:59 P.M. on primary days but allows access during these hours on secondary days

To specify access hours for specific types of access, see the /BATCH, /DIALUP, /INTERACTIVE, /LOCAL, /NETWORK, and /REMOTE qualifiers.

Refer to *HP OpenVMS Guide to System Security* for information about the effects of login class restrictions.

**/ACCOUNT=account-name**

Specifies the default name for the account (for example, a billing name or number). The name can be a string of 1 to 8 alphanumeric characters. By default, AUTHORIZE does not assign an account name.

**/ALGORITHM=keyword=type [=value]**

Sets the password encryption algorithm for a user. The keyword VMS refers to the algorithm used in the operating system version that is running on your system, whereas a customer algorithm is one that is added through the \$HASH\_PASSWORD system service by a customer site, by a layered product, or by a third party. The customer algorithm is identified in \$HASH\_PASSWORD by an integer in the range of 128 to 255. It must correspond with the number used in the AUTHORIZE command MODIFY/ALGORITHM. By default, passwords are encrypted with the VMS algorithm for the current version of the operating system.

Keyword	Function
BOTH	Set the algorithm for primary and secondary passwords.
CURRENT	Set the algorithm for the primary, secondary, both, or no passwords, depending on account status. CURRENT is the default value.
PRIMARY	Set the algorithm for the primary password only.
SECONDARY	Set the algorithm for the secondary password only.

The following table lists password encryption algorithms:

Type	Definition
VMS	The algorithm used in the version of the operating system that is running on your system.
CUSTOMER	A numeric value in the range of 128 to 255 that identifies a customer algorithm.

The following example selects the VMS algorithm for Sontag's primary password:

```
UAF> MODIFY SONTAG/ALGORITHM=PRIMARY=VMS
```

If you select a site-specific algorithm, you must give a value to identify the algorithm, as follows:

```
UAF> MODIFY SONTAG/ALGORITHM=CURRENT=CUSTOMER=128
```

**/ASTLM=value**

Specifies the AST queue limit, which is the total number of asynchronous system trap (AST) operations and scheduled wake-up requests that the user can have queued at one time. The default is 40 on VAX systems and 300 on Alpha and I64 systems.

**/BATCH[=(range[,...])]**

Specifies the hours of access permitted for batch jobs. For a description of the range specification, see the /ACCESS qualifier. By default, a user can submit batch jobs any time.

**/BIOLM=value**

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. The default is 40 on VAX systems and 150 on Alpha and I64 systems.

## AUTHORIZE DEFAULT

### **/BYTLM=value**

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 can consume at one time. Nonpaged dynamic memory is used for operations such as I/O buffering, mailboxes, and file-access windows. The default is 32768 on VAX systems and 128,000 on Alpha and I64 systems.

### **/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 a string of 1 to 31 alphanumeric characters and should be DCL, which is the default. This setting is ignored for network jobs.

### **/CLITABLES=filespec**

Specifies user-defined CLI tables for the account. The *filespec* can contain 1 to 31 characters. The default is SYS\$LIBRARY:DCLTABLES. Note that this setting is ignored for network jobs to guarantee that the system-supplied command procedures used to implement network objects function properly.

### **/CPUTIME=time**

Specifies the maximum process CPU time for the CPU field of the UAF record. The maximum process CPU time is the maximum amount of CPU time a user's process can take per session. You must specify a delta time value. For a discussion of delta time values, refer to the *OpenVMS User's Manual*. The default is 0, which means an infinite amount of time.

### **/DEFPRIVILEGES=(**[NO]**privname[,...])**

Specifies default privileges for the user; that is, those enabled at login time. A NO prefix removes a privilege from the user. By specifying the keyword [NO]ALL with the /DEFPRIVILEGES qualifier, you can disable or enable all user privileges. The default privileges are TMPMBX and NETMBX. *Privname* is the name of the privilege.

### **/DEVICE=device-name**

Specifies the name of the user's default device at login. The *device-name* is a string of 1 to 31 alphanumeric characters. If you omit the colon from the *device-name* value, AUTHORIZE appends a colon. The default device is SYS\$SYSDISK.

If you specify a logical name as the *device-name* (for example, DISK1: for DUA1:), you must make an entry for the logical name in the LNM\$SYSTEM\_TABLE in executive mode by using the DCL command DEFINE/SYSTEM/EXEC.

### **/DIALUP[=(range[,...])]**

Specifies hours of access permitted for dialup logins. For a description of the range specification, see the /ACCESS qualifier. The default is full access.

### **/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 default is 40 on VAX systems and 150 on Alpha and I64 systems.



**/DIRECTORY=directory-name**

Specifies the default directory name for the DIRECTORY field of the UAF record. The *directory-name* can be 1 to 39 alphanumeric characters. If you do not enclose the directory name in brackets, AUTHORIZE adds the brackets for you. The default directory name is [USER].

**/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 by the user at one time. The default is 200 on VAX systems and 4000 on Alpha and I64 systems.

**/EXPIRATION=time (default)**

**/NOEXPIRATION**

Specifies the expiration date and time of the account. The /NOEXPIRATION qualifier removes the expiration date on the account. If you do not specify an expiration time when you add a new account, AUTHORIZE copies the expiration time from the DEFAULT account. (The expiration time on the DEFAULT account is "none" by default.)

**/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. The default is 300 on VAX systems and 128 on Alpha and I64 systems.

**/FLAGS=(*[NO]option[,...]*)**

Specifies login flags for the user. The prefix NO clears the flag. The options are as follows:

AUDIT	Enables or disables mandatory security auditing for a specific user. By default, the system does not audit the activities of specific users (NOAUDIT).
AUTOLOGIN	Restricts the user to the automatic login mechanism when logging in to an account. When set, the flag disables login by any terminal that requires entry of a user name and password. The default is to require a user name and password (NOAUTOLOGIN).
CAPTIVE	Prevents the user from changing any defaults at login, for example, /CLI or /LGICMD. It prevents the user from escaping the captive login command procedure specified by the /LGICMD qualifier and gaining access to the DCL command level. Refer to "Guidelines for Captive Command Procedures" in the <i>HP OpenVMS Guide to System Security</i> . The CAPTIVE flag also establishes an environment where Ctrl/Y interrupts are initially turned off; however, command procedures can still turn on Ctrl/Y interrupts with the DCL command SET CONTROL=Y. By default, an account is not captive (NOCAPTIVE).
DEFCLI	Restricts the user to the default command interpreter by prohibiting the use of the /CLI qualifier at login. By default, a user can choose a CLI (NODEFCLI).

## AUTHORIZE DEFAULT

DISCTLY	Establishes an environment where Ctrl/Y interrupts are initially turned off and are invalid until a SET CONTROL=Y is encountered. This could happen in SYLOGIN.COM or in a procedure called by SYLOGIN.COM. Once a SET CONTROL=Y is executed (which requires no privilege), a user can enter a Ctrl/Y and reach the DCL prompt (\$). If the intent of DISCTLY is to force execution of the login command files, then SYLOGIN.COM should issue the DCL command SET CONTROL=Y to turn on Ctrl/Y interrupts before exiting. By default, Ctrl/Y is enabled (NODISCTLY).
DISFORCE_PWD_CHANGE	Removes the requirement that a user must change an expired password at login. By default, a person can use an expired password only once (NODISFORCE_PWD_CHANGE) and then is forced to change the password after logging in. If the user does not select a new password, the user is locked out of the system.  To use this feature, set a password expiration date with the /PWDLIFETIME qualifier.
DISIMAGE	Prevents the user from executing RUN and foreign commands. By default, a user can execute RUN and foreign commands (NODISIMAGE).
DISMAIL	Disables mail delivery to the user. By default, mail delivery is enabled (NODISMAIL).
DISNEWMAIL	Suppresses announcements of new mail at login. By default, the system announces new mail (NODISNEWMAIL).
DISPWDDIC	Disables automatic screening of new passwords against a system dictionary. By default, passwords are automatically screened (NODISPWDDIC).
DISPWDHIS	Disables automatic checking of new passwords against a list of the user's old passwords. By default, the system screens new passwords (NODISPWDHIS).
DISPWDSYNCH	Suppresses synchronization of the external password for this account. See bit 9 in the SECURITY_POLICY system parameter for systemwide password synchronization control.
DISRECONNECT	Disables automatic reconnection to an existing process when a terminal connection has been interrupted. By default, automatic reconnection is enabled (NODISRECONNECT).
DISREPORT	Suppresses reports of the last login time, login failures, and other security reports. By default, login information is displayed (NODISREPORT).
DISUSER	Disables the account so the user cannot log in. For example, the DEFAULT account is disabled. By default, an account is enabled (NODISUSER).

DISWELCOME	Suppresses the welcome message (an informational message displayed during a local login). This message usually indicates the version number of the operating system that is running and the name of the node on which the user is logged in. By default, a system login message appears (NODISWELCOME).
EXTAUTH	Considers user to be authenticated by an external user name and password, not by the SYSUAF user name and password. (The system still uses the SYSUAF record to check a user's login restrictions and quotas and to create the user's process profile.)
GENPWD	Restricts the user to generated passwords. By default, users choose their own passwords (NOGENPWD).
LOCKPWD	Prevents the user from changing the password for the account. By default, users can change their passwords (NOLOCKPWD).
PWD_EXPIRED	Marks a password as expired. The user cannot log in if this flag is set. The LOGINOUT.EXE image sets the flag when both of the following conditions exist: a user logs in with the DISFORCE_PWD_CHANGE flag set, and the user's password expires. A system manager can clear this flag. By default, passwords are not expired after login (NOPWD_EXPIRED).
PWD2_EXPIRED	Marks a secondary password as expired. Users cannot log in if this flag is set. The LOGINOUT.EXE image sets the flag when both of the following conditions exist: a user logs in with the DISFORCE_PWD_CHANGE flag set, and the user's password expires. A system manager can clear this flag. By default, passwords are not set to expire after login (NOPWD2_EXPIRED).
PWDMIX	<p>Enables case-sensitive and extended-character passwords. After PWDMIX is specified, you can then use mixed-case and extended characters in passwords. Be aware that before the PWDMIX flag is enabled, the system stores passwords in all upper-case. Therefore, until you change passwords, you must enter your pre-PWDMIX passwords in upper-case.</p> <p>To change the password after PWDMIX is enabled:</p> <ul style="list-style-type: none"><li>• You (the user) can use the DCL command SET PASSWORD, specifying the new mixed-case password (omitting quotation marks).</li><li>• You (the system manager) can use the AUTHORIZE command MODIFY/PASSWORD, and enclose the user's new mixed-case password in quotation marks " ".</li></ul>

## AUTHORIZE DEFAULT

RESTRICTED	Prevents the user from changing any defaults at login (for example, by specifying /LGICMD) and prohibits user specification of a CLI with the /CLI qualifier. The RESTRICTED flag establishes an environment where Ctrl/Y interrupts are initially turned off; however, command procedures can still turn on Ctrl/Y interrupts with the DCL command SET CONTROL=Y. Typically, this flag is used to prevent an applications user from having unrestricted access to the CLI. By default, a user can change defaults (NORESTRICTED).
VMSAUTH	Allows account to use standard (SYSUAF) authentication when the EXTAUTH flag would otherwise require external authentication. This depends on the application. An application specifies the VMS domain of interpretation when calling SYS\$ACM to request standard VMS authentication for a user account that normally uses external authentication.

### **/GENERATE\_PASSWORD[=keyword]**

### **/NOGENERATE\_PASSWORD (default)**

Invokes the password generator to create user passwords. Generated passwords can consist of 1 to 10 characters. Specify one of the following keywords:

BOTH	Generate primary and secondary passwords.
CURRENT	Do whatever the DEFAULT account does (for example, generate primary, secondary, both, or no passwords). This is the default keyword.
PRIMARY	Generate primary password only.
SECONDARY	Generate secondary password only.

When you modify a password, the new password expires automatically; it is valid only once (unless you specify /NOPWDEXPIRED). On login, users are forced to change their passwords (unless you specify /FLAGS=DISFORCE\_PWD\_CHANGE).

Note that the /GENERATE\_PASSWORD and /PASSWORD qualifiers are mutually exclusive.

### **/INTERACTIVE[ =(range[,...])]**

### **/NOINTERACTIVE**

Specifies the hours of access for interactive logins. For a description of the range specification, see the /ACCESS qualifier. By default, there are no access restrictions on interactive logins.

### **/JTQUOTA=value**

Specifies the initial byte quota with which the jobwide logical name table is to be created. By default, the value is 4096 on VAX systems and 4096 on Alpha and I64 systems.

### **/LGICMD=filespec**

Specifies the name of the default login command file. The file name defaults to the device specified for /DEVICE, the directory specified for /DIRECTORY, a file name of LOGIN, and a file type of .COM. If you select the defaults for all these values, the file name is SYS\$SYSTEM:[USER]LOGIN.COM.

**/LOCAL[=(range[,...])]**

Specifies hours of access for interactive logins from local terminals. For a description of the range specification, see the /ACCESS qualifier. By default, there are no access restrictions on local logins.

**/MAXACCTJOBS=value**

Specifies the maximum number of batch, interactive, and detached processes that can be active at one time for all users of the same account. By default, a user has a maximum of 0, which represents an unlimited number.

**/MAXDETACH=value**

Specifies the maximum number of detached processes with the cited user name that can be active at one time. To prevent the user from creating detached processes, specify the keyword NONE. By default, a user has a value of 0, which represents an unlimited number.

**/MAXJOBS=value**

Specifies the maximum number of processes (interactive, batch, detached, and network) with the cited user name that can be active simultaneously. The first four network jobs are not counted. By default, a user has a maximum value of 0, which represents an unlimited number.

**/MODIFY\_IDENTIFIER (default)****/NOMODIFY\_IDENTIFIER**

Specifies whether the identifier associated with the user is to be modified in the rights database. This qualifier applies only when you modify the UIC or user name in the UAF record. By default, the associated identifiers are modified.

**/NETWORK[=(range[,...])]**

Specifies hours of access for network batch jobs. For a description of how to specify the range, see the /ACCESS qualifier. By default, network logins have no access restrictions.

**/OWNER=owner-name**

Specifies the name of the owner of the account. You can use this name for billing purposes or similar applications. The owner name is 1 to 31 characters. No default owner name exists.

**/PASSWORD=(password1[,password2])****/NOPASSWORD**

Specifies up to two passwords for login. Passwords can be from 0 to 32 alphanumeric characters in length. The dollar sign (\$) and underscore (\_) are also permitted.

Uppercase and lowercase characters are equivalent. All lowercase characters are converted to uppercase before the password is encrypted. Avoid using the word *password* as the actual password.

Use the /PASSWORD qualifier as follows:

- To set only the first password and clear the second, specify /PASSWORD=password.
- To set both the first and second password, specify /PASSWORD=(password1, password2).
- To change the first password without affecting the second, specify /PASSWORD=(password, "").

## AUTHORIZE DEFAULT

- To change the second password without affecting the first, specify `/PASSWORD="(", password)`.
- To set both passwords to null, specify `/NOPASSWORD`.

When you modify a password, the new password expires automatically; it is valid only once (unless you specify `/NOPWDEXPIRED`). On login, the user is forced to change the password (unless you specify `/FLAGS=DISFORCE_PWD_CHANGE`).

Note that the `/GENERATE_PASSWORD` and `/PASSWORD` qualifiers are mutually exclusive.

### **/PBYTLM**

This flag is reserved for HP.

### **/PGFLQUOTA=value**

Specifies the paging file limit. This is the maximum number of pages that the person's process can use in the system paging file. By default, the value is 32768 pages on VAX systems and 256,000 pagelets on Alpha and I64 systems.

If decompressing libraries, make sure to set `PGFLQUOTA` to twice the size of the library.

### **/PRCLM=value**

Specifies the subprocess creation limit. This is the maximum number of subprocesses that can exist at one time for the specified user's process. By default, the value is 2 on VAX systems and 8 on Alpha and I64 systems.

### **/PRIMEDAYS=(*[NO]*day[,...])**

Defines the primary and secondary days of the week for logging in. Specify the days as a list separated by commas, and enclose the list in parentheses. To specify a secondary day, prefix the day with `NO` (for example, `NOFRIDAY`). To specify a primary day, omit the `NO` prefix.

By default, primary days are Monday through Friday and secondary days are Saturday and Sunday. If you omit a day from the list, `AUTHORIZE` uses the default value. (For example, if you omit Monday from the list, `AUTHORIZE` defines Monday as a primary day.)

Use the primary and secondary day definitions in conjunction with such qualifiers as `/ACCESS`, `/INTERACTIVE`, and `/BATCH`.

### **/PRIORITY=value**

Specifies the default base priority. The value is an integer in the range of 0 to 31 on VAX systems and 0 to 63 on Alpha and I64 systems. By default, the value is set to 4 for timesharing users.

### **/PRIVILEGES=(*[NO]*privname[,...])**

Specifies which privileges the user is authorized to hold, although these privileges are not necessarily enabled at login. (The `/DEFPRIVILEGES` qualifier determines which ones are enabled.) A `NO` prefix removes the privilege from the user. The keyword `NOALL` disables all user privileges. Many privileges have varying degrees of power and potential system impact (see the *HP OpenVMS Guide to System Security* for a detailed discussion). By default, a user holds `TMPMBX` and `NETMBX` privileges. *Privname* is the name of the privilege.

**/PWDEXPIRED (default)****/NOPWDEXPIRED**

Specifies the password is valid for only one login. A user must change a password immediately after login or be locked out of the system. The system warns users of password expiration. A user can either specify a new password, with the DCL command SET PASSWORD, or wait until expiration and be forced to change. By default, a user must change a password when first logging in to an account. The default is applied to the account only when the password is being modified.

**/PWDLIFETIME=time (default)****/NOPWDLIFETIME**

Specifies the length of time a password is valid. Specify a delta time value in the form [dddd-] [hh:mm:ss.cc]. For example, for a lifetime of 120 days, 0 hours, and 0 seconds, specify /PWDLIFETIME="120-". For a lifetime of 120 days 12 hours, 30 minutes and 30 seconds, specify /PWDLIFETIME="120-12:30:30". If a period longer than the specified time elapses before the user logs in, the system displays a warning message. The password is marked as expired.

To prevent a password from expiring, specify the time as NONE. By default, a password expires in 90 days.

**/PWDMINIMUM=value**

Specifies the minimum password length in characters. Note that this value is enforced only by the DCL command SET PASSWORD. It does not prevent you from entering a password shorter than the minimum length when you use AUTHORIZE to create or modify an account. By default, a password must have at least 6 characters. The value specified by the /PWDMINIMUM qualifier conflicts with the value used by the /GENERATE\_PASSWORD qualifier or the DCL command SET PASSWORD/GENERATE, the operating system chooses the lesser value. The maximum value for generated passwords is 10.

**/QUEPRIO=value**

Reserved for future use.

**/REMOTE[=(range[,...])]**

Specifies hours during which access is permitted for interactive logins from network remote terminals (with the DCL command SET HOST). For a description of the range specification, see the /ACCESS qualifier. By default, remote logins have no access restrictions.

**/SHRFILLM=value**

Specifies the maximum number of shared files that the user can have open at one time. By default, the system assigns a value of 0, which represents an infinite number.

**/TQELM**

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. By default, a user can have 100.

**/UIC=value**

Specifies the user identification code (UIC). The UIC value is a group number in the range from 1 to 37776 (octal) and a member number in the range from 0 to 17776 (octal), which are separated by a comma and enclosed in brackets. HP reserves group 1 and groups 300–377 for its own use.

Each user must have a unique UIC. By default, the UIC value is [200,200].

## AUTHORIZE DEFAULT

### **/WSDEFAULT=value**

Specifies the default working set limit. This represents the initial limit to the number of physical pages the process can use. (The user can alter the default quantity up to WSQUOTA with the DCL command SET WORKING\_SET.) By default, a user has 256 pages on VAX systems and 4096 pagelets on Alpha and I64 systems.

The value cannot be greater than WSMAX. This quota value replaces smaller values of PQL\_MWSDEFAULT.

### **/WSEXTENT=value**

Specifies the working set maximum. This represents the maximum amount of physical memory allowed to the process. The system provides memory to a process beyond its working set quota only when it has excess free pages. The additional memory is recalled by the system if needed.

The value is an integer equal to or greater than WSQUOTA. By default, the value is 1024 pages on VAX systems and 16384 pagelets on Alpha and I64 systems. The value cannot be greater than WSMAX. This quota value replaces smaller values of PQL\_MWSEXTENT.

### **/WSQUOTA=value**

Specifies the working set quota. This is the maximum amount of physical memory a user process can lock into its working set. It also represents the maximum amount of swap space that the system reserves for this process and the maximum amount of physical memory that the system allows the process to consume if the systemwide memory demand is significant.

The value cannot be greater than the value of WSMAX and cannot exceed 8,192 pagelets on Alpha and I64 systems. This quota value replaces smaller values of PQL\_MWSQUOTA.

## Description

Modify the DEFAULT record when qualifiers normally assigned to a new user differ from the HP-supplied values. The following qualifiers correspond to fields in the default record that are commonly modified:

Qualifier	Reason for Modification
/CLI	Specifies the default Command Line Interpreter to be used for this user. (Most OpenVMS users use the DCL command interpreter.)
/DEVICE	If most users have the same default login device, allows you to specify a default login device for newly-created users. The use of a logical name is recommended.



Qualifier	Reason for Modification
/LGICMD	<p>Specifies the filename of a command procedure to be invoked during the login of the user.</p> <ol style="list-style-type: none"> <li>OpenVMS first looks for a systemwide login command procedure, using the systemwide logical name SYS\$SYLOGIN. If this logical name successfully translates to a valid file specification, the command interpreter invokes the resulting command procedure during login. If the file specification does not include a file extension, the command interpreter applies a default value that is specific to that command interpreter. In the case of the DCL interpreter, the default file extension is .COM.</li> <li>OpenVMS then looks for a LGICMD specification. If it finds this specification, OpenVMS invokes the command procedure. If the LGICMD specification does not include a file extension, the current command interpreter applies a default value. In the case of the DCL interpreter, the default file extension is .COM.</li> </ol> <p>You can disable or override the command procedure invocation during login by specifying qualifiers such as /NOCOMMAND or /LGICMD at the login username prompt. Also see the CAPTIVE and RESTRICTED flags.</p>
/PRIVILEGES	When users are given different privileges than those supplied by HP.
Quota qualifiers	When the default quotas are insufficient or inappropriate for mainstream work.

### Example

```
UAF> DEFAULT /DEVICE=SYS$USER/LGICMD=SYS$MANAGER:SECURELGN -
_UAF> /PRIVILEGES=(TMPMBX,GRPNAM,GROUP)
%UAF-I-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.

## **AUTHORIZE EXIT**

---

### **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

### **Parameters**

None.

### **Qualifiers**

None.

---

## **GRANT/IDENTIFIER**

Assigns the specified identifier to the user and documents the user as a holder of the identifier in the rights database.

### **Format**

GRANT/IDENTIFIER id-name user-spec

### **Parameters**

#### **id-name**

Specifies the identifier name. The identifier name is a string of 1 to 31 alphanumeric characters that can contain underscores and dollar signs. The name must contain at least one nonnumeric character.

#### **user-spec**

Specifies the UIC identifier that uniquely identifies the user on the system. This type of identifier appears in alphanumeric format. For example: [GROUP1,JONES].

### **Qualifier**

#### **/ATTRIBUTES=(keyword[,...])**

Specifies attributes to be associated with the identifier. The following are valid keywords:

DYNAMIC	Allows unprivileged holders of the identifier to remove and to restore the identifier from the process rights list by using the DCL command SET RIGHTS_LIST.
HOLDER_HIDDEN	Prevents people from getting a list of users who hold an identifier, unless they own the identifier themselves.
NAME_HIDDEN	Allows holders of an identifier to have it translated, either from binary to ASCII or from ASCII to binary, but prevents unauthorized users from translating the identifier.
NOACCESS	Makes any access rights of the identifier null and void. If a user is granted an identifier with the No Access attribute, that identifier has no effect on the user's access rights to objects. This attribute is a modifier for an identifier with the Resource or Subsystem attribute.
RESOURCE	Allows holders of an identifier to charge disk space to the identifier. Used only for file objects.
SUBSYSTEM	Allows holders of the identifier to create and maintain protected subsystems by assigning the Subsystem ACE to the application images in the subsystem. Used only for file objects.

To remove an attribute from the identifier, add a NO prefix to the attribute keyword. For example, to remove the Resource attribute, specify /ATTRIBUTES=NORESOURCE.

## AUTHORIZE GRANT/IDENTIFIER

### 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 the user named Cramer who has UIC [300,015]. 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**

Displays information concerning the use of AUTHORIZE, including formats and explanations of commands, parameters, and qualifiers.

### **Format**

HELP [keyword[,...]]

### **Parameter**

#### **keyword[,...]**

Specifies one or more keywords that refer to the topic, command, qualifier, or parameter on which you want information from the AUTHORIZE HELP command.

### **Qualifiers**

None.

### **Description**

If you do not specify a keyword, HELP displays information about the topics and commands for which help is available. It then prompts you with "Topic?". You can supply a topic or a command name, or press Return. When you specify a command name and qualifiers, you get detailed information about that command. If you respond by pressing Return, you exit from help. You can also exit from help by pressing Ctrl/Z.

If the command you request accepts qualifiers, the display of the help information about the command is followed by the prompt "Subtopic?". Respond to this prompt with a qualifier name, or press Return. If you respond by pressing Return, HELP prompts with "Topic?". If you want to exit from help directly from this level, press Ctrl/Z.

### **Examples**

1. UAF> HELP ADD

The HELP command in this example displays information about the ADD command:

ADD

    Adds a user record to the SYSUAF and corresponding identifiers to the rights database.

    Format

    ADD newusername

    Additional information available:

# AUTHORIZE HELP

Parameter	Qualifiers
/ACCESS	/ACCOUNT /ADD_IDENTIFIER /ALGORITHM /ASTLM /BATCH
/BIOLM	/BYTLM /CLI /CLITABLES /CPUTIME /DEFPRIVILEGES
/DEVICE	/DIALUP /DIOLM /DIRECTORY /ENQLM /EXPIRATION
/FILLM	/FLAGS /GENERATE_PASSWORD /INTERACTIVE /JTQUOTA
/LGICMD	/LOCAL /MAXACCTJOBS /MAXDETACH /MAXJOBS /NETWORK
/OWNER	/PASSWORD /PBYTLM /PGFLQUOTA /PRCLM /PRIMEDAYS /PRIORITY
/PRIVILEGES	/PWDEXPIRED /PDDLIFETIME
/PDMINIMUM	/REMOTE /SHRFILLM /TQELM /UIC
/WSDEFAULT	/WSEXTENT /WSQUOTA
Examples	/IDENTIFIER /PROXY

ADD Subtopic?

2. UAF> HELP ADD/ACCOUNT

The command in this example displays information about the /ACCOUNT qualifier:

ADD

/ACCOUNT=account-name

Specifies the default name for the account (for example, a billing name or number). The name can be a string of 1 to 8 alphanumeric characters. By default, AUTHORIZE does not assign an account name.

## LIST

Writes reports for selected UAF records to a listing file, SYSUAF.LIS, which is placed in the current default directory.

---

**Note**

---

LIST/IDENTIFIER, LIST/PROXY, and LIST/RIGHTS are documented as separate commands.

---

### Format

LIST [user-spec]

### Parameter

#### **user-spec**

Specifies the user name or UIC of the requested UAF record. Without the **user-spec** parameter, AUTHORIZE lists the user records of all users. The asterisk (\*) and percent sign (%) wildcards are permitted in the user name.

### Qualifiers

#### **/BRIEF**

Specifies that a brief report be written to SYSUAF.LIS. The /BRIEF qualifier is the default qualifier. SYSUAF.LIS is placed in the default directory.

#### **/FULL**

Specifies that a full report be written to SYSUAF.LIS, including identifiers held by the user. SYSUAF.LIS is placed in the SYS\$SYSTEM directory.

### Description

The LIST command creates a listing file of reports for selected UAF records. Print the listing file, SYSUAF.LIS, with the DCL command PRINT.

Specification of a user name results in a single-user report. Specification of the asterisk wildcard character following the LIST command results in reports for all users in ascending sequence by user name. Specification of a UIC results in reports for all users with that UIC. (HP recommends that you assign each user a unique UIC, but if users share a UIC, the report will show all users with that UIC.) You can use the asterisk wildcard character to specify the UIC.

The following table shows how to specify a UIC with the LIST command and use the asterisk wildcard character with the UIC specification to produce various types of reports:

Command	Description
LIST [14,6]	Lists a full report for the user (or users) with member number 6 in group 14.
LIST [14,*] /BRIEF	Lists a brief report for all users in group 14, in ascending sequence by member number.

## AUTHORIZE LIST

Command	Description
LIST [*,6] /BRIEF	Lists a brief report for all users with a member number of 6.
LIST [*,*] /BRIEF	Lists a brief report for all users, in ascending sequence by UIC.

Although you must provide separate UICs for each user, the LIST command reports users with the same UIC in the order in which they were added to the SYSUAF. Full reports list the details of the limits, privileges, login flags, and command interpreter. Brief reports do not include the limits, login flags, or command interpreter, nor do they summarize the privileges. AUTHORIZE never displays the password for an account.

See the SHOW command for examples of brief and full reports.

### Examples

1. 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.

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

This command results in brief reports for all users in ascending sequence by user name. Note, however, that this is the same result you would produce had you omitted the asterisk wildcard.

3. UAF> LIST [300,\*]  
%UAF-I-LSTMSG1, writing listing file  
%UAF-I-LSTMSG2, listing file SYSUAF.LIS complete

This command lists a brief report for all user records with a group UIC of 300.



---

## **LIST/IDENTIFIER**

Creates a listing file (RIGHTSLIST.LIS) in which identifier names, attributes, values, and holders are 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. The /FULL qualifier specifies the default listing format.

#### **/USER=user-spec**

Specifies one or more users whose identifiers are to be listed. The *user-spec* can be a user name or UIC. You can use the asterisk wildcard character (\*) to specify multiple user names or UICs. 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 formats are valid for the *value-specifier*:

IDENTIFIER:n	An integer value in the range 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). To differentiate general identifiers from UIC identifiers, %X80000000 is added to the value you specify.
GID:n	GID is the POSIX group identifier. It is an integer value in the range 0 to 16,777,215 (%XFFFFFF). The system will add %XA400.0000 to the value you specify and then enter this new value into the system RIGHTSLIST as an identifier.
UIC:uic	A UIC value in the standard UIC format.

# AUTHORIZE LIST/IDENTIFIER

## Description

The LIST/IDENTIFIER command creates a listing file in which identifier names, attributes, values, and holders are displayed in various formats depending on the qualifiers specified. Two of these formats are illustrated in the description of the SHOW/IDENTIFIER command.

Print the listing file named RIGHTSLIST.LIS with the DCL command PRINT.

## Examples

1. 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.

2. UAF> LIST/IDENTIFIER/USER=ANDERSON  
%UAF-I-LSTMSG1, writing listing file  
%UAF-I-RLSTMSG, listing file RIGHTSLIST.LIS complete

This command lists an identifier associated with the user ANDERSON, along with its value and attributes. Note, however, that this is the same result you would produce had you specified ANDERSON's UIC with the following forms of the command:

```
UAF> LIST/IDENTIFIER/USER=[300,015]
```

```
UAF> LIST/IDENTIFIER/VALUE=UIC:[300,015]
```

---

## **LIST/PROXY**

Creates a listing file of the network proxy database entries from the network database file NET\$PROXY.DAT.

### **Format**

LIST/PROXY

### **Parameters**

None.

### **Qualifiers**

#### **/OLD**

Directs AUTHORIZE to display information from the NETPROXY.DAT file rather than from the default file NET\$PROXY.DAT.

If someone modifies the proxy database on a cluster node that is not running the current OpenVMS VAX system, then you can use the /OLD qualifier to list the contents of the old database: NETPROXY.DAT.

### **Description**

Use the DCL command PRINT to print the listing file, NETPROXY.LIS. The output assumes the same format as that of the SHOW/PROXY command. For an example of the output format, see the description of the SHOW/PROXY command.

### **Example**

```
UAF> LIST/PROXY/OLD
%UAF-I-LSTMSG1, writing listing file
%UAF-I-NETLSTMSG, listing file NETPROXY.LIS complete
```

The command in this example creates a listing file of all the entries in the network proxy database NETPROXY.DAT.

# AUTHORIZE LIST/RIGHTS

---

## 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. 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. The *user-spec* can be a user name or UIC. You can use the asterisk wildcard character (\*) to specify multiple UICs or all user names. UICs must be in the form [\*,\*], [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.

### Description

Use the DCL command PRINT to print the listing file (RIGHTSLIST.LIS) produced by the LIST/RIGHTS command. For an example of the output format, see the description of the SHOW/RIGHTS command.

### Example

```
UAF> LIST/RIGHTS PAYROLL
%UAF-I-LSTMSG1, writing listing file
%UAF-I-RLSTMSG, listing file RIGHTSLIST.LIS complete
```

The command in this example lists identifiers held by PAYROLL, providing PAYROLL is the name of a UIC format identifier.

## MODIFY

Changes values in a SYSUAF user record. Qualifiers not specified in the command remain unchanged.

---

**Note**

---

MODIFY/IDENTIFIER, MODIFY/PROXY, and MODIFY/SYSTEM\_PASSWORD are documented as separate commands.

---

### Format

MODIFY username /qualifier[,...]

### Parameter

**username**

Specifies the name of a user in the SYSUAF. The asterisk (\*) and percent sign (%) wildcard 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

**/ACCESS[=(range[,...])]**

**/NOACCESS[=(range[,...])]**

Specifies hours of access for all modes of access. The syntax for specifying the range is:

`/[NO]ACCESS=([PRIMARY], [n-m], [n], [...],[SECONDARY], [n-m], [n], [...])`

Specify hours as integers from 0 to 23, inclusive. You can specify single hours (n) or 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 *inclusive*; that is, if you grant access during a given hour, access extends to the end of that hour.

By default, a user has full access every day. See the DCL command SET DAY in the *HP OpenVMS DCL Dictionary* for information about overriding the defaults for primary and secondary day types.

All the list elements are optional. Unless you specify hours for a day type, access is permitted for the entire day. By specifying an access time, you prevent access at all other times. Adding NO to the qualifier denies the user access to the system for the specified period of time. See the following examples.

<code>/ACCESS</code>	Allows unrestricted access
<code>/NOACCESS=SECONDARY</code>	Allows access on primary days only
<code>/ACCESS=(9-17)</code>	Allows access from 9 A.M. to 5:59 P.M. on all days

## AUTHORIZE MODIFY

**/NOACCESS=(PRIMARY, 9-17,  
SECONDARY, 18-8)**

Disallows access between 9 A.M. to 5:59 P.M. on primary days but allows access during these hours on secondary days

To specify access hours for specific types of access, see the **/BATCH**, **/DIALUP**, **/INTERACTIVE**, **/LOCAL**, **/NETWORK**, and **/REMOTE** qualifiers.

Refer to *HP OpenVMS Guide to System Security* for information about the effects of login class restrictions.

### **/ACCOUNT=account-name**

Specifies the default name for the account (for example, a billing name or number). The name can be a string of 1 to 8 alphanumeric characters. By default, AUTHORIZE does not assign an account name.

### **/ALGORITHM=keyword=type [=value]**

Sets the password encryption algorithm for a user. The keyword VMS refers to the algorithm used in the operating system version that is running on your system, whereas a customer algorithm is one that is added through the \$HASH\_PASSWORD system service by a customer site, by a layered product, or by a third party. The customer algorithm is identified in \$HASH\_PASSWORD by an integer in the range of 128 to 255. It must correspond with the number used in the AUTHORIZE command MODIFY/ALGORITHM. By default, passwords are encrypted with the VMS algorithm for the current version of the operating system.

Keyword	Function
BOTH	Set the algorithm for primary and secondary passwords.
CURRENT	Set the algorithm for the primary, secondary, both, or no passwords, depending on account status. CURRENT is the default value.
PRIMARY	Set the algorithm for the primary password only.
SECONDARY	Set the algorithm for the secondary password only.

The following table lists password encryption algorithms:

Type	Definition
VMS	The algorithm used in the version of the operating system that is running on your system.
CUSTOMER	A numeric value in the range of 128 to 255 that identifies a customer algorithm.

The following example selects the VMS algorithm for Sontag's primary password:

```
UAF> MODIFY SONTAG/ALGORITHM=PRIMARY=VMS
```

If you select a site-specific algorithm, you must give a value to identify the algorithm, as follows:

```
UAF> MODIFY SONTAG/ALGORITHM=CURRENT=CUSTOMER=128
```

**/ASTLM=value**

Specifies the AST queue limit, which is the total number of asynchronous system trap (AST) operations and scheduled wake-up requests that the user can have queued at one time. The default is 40 on VAX systems and 300 on Alpha and I64 systems.

**/BATCH[=(range[,...])]**

Specifies the hours of access permitted for batch jobs. For a description of the range specification, see the /ACCESS qualifier. By default, a user can submit batch jobs any time.

**/BIOLM=value**

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. The default is 40 on VAX systems and 150 on Alpha and I64 systems.

**/BYTLM=value**

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 can consume at one time. Nonpaged dynamic memory is used for operations such as I/O buffering, mailboxes, and file-access windows. The default is 32768 on VAX systems and 128,000 on Alpha and I64 systems.

**/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 a string of 1 to 31 alphanumeric characters and should be DCL, which is the default. This setting is ignored for network jobs.

**/CLITABLES=filespec**

Specifies user-defined CLI tables for the account. The *filespec* can contain 1 to 31 characters. The default is SYS\$LIBRARY:DCLTABLES. Note that this setting is ignored for network jobs to guarantee that the system-supplied command procedures used to implement network objects function properly.

**/CPUTIME=time**

Specifies the maximum process CPU time for the CPU field of the UAF record. The maximum process CPU time is the maximum amount of CPU time a user's process can take per session. You must specify a delta time value. For a discussion of delta time values, refer to the *OpenVMS User's Manual*. The default is 0, which means an infinite amount of time.

**/DEFPRIVILEGES=([NO]privname[,...])**

Specifies default privileges for the user; that is, those enabled at login time. A NO prefix removes a privilege from the user. By specifying the keyword [NO]ALL with the /DEFPRIVILEGES qualifier, you can disable or enable all user privileges. The default privileges are TMPMBX and NETMBX. *Privname* is the name of the privilege.

**/DEVICE=device-name**

Specifies the name of the user's default device at login. The *device-name* is a string of 1 to 31 alphanumeric characters. If you omit the colon from the *device-name* value, AUTHORIZE appends a colon. The default device is SYS\$SYSDISK.

## AUTHORIZE MODIFY

If you specify a logical name as the *device-name* (for example, DISK1: for DUA1:), you must make an entry for the logical name in the LNM\$SYSTEM\_TABLE in executive mode by using the DCL command DEFINE/SYSTEM/EXEC.

### **/DIALUP[=(range[,...])]**

Specifies hours of access permitted for dialup logins. For a description of the range specification, see the /ACCESS qualifier. The default is full access.

### **/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 default is 40 on VAX systems and 150 on Alpha and I64 systems.

### **/DIRECTORY=directory-name**

Specifies the default directory name for the DIRECTORY field of the UAF record. The *directory-name* can be 1 to 39 alphanumeric characters. If you do not enclose the directory name in brackets, AUTHORIZE adds the brackets for you. The default directory name is [USER].

### **/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 by the user at one time. The default is 200 on VAX systems and 4000 on Alpha and I64 systems.

### **/EXPIRATION=time (default)**

#### **/NOEXPIRATION**

Specifies the expiration date and time of the account. The /NOEXPIRATION qualifier removes the expiration date on the account. If you do not specify an expiration time when you add a new account, AUTHORIZE copies the expiration time from the DEFAULT account. (The expiration time on the DEFAULT account is "none" by default.)

### **/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. The default is 300 on VAX systems and 128 on Alpha and I64 systems.

### **/FLAGS=([NO]option[,...])**

Specifies login flags for the user. The prefix NO clears the flag. The options are as follows:

AUDIT	Enables or disables mandatory security auditing for a specific user. By default, the system does not audit the activities of specific users (NOAUDIT).
AUTOLOGIN	Restricts the user to the automatic login mechanism when logging in to an account. When set, the flag disables login by any terminal that requires entry of a user name and password. The default is to require a user name and password (NOAUTOLOGIN).



CAPTIVE	Prevents the user from changing any defaults at login, for example, /CLI or /LGICMD. It prevents the user from escaping the captive login command procedure specified by the /LGICMD qualifier and gaining access to the DCL command level. Refer to “Guidelines for Captive Command Procedures” in the <i>HP OpenVMS Guide to System Security</i> . The CAPTIVE flag also establishes an environment where Ctrl/Y interrupts are initially turned off; however, command procedures can still turn on Ctrl/Y interrupts with the DCL command SET CONTROL=Y. By default, an account is not captive (NOCAPTIVE).
DEFCLI	Restricts the user to the default command interpreter by prohibiting the use of the /CLI qualifier at login. By default, a user can choose a CLI (NODEFCLI).
DISCTLY	Establishes an environment where Ctrl/Y interrupts are initially turned off and are invalid until a SET CONTROL=Y is encountered. This could happen in SYLOGIN.COM or in a procedure called by SYLOGIN.COM. Once a SET CONTROL=Y is executed (which requires no privilege), a user can enter a Ctrl/Y and reach the DCL prompt (\$). If the intent of DISCTLY is to force execution of the login command files, then SYLOGIN.COM should issue the DCL command SET CONTROL=Y to turn on Ctrl/Y interrupts before exiting. By default, Ctrl/Y is enabled (NODISCTLY).
DISFORCE_PWD_CHANGE	Removes the requirement that a user must change an expired password at login. By default, a person can use an expired password only once (NODISFORCE_PWD_CHANGE) and then is forced to change the password after logging in. If the user does not select a new password, the user is locked out of the system.  To use this feature, set a password expiration date with the /PWDLIFETIME qualifier.
DISIMAGE	Prevents the user from executing RUN and foreign commands. By default, a user can execute RUN and foreign commands (NODISIMAGE).
DISMAIL	Disables mail delivery to the user. By default, mail delivery is enabled (NODISMAIL).
DISNEWMAIL	Suppresses announcements of new mail at login. By default, the system announces new mail (NODISNEWMAIL).
DISPWDDIC	Disables automatic screening of new passwords against a system dictionary. By default, passwords are automatically screened (NODISPWDDIC).
DISPWDHIS	Disables automatic checking of new passwords against a list of the user’s old passwords. By default, the system screens new passwords (NODISPWDHIS).

## AUTHORIZE MODIFY

DISPWDSYNCH	Suppresses synchronization of the external password for this account. See bit 9 in the SECURITY_POLICY system parameter for systemwide password synchronization control.
DISRECONNECT	Disables automatic reconnection to an existing process when a terminal connection has been interrupted. By default, automatic reconnection is enabled (NODISRECONNECT).
DISREPORT	Suppresses reports of the last login time, login failures, and other security reports. By default, login information is displayed (NODISREPORT).
DISUSER	Disables the account so the user cannot log in. For example, the DEFAULT account is disabled. By default, an account is enabled (NODISUSER).
DISWELCOME	Suppresses the welcome message (an informational message displayed during a local login). This message usually indicates the version number of the operating system that is running and the name of the node on which the user is logged in. By default, a system login message appears (NODISWELCOME).
EXTAUTH	Considers user to be authenticated by an external user name and password, not by the SYSUAF user name and password. (The system still uses the SYSUAF record to check a user's login restrictions and quotas and to create the user's process profile.)
GENPWD	Restricts the user to generated passwords. By default, users choose their own passwords (NOGENPWD).
LOCKPWD	Prevents the user from changing the password for the account. By default, users can change their passwords (NOLOCKPWD).
PWD_EXPIRED	Marks a password as expired. The user cannot log in if this flag is set. The LOGINOUT.EXE image sets the flag when both of the following conditions exist: a user logs in with the DISFORCE_PWD_CHANGE flag set, and the user's password expires. A system manager can clear this flag. By default, passwords are not expired after login (NOPWD_EXPIRED).
PWD2_EXPIRED	Marks a secondary password as expired. Users cannot log in if this flag is set. The LOGINOUT.EXE image sets the flag when both of the following conditions exist: a user logs in with the DISFORCE_PWD_CHANGE flag set, and the user's password expires. A system manager can clear this flag. By default, passwords are not set to expire after login (NOPWD2_EXPIRED).

**PWDMIX** Enables case-sensitive and extended-character passwords. After PWDMIX is specified, you can then use mixed-case and extended characters in passwords. Be aware that before the PWDMIX flag is enabled, the system stores passwords in all upper-case. Therefore, until you change passwords, you must enter your pre-PWDMIX passwords in upper-case.

To change the password after PWDMIX is enabled:

- You (the user) can use the DCL command SET PASSWORD, specifying the new mixed-case password (omitting quotation marks).
- You (the system manager) can use the AUTHORIZE command MODIFY/PASSWORD, and enclose the user's new mixed-case password in quotation marks " ".

**RESTRICTED** Prevents the user from changing any defaults at login (for example, by specifying /LGICMD) and prohibits user specification of a CLI with the /CLI qualifier. The RESTRICTED flag establishes an environment where Ctrl/Y interrupts are initially turned off; however, command procedures can still turn on Ctrl/Y interrupts with the DCL command SET CONTROL=Y. Typically, this flag is used to prevent an applications user from having unrestricted access to the CLI. By default, a user can change defaults (NORESTRICTED).

**VMSAUTH** Allows account to use standard (SYSUAF) authentication when the EXTAUTH flag would otherwise require external authentication. This depends on the application. An application specifies the VMS domain of interpretation when calling SYS\$ACM to request standard VMS authentication for a user account that normally uses external authentication.

**/GENERATE\_PASSWORD[=keyword]**

**/NOGENERATE\_PASSWORD (default)**

Invokes the password generator to create user passwords. Generated passwords can consist of 1 to 10 characters. Specify one of the following keywords:

<b>BOTH</b>	Generate primary and secondary passwords.
<b>CURRENT</b>	Do whatever the DEFAULT account does (for example, generate primary, secondary, both, or no passwords). This is the default keyword.
<b>PRIMARY</b>	Generate primary password only.
<b>SECONDARY</b>	Generate secondary password only.

When you modify a password, the new password expires automatically; it is valid only once (unless you specify /NOPWDEXPIRED). On login, users are forced to change their passwords (unless you specify /FLAGS=DISFORCE\_PWD\_CHANGE).

Note that the /GENERATE\_PASSWORD and /PASSWORD qualifiers are mutually exclusive.

## AUTHORIZE MODIFY

**/INTERACTIVE[ =(range[,...])]**

**/NOINTERACTIVE**

Specifies the hours of access for interactive logins. For a description of the range specification, see the /ACCESS qualifier. By default, there are no access restrictions on interactive logins.

**/JTQUOTA=value**

Specifies the initial byte quota with which the jobwide logical name table is to be created. By default, the value is 4096 on VAX systems and 4096 on Alpha and I64 systems.

**/LGICMD=filespec**

Specifies the name of the default login command file. The file name defaults to the device specified for /DEVICE, the directory specified for /DIRECTORY, a file name of LOGIN, and a file type of .COM. If you select the defaults for all these values, the file name is SYS\$SYSTEM:[USER]LOGIN.COM.

**/LOCAL[=(range[,...])]**

Specifies hours of access for interactive logins from local terminals. For a description of the range specification, see the /ACCESS qualifier. By default, there are no access restrictions on local logins.

**/MAXACCTJOBS=value**

Specifies the maximum number of batch, interactive, and detached processes that can be active at one time for all users of the same account. By default, a user has a maximum of 0, which represents an unlimited number.

**/MAXDETACH=value**

Specifies the maximum number of detached processes with the cited user name that can be active at one time. To prevent the user from creating detached processes, specify the keyword NONE. By default, a user has a value of 0, which represents an unlimited number.

**/MAXJOBS=value**

Specifies the maximum number of processes (interactive, batch, detached, and network) with the cited user name that can be active simultaneously. The first four network jobs are not counted. By default, a user has a maximum value of 0, which represents an unlimited number.

**/MODIFY\_IDENTIFIER (default)**

**/NOMODIFY\_IDENTIFIER**

Specifies whether the identifier associated with the user is to be modified in the rights database. This qualifier applies only when you modify the UIC or user name in the UAF record. By default, the associated identifiers are modified.

**/NETWORK[=(range[,...])]**

Specifies hours of access for network batch jobs. For a description of how to specify the range, see the /ACCESS qualifier. By default, network logins have no access restrictions.

**/OWNER=owner-name**

Specifies the name of the owner of the account. You can use this name for billing purposes or similar applications. The owner name is 1 to 31 characters. No default owner name exists.

**/PASSWORD=(password1[,password2])****/NOPASSWORD**

Specifies up to two passwords for login. Passwords can be from 0 to 32 alphanumeric characters in length. The dollar sign (\$) and underscore (\_) are also permitted.

Uppercase and lowercase characters are equivalent. All lowercase characters are converted to uppercase before the password is encrypted. Avoid using the word *password* as the actual password.

Use the /PASSWORD qualifier as follows:

- To set only the first password and clear the second, specify /PASSWORD=password.
- 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.

When you modify a password, the new password expires automatically; it is valid only once (unless you specify /NOPWDEXPIRED). On login, the user is forced to change the password (unless you specify /FLAGS=DISFORCE\_PWD\_CHANGE).

Note that the /GENERATE\_PASSWORD and /PASSWORD qualifiers are mutually exclusive.

**/PBYTLM**

This flag is reserved for HP.

**/PGFLQUOTA=value**

Specifies the paging file limit. This is the maximum number of pages that the person's process can use in the system paging file. By default, the value is 32768 pages on VAX systems and 256,000 pagelets on Alpha and I64 systems.

If decompressing libraries, make sure to set PGFLQUOTA to twice the size of the library.

**/PRCLM=value**

Specifies the subprocess creation limit. This is the maximum number of subprocesses that can exist at one time for the specified user's process. By default, the value is 2 on VAX systems and 8 on Alpha and I64 systems.

**/PRIMEDAYS=(**[NO]**day[,...])**

Defines the primary and secondary days of the week for logging in. Specify the days as a list separated by commas, and enclose the list in parentheses. To specify a secondary day, prefix the day with NO (for example, NOFRIDAY). To specify a primary day, omit the NO prefix.

By default, primary days are Monday through Friday and secondary days are Saturday and Sunday. If you omit a day from the list, AUTHORIZE uses the default value. (For example, if you omit Monday from the list, AUTHORIZE defines Monday as a primary day.)

## AUTHORIZE MODIFY

Use the primary and secondary day definitions in conjunction with such qualifiers as /ACCESS, /INTERACTIVE, and /BATCH.

### **/PRIORITY=value**

Specifies the default base priority. The value is an integer in the range of 0 to 31 on VAX systems and 0 to 63 on Alpha and I64 systems. By default, the value is set to 4 for timesharing users.

### **/PRIVILEGES=(*[NO]*privname[,...])**

Specifies which privileges the user is authorized to hold, although these privileges are not necessarily enabled at login. (The /DEFPRIVILEGES qualifier determines which ones are enabled.) A NO prefix removes the privilege from the user. The keyword NOALL disables all user privileges. Many privileges have varying degrees of power and potential system impact (see the *HP OpenVMS Guide to System Security* for a detailed discussion). By default, a user holds TMPMBX and NETMBX privileges. *Privname* is the name of the privilege.

### **/PWDEXPIRED (default)**

#### **/NOPWDEXPIRED**

Specifies the password is valid for only one login. A user must change a password immediately after login or be locked out of the system. The system warns users of password expiration. A user can either specify a new password, with the DCL command SET PASSWORD, or wait until expiration and be forced to change. By default, a user must change a password when first logging in to an account. The default is applied to the account only when the password is being modified.

### **/PWDLIFETIME=time (default)**

#### **/NOPWDLIFETIME**

Specifies the length of time a password is valid. Specify a delta time value in the form [dddd-] [hh:mm:ss.cc]. For example, for a lifetime of 120 days, 0 hours, and 0 seconds, specify /PWDLIFETIME="120-". For a lifetime of 120 days 12 hours, 30 minutes and 30 seconds, specify /PWDLIFETIME="120-12:30:30". If a period longer than the specified time elapses before the user logs in, the system displays a warning message. The password is marked as expired.

To prevent a password from expiring, specify the time as NONE. By default, a password expires in 90 days.

### **/PWDMINIMUM=value**

Specifies the minimum password length in characters. Note that this value is enforced only by the DCL command SET PASSWORD. It does not prevent you from entering a password shorter than the minimum length when you use AUTHORIZE to create or modify an account. By default, a password must have at least 6 characters. The value specified by the /PWDMINIMUM qualifier conflicts with the value used by the /GENERATE\_PASSWORD qualifier or the DCL command SET PASSWORD/GENERATE, the operating system chooses the lesser value. The maximum value for generated passwords is 10.

### **/QUEPRIO=value**

Reserved for future use.

### **/REMOTE[=(range[,...])]**

Specifies hours during which access is permitted for interactive logins from network remote terminals (with the DCL command SET HOST). For a description of the range specification, see the /ACCESS qualifier. By default, remote logins have no access restrictions.

**/SHRFILLM=value**

Specifies the maximum number of shared files that the user can have open at one time. By default, the system assigns a value of 0, which represents an infinite number.

**/TQELM**

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. By default, a user can have 100.

**/UIC=value**

Specifies the user identification code (UIC). The UIC value is a group number in the range from 1 to 37776 (octal) and a member number in the range from 0 to 177776 (octal), which are separated by a comma and enclosed in brackets. HP reserves group 1 and groups 300–377 for its own use.

Each user must have a unique UIC. By default, the UIC value is [200,200].

**/WSDEFAULT=value**

Specifies the default working set limit. This represents the initial limit to the number of physical pages the process can use. (The user can alter the default quantity up to WSQUOTA with the DCL command SET WORKING\_SET.) By default, a user has 256 pages on VAX systems and 4096 pagelets on Alpha and I64 systems.

The value cannot be greater than WSMAX. This quota value replaces smaller values of PQL\_MWSDEFAULT.

**/WSEXTENT=value**

Specifies the working set maximum. This represents the maximum amount of physical memory allowed to the process. The system provides memory to a process beyond its working set quota only when it has excess free pages. The additional memory is recalled by the system if needed.

The value is an integer equal to or greater than WSQUOTA. By default, the value is 1024 pages on VAX systems and 16384 pagelets on Alpha and I64 systems. The value cannot be greater than WSMAX. This quota value replaces smaller values of PQL\_MWSEXTENT.

**/WSQUOTA=value**

Specifies the working set quota. This is the maximum amount of physical memory a user process can lock into its working set. It also represents the maximum amount of swap space that the system reserves for this process and the maximum amount of physical memory that the system allows the process to consume if the systemwide memory demand is significant.

The value cannot be greater than the value of WSMAX and cannot exceed 8,192 pagelets on Alpha and I64 systems. This quota value replaces smaller values of PQL\_MWSQUOTA.

## **Description**

The MODIFY command changes values in a SYSUAF user record. Most values not in the command remain unchanged. If the UIC is changed, the value of the corresponding identifier is also changed.

Modifications to the user record are not retroactive; thus, any changes to quota values apply to the next process that is created but not to the current one.

## AUTHORIZE MODIFY

### Examples

1. 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.

2. UAF> MODIFY ROBIN/FLAGS=RESTRICTED  
%UAF-I-MDFYMSG, user record(s) updated

The command in this example modifies the UAF record for user ROBIN by adding the login flag RESTRICTED.



---

## **MODIFY/IDENTIFIER**

Modifies an identifier name, its associated value, or its attributes 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 keywords are valid:

DYNAMIC	Allows unprivileged holders of the identifier to remove and to restore the identifier from the process rights list by using the DCL command SET RIGHTS_LIST.
HOLDER_HIDDEN	Prevents people from getting a list of users who hold an identifier, unless they own the identifier themselves.
NAME_HIDDEN	Allows holders of an identifier to have it translated, either from binary to ASCII or from ASCII to binary, but prevents unauthorized users from translating the identifier.
NOACCESS	Makes any access rights of the identifier null and void. If a user is granted an identifier with the No Access attribute, that identifier has no effect on the user's access rights to objects. This attribute is a modifier for an identifier with the Resource or Subsystem attribute.
RESOURCE	Allows holders of an identifier to charge disk space to the identifier. Used only for file objects.
SUBSYSTEM	Allows holders of the identifier to create and maintain protected subsystems by assigning the Subsystem ACE to the application images in the subsystem. Used only for file objects.

To remove an attribute from the identifier, add a NO prefix to the attribute keyword. For example, to remove the Resource attribute, specify /ATTRIBUTES=NORESOURCE.

---

#### **Note**

---

If you specify the NORESOURCE keyword without naming any holder with the /HOLDER qualifier, all holders lose the right to charge resources.

---

#### **/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.

## AUTHORIZE MODIFY/IDENTIFIER

If you specify /HOLDER, the /NAME and /VALUE qualifiers are ignored.

### **/NAME=new-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 formats are valid for the *value-specifier*:

IDENTIFIER:n	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). To differentiate general identifiers from UIC identifiers, %X80000000 is added to the value you specify.
GID:n	GID is the POSIX group identifier. It is an integer value in the range 0 to 16,777,215 (%XFFFFFF). The system will add %XA400.0000 to the value you specify and then enter this new value into the system RIGHTSLIST as an identifier.
UIC:uic	A UIC value in the standard UIC format.

## Description

The MODIFY/IDENTIFIER command changes identifier names, associated values, and attributes in the rights database. Values not specified in the command remain unchanged.

## Examples

1. 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.

2. UAF> MODIFY/IDENTIFIER/VALUE=UIC:[300,21] ACCOUNTING  
%UAF-I-RDBMDFYMSG, identifier ACCOUNTING modified

The command in this example changes the old UIC value of the identifier ACCOUNTING to a new value.

3. UAF> MODIFY/IDENTIFIER/ATTRIBUTES=NORESOURCE-  
UAF> /HOLDER=CRAMER ACCOUNTING  
%UAF-I-RDBMDFYMSG, identifier ACCOUNTING modified

The command in this example associates the attribute NORESOURCE with the identifier ACCOUNTING in CRAMER's holder record. The identifier ACCOUNTING is not changed.

---

## **MODIFY/PROXY**

Modifies an entry in the network proxy authorization file to specify a different local account as the default proxy account for the remote user or to specify no default proxy account for the remote user.

The command modifies an entry in the network proxy authorization file NET\$PROXY.DAT and, to maintain compatibility with other systems, modifies an entry in NETPROXY.DAT.

---

### **Note**

---

You must modify the proxy database from a system running the current OpenVMS system.

---

## **Format**

MODIFY/PROXY node::remote-user

## **Parameters**

### **node**

Specifies a node name. If you specify an asterisk wildcard character (\*), 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 wildcard character, all users at the specified node are served by the local user.

For systems that are not OpenVMS systems that implement DECnet, specifies the UIC of a user at a remote node. You can specify an asterisk wildcard 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.

## **Description**

Use the MODIFY/PROXY command to specify a different local account as the default proxy account for the remote user or to specify that there is no default proxy account for the remote user. Whenever you modify user entries, AUTHORIZE signals DECnet to update its volatile database. Proxy modifications take effect immediately on all nodes in a cluster that share the proxy database.

The first command in the following example grants remote user STIR::YETTA proxy access to the PROXY1 and PROXY2 local accounts. The default proxy account is PROXY1. The second command changes the default proxy account to PROXY2.

## AUTHORIZE MODIFY/PROXY

```
UAF> ADD/PROXY STIR::YETTA PROXY1/DEFAULT, PROXY2
.
.
.
UAF> MODIFY/PROXY STIR::YETTA /DEFAULT=PROXY2
```

The next example shows the command used to remove the default proxy designation.

```
UAF> MODIFY/PROXY STIR::YETTA /NODEFAULT
```

If you remove the default proxy designation as shown in the last command, remote user STIR::YETTA must include the name of the proxy account (PROXY1 or PROXY2) in the access control string of each network operation to gain proxy access to the local system.

If no default proxy account is specified either in the network proxy database or in the access control string of the DCL command, the system attempts to perform the network operation using the default DECnet account.

### Example

```
UAF> MODIFY/PROXY MISHA::MARCO /DEFAULT=JOHNSON
%UAF-I-NAFADMSG, 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 systemwide password.

---

### Note

---

The systemwide password is different from the password for the SYSTEM user name. See the note in the Description.

---

This command operates similarly to the DCL command SET PASSWORD/SYSTEM.

### Format

MODIFY/SYSTEM\_PASSWORD=system-password

### Parameter

**system-password**  
Specifies the new systemwide password.

### Qualifiers

None.

### Description

For a detailed description of the effects of this command, refer to the discussion of the SET PASSWORD/SYSTEM command in the *HP OpenVMS Guide to System Security*.

### Example

```
UAF> MODIFY/SYSTEM_PASSWORD=ABRACADABRA  
UAF>
```

This command changes the systemwide password to ABRACADABRA.

# AUTHORIZE REMOVE

---

## REMOVE

Deletes a SYSUAF user record and corresponding identifiers in the rights database. The DEFAULT and SYSTEM records cannot be deleted.

---

### Note

---

REMOVE/IDENTIFIER and REMOVE/PROXY are documented as separate commands.

---

## Format

REMOVE username

## Parameter

### username

Specifies the name of a user in the SYSUAF.

## Qualifier

### /REMOVE\_IDENTIFIER (default)

### /NOREMOVE\_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 two UAF records have 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.

## Description

If you remove a SYSUAF record for a user who also appears as a local user in the network user authorization file, every network authorization record for that user is also removed.

## Example

```
UAF> REMOVE ROBIN
%UAF-I-REMSG, record removed from SYSUAF.DAT
%UAF-I-RDBREMSGU, identifier ROBIN value: [000014,000006] removed from
RIGHTSLIST.DAT
```

The command in this example deletes the record for user ROBIN from the SYSUAF 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.

### **Qualifiers**

None.

### **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.

## AUTHORIZE REMOVE/PROXY

---

### REMOVE/PROXY

Deletes network proxy access for the specified remote user.

#### Format

```
REMOVE/PROXY node::remote-user [local-user,...]
```

#### Parameters

**node**

Specifies the name of a network node in the network proxy authorization file.

**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.

#### Qualifiers

None.

#### Example

```
UAF> REMOVE/PROXY MISHA::MARCO
%UAF-I-NAFREMSG, proxy from MISHA::MARCO to * removed
```

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**

Changes the user name of the SYSUAF record (and, if specified, the corresponding identifier) while retaining the characteristics of the old record.

---

**Note**

---

RENAME/IDENTIFIER is documented as a separate command.

---

### **Format**

RENAME oldusername newusername

### **Parameters**

**oldusername**

Specifies the current user name in the SYSUAF.

**newusername**

Specifies the new name for the user. It can contain 1 to 12 alphanumeric characters and underscores. Although dollar signs are permitted, they are usually reserved for system names.

### **Qualifiers**

**/GENERATE\_PASSWORD[=keyword]**

**/NOGENERATE\_PASSWORD (default)**

Invokes the password generator to create user passwords. Generated passwords can consist of 1 to 10 characters. Specify one of the following keywords:

<b>BOTH</b>	Generate primary and secondary passwords.
<b>CURRENT</b>	Do whatever the DEFAULT account does (for example, generate primary, secondary, both, or no passwords). This is the default keyword.
<b>PRIMARY</b>	Generate primary password only.
<b>SECONDARY</b>	Generate secondary password only.

When you modify a password, the new password expires automatically; it is valid only once (unless you specify /NOPWDEXPIRED). On login, users are forced to change their passwords (unless you specify /FLAGS=DISFORCE\_PWD\_CHANGE).

Note that the /GENERATE\_PASSWORD and /PASSWORD qualifiers are mutually exclusive.

**/MODIFY\_IDENTIFIER (default)**

**/NOMODIFY\_IDENTIFIER**

Specifies whether the identifier associated with the user is to be modified in the rights database. This qualifier applies only when you modify the UIC or user name in the UAF record. By default, the associated identifiers are modified.

## AUTHORIZE RENAME

**/PASSWORD=(password1[,password2])  
/NOPASSWORD**

Specifies up to two passwords for login. Passwords can be from 0 to 32 alphanumeric characters in length. The dollar sign (\$) and underscore (\_) are also permitted.

Uppercase and lowercase characters are equivalent. All lowercase characters are converted to uppercase before the password is encrypted. Avoid using the word *password* as the actual password.

Use the /PASSWORD qualifier as follows:

- To set only the first password and clear the second, specify /PASSWORD=password.
- 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.

When you modify a password, the new password expires automatically; it is valid only once (unless you specify /NOPWDEXPIRED). On login, the user is forced to change the password (unless you specify /FLAGS=DISFORCE\_PWD\_CHANGE).

Note that the /GENERATE\_PASSWORD and /PASSWORD qualifiers are mutually exclusive.

When you create a new UAF record with the RENAME command, you must specify a password.

### Description

The RENAME command renames a SYSUAF record. It changes the user name of the SYSUAF record (and, if specified, the corresponding identifier) while retaining the characteristics of the old record. Retention of these characteristics can be particularly helpful when a user's name changes.

Note that because password verification includes the user name as well as the password, an attempted login will fail when the user whose name has been changed attempts to log in with an old password. (Only null passwords can be effectively transferred from one user record to another by the RENAME command.) Make it a practice to include a new password when you use the RENAME command, and notify the user of the change. If you omit the /PASSWORD qualifier, you receive a warning message reminding you that the old password must be changed.

The user's network authorization records are automatically changed to the new name.

## Examples

1. UAF> RENAME HAWKES KRAMERDOVE/PASSWORD=MARANNKRA  
%UAF-I-PRACREN, 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.

2. UAF> RENAME HAWKES KRAMERDOVE  
%UAF-I-PRACREN, proxies to HAWKES renamed  
%UAF-I-RENMSG, user record renamed  
%UAF-W-DEFPWD, Warning: copied or renamed records must receive  
new password  
%UAF-I-RDBMDFYMSG, identifier HAWKES modified

This example shows the warning message that the system displays if you fail to specify a new password with the RENAME command.

## AUTHORIZE RENAME/IDENTIFIER

---

### RENAME/IDENTIFIER

Renames an identifier in the rights database.

#### Format

RENAME/IDENTIFIER current-id-name new-id-name

#### Parameters

**current-id-name**

Specifies the name of an identifier to be renamed.

**new-id-name**

Specifies the new name for the identifier.

#### Qualifiers

None.

#### Description

The RENAME/IDENTIFIER command is functionally equivalent to the following AUTHORIZE command:

```
MODIFY/IDENTIFIER/NAME=new-id-name id-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**

Takes an identifier away from a user.

### **Format**

REVOKE/IDENTIFIER id-name user-spec

### **Parameters**

#### **id-name**

Specifies the identifier name. The identifier name is a string of 1 to 31 alphanumeric characters. The name can contain underscores and dollar signs. It must contain at least one nonnumeric character.

#### **user-spec**

Specifies the UIC identifier that uniquely identifies the user on the system. This type of identifier appears in alphanumeric format, not numeric format; for example, [GROUP1,JONES].

### **Description**

The REVOKE/IDENTIFIER command edits RIGHTSLLIST.DAT, removing the user's name from the list of those who hold a given identifier. The change does not affect the process rights list of any current processes.

### **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 because rights identifiers are stored in numeric format, it is not necessary to change records for users holding a renamed identifier.

# AUTHORIZE SHOW

---

## SHOW

Displays reports for selected UAF records on the current SYS\$OUTPUT device.

---

### Note

SHOW/IDENTIFIER, SHOW/PROXY, and SHOW/RIGHTS are documented as separate commands.

---

## Format

SHOW user-spec

## Parameter

### user-spec

Specifies the user name or UIC of the requested 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. In the report, the Directory field displays one of the following items:

- Disuser—The account has been disabled.
- Expired—The account has expired.
- A device and directory name—The login device and directory for the account (for example, DOCD\$:[SMITH]).

If you omit the /BRIEF qualifier, AUTHORIZE displays a full report.

### /FULL

Specifies that a full report be displayed, including identifiers held by the user. Full reports include the details of the limits, privileges, login flags, and the command interpreter as well as the identifiers held by the user. The password is not listed.

### /EXACT

Controls whether the SHOW command matches the search string exactly or treats uppercase and lowercase letters as equivalents. Enclose the specified string within quotation marks (" "). Use /EXACT with the /PAGE=SAVE and /SEARCH qualifiers.

### /HIGHLIGHT[=keyword]

### /NOHIGHLIGHT (default)

Identifies how to display the line that contains a string once it is found. The following keywords are valid:

BLINK  
BOLD (default)  
REVERSE  
UNDERLINE

Use the /HIGHLIGHT qualifier with the /PAGE=SAVE and /SEARCH qualifiers.

**/PAGE[=keyword]**

**/NOPAGE (default)**

Controls the information display on a screen. The following keywords are valid:

CLEAR_SCREEN	Clear the screen before displaying the next page.
SCROLL	Display a continuous stream of information.
SAVE[=n]	Store information and enable the navigational keys listed in Table 5–1. By default, the command saves 5 pages. The maximum page width is 255 columns.

**Table 5–1 Screen Control Keys**

Key or Key Sequence	Action Taken When Key or Key Sequence Is Pressed
DOWN ARROW KEY	Scroll the display down one line
LEFT ARROW KEY	Scroll the display one column to the left
RIGHT ARROW KEY	Scroll the display one column to the right
UP ARROW KEY	Scroll the display up one line
Find (E1)	Search for a new string in the information being displayed
Insert Here (E2)	Move the display to the right by half a screen
Remove (E3)	Move the display to the left by half a screen
Select (E4)	Switch from 80-column displays to 132-column displays
Prev Screen (E5)	Return to the previous page
Next Screen (E6)	Display the next page
CTRL/Z	Return to the UAF> prompt
Help	Display AUTHORIZE help text
F16 (Do)	Switch from the oldest to the newest page
Ctrl/W	Refresh the display

**/SEARCH=string**

Used with the /PAGE=SAVE qualifier to specify a string to find in the information being displayed. You can dynamically change the search string by pressing the Find key (E1) while the information is being displayed.

**/WRAP**

**/NOWRAP (default)**

Used with the /PAGE=SAVE qualifier to limit the number of columns to the width of the screen and wrap lines that extend beyond the width of the screen to the next line.

The /NOWRAP qualifier extends lines beyond the width of the screen. Use the /PAGE=SAVE qualifier and the screen control keys listed in Table 5–1 to view the entire screen.

# AUTHORIZE SHOW

## Description

The SHOW command produces reports on user authorization records. You can select the reports to be displayed, as follows:

- To display a single-user report, specify a user name.
- To display reports for all users in ascending sequence by user name, specify an asterisk wildcard character (\*).
- To display reports for all users with a common UIC, specify the UIC. Users with the same UIC are listed in the order in which they were added to the SYSUAF.

You can also use the asterisk wildcard character to specify all or part of the UIC, as shown in the following examples:

Command	Description
SHOW [14,*] /BRIEF	Displays a brief report for all users in group 14, in ascending sequence by member number.
SHOW [* ,6] /BRIEF	Displays a brief report for all users with a member number of 6.
SHOW [* ,*] /BRIEF	Displays a brief report for all users, in ascending sequence by UIC.

## Examples

1. UAF> SHOW ROBIN

The command in this VAX example displays a full report for the user ROBIN. The display corresponds to the first example in the description of the ADD command. 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-JAN-2000 14:08
Last Login: (none) (interactive), (none) (non-interactive)
Maxjobs: 0 Fillm: 300 Byt1m: 32768
Maxacctjobs: 0 Shrfillm: 0 Pbyt1m: 0
Maxdetach: 0 BI01m: 40 JTquota: 4096
Prclm: 2 DI01m: 40 WSdef: 256
Prio: 4 AST1m: 40 WSquo: 512
Queprio: 0 TQElm: 10 WSextent: 1024
CPU: (none) Enqlm: 200 Pgflquo: 32768
Authorized Privileges:
TMPMBX NETMBX
Default Privileges:
TMPMBX NETMBX
Identifier Value Attributes
CLASS_CA101 %X80010032 NORESOURCE NODYNAMIC
CLASS_PY102 %X80010049 NORESOURCE NODYNAMIC
```



---

**Note**

---

The quotas Pbytlm and Queprio are placeholders only.

---

2. UAF> SHOW [360,\*] /BRIEF

The command in this example displays a brief report for every user with a group UIC of 360.

Owner	Username	UIC	Account	Privs	Pri	Default	Directory
JOHN JAMES	JAMES	[360,201]	USER	Normal	4	DOCD\$:[JAMES]	
SUZY JONES	JONES	[360,203]	DOC	Devour	4	DOCD\$:[JONES]	
CLIFF BROWN	BROWN	[360,021]	DOC	All	4	disuser	
JOY CARTER	CARTER	[360,005]	DOCSEC	Group	4	expired	

3. UAF> SHOW WELCH

This command displays a full report for the restricted user WELCH. This display corresponds to the second example in the description of the ADD command.

```

Username: WELCH                               Owner: ROB WELCH
Account: INV                                   UIC: [14,51] ([14,51])
CLI: DCL                                       Tables: DCLTABLES
Default: SYS$USER:[WELCH]
LGICMD: SECUREIN
Login Flags: Restricted Diswelcome Disnewmail ExtAuth
Primary days: Mon Tue Wed Thu Fri
Secondary days:                               Sat Sun
Primary 000000000001111111112222           Secondary 00000000001111111112222
Day Hours 012345678901234567890123         Day Hours 012345678901234567890123
Network: ---- No access ----                ##### Full access #####
Batch: #####-----#####                 -----#####-----
Local: #####-----#####                  -----#####-----
Dialup: ##### Full access #####            ----- No access -----
Remote: #####-----#####                 -----#####-----
Expiration: (none) Pwdminimum: 6 Login Fails: 0
Pwdlifetime: (none) Pwdchange: (pre-expired)
Last Login: (none) (interactive), (none) (non-interactive)
Maxjobs: 0 Fillm: 300 Bytlm: 32768
Maxacctjobs: 0 Shrfillm: 0 Pbytlm: 0
Maxdetach: 0 BIolm: 40 JTquota: 4096
Prclm: 2 DIolm: 40 WSdef: 256
Prio: 4 ASTlm: 40 WSquo: 512
Queprio: 4 TQElm: 10 WSextent: 1024
CPU: (none) Enqlm: 200 Pgflquo: 32768
Authorized Privileges:
  TMPMBX NETMBX
Default Privileges:
  TMPMBX NETMBX

```

Note that WELCH is a captive user who does not receive announcements of new mail or the welcome message when logging in. His login command file, SECUREIN.COM, is presumably a captive command file that controls all of his operations. (Such a command file never exits, but performs operations for its user and logs him out when appropriate.) The CAPTIVE flag prevents WELCH from escaping control of the command file by using Ctrl/Y or other means. Furthermore, he is restricted to logging in between the hours of 5:00 P.M. and 8:59 A.M. on weekdays and 9:00 A.M. and 5:59 P.M. on weekends. Although he is allowed to use dial-up lines at all times during the week, he is not allowed to log in over the network. On weekends, he is further restricted

## **AUTHORIZE SHOW**

so that he cannot dial in at any time or use the DCL command SET HOST between the hours of 6:00 P.M. and 8:59 A.M.

---

## SHOW/IDENTIFIER

Displays information about an identifier, such as its name, value, attributes, and holders, on the current SYS\$OUTPUT device.

### Format

SHOW/IDENTIFIER [id-name]

### Parameter

#### **id-name**

Specifies an identifier name. The identifier name is a string of 1 to 31 alphanumeric characters. The name can contain underscores and dollar signs. It must contain at least one nonnumeric character. 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. The default format is /BRIEF.

#### **/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. The *user-spec* can be a user name or a UIC. You can use the asterisk wildcard character (\*) to specify multiple UICs or all 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 the value of the identifier to be listed. The following formats are valid for the *value-specifier*:

IDENTIFIER:n	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). To differentiate general identifiers from UIC identifiers, %X80000000 is added to the value you specify.
GID:n	GID is the POSIX group identifier. It is an integer value in the range 0 to 16,777,215 (%XFFFFFF). The system will add %XA400.0000 to the value you specify and then enter this new value into the system RIGHTSLIST as an identifier.
UIC:uic	A UIC value in the standard UIC format.

See also the screen control qualifiers listed under the SHOW command:

# AUTHORIZE SHOW/IDENTIFIER

**/EXACT**

**/HIGHLIGHT[=keyword]**  
**/NOHIGHLIGHT (default)**

**/PAGE[=keyword]**  
**/NOPAGE (default)**

**/SEARCH=string**

**/WRAP**  
**/NOWRAP (default)**

## Description

The SHOW/IDENTIFIER command displays identifier names, values, attributes, and holders in various formats depending on the qualifiers specified. Two of these formats are illustrated in the following examples.

## Examples

1. UAF> SHOW/IDENTIFIER/FULL INVENTORY

This command would produce output similar to the following example:

Name	Value	Attributes
INVENTORY	%X80010006	NORESOURCE NODYNAMIC
Holder	Attributes	
ANDERSON	NORESOURCE	NODYNAMIC
BROWN	NORESOURCE	NODYNAMIC
CRAMER	NORESOURCE	NODYNAMIC

2. UAF> SHOW/IDENTIFIER/USER=ANDERSON

This command displays the identifier associated with the user ANDERSON, as follows:

Name	Value	Attributes
ANDERSON	[000300,000015]	NORESOURCE NODYNAMIC

The identifier is shown, along with its value and attributes. Note, however, that this is the same result you would produce had you specified ANDERSON's UIC with the following forms of the command:

```
UAF> SHOW/IDENTIFIER/USER=[300,015]
```

```
UAF> SHOW/IDENTIFIER/VALUE=UIC:[300,015]
```

---

## **SHOW/PROXY**

Displays all authorized proxy access for the specified remote user.

### **Format**

SHOW/PROXY node::remote-user

### **Parameters**

#### **node**

Specifies the name of a network node in the network proxy authorization file. The asterisk wildcard character (\*) is permitted in the node specification.

#### **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.

### **Qualifiers**

#### **/OLD**

Directs AUTHORIZE to display information from NETPROXY.DAT rather than the default file NET\$PROXY.DAT.

If someone modifies the proxy database on a cluster node that is running an OpenVMS system prior to Version 6.1, you can use the /OLD qualifier to display the contents of the old database, NETPROXY.DAT.

See also the screen control qualifiers listed under the SHOW command:

#### **/EXACT**

**/HIGHLIGHT[=keyword]**  
**/NOHIGHLIGHT (default)**

**/PAGE[=keyword]**  
**/NOPAGE (default)**

**/SEARCH=string**

**/WRAP**  
**/NOWRAP (default)**

### **Description**

The SHOW/PROXY command displays the first 255 characters of a node name although the command can handle a maximum of 1024 characters.

# AUTHORIZE SHOW/PROXY

## Examples

1. UAF> SHOW/PROXY SAMPLE::<200,100]  
Default proxies are flagged with an \*  
SAMPLE::<200,100]  
MARCO \* PROXY2  
PROXY3

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.

2. UAF> SHOW/PROXY \*::\*  
Default proxies are flagged with (D)  
TAO:.TWA.RANCH::MARTINEZ  
MARTINEZ (D) SALES\_READER  
UAF> show/proxy/old \*::\*  
Default proxies are flagged with (D)  
RANCH::MARTINEZ  
MARTINEZ (D) SALES\_READER

The command in this example displays information about local authorized proxy access on a system running DECnet-Plus. The first command draws information from the file NET\$PROXY.DAT. By including the /OLD qualifier on the SHOW/PROXY command, AUTHORIZE displays information from the file NETPROXY.DAT.

---

## 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 [id-name]

### Parameter

#### id-name

Specifies the name of the identifier associated with the user. 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. The *user-spec* can be a user name or a UIC. You can use the asterisk wildcard character (\*) to specify multiple UICs or all user names. UICs must be in the form [\*,\*], [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.

See also the screen control qualifiers listed under the SHOW command:

#### /EXACT

/HIGHLIGHT[=keyword]  
/NOHIGHLIGHT (default)

/PAGE[=keyword]  
/NOPAGE (default)

#### /SEARCH=string

/WRAP  
/NOWRAP (default)

### Description

Output displayed from the SHOW/RIGHTS command is identical to that written to RIGHTS.LIST.LIS when you use the LIST/RIGHTS command.

### 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

## **AUTHORIZE SHOW/RIGHTS**

Note that the following formats of the command produce the same result:

SHOW/RIGHTS/USER=ANDERSON

SHOW/RIGHTS/USER=[300,015]



---

**AUTOGEN Command Procedure**



## 6.1 AUTOGEN Description

The AUTOGEN command procedure (SYS\$UPDATE:AUTOGEN.COM) sets appropriate values for system parameters and sizes for system page, swap, and dump files. AUTOGEN runs automatically when you install or upgrade the operating system.

In addition, you can use AUTOGEN to reset system parameter values, system file sizes, or both. The new values and file sizes take effect the next time the system is booted.

HP recommends that you run AUTOGEN on a weekly basis to adjust system parameters according to your system's work load. For a list and description of all system parameters, refer to

AUTOGEN executes in phases, with each phase performing a separate task. You control which tasks AUTOGEN performs by specifying a start phase and an end phase when you invoke AUTOGEN. For more information about the AUTOGEN phases, see Section 6.4.

You can add commands to the file SYS\$SYSTEM:MODPARAMS.DAT to control the system parameter values and file sizes that AUTOGEN sets. AUTOGEN uses the information in this file to determine final values for system parameters or page, swap or dump file sizes. For more information, refer to the chapter about managing system parameters in the *HP OpenVMS System Manager's Manual*.

AUTOGEN can improve system performance by using dynamic information, called **feedback**, which is gathered from the running system.

---

### Note

---

When making major configuration changes, do not use feedback. Specify *nofeedback* to assure the use of the initial AUTOGEN settings. See Table 6-2 for more information about *nofeedback*.

---

You control how AUTOGEN uses feedback by specifying an execution mode when you invoke AUTOGEN. To direct AUTOGEN to use feedback to make its calculations, run AUTOGEN in **feedback mode**. After a period of time, you can execute AUTOGEN in feedback mode to further refine system parameter settings. For more information about AUTOGEN feedback, refer to Section 6.3.

### 6.1.1 NEWPARAMS.DAT

The following sections explain how AUTOGEN uses NEWPARAMS.DAT and how layered product installation procedures can use it. If you are not involved in making layered product kits, you might not be interested in the sections preceding Section 6.1.1.3.

The basic reason for developing NEWPARAMS.DAT has been to provide a way for layered product installation procedures to easily supply AUTOGEN with the necessary parameter changes for the product. NEWPARAMS.DAT greatly reduces (if not eliminates) the need to modify MODPARAMS.DAT after installing a new layered product and avoids having installation procedures attempt to edit MODPARAMS.DAT.

## AUTOGEN Command Procedure

### 6.1 AUTOGEN Description

NEWPARAMS.DAT does not replace or make MODPARAMS.DAT obsolete, and it does not remove the requirement to run AUTOGEN after installing a layered product. (AUTOGEN can, however, process several versions of NEWPARAMS.DAT, which you might have if you install several layered products without running AUTOGEN between installations).

#### 6.1.1.1 How NEWPARAMS.DAT Works

Three files are involved in the operation of NEWPARAMS.DAT:

- **NEWPARAMS.DAT**  
This file contains the parameter requirements for a particular layered product.
- **NEWPARAMS.DONE**  
AUTOGEN renames each NEWPARAMS.DAT to NEWPARAMS.DONE as soon as it has finished processing the file.

---

#### Note

---

If AUTOGEN finds three versions of NEWPARAMS.DAT, it processes version 3, then version 2, and then version 1. After AUTOGEN has renamed the files to NEWPARAMS.DONE, their version numbers ARE reversed, reflecting the fact that the oldest file was processed and renamed the most recently.

The system manager can purge these files at any time. AUTOGEN, however, does not delete these files immediately so that you can examine them if some problem occurs with the layered product installation or with a subsequent run of AUTOGEN.

---

- **CLU\$PARAMS.DAT**  
This file receives the parameter values of the NEWPARAMS.DAT files that AUTOGEN processes.

In NEWPARAMS.DAT the AUTOGEN code expects to find records that are comments (which are not passed to CLU\$PARAMS.DAT), and parameter assignments very much like those in MODPARAMS.DAT. These are usually assignments similar to ADD\_parameter and MIN\_parameter:

- The first setting, ADD\_parameter, defines the amount of a particular resource (for example, NPAGEDYN, GBLPAGES) the layered product requires.
- The second setting, MIN\_parameter, provides floor for the calculation of a parameter (for example, PQL\_DWSDEFAULT).

The parameter settings in NEWPARAMS.DAT are integrated into CLU\$PARAMS.DAT. This file is then used, along with MODPARAMS.DAT and the feedback and hardware configuration data, to calculate parameter values in the GENPARAMS phase. System managers do not need to modify CLU\$PARAMS.DAT; MODPARAMS.DAT continues to be the proper file to contain system-specific parameter changes.

### 6.1.1.2 What Goes into NEWPARAMS.DAT

The following sections describe what is placed in the NEWPARAMS.DAT file.

**6.1.1.2.1 Product Name** A very important difference between the parameter assignments in MODPARAMS.DAT, which is familiar to most system managers, and those in NEWPARAMS.DAT: the name of the layered product that makes the assignments.

---

**Notes**

---

If the name of the product changes from one kit to the next, the system ends up with parameter changes made under **both** names. Therefore, choose the name carefully so that you do not need to change it in future kits. In addition, do not include version numbers.

AUTOGEN no longer allows layered product kits to provide NEWPARAMS.DAT records that do not include a product name.

---

You can pass the layered product name to AUTOGEN in either of the following ways:

- Prepend the name and a dollar-sign (\$) to each parameter name; for example:

```
DW-MOTIF$ADD_GBLPAGES=28000
```

- Include the name as a comment, which must begin as follows:

```
!Set by
```

The entire remainder of the comment is used as the product name for example:

```
MIN_GBLPAGES=62000 !Set by DW-MOTIF
```

In this example, DW-MOTIF becomes the product name.

The first method takes precedence over the second method. For example, someone might enter a value like the following:

```
ABBOTT$add_npagedyn=1000000 !Set by COSTELLO
```

In this example, the prefix (ABBOTT) will be used, and the product name specified in the comment (COSTELLO) will be ignored.

HP recommends that you use one method or the other to avoid confusion.

**6.1.1.2.2 Parameter Assignment** Except for specifying the product name, parameter assignment with NEWPARAMS.DAT works the same as it does in MODPARAMS.DAT, where you can set values, floors and ceilings, and specify amounts to add to a parameter; for example:

```
WINDOW_SYSTEM = 1  
MIN_GBLSECTIONS = 600  
MAX_WSMAX = 250000  
ADD_GBLPAGES = 25000
```

## AUTOGEN Command Procedure

### 6.1 AUTOGEN Description

**6.1.1.2.3 How to Remove Assignments from CLU\$PARAMS.DAT** You might want to remove one or more parameter assignments from CLU\$PARAMS.DAT for a number of reasons. A layered product might no longer need to use a value other than the default value of a parameter. Also, occasionally OpenVMS Engineering makes a parameters obsolete (for example, VIRTUALPAGECNT), and layered product kits need to include a way to remove these parameters from a system.

To remove an assignment from CLU\$PARAMS.DAT, create a NEWPARAMS.DAT that includes either of the following syntaxes:

```
AGEN$REMOVE_PARAM <parameter> !Set by <product>
AGEN$REMOVE_PARAM <product>$<parameter>
```

The rules explained in Section 6.1.1.2.1 for specifying product names also apply here. Also, you can mix such removals with assignments in a single NEWPARAMS.DAT file. Use removals to remove assignments of parameters that will no longer be needed; to change a parameter value, simply assign the new value to the parameter in NEWPARAMS.DAT, and the new value will replace the old value.

Note that AUTOGEN does not parse anything between the parameter name and the *!Set by* comment. This allows the kit producer to use NEWPARAMS.DAT for installation and to create a NEWPARAMS.DAT for deinstallation by prepending "AGEN\$REMOVE\_PARAM " to the beginning of each line.

#### 6.1.1.3 What CLU\$PARAMS.DAT Looks Like

The format of CLU\$PARAMS.DAT, which is used to define layered- product-driven parameter modifications, is as follows:

```
! The file contains parameters supplied by layered products.
! It should not be modified. Customer parameters should be placed in
! SYS$SYSTEM:MODPARAMS.DAT.
!=====
!
DW_MOTIF$MIN_CHANNELCNT = 255 !Set by DW-MOTIF
DW_MOTIF$ADD_GBLPAGES = 28000 !Set by DW-MOTIF
DW_MOTIF$MIN_GBLPAGES = 62000 !Set by DW-MOTIF
DW_MOTIF$ADD_GBLPAGFIL = 5000 !Set by DW-MOTIF
DW_MOTIF$MIN_GBLPAGFIL = 6024 !Set by DW-MOTIF
DECNET_PLUS$MIN_GBLPAGES = 55000 !Set by DECnet-Plus
DECNET_PLUS$ADD_GBLPAGES = 24000 !Set by DECnet-Plus
```

This example shows two layered product installations that use NEWPARAMS.DAT: one is DW-MOTIF (the DECWindows kit), and the other is DECnet-Plus.

A subsequent installation of DW-MOTIF replaces the value of each parameter assignment in CLU\$PARAMS.DAT with the value found in NEWPARAMS.DAT. Whether the new kit has 255, 200 or 300 for MIN\_CHANNELCNT, the value that is supplied is the value found in the new copy of CLU\$PARAMS.DAT.

Similarly, the new value that DW\_MOTIF provides for ADD\_GBLPAGES replaces the 28000 in the assignment of ADD\_GBLPAGES shown in the last example. These values are not cumulative for a given product; they do, however, accumulate across layered products; therefore, the total ADD\_GBLPAGES value are as follows:

```
28000 + 24000, or 52000
```

Both NEWPARAMS.DAT and CLU\$PARAMS.DAT are ordinary text files created by text editors. HP does not recommend editing these files.

## 6.2 AUTOGEN Usage Summary

HP recommends that you run AUTOGEN in the following circumstances:

- During a new installation or upgrade.
- Whenever your work load changes significantly.
- When you add an optional (layered) software product. Certain layered products might require you to execute AUTOGEN to adjust parameter values and page and swap file sizes. (For information about using AUTOGEN to modify page and swap files, refer to the chapter on managing page, swap, and dump files in the *HP OpenVMS System Manager's Manual*.) Refer to specific product documentation for installation requirements.
- When you install images with the /SHARED attribute. The GBLSECTIONS and GBLPAGES parameters might need to be increased to accommodate additional use of global pages and global sections.
- During normal operation, as part of a batch-oriented command procedure that runs AUTOGEN on a regular basis and automatically sends a report to an appropriate Mail account. The recommended procedure is described in the chapter on managing system parameters in the *HP OpenVMS System Manager's Manual*.

After a new operating system installation or upgrade, examine the results of calculations that AUTOGEN made to determine whether AUTOGEN has set system parameter values that are appropriate for your workload requirements.

The system parameters listed in an appendix to this manual indicate whether they are affected by AUTOGEN calculations. AUTOGEN calculations also affect the size of page, swap and dump files.

## 6.3 Feedback

AUTOGEN feedback minimizes the necessity to modify parameter values or system file sizes. Feedback allows AUTOGEN to automatically size the operating system based on your actual work load. **Sizing** is the process of matching the allocation of system resources (memory and disk space) with the workload requirements of your site.

Feedback is information about how various resources are used by the system's work load. This information is continuously collected by the operating system executive. Because the system collects feedback when exception events occur, feedback collection does not affect system performance.

You control how AUTOGEN uses feedback by specifying an execution mode when you invoke AUTOGEN. When run in feedback mode, AUTOGEN analyzes this information and adjusts any related parameter values. For information about controlling AUTOGEN's use of feedback, see Section 6.5.

AUTOGEN collects feedback during the SAVPARAMS phase by executing the image SYS\$SYSTEM:AGEN\$FEEDBACK.EXE. AUTOGEN writes feedback information to the file SYS\$SYSTEM:AGEN\$FEEDBACK.DAT. This file is then read during the GETDATA phase. For more information about AUTOGEN phases, see Section 6.4.

The system parameters listed in an appendix to this manual indicate whether they are affected by AUTOGEN feedback.

# AUTOGEN Command Procedure

## 6.4 Phases

### 6.4 Phases

AUTOGEN executes in phases. You control which tasks AUTOGEN performs by specifying a **start phase** and an **end phase** when you invoke AUTOGEN. Table 6–1 lists the phases AUTOGEN can execute in order.

**Table 6–1 AUTOGEN Phases**

Phase	Description
SAVPARAMS	Saves dynamic feedback from the running system.
GETDATA	Collects all data to be used in AUTOGEN calculations.
GENPARAMS	Generates new system parameters; creates the installed image list.
TESTFILES	Displays the system page, swap, and dump file sizes calculated by AUTOGEN (cannot be used as a start phase).
GENFILES	Generates new system page, swap, and dump files if appropriate (cannot be used as a start phase).
SETPARAMS	Runs SYSMAN to set the new system parameters in the default parameter file, saves the original parameters, and generates a new parameter file, AUTOGEN.PAR.
SHUTDOWN	Prepares the system to await a manual reboot.
REBOOT	Automatically shuts down and reboots the system.
HELP	Displays help information to the screen.

The following sections describe each phase in detail.

#### 6.4.1 SAVPARAMS

The SAVPARAMS phase records feedback in the file AGEN\$FEEDBACK.DAT, which can be used in subsequent AUTOGEN phases. If you specify NOFEEDBACK as the **execution-mode** parameter, the information collected is not used.

The SAVPARAMS phase is valid as a start phase and end phase. SAVPARAMS requires the SYSPRV and CMKRNL privileges.

---

#### Note

You can specify the SAVE\_FEEDBACK option during an interactive orderly shutdown with SYS\$SYSTEM:SHUTDOWN.COM. Entering this option in response to the prompt “Shutdown options:” records feedback collected since the system was last booted. Using the SAVE\_FEEDBACK option creates a new version of SYS\$SYSTEM:AGEN\$FEEDBACK.DAT. Run AUTOGEN from the GETDATA phase after the system reboots to use this new version of the feedback.

---



### 6.4.2 GETDATA

The GETDATA phase collects the following information required for AUTOGEN calculations and places it in the file PARAMS.DAT:

- Hardware configuration data
- HP-supplied data from CLU\$PARAMS.DAT
- Feedback from AGEN\$FEEDBACK.DAT (if run in feedback mode)
- User-supplied data from MODPARAMS.DAT

The GETDATA phase also attempts to configure devices on the system, by executing the following procedure and command:

- The command procedure SYS\$MANAGER:SYCONFIG.COM. (For more information about this procedure, refer to the chapter on managing devices in the *HP OpenVMS System Manager's Manual*.)
- The SYSGEN command AUTOCONFIGURE ALL (unless the symbol STARTUP\$AUTOCONFIGURE\_ALL is set to 0 in SYCONFIG.COM).

The GETDATA phase is valid as a start phase and an end phase. GETDATA requires the SYSPRV and CMKRNL privileges.

### 6.4.3 GENPARAMS

In the GENPARAMS phase, AUTOGEN calculates the parameter values based on data stored in PARAMS.DAT and produces SETPARAMS.DAT as output. AUTOGEN checks to see if feedback is included, and if so, uses it in the calculations unless the NOFEEDBACK execution mode was specified when AUTOGEN was invoked. Also during this phase, AUTOGEN generates the known image file list (VMSIMAGES.DAT).

The GENPARAMS phase is valid as a start phase and an end phase. GENPARAMS requires the SYSPRV and OPER privileges.

### 6.4.4 TESTFILES

The TESTFILES phase displays system page, swap, and dump file sizes calculated by AUTOGEN. (This phase does not change the file sizes.)

File sizes for all currently installed primary and secondary page and swap files are displayed. The information is directed to SYS\$OUTPUT and the AGEN\$PARAMS.REPORT file by default.

Specify the TESTFILES phase to display AUTOGEN's file size calculations; to generate new sized files, specify the GENFILES phase. You cannot specify both of these phases when invoking AUTOGEN. HP recommends that you use TESTFILES to display the file size changes before actually generating new sized files on your system.

The TESTFILES phase is valid only as an end phase. TESTFILES requires the SYSPRV privilege.

## AUTOGEN Command Procedure

### 6.4 Phases

#### 6.4.5 GENFILES

The GENFILES phase generates the new page, swap, and dump files on the system. This phase changes the file sizes based on AUTOGEN's calculations.

The GENFILES phase does not modify a file if the calculated size change is within ten percent of the existing file size. The following files are affected: PAGEFILE.SYS, SWAPFILE.SYS, SYSDUMP.DMP, and all other currently installed page and swap files. For more information, refer to the chapter on managing page, swap and dump files in the *HP OpenVMS System Manager's Manual*.

GENFILES is valid only as an end phase. GENFILES requires the SYSPRV privilege.

#### 6.4.6 SETPARAMS

The SETPARAMS phase uses as its input the SETPARAMS.DAT file created during the GENPARAMS phase. In this phase, AUTOGEN runs SYSMAN to update the system parameter values in the default parameter file.

On VAX systems, the default parameter file is SYS\$SYSTEM:VAXVMSSYS.PAR. AUTOGEN saves the current system parameters in the file SYS\$SYSTEM:VAXVMSSYS.OLD before updating these parameters in SYS\$SYSTEM:VAXVMSSYS.PAR. The new values are also saved in SYS\$SYSTEM:AUTOGEN.PAR.

On Alpha systems, SYS\$SYSTEM:ALPHAVMSSYS.PAR is the default parameter file. AUTOGEN saves the current system parameters in the file SYS\$SYSTEM:ALPHAVMSSYS.OLD before updating these parameters in SYS\$SYSTEM:ALPHAVMSSYS.PAR. The new values are also saved in SYS\$SYSTEM:AUTOGEN.PAR.

On I64 systems, SYS\$SYSTEM:IA64VMSSYS.PAR is the default parameter file. AUTOGEN saves the current system parameters in the file SYS\$SYSTEM:IA64VMSSYS.OLD before updating these parameters in SYS\$SYSTEM:IA64VMSSYS.PAR. The new values are also saved in SYS\$SYSTEM:AUTOGEN.PAR.

The SETPARAMS phase is valid as a start phase and an end phase. SETPARAMS requires the SYSPRV and OPER privileges.

#### 6.4.7 SHUTDOWN

SHUTDOWN shuts down the system and awaits a manual reboot. To use the new system parameter values generated in the SETPARAMS phase, specify either SHUTDOWN or REBOOT as the end phase. You can define the logical name AGEN\$SHUTDOWN\_TIME (using the DCL command DEFINE) to specify the number of minutes before shutdown occurs.

SHUTDOWN requires the SETPRV privilege.

#### 6.4.8 REBOOT

REBOOT automatically shuts down and reboots the system, thus installing the new parameter values. To install the new system parameter values generated in the SETPARAMS phase, specify either SHUTDOWN or REBOOT as the end phase. You can define the logical name AGEN\$SHUTDOWN\_TIME (using the DCL command DEFINE) to specify the number of minutes before shutdown occurs.

REBOOT requires the SETPRV privilege.

### 6.4.9 HELP

HELP displays help information about AUTOGEN to the screen. The HELP phase is only valid as the start phase command line parameter. When you specify HELP for the start phase, the end phase and execution mode parameters are ignored.

## 6.5 Execution Modes

Specify an execution mode when you invoke AUTOGEN to control how AUTOGEN uses feedback. Table 6–2 lists the execution-mode options.

**Table 6–2 AUTOGEN Execution Modes**

Option	Description
FEEDBACK	Specifies that AUTOGEN run in feedback mode, using dynamic feedback collected during the SAVPARAMS phase to make its calculations.
NOFEEDBACK	Specifies that AUTOGEN not use feedback in the calculations. The feedback from the SAVPARAMS phase is ignored. Use NOFEEDBACK mode for the initial system installation or upgrade. NOFEEDBACK supersedes the execution-mode option INITIAL, which was used in a previous version of the operating system.
CHECK_FEEDBACK	Specifies that AUTOGEN use feedback in its calculations as long as the feedback is valid. If feedback is suspect, AUTOGEN does not use feedback in the calculations, but continues through the specified end phase.
Blank	If you do not specify an execution mode, AUTOGEN uses feedback in the calculations by default. However, if AUTOGEN determines that the feedback might be suspect, it performs the calculations, issues the feedback report, and stops before modifying any parameters or system files, even if you specified an end phase of GENFILES, SETPARAMS, SHUTDOWN or REBOOT.

## 6.6 Files Used by AUTOGEN

Table 6–3 lists the files AUTOGEN uses during each phase.

**Table 6–3 Files Used by AUTOGEN**

AUTOGEN Phase	Input Files <sup>1</sup>	Output Files <sup>1</sup>
SAVPARAMS	None	AGEN\$FEEDBACK.DAT
GETDATA	NEWPARAMS.DAT <sup>2</sup> CLU\$PARAMS.DAT	CLU\$PARAMS.DAT

<sup>1</sup>All files except VMSIMAGES.DAT, which contains the installed image list, reside in the SYS\$SYSTEM directory. VMSIMAGES.DAT resides in the SYS\$MANAGER directory.

<sup>2</sup>From software installation kit

(continued on next page)

## AUTOGEN Command Procedure

### 6.6 Files Used by AUTOGEN

**Table 6–3 (Cont.) Files Used by AUTOGEN**

AUTOGEN Phase	Input Files <sup>1</sup>	Output Files <sup>1</sup>
	AGEN\$FEEDBACK.DAT CLU\$PARAMS.DAT MODPARAMS.DAT	PARAMS.DAT <sup>3</sup>
GENPARAMS	PARAMS.DAT	SETPARAMS.DAT VMSIMAGES.DAT AGEN\$PARAMS.REPORT
TESTFILES	PARAMS.DAT	SYS\$OUTPUT
GENFILES	PARAMS.DAT	PAGEFILE.SYS SWAPFILE.SYS (and secondary page and swap files) SYSDUMP.DMP AGEN\$PARAMS.REPORT
SETPARAMS	SETPARAMS.DAT	‡ VAXVMSSYS.PAR (VAX) ‡ ALPHAVMSSYS.PAR (Alpha) *IA64VMSSYS.PAR (I64) AUTOGEN.PAR ‡ VAXVMSSYS.OLD (VAX) ‡ ALPHAVMSSYS.OLD (Alpha) *IA64VMSSYS.OLD (I64)
SHUTDOWN	None	None
REBOOT	None	None

<sup>1</sup>All files except VMSIMAGES.DAT, which contains the installed image list, reside in the SYS\$SYSTEM directory. VMSIMAGES.DAT resides in the SYS\$MANAGER directory.

<sup>3</sup>Also includes collected hardware configuration information

## 6.7 AUTOGEN Usage Summary

The AUTOGEN command procedure runs automatically when your system is installed or upgraded to set appropriate values for system parameters and sizes for system page, swap, and dump files.

Execute AUTOGEN to reset system parameter values and system file sizes. The new values and file sizes take effect the next time the system is booted.

### Format

```
@SYS$UPDATE:AUTOGEN [start-phase] [end-phase] [execution-mode]
```

#### Parameters

##### start-phase

Specify the phase where AUTOGEN is to begin executing. Table 6–1 lists the options for the *end-phase* parameter.

The phase specified for *start-phase* must either precede or be identical to the phase specified for *end-phase*, according to the sequence shown in Table 6–1. If you do not supply an option for the *start-phase* parameter, enter a null argument (that is, " "). If you do not specify a start phase, GENPARAMS is the default.

**end-phase**

Specify the phase where AUTOGEN is to complete executing. Table 6–1 lists the options for the *end-phase* parameter. If you do not specify an end phase, the end phase has the same value as the start phase by default.

**execution-mode**

Specify one of the following execution-mode options to control how AUTOGEN uses feedback:

- FEEDBACK
- NOFEEDBACK
- CHECK\_FEEDBACK
- Blank

Table 6–2 describes each execution-mode option.

**Description**

To invoke AUTOGEN, use the following syntax to enter a command at the DCL command prompt:

```
$ @SYS$UPDATE:AUTOGEN [start-phase] [end-phase] [execution-mode]
```

You are returned to DCL level when the command has finished processing unless you specify SHUTDOWN or REBOOT as the end-phase parameter.









## 7.1 BACKUP Description

The Backup utility (BACKUP) helps you prevent data loss or corruption by creating copies of your files, directories, and disks. In case of a problem, for example, a disk drive failure, you can restore the backup copy and continue your work with minimal disruption.

When you save files with BACKUP, it writes the files to a special file called a **save set**. Save sets are written in a format that only BACKUP can interpret. (A save set stored on a Files-11 disk is a standard OpenVMS file, however, and can be copied, renamed, deleted, or backed up. A save set stored on magnetic tape should only be processed with the BACKUP command; do not use the DCL command COPY to copy a magnetic tape save set to disk.)

Use BACKUP to perform the following tasks:

- Save disk files to a BACKUP save set.
- Restore files to disk from a BACKUP save set.
- Copy disk files to disk files.
- Compare disk files created by BACKUP or files in a BACKUP save set with disk files.
- List information about the files in a BACKUP save set.
- Create and list journal files that record the results of BACKUP save operations.
- Convert ODS-5 file names to ODS-2 file names.

For specific information about performing these tasks, refer to the *HP OpenVMS System Manager's Manual*.

---

**Note**

---

Some layered products have their own special backup procedures. For more information, refer to the layered product documentation.

---

Using BACKUP also eliminates disk fragmentation. Fragmentation can occur as you create and extend files on a disk. If the file system cannot store files in contiguous blocks, it stores them in noncontiguous pieces. Eventually, the disk can become severely fragmented and system performance suffers. To eliminate fragmentation, perform an image backup of the disk and restore the backup copy. When you restore the image backup, BACKUP places the files on the disk contiguously.

Besides backing up your own files, directories, and disks, remember to back up your OpenVMS system disk. Depending on the policy at your site, individuals may be responsible for backing up their system disks, or an operator or system manager may perform the backup (as would likely be the case in a large, clustered computer system).

## BACKUP

### 7.1 BACKUP Description

The two ways to back up your system disk are:

- If you have access to the OpenVMS Alpha, I64, or VAX system CD-ROM, you can use a menu system supplied on the CD-ROM to back up your system disk.
- If you do *not* have access to the OpenVMS VAX system CD-ROM, you must use standalone BACKUP to back up your system disk (VAX only).

Refer to the *HP OpenVMS System Manager's Manual* for more information about standalone BACKUP and the menu-driven procedure.

Types of backup operations are:

- An **image backup** (also called a full backup) saves a copy of all the files on a disk (or volume) to a save set. The first backup that you do on a disk must be an image backup; you cannot perform an incremental backup first.
- An **image restore** initializes the output disk and restores an entire volume.
- An **image copy** operation initializes the output disk and copies an entire volume; the image backup is a logical duplicate of the contents of the disk.
- An **image compare** operation compares the contents of entire volumes.

---

#### Note

---

Because an image copy or backup operation processes all files on the input volume, you cannot specify file-selection qualifiers for these operations. You can, however, restore files and directories selectively from an image save set.

---

If the output volume of an image operation is a disk, BACKUP stores all files on the output volume contiguously, eliminating disk fragmentation and creating contiguous free blocks of disk space.

- An **incremental backup** saves only those files that have been created or modified since the most recent backup that was performed using the /RECORD qualifier. (The /RECORD qualifier records the date and time that the files are backed up.)
- An **incremental restore** operation restores an incremental save set. Specify the command qualifier /INCREMENTAL in an incremental restore operation.
- A **file operation** processes individual files or directories.
- A **selective operation** process files or volumes selectively, according to criteria such as version number, file type, UIC, date and time of creation, expiration date, or modification date.

Perform selective save operations by using wildcard characters and input file-selection qualifiers (for example, /BACKUP, /BEFORE, /BY\_OWNER (use instead of /OWNER\_UIC), /CREATED, /EXCLUDE, /EXPIRED, /MODIFIED, and /SINCE).

- A **physical operation** copies, saves, restores, or compares an entire volume in terms of logical blocks, ignoring any file structure.

BACKUP allocates virtual memory to hold copies of the index file and storage bitmaps. With larger bitmaps, the virtual memory requirement of this utility increases correspondingly. To use BACKUP on volumes with large bitmaps, you might need to increase your page file quota. On OpenVMS VAX systems, you might also need to increase the system parameter VIRTUALPAGECNT.

Sizes of virtual memory requirements for the bitmaps are VAX pages (or Alpha and I64 512-byte pagelets) per block of bitmap. For the BACKUP utility, the virtual memory requirement for the bitmaps is equal to the sum of the sizes of all index file bitmaps on the volume set. (Note that this memory requirement is in addition to the BACKUP utility's substantial buffer pool.)

The following sections describe the BACKUP command line format.

## 7.2 BACKUP Command Line Format

To perform BACKUP operations, enter the DCL command BACKUP in the following format:

BACKUP input-specifier output-specifier

BACKUP evaluates the input and output specifiers to determine which type of operation to perform. BACKUP also uses the input specifier to locate the input and directs output to the output specifier.

## 7.3 BACKUP Input and Output Specifiers

BACKUP can process several different types of input and output. Depending on the type of operation being executed, input and output specifiers can be standard OpenVMS file specifications, BACKUP save-set specifications, or device specifications. Device specifications can refer to disk or magnetic tape volumes.

You can specify any valid OpenVMS file specification as BACKUP input or output specifiers; however, BACKUP does not allow node names in BACKUP file specifications. You can use wildcard characters, and you can list multiple file specifications as input to a single BACKUP operation.

A BACKUP save-set specification is the file specification of a BACKUP save set. When you use BACKUP to save files or volumes, BACKUP writes your files to a save set. You can specify the save set as input to other BACKUP operations. When specifying a save set, follow the rules for specifying a OpenVMS file. The *OpenVMS User's Manual* describes valid specifications for disk files; the *HP OpenVMS System Manager's Manual* explains the rules for specifying magnetic tape files. A save-set specification has no default file type, although you can use BCK or SAV.

The save-set name can be any valid OpenVMS file name and type. However, when you create a save set on magnetic tape, the save-set name has the following restrictions:

- The entire save-set name cannot exceed 17 characters, including the period delimiter.
- You cannot specify a version number.
- You cannot specify a directory name.

Device specifications used as BACKUP input or output specifiers follow the conventions for specifying devices outlined in the *OpenVMS User's Manual*.

## BACKUP

### 7.3 BACKUP Input and Output Specifiers

By default, BACKUP treats an input or output specifier referring to a Files–11 disk as a file specification. Therefore, to identify a save set on a Files–11 volume, you must include the /SAVE\_SET qualifier with the specifier (see /SAVE\_SET). BACKUP treats input and output specifiers referring to magnetic tape as save sets.

---

#### Note

---

You cannot specify a save set for both the input and output specifier of a BACKUP command. For this reason, you cannot perform a BACKUP operation from one magnetic tape to another.

---

Table 7–1 shows input and output specifiers for each type of BACKUP operation.

**Table 7–1 BACKUP Input and Output by Operation Type**

Operation	Format
Save	BACKUP file-spec save-set-spec
Save (image)	BACKUP/IMAGE device-spec save-set-spec
Save (physical to disk)	BACKUP/PHYSICAL device-spec device-spec
Restore	BACKUP save-set-spec file-spec
Restore (image)	BACKUP/IMAGE save-set-spec device-spec
Restore (physical from disk)	BACKUP/PHYSICAL save-set-spec device-spec
Restore (physical from tape)	BACKUP/PHYSICAL save-set-spec device-spec
Copy	BACKUP file-spec file-spec
Copy (image)	BACKUP/IMAGE device-spec device-spec
Copy (physical to tape)	BACKUP/PHYSICAL device-spec save-set-spec
Compare	BACKUP/COMPARE file-spec file-spec BACKUP/COMPARE save-set-spec file-spec
Compare (image)	BACKUP/COMPARE/IMAGE save-set-spec device-spec BACKUP/COMPARE/IMAGE device-spec device-spec
Compare (physical)	BACKUP/COMPARE/PHYSICAL device-spec device-spec BACKUP/COMPARE/PHYSICAL save-set-spec device-spec
List <sup>1</sup>	BACKUP/LIST[=file-spec] save-set-spec BACKUP/LIST[=file-spec] device-spec
Create Journal	BACKUP/JOURNAL[=file-spec] file-spec save-set-spec
List Journal	BACKUP/JOURNAL[=file-spec]/LIST[=file-spec]

---

<sup>1</sup>Can also be used with any other operation listed here.

---

#### 7.3.1 Input and Output Specifier Element Lists

An **element list** is a list of arguments specified with a command or qualifier. The arguments, or elements, in the list are separated by commas. Element lists relating to input or output specifiers are allowed only in the following circumstances:

7.3 BACKUP Input and Output Specifiers

- If an input specifier refers to a Files-11 disk, you can construct lists from standard OpenVMS file specifications, as follows:

```
$ BACKUP
  _From: DUA0:[DATA]A.DAT,B.DAT,[PROGRAMS]TEST.EXE
  _To: MSA0:TEST.SAV/LABEL=DLY101
```

- If an input specifier or an output specifier refers to a BACKUP save set on magnetic tape or sequential disk, you can specify more than one device name to be used in the operation. This allows you to process multivolume save sets efficiently by specifying the order in which devices will be used. The first volume is processed until it is full. The second (or subsequent) volume is processed while the media in the first (or previous) volume is being changed. However, the save-set name must appear with the first element in the list and must not appear in subsequent elements in the list.

In the following example, BACKUP first saves data to a tape in drive MSA0, then to a tape in drive MSA1. When the tape in drive MSA1 is full, BACKUP saves data to a fresh tape in MSA0.

```
$ BACKUP
  _From: DUA0:[DATA]*.*,DUA0:[PROGRAMS]*.*
  _To: MSA0:TEST.SAV,MSA1:/LABEL=WKLY01
```

- If you are performing an image operation on a volume set, you can specify element lists in the input and output specifiers. In the following example, BACKUP first restores the save set TEST.SAV from the tape in drive MSA0, and then continues to restore the save set from the tape in drive MSA1. BACKUP first restores this save set to DUA0. When DUA0 is full, BACKUP continues the restore operation to DUA1.

```
$ BACKUP/IMAGE
  _From: MSA0:TEST.SAV,MSA1:
  _To: DUA0:[DATA...],DUA1:
```

7.3.2 BACKUP Qualifiers

You can also affect BACKUP operations by specifying qualifiers. BACKUP has five types of qualifiers:

- **Command qualifiers** 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 or compare 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.

---

## BACKUP

### 7.3 BACKUP Input and Output Specifiers

It is important to understand the differences between the types of qualifiers. The position of qualifiers in the BACKUP command line affects the results of the command. Although command qualifiers can be placed anywhere in the command line, input- and output-specifier qualifiers are position-dependent. That is, input-specifier qualifiers must be placed immediately after the input specifier, and output-specifier qualifiers must be placed immediately after the output specifier.

Additionally, several BACKUP qualifiers are both input-specifier qualifiers and output-specifier qualifiers. To achieve the results you want from a BACKUP command, ensure that you place position-dependent qualifiers correctly. For example, use the /SAVE\_SET qualifier as an output save-set qualifier in a BACKUP save operation and as an input save-set qualifier in a BACKUP restore operation.

Appendix G contains more information about valid combinations of BACKUP qualifiers.

#### 7.3.3 Using Wildcard Characters with BACKUP

BACKUP allows you to use wildcard characters in file specifications to represent directories, file names, file types, and version numbers. Omitted file names, file types, or version numbers are assumed to be the asterisk wildcard character (\*). For instance, if you omit the version number, BACKUP processes all versions. (For introductory information about wildcard characters, refer to the *OpenVMS User's Manual*.)

You can use any valid DCL wildcard character with input specifiers that are Files-11 media or with the /SELECT and /EXCLUDE qualifiers. Note, however, that the symbols denoting the latest versions of files (;) and relative versions of files (;-n) are processed as the asterisk wildcard character (;\*) when they are used with the /EXCLUDE and /SELECT qualifiers.

You cannot use wildcard characters in BACKUP save-set specifications unless the save sets are input specifiers on tape.

##### Using Wildcard Characters to Represent Directories

The following table lists the types of directory wildcards allowed for output specifiers that are Files-11 media:

Directory Wildcard	Result
omitted	If a directory name is omitted, BACKUP restores file to the current default directory [].
[* . . . ]	BACKUP restores files to the directory from which they were saved.
[directory]	BACKUP restores files to the named directory.
[directory . . . ]	The wildcard characters used in the specification of the input files determine the directory to which BACKUP restores the files.

---

##### Note

If you specify directory wildcard characters incorrectly and your directories contain many levels of subdirectories, you risk losing the lower level subdirectories in BACKUP operations because OpenVMS directory

## BACKUP

### 7.3 BACKUP Input and Output Specifiers

trees can have only 8 levels with ODS-2 files. ODS-5 files, however, do not have this 8-level restriction.

---

The following example uses the directory wildcard format [directory . . . ] for both the input and the output specifiers:

```
$ BACKUP [OSCAR...] [JOE.RECEIVED...]
```

In this example, BACKUP creates a directory named [JOE.RECEIVED] (if it does not already exist) as well as subdirectories that correspond to the subdirectories of [OSCAR]. BACKUP copies all files from the directory [OSCAR] and its subdirectories to [JOE.RECEIVED] and its subdirectories. If [OSCAR] has 8 levels of directories, however, and files in it are ODS-2, BACKUP is unable to create a corresponding 9-level subdirectory to [JOE.RECEIVED]; the 8-level subdirectory to [OSCAR] is not copied. (This restriction does not apply to ODS-5 files.)

If you use the asterisk wildcard character (\*) to represent subdirectories in the input specifier of a copy operation, BACKUP creates subdirectories to the directory specified in the output specifier that correspond to the subdirectories in the input specifier. BACKUP then copies all files from the lowest level subdirectory in the input specifier to the lowest level subdirectory in the output specifier. In the following example, the asterisk represents subdirectories named MONDAY and TUESDAY:

```
$ BACKUP [SAM.WORK.*.WEDNESDAY] [JAMES...]
```

In this example, BACKUP creates a subdirectory named [JAMES.MONDAY.TUESDAY.WEDNESDAY]. In doing so, BACKUP copies the file MONDAY.DIR to [JAMES], copies the file TUESDAY.DIR to [JAMES.MONDAY], and copies the file WEDNESDAY.DIR to [JAMES.MONDAY.TUESDAY]. Then BACKUP copies all files from [SAM.WORK.MONDAY.TUESDAY.WEDNESDAY] to [JAMES.MONDAY.TUESDAY.WEDNESDAY].

In a restore operation, the input specifier defaults to [\* . . . ] if the input save-set qualifier /SELECT is not used; this is important if you use the form [directory . . . ] in the output specifier. The function of the wildcard [\* . . . ] is to carry over the entire directory name from the first level on and to place it before the ellipsis in the output specifier. Thus, if the save set in the following example contains the directory tree [SAVE . . . ], the restored directory tree will be [WORK.SAVE . . . ]:

```
$ BACKUP MTA0:SAVE.BCK [WORK...]
```

Note that the result will be the same, even if your output specifier has the same name as the directory in the input specifier, as shown in the following example:

```
$ BACKUP MTA0:SAVE.BCK [SAVE...]
```

The preceding command restores the directory tree [SAVE . . . ] to a directory tree named [SAVE.SAVE . . . ].

The following command restores the directory tree [SAVE . . . ] to a directory tree named [WORK . . . ]:

```
$ BACKUP MTA0:SAVE.BCK/SELECT=[SAVE...] [WORK...]
```

## BACKUP

### 7.3 BACKUP Input and Output Specifiers

There are two ways to retain the original directory name when you restore files. You must either use the form [\* . . . ] for the output specifier, or you must specify the input save-set qualifier /SELECT. The following example uses the form [\* . . . ] in the output specifier to restore the directory tree [SAVE . . . ] in save set SAVE.BCK to the directory tree [SAVE . . . ]:

```
$ BACKUP MTA0:SAVE.BCK [*...]
```

The input save-set qualifier /SELECT causes only the ellipsis portion of the selected file specification to be carried over to the directory tree named in the output specifier [directory . . . ]. The following command restores [SAVE . . . ] to [SAVE . . . ]:

```
$ BACKUP MTA0:SAVE.BCK/SELECT=[SAVE...] [SAVE...]
```

### 7.4 BACKUP Usage Summary

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 OpenVMS systems.

The two ways to back up your system disk are:

- If you have access to the CD-ROM of the current version of OpenVMS Alpha, I64, or VAX, you can use a menu-driven procedure to back up your system disk.
- If you do not have access to the CD-ROM of the current version of OpenVMS Alpha, I64, or VAX, you must use standalone BACKUP to back up your system disk. Standalone BACKUP is a version of the Backup utility that is bootstrapped into main memory instead of running under the control of the OpenVMS VAX operating system. Standalone BACKUP uses a subset of BACKUP qualifiers to perform image and physical BACKUP operations.

### Format

```
BACKUP input-specifier output-specifier
```

#### Parameters

##### input specifier

Specifies the input for the BACKUP operation. The input specifier can be a standard OpenVMS 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 node names are allowed only in save-set specifications.

Wildcards are permitted in standard OpenVMS 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 OpenVMS 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 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. Refer to Section 7.3.3 for restrictions on the use of wildcard characters in BACKUP commands.

## Description

To invoke online BACKUP, enter an appropriate BACKUP command at the DCL prompt. For instructions on invoking standalone BACKUP, refer to the *HP OpenVMS System Manager's Manual*.

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.

## 7.5 BACKUP Qualifiers

This section describes and provides examples of each BACKUP qualifier. Make sure that you understand how the position of BACKUP qualifiers affects BACKUP operations. See Section 7.2 for information about the BACKUP command line format. Table 7–2 summarizes the BACKUP qualifiers.

**Table 7–2 BACKUP Qualifier Summary**

Qualifier	Description
/ALIAS	Specifies whether to maintain the previous behavior of multiple processing of alias and primary file entries.
/ASSIST	Allows operator or user intervention if a request to mount a magnetic tape fails during a BACKUP operation.
/BACKUP	Selects files according to the BACKUP date written in the file header record by the BACKUP/RECORD command.
/BEFORE	Selects files dated earlier than the date and time you specify.
/BLOCK_SIZE	Specifies the output block size in bytes for data records in a BACKUP save set.
/BRIEF	Causes the /LIST qualifier to display the file specification, size (in blocks), and creation date for each file in the save set.
/BUFFER_COUNT	This qualifier is obsolete and has no effect.

(continued on next page)

## BACKUP

### 7.5 BACKUP Qualifiers

**Table 7–2 (Cont.) BACKUP Qualifier Summary**

Qualifier	Description
/BY_OWNER	As an input file-selection qualifier, /BY_OWNER causes BACKUP to process files owned by the specified UIC. As an output file qualifier, /BY_OWNER redefines the owner user identification code (UIC) for restored files. As an output save-set qualifier, /BY_OWNER specifies the owner user identification code (UIC) of the save set.
/COMMENT	Places the string that you supply into the BACKUP summary record of the output save set.
/COMPARE	Causes BACKUP to compare the contents of the first parameter with the contents of the second parameter.
/CONFIRM	Displays prompts on your terminal for confirmation before processing each file.
/CONVERT	Converts ODS-5 file names to ODS-2 file names.
/CRC	As an input save-set qualifier, /CRC checks the software cyclic redundancy check (CRC) encoded in the save set's data blocks. As an output save-set qualifier, /CRC specifies that the CRC is to be computed and stored in the data blocks of the output save set.
/CREATED	Selects files according to the value of the creation date field in each file header record.
/DELETE	Specifies that a BACKUP save or copy operation is to delete the selected input files from the input volume after all files have been successfully processed.
/DENSITY	Specifies the recording density of the output magnetic tape.
/EXACT_ORDER	Specifies the exact order of tape volume labels that you want to use in a BACKUP operation.
/EXCLUDE	Excludes files from processing that otherwise meet the selection criteria for a save or copy operation.
/EXPIRED	Selects files according to the value of the expiration date field in each file header record.
/FAST	Processes the input specifier using a fast file scan to reduce processing time.
/FILES_SELECTED	Specifies a file that contains a list of the files that are to be selected when a save set is restored.
/FULL	Displays the information produced by the /LIST command qualifier in a format similar to that displayed by the DCL command DIRECTORY/FULL.
/GROUP_SIZE	Defines the number of blocks BACKUP places in each redundancy group.
/HEADER_ONLY	Controls whether BACKUP saves only the file header of shelved and preshelved files.
/IGNORE	Specifies that a BACKUP save or copy operation will override restrictions placed on files or will not perform tape label processing checks.
/IMAGE	Directs BACKUP to process an entire volume or volume set.

(continued on next page)

**Table 7–2 (Cont.) BACKUP Qualifier Summary**

Qualifier	Description
/INCREMENTAL	Allows you to restore a disk volume from a series of incremental save sets.
/INITIALIZE	Initializes an output disk volume, making its entire previous contents unavailable.
/INPUT_FILES	Directs BACKUP to treat the input-specifier as the file name of a list of files. This file specifies the input files for a BACKUP operation.
/INTERCHANGE	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.
/JOURNAL	Specifies that a BACKUP save operation is to create, or append information to, a BACKUP journal file.
/LABEL	Specifies the 1- to 6- character volume labels for the magnetic tapes and 1- to 12- character volume labels for disks to which the save set is written.
/LIST	Lists information about a BACKUP save set and about the files in a save set.
/LOG	Displays the file specification of each file processed during the operation on SYS\$OUTPUT.
/MEDIA_FORMAT	Controls whether data records are automatically compacted and blocked together.
/MODIFIED	Selects files according to the value of the modified date field (the date the file was last modified) in each file header record.
/NEW_VERSION	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.
/NOINCREMENTAL	Allows you to control the amount of file data that is saved in a save operation.
/OVERLAY	Writes over an existing file when an identically named file is encountered during the restore operation.
/OWNER_UIC	The /OWNER_UIC qualifier has been superseded by the /BY_OWNER qualifier. HP recommends that you substitute /BY_OWNER for OWNER_UIC in command procedures and operator instructions. See the description of the /BY_OWNER qualifier for more information.
/PHYSICAL	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.
/PROTECTION	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.
/RECORD	Records the current date and time in the BACKUP date field of each file header once a file is successfully saved or copied.
/RELEASE_TAPE	Dismounts and unloads a tape after a BACKUP save operation either writes and verifies the save set, or reaches the end of the tape.

(continued on next page)

## BACKUP

### 7.5 BACKUP Qualifiers

**Table 7–2 (Cont.) BACKUP Qualifier Summary**

Qualifier	Description
/REPLACE	Replaces a file on the output specifier with an identically named file from the input specifier.
/REWIND	As an input save-set qualifier, /REWIND rewinds the input tape reel to the beginning-of-tape marker before reading the input volume. As an output save-set qualifier, /REWIND rewinds the output tape to the beginning-of-tape marker and initializes the output tape.
/SAVE_SET	As an input save-set qualifier, /SAVE_SET directs BACKUP to treat the input file as a BACKUP save set. As an output save-set qualifier, /SAVE_SET directs BACKUP to treat the output file as a BACKUP save set.
/SELECT	Selects the specified files for processing.
/SINCE	Selects files dated equal to or later than the specified date and time.
/TAPE_EXPIRATION	Writes a file expiration date other than the current date to the file header label of the save set.
/TRUNCATE	Controls whether a copy or restore operation truncates a sequential output file at the end-of-file (EOF) when creating it.
/VERIFY	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.
/VOLUME	Indicates that a specific disk volume in a disk volume set is to be processed.

---

## **/ALIAS**

### **Command Qualifier**

Specifies that the previous behavior of multiple processing of alias and primary file entries be maintained.

---

#### **Note**

---

Use the /ALIAS qualifier **only** when you are restoring very old save sets (from OpenVMS Version 6.2 or earlier). The current default behavior is correct in nearly every other situation. If you are in doubt about using this qualifier, contact your HP support representative.

---

## **Description**

The /ALIAS qualifier maintains the previous BACKUP behavior of treating alias file entries the same as primary file entries. Therefore, a primary file may be processed multiple times by BACKUP if one or more alias file entries reference the same primary file entry.

If you specify /NOALIAS, alias directory and file entries are ignored. Therefore, multiple processing of primary files may be avoided, which saves time and save-set file space. If a restore operation is performed using the /ALIAS qualifier but the save set was created by using the /NOALIAS qualifier, a message is displayed that the /ALIAS qualifier will be ignored.

## **Format**

**/ALIAS** save-set-spec (default)

**/NOALIAS**

## BACKUP /ASSIST

---

### /ASSIST

#### Command Qualifier

Allows operator or user intervention during a BACKUP operation if a magnetic tape mount request fails or if an operation requires another volume.

#### 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 or when an operation requires another volume. BACKUP sends messages to operator terminals enabled to receive TAPES and CENTRAL messages. (See the description of the REPLY command in the *HP OpenVMS DCL Dictionary* 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.

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. The /NOASSIST qualifier has no effect if the logical name SYS\$COMMAND points to a device that is not a terminal (as is the case when you run BACKUP in a batch job). 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 Levels 2 and 5 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.

The /BACKUP qualifier 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 (that is, /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 (that is, /RECORD was not specified when the file was saved or copied) are *not* selected.

**Examples**

1. \$ BACKUP/RECORD  
\_From: [PAYROLL]\*.\*;\*/BEFORE=01-SEP-2002/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, 2002. The command qualifier /RECORD writes the date and time of the save operation to the file header record of each saved file.

2. \$ BACKUP/RECORD [ACCOUNTS...]/SINCE=YESTERDAY/BACKUP MTA1:ACC.BCK

In this command, the /BACKUP qualifier combined with the /SINCE qualifier saves all files in all subdirectories of [ACCOUNTS] that have a BACKUP date written since yesterday (24 hours before midnight last night). The command qualifier /RECORD writes the date and time of the save operation to the file header record of each saved file.

## BACKUP /BEFORE

---

### /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 (and their functions) that you can use with the /BEFORE qualifier. Use these other qualifiers only one 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.

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 Levels 2 or 5 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-spec/BLOCK\_SIZE=n

### **Description**

You can specify a block size between 2048 and 65,535 bytes. BACKUP may adjust this value according to the constraints of the BACKUP format. Although BACKUP may adjust the block size you specify, it does not adjust the block size over the maximum of 65,535.

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).

# BACKUP /BRIEF

---

## /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 the information in a format similar to that displayed by the DCL command DIRECTORY/FULL.

### Format

```
/LIST/BRIEF save-set-spec
```

### Example

```
$ BACKUP/LIST/BRIEF DBA2:[SAVE]23MAR02.BCK/SAVE_SET
Listing of save set(s)
Save set:          23MAR02.BCK
Written by:       MOROCI
UIC:              [000200,000200]
Date:             23-MAR-2002 14:18:16.00
Command:          BACKUP [SAVE] DBA2:[SAVE]23MAR02.BCK/SAVE_SET
Operating system: OpenVMS Alpha Version 7.3-1
BACKUP version:   V7.3-1
CPU ID register:  08000000
Node name:        _SUZI::
Written on:       _DBA2:
Block size:       32,256
Group size:       10
Buffer count:     3

[SAVE]INFO.TXT;4          5   4-FEB-2002 13:12
[SAVE]LAST.DAT;1         1   18-JAN-2002 14:11
[SAVE]WORK.DAT;3        33   1-JAN-2002 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 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.) HP recommends that you remove the /BUFFER\_COUNT qualifier from command procedures.

## BACKUP /BY\_OWNER

---

### /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 to 37776 representing the group number or an alphanumeric group name

**m** An octal number in the range 0 to 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.

#### Examples

1. \$ 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.

2. \$ BACKUP [SUNDANCE]/BY\_OWNER=[727,46] DBA1:STABLE.BCK/SAVE\_SET

In this example, all files in the directory [SUNDANCE] with an owner UIC of [727,46] are saved to the sequential-disk save set STABLE.BCK on DBA1.

---

## **/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 options are available:

DEFAULT	Sets the owner UIC to the user's current default UIC. This option is the default if you do not specify the /BY_OWNER qualifier, 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 you specify the /BY_OWNER qualifier with no option. 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.

## 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=uc

#### 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 to 37776 representing the group number or alphanumeric group name

**m** An octal number in the range 0 to 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, you must enclose it in quotation marks ( " "). A DCL command can contain a maximum of 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, 2002"
$ BACKUP/LIST DMA1:20JULREM.BCK/SAVE_SET
Listing of save set

Save set:          20JULREM.BCK
Written by:        WALRUS
UIC:               [360,054]
Date:              20-JUL-2002 15:22:06.62
Command:           BACKUP [REMARKS] DMA1:20JULREM.BCK/SAVE_SET/COMMENT=Remote
operations for July 20, 2002

Operating system:  OpenVMS Alpha Version V7.3-1

BACKUP version:   V7.3-1
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  20-JUL-2002 14:13
[REMARKS]COM.LIS;1         1  20-JUL-2002 14:04
[REMARKS]DTOP.DIR;1       1  20-JUL-2002 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.

## BACKUP /COMPARE

---

### /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  
/IMAGE/COMPARE save-set-spec device-spec  
/PHYSICAL/COMPARE device-spec device-spec  
/PHYSICAL/COMPARE save-set-spec device-spec
```

#### 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.



## **Examples**

1. \$ BACKUP/COMPARE JAZZ.DAT BLUES.DAT

This example compares two Files-11 files. Because no version number is specified, BACKUP compares all versions of each file.

2. \$ BACKUP/COMPARE/IMAGE MTA0:SWING.BCK DBA2:

This example compares an image save set stored on magnetic tape and a Files-11 volume.

## BACKUP /CONFIRM

---

### /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 you must use the output save-set qualifier /SAVE\_SET when creating a save set on disk.

---

**/CONVERT**

**Input File-Selection Qualifier**

Converts ODS-5 file names to ODS-2 file names. To preserve the output volume as ODS-2, you must also use the /NOINIT qualifier.

Be aware that all ODS-5 file attributes are lost if you convert from an ODS-5 file name to an ODS-2 file name.

**Format**

input-specifier/CONVERT output-specifier

**Example**

```
$ BACKUP/LOG/CONVERT/IMAGE DKA500: DKA200:[000000]IMAGE.BCK/SAVE
```

The command in this example creates an ODS-2 image save set from an ODS-5 disk. The save set can be read by a system running a version of OpenVMS prior to Version 7.2.

## **BACKUP /CRC**

---

### **/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

#### **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.

#### **Example**

```
$ BACKUP MTA2:988SAVE.BCK/NOCRC []
```

This command restores the save set 988SAVE.BCK to the current default directory, indicated by ([]); the input save-set qualifier /NOCRC disables CRC.

---

**/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.

**Example**

```
$ BACKUP/RECORD [ ]/SINCE=BACKUP MTA2:988SAVE.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 988SAVE.BCK; the output save-set qualifier /NOCRC disables cyclic redundancy checking.

## **BACKUP /CREATED**

---

### **/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 with /BEFORE or /SINCE determine which files should be processed.

You cannot use /CREATED with the /BACKUP, /MODIFIED, or /EXPIRED qualifiers.

#### **Example**

```
$ BACKUP *.SDML/SINCE=YESTERDAY/CREATED DLA2:[SAVEDIR]/SAVE_SET
```

The command in this example saves all files with a file type of .SDML created since yesterday (24 hours before midnight last night).

---

**/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 successfully 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 `/PHYSICAL`, `/RECORD` or `/COMPARE` command qualifiers.

**Examples**

1. `$ 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`.

2. `$ BACKUP/VERIFY/DELETE RAY.DAT,JOE.DAT,ELLA.DAT MTA0:OSCAR.BCK/LABEL=FRIEND`

The BACKUP command deletes the selected list of files in this example after saving them to `OSCAR.BCK` on `MTA0` and comparing the output save set with the input files. If BACKUP detects a difference between the contents of the output save set and the input file, the input file is not deleted.

---

**/DENSITY**

**Output Save-Set Qualifier**

Specifies the recording density of the output magnetic tape. Use a value that is supported by the magnetic tape drive.

If you do not specify the /DENSITY qualifier, the default density is the current density of the magnetic tape drive. You must specify the output save-set qualifier /REWIND with /DENSITY.

**Format**

input-specifier output-save-set-spec/DENSITY=keyword

The following table shows shows the densities that are supported for tapes.

<b>Keyword</b>	<b>Meaning</b>
DEFAULT	Default density
800	NRZI 800 bits per inch (BPI)
1600	PE 1600 BPI
6250	GRC 6250 BPI
3480	IBM 3480 HPC 39872 BPI
3490E	IBM 3480 compressed
833	DLT TK50: 833 BPI
TK50	DLT TK50: 833 BPI
TK70	DLT TK70: 1250 BPI
6250	RV80 6250 BPI EQUIVALENT

NOTE: Only the symbols listed above are understood by TMSCP/TUDRIVER code prior to OpenVMS Version 7.2. The remaining values in this table are supported only on Alpha and I64 systems.

TK85	DLT Tx85: 10625 BPI—Cmpt III - Alpha and I64 only
TK86	DLT Tx86: 10626 BPI—Cmpt III - Alpha and I64 only
TK87	DLT Tx87: 62500 BPI—Cmpt III - Alpha and I64 only
TK88	DLT Tx88: (Quantum 4000)—Cmpt IV - Alpha and I64 only
TK89	DLT Tx89: (Quantum 7000)—Cmpt IV - Alpha and I64 only
QIC	All QIC drives are drive-settable only - Alpha and I64 only
8200	Exa-Byte 8200 - Alpha and I64 only
8500	Exa-Byte 8500 - Alpha and I64 only
DDS1	Digital Data Storage 1—2G - Alpha and I64 only
DDS2	Digital Data Storage 2—4G - Alpha and I64 only
DDS3	Digital Data Storage 3—8-10G - Alpha and I64 only
DDS4	Digital Data Storage 4 - Alpha and I64 only
AIT1	Sony Advanced Intelligent Tape 1 - Alpha and I64 only



<b>Keyword</b>	<b>Meaning</b>
AIT2	Sony Advanced Intelligent Tape 2 - Alpha and I64 only
AIT3	Sony Advanced Intelligent Tape 3 - Alpha and I64 only
AIT4	Sony Advanced Intelligent Tape 4 - Alpha and I64 only
DLT8000	DLT 8000 - Alpha and I64 only
8900	Exabyte 8900 - Alpha and I64 only
SDLT	SuperDLT1 - Alpha and I64 only
SDLT320	SuperDLT320 - Alpha and I64 only

Note that tape density keywords cannot be abbreviated.

## Description

The value that 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 bits/in.

## BACKUP /EXACT\_ORDER

---

### /EXACT\_ORDER

#### Output Save-Set Qualifier

Depending on the other qualifiers you specify on the command line, the /EXACT\_ORDER qualifier allows you to perform the following actions:

- Specify the exact order of tape volume labels that you want to use in a BACKUP operation.
- Preserve the existing volume label on a tape.
- Prevent previous volumes of a multivolume save operation from being overwritten.

#### Format

input-specifier output-save-set-spec/EXACT\_ORDER

#### Description

The /EXACT\_ORDER qualifier allows you to perform the following actions:

- Specify the exact order of tape volume labels that you want to use in a BACKUP operation. You must use the /LABEL=(label1,label2,...) qualifier to specify the order of the labels. BACKUP continues the operation as long as the label of the tape in the drive matches the corresponding label on the command line. If you do not specify enough labels on the command line to complete the operation, BACKUP prompts you to enter a label for the tape in the drive.
- Preserve the existing volume label on a tape. If you do not use the /LABEL qualifier on the command line and the tape has an ANSI label, BACKUP uses the existing label.
- Prevent previous volumes of a multivolume save operation from being overwritten. BACKUP keeps track of the volume labels you have already used in the operation. If you accidentally mount one of the previous volumes, BACKUP displays the following error message:

```
%BACKUP-W-MOUNTERR, volume 1 on MKB100: was not mounted
because its label does not match the one requested
Volume with label TAPE1 was already used in this save
operation. Specify option (QUIT or NEW tape)
BACKUP>
```

Note the following restrictions when you use the /EXACT\_ORDER output qualifier:

- If you use the /EXACT\_ORDER qualifier, you cannot specify a label longer than six characters on the command line. If you specify a label longer than six characters, BACKUP displays the following error message:

```
%BACKUP-F-INVQUAVAL, value 'label_name' invalid for
/LABEL qualifier
```

- You cannot use the /IGNORE=LABEL\_PROCESSING qualifier with the /EXACT\_ORDER qualifier.
- If you use the /LABEL qualifier with the /EXACT\_ORDER qualifier, you cannot specify duplicate labels.

The default is /NOEXACT\_ORDER.

## Examples

1. 

```
$ BACKUP/IMAGE/RECORD/VERIFY/NOASSIST
  _From: DKA100:
  _To: MKB100:MAR11.SAV/LABEL=(TAPE1,TAPE2,TAPE3)/EXACT_ORDER
```

This example uses the /EXACT\_ORDER qualifier to specify the exact order of labels for the BACKUP operation. Note that if you specify the /ASSIST qualifier, BACKUP would display messages on the operator terminal.

BACKUP performs the following actions:

- a. Compares the volume label of the tape in MKB100: with the first label that you specified on the command line (TAPE1). If the labels match exactly, BACKUP begins the save operation. If the labels do not match or if the tape does not have an ANSI label, BACKUP displays the following message:

```
%BACKUP-W-MOUNTERR, volume 1 on MKB100: was not mounted
  because its label does not match the one requested
%BACKUP-W-EXLABEER, volume label processing failed
  because volume TAPE4 is out of order, Volume label
  TAPE1 was expected. Specify option (QUIT, NEW tape,
  OVERWRITE tape, USE loaded tape)
BACKUP> OVERWRITE
```

Depending on the option you specify, you can quit the backup operation (QUIT), dismount the old tape and mount a new one (NEW), overwrite the label and the data on the tape (OVERWRITE), or write the data to the tape using the loaded tape's label (USE).

- b. When the operation fills the first tape, it displays the following message:

```
%BACKUP-I-RESUME, resuming operation on volume 2
%BACKUP-I-READYWRITE, mount volume TAPE2 on MKB100:
  for writing. Respond with YES when ready:
```

- c. When you load the second tape and enter YES, BACKUP compares the label of the second tape with the second label you specified on the command line (TAPE2) just as it did in step 1a.
- d. Assuming the volume labels match, BACKUP continues processing until it completes the operation or runs out of volume labels. If you do not specify enough labels on the command line to complete the operation, BACKUP prompts you to enter a label for the tape in the drive as follows:

```
%BACKUP-W-MOUNTERR, volume 4 on MKB100: was not mounted
  because the label was not specified
  Specify EXACT_ORDER label (up to 6 characters)
BACKUP>
```

BACKUP then compares the label on the tape with label you specify as described previously.

## BACKUP /EXACT\_ORDER

```
2. $ BACKUP/IMAGE/RECORD/VERIFY/NOASSIST
   _From: DKA100:[TEST]
   _To: MKB100:MAR11.SAV/EXACT_ORDER
```

Because this example does not use the /LABEL qualifier, BACKUP uses the existing label on the tape. If the tape does not have an ANSI label, and it is the first tape in the operation, BACKUP displays the following error message:

```
%BACKUP-F-NOTANSI, tape is not valid ANSI format
```

If the tape does not have an ANSI label, and is not the first tape in the operation, BACKUP displays the following error message prompting you to specify a label:

```
%BACKUP-W-MOUNTERR, volume 2 on MKB100: was not mounted
because the label was not specified
Specify EXACT_ORDER label (up to 6 characters)
BACKUP>
```

---

### Note

---

BACKUP checks to make sure you specify a valid label. If the label is not valid (for example, longer than six characters), BACKUP displays an error message. In previous versions of the OpenVMS operating system, BACKUP truncated long volume labels.

---

---

**/EXCLUDE****Input File-Selection Qualifier**

Excludes files that otherwise meet the selection criteria for a save or copy 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 defining the files to be excluded. 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.

You cannot use the /EXCLUDE qualifier in image 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.

## **BACKUP /EXPIRED**

---

### **/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.

When you use the `/FAST` qualifier to save a disk, ALIAS directory trees are not processed. Only the primary files that the ALIAS points to are saved. Depending on the number of ALIAS directory specifications there are on the disk, this may increase performance by reducing the number of files BACKUP checks for processing. A message is displayed for each ALIAS directory or file that is not processed.

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.

## BACKUP /FILES\_SELECTED

---

### /FILES\_SELECTED

#### Input File-Selection Qualifier

Specifies a file that contains a list of the files that will be selected when a save set is restored.

#### Format

input-specifier /FILES\_SELECTED=file-spec output-specifier

#### Description

The /FILES\_SELECTED qualifier allows you to specify a file that contains a list of the files that are to be selected when a save set is restored. You can use this qualifier in place of the /SELECT qualifier to select files to restore from a save set.

Do not use a device specification when you list the files to be selected. In the list of files, enter one OpenVMS file specification per line. You can use most standard wildcard characters, but you cannot use wildcard characters denoting the latest version of files (;) and relative versions of files (;-n).

#### Example

```
$ BACKUP INFO.BCK/SAVE_SET/FILES_SELECTED=RFILE.DAT [ ]
```

The command in this example selects the files in RFILE.DAT and restores them to the current default directory. The RFILE.DAT file contains the following entries:

```
[ INFO]RESTORE.COM  
[ PAYROLL]BADGE.DAT  
EMPLOYEE.DAT
```



---

**/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)

Save set:          ROCK.BCK
Written by:        RINGO
UIC:               [000200,000300]
Date:              20-AUG-2002 15:39:38.89
Command:           BACKUP [.STONES] MTA0:ROCK.BCK/LABEL=BACKUP

Operating system:  OpenVMS Alpha Version V7.3-1

BACKUP version:   V7.3-1
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-2002 14:10
      Owner: [000200,000200] Revised: 18-AUG-2002 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-2002 14:11
      Owner: [000200,000200] Revised: 18-AUG-2002 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
```

## **BACKUP /FULL**

```
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 0 to 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.

## BACKUP /HEADER\_ONLY

---

### /HEADER\_ONLY

#### Input File-Selection Qualifier

Specifies that only the file headers of a file are to be saved in a BACKUP operation.

#### Format

input-specifier /HEADER\_ONLY=option output-specifier

#### Description

The /HEADER\_ONLY qualifier specifies that the Backup utility is to save only the file header of a shelved or a preshelved file in a BACKUP operation.

When a file is shelved, the data in the file is shelved, but the file header is retained. Users shelve files to save disk space. (In addition, users might *preselve* files to save time by performing shelving operations ahead of time.)

In a BACKUP save operation, the default behavior is to unshelve files before backing them up. This brings back the file data online, so that, when the BACKUP operation is performed, the entire file is backed up (not just the file header). The only exception to the BACKUP default behavior is in operations that use the /PHYSICAL or /IMAGE qualifier. For those operations, the file remains in the file shelved state.

For more information about file shelving and preshelving, refer to Hierarchical Storage Management (HSM) documentation.

Use the following options with the /HEADER\_ONLY qualifier:

Option	Description
SHELVED	Saves only the file header of a shelved file.
NOSHELVED	Saves both the file header and the file data of a shelved file. (This causes the file to be unshelved.)
PRESHELVED	Saves only the file header of a preshelved file.
NOPRESHELVE	Saves both the file header and the file data of a preshelved file.

#### Examples

1. `$ BACKUP [INFO]/HEADER_ONLY=(SHELVED) MKA600:INFO.BCK/SAVE_SET`

The command in this example saves all files in the directory [INFO] to a tape drive save set named INFO.BCK. The shelved files in [INFO] will not be unshelved. Only their file headers will be saved to save set INFO.BCK because the /HEADER\_ONLY=(SHELVED) qualifier is specified.

2. \$ BACKUP [INFO]/HEADER\_ONLY=(SHELVED,PRESHELVED)  
MKA600:INFO.BCK/SAVE\_SET

This command saves all files in the directory [INFO] to a tape drive save set named INFO.BCK. The files saved from [INFO] will not be unshelved because the HEADER\_ONLY=(SHELVED,PRESHELVED) qualifier is specified. The save set INFO.BCK will contain only the file headers of files that are shelved or preshelved.

3. \$ BACKUP/IMAGE DUA0: MKA600:INFO.BCK/SAVE\_SET

The command in this example saves all files on the disk DKA0:. Because the /IMAGE qualifier is specified, only the file headers of files that are shelved or preshelved are saved to INFO.BCK.

4. \$ BACKUP [INFO] MKA600:INFO.BCK/SAVE\_SET

The command in this example saves all files in the directory [INFO] to a tape drive save set named INFO.BCK. The files saved from [INFO] will be unshelved (the default). The save set INFO.BCK will contain both the file header and the data of files that are shelved or preshelved.

## BACKUP /IGNORE=option

---

### /IGNORE=option

#### Command Qualifier

Specifies that a BACKUP save or copy operation will override restrictions placed on files or will not perform tape label processing checks.

---

#### Note

---

File system interlocks are expressly designed to prevent data corruptions, and to allow applications to detect and report data access conflicts.

Use of the INTERLOCK keyword overrides these file data integrity interlocks. The data that BACKUP subsequently transfers can then contain corrupted data for open files. Also, all cases in which these data corruptions can occur in the data that BACKUP transfers are not reliably reported to you; in other words, silent data corruptions are possible within the transferred data.

---

#### Format

/IGNORE=option input-specifier output-specifier

#### Description

The /IGNORE= qualifier has the following options:

ACCESSIBILITY	Processes files on a tape that is protected by a volume accessibility character, or on a tape created by HSC Backup. The option applies only to tapes. It affects the first tape mounted and all subsequent tapes in the save set.
INTERLOCK	Processes files that otherwise cannot be processed due to file access conflicts. Use this option to save or copy files currently open for writing. 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 following 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. See the Note before this table for more information about this keyword.

<b>LABEL_</b> <b>PROCESSING</b>	Saves or copies the contents of files to the specified magnetic tape volume regardless of the information contained in the volume header record. BACKUP does not verify the volume label or expiration date before writing information to the tape volume. Note that you cannot use this option with the /EXACT_ORDER qualifier.
<b>NOBACKUP</b>	Saves or copies 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.

## Examples

1. \$ 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.

2. \$ 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.

3. \$ INITIALIZE/LABEL=VOLUME\_ACCESSIBILITY:"K" MUA1: 29JUN  
    \$ BACKUP/IGNORE=(ACCESSIBILITY)  
    From: DUA0:[BOOKS...]  
    To: MUA1:BACKUP.SAV /LABEL=29JUN

The INITIALIZE command in this example initializes the tape with an accessibility character (K) and a volume label (29JUN). The BACKUP command mounts the tape, regardless of the accessibility, and performs the BACKUP operation. For more information about tape protection, refer to the *HP OpenVMS System Manager's Manual*.

4. \$ BACKUP/LOG/IMAGE/CONVERT DKA500:[000000]IMAGE.BCK/SAVE DKA200:/NOINIT  
    %BACKUP-I-ODS5CONV, structure level 5 files will be converted to structure level 2 on DKA200:  
    -BACKUP-I-ODS5LOSS, conversion may result in loss of structure level 5 file attributes  
    %BACKUP-S-CREATED, created DKA200:[000000]000000.DIR;1  
    %BACKUP-S-CREATED, created DKA200:[000000]BACKUP.SYS;1  
    %BACKUP-S-CREATED, created DKA200:[000000]CONTIN.SYS;1  
    %BACKUP-S-CREATED, created DKA200:[000000]CORIMG.SYS;1  
    %BACKUP-S-CREATED, created DKA200:[000000]SECURITY.SYS;1  
    %BACKUP-S-CREATED, created MDA2:[000000]TEST FILES.DIR;1  
    %BACKUP-S-CREATEDAS, created DKA200:[TEST\_FILES]SUB^\_{DIR^}.DIR;1 as DKA200:[TEST\_FILES]SUB\$DIR\$.DIR;1

You can use commands like the ones in the example if you have an image backup of an ODS-5 disk, and you want to restore it to an ODS-2 disk.

**BACKUP**  
**/IGNORE=option**

In the command line in the example, IMAGE.BCK is the ODS-5 save set, and DKA200: is the ODS-2 disk. When you use this conversion method, you must preinitialize the output disk to ODS-2 and then include the /NOINIT qualifier in your command line.



---

**/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.

You can receive a fatal error if you use /IMAGE with the /PHYSICAL qualifier.

When you use the /IMAGE qualifier to save to a disk, alias directory trees are not processed.

---

**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). By default, a BACKUP image operation saves or copies the attributes but not the contents of files flagged as NOBACKUP.

Also by default, BACKUP does not save the attributes nor the contents of files 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.

## 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. A BACKUP operation to mixed tape and disk save sets, as shown in the following command, is unsupported:

```
$ BACKUP SYS$DISK:/IMAGE dka0:FUN,MKA0:/SAVE/REW
```

### Examples

```
1. $ 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.

```
2. $ BACKUP/IMAGE DBA2: MTA0:ET.BCK,MTA1:
```

This command saves an entire disk volume to a multivolume save set named ET.BCK using two magnetic tape drives.

```
3. $ MOUNT/FOREIGN DBA1:
%MOUNT-I-MOUNTED, mounted on NODE$DBA1:
$ BACKUP/IMAGE WORKDISK DBA1:28SEP.BCK/SAVE_SET
```

The MOUNT command prepares the target disk for the image save operation. The BACKUP command performs an image save operation to a Files-11 save set named 28SEP.BCK.

---

## **/INCREMENTAL**

### **Command Qualifier**

Allows you to restore an incremental save set.

---

#### **Note**

`/INCREMENTAL` is valid only in restore operations. It is not related to the `/NOINCREMENTAL` qualifier, which is valid only in `BACKUP` save operations.

---

### **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. (HP 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 volume using the latest (most recent) image backup save set. (The saveset must have been created using the `/IMAGE` and `/RECORD BACKUP` command qualifiers.)
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, specifying `/BY_OWNER` is unnecessary unless you want to change the original UICs. The `/INCREMENTAL` qualifier can be used only on Files-11 Structure Level 2 or 5 volumes.

You can receive a fatal error if you use the `/PHYSICAL` qualifier with `/INCREMENTAL`.

## BACKUP /INCREMENTAL

### 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 one to create a new copy of the public volume. First, restore the volume from the latest full backup with an image restore operation.

The section “Formulating a Backup Strategy” in the BACKUP chapter of the *HP OpenVMS System Manager’s Manual* discusses the importance of using the /IMAGE and /RECORD qualifiers the first time you back up a disk, before you perform incremental backups.

```
$ MOUNT/FOREIGN DRA0:
%MOUNT-I-MOUNTED, mounted on DRA0:
$ BACKUP/IMAGE/RECORD MTA0:FULLJUN02,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 or tape volume, making its entire previous contents unavailable. (/REWIND performs the same function for output tapes.)

**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.

The BACKUP/IMAGE/INITIALIZE command sizes the storage bitmap to correspond to the entire physical volume. Beginning with OpenVMS Version 7.2, the file system also correctly handles a volume whose storage bitmap is smaller than required. The space on the volume available for allocation is the space the bitmap describes; as a result, if the bitmap is smaller than the volume requires, not all the volume is available for file allocation. A SHOW DEVICE /FULL command continues to display the actual physical volume size; however, the free blocks displayed are the number of blocks actually available for allocation.

**Examples**

1. \$ 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.

## **BACKUP /INITIALIZE**

2. `$ BACKUP/IMAGE/INITIALIZE DBA2:OLDFILES.BCK/SAVE_SET DBA6:`

This command directs BACKUP to initialize the output volume DBA6 using volume initialization parameters in the save-volume summary record on DBA2. The image save set OLDFILES.BCK is then restored to DBA6.

---

## **/INPUT\_FILES**

### **Input Save-Set Qualifier**

Directs BACKUP to treat the input-specifier as the file name of a list of files. This file specifies the input files for a BACKUP operation.

### **Format**

input-specifier /INPUT\_FILES output-specifier/SAVE\_SET

### **Description**

The /INPUT\_FILES qualifier allows you to specify a list of files to be processed for input. The input-specifier is the name of a file that contains one standard OpenVMS file specification per line.

### **Example**

```
$ BACKUP FILE.DAT/INPUT_FILES MKA600:INFO.BCK/SAVE_SET
```

The command in this example backs up the files listed in the FILE.DAT file to a tape drive save set named INFO.BCK. If the disk, directory, or file extension is not specified, the defaults are copied from the previous entry or from the default if this is the first entry. The FILE.DAT file contains the following entries:

```
$1$DKA0:[INFO]*.COM  
INFO.TEXT  
[PAYROLL]*.DAT
```

## BACKUP /INTERCHANGE

---

### /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.

The /INTERCHANGE qualifier implies /CONVERT when the input is an ODS-5 disk or file.

#### 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 8192 bytes.
- Normal error recovery is used to write magnetic tapes so that no bad records exist on the resulting magnetic tape.

#### 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`). You can receive a fatal error if you use the `/PHYSICAL` qualifier with `/JOURNAL`.

**Examples**

1. `$ 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.

## BACKUP /JOURNAL

```
2. $ BACKUP/LIST/JOURNAL=ARCH.BJL/SELECT=[SMITH.PROGS]/SINCE=5-OCT-2002
Listing of BACKUP journal
Journal file_DB1:[SYSMGR]:ARCH.BJL;1 ON 7-OCT-2002 00:45:43.01
Save set WKLY.BCK, created on 6-OCT-2002 00:01:34.54
Volume number 1, volume label WKL101

      [SMITH.PROGS]REMINDER.FOR;46
      [SMITH.PROGS]RUNTHIS.FOR;4
      [SMITH.PROGS]TIMER.PAS;5

.
.
.
```

This example displays all files in the directory [SMITH.PROGS] that were saved after October 5, 2002, and listed in the BACKUP journal file ARCH.BJL.

```
3. $ BACKUP/JOURNAL/LOG/IMAGE DRA2: MTA0:3OCT.FUL
%BACKUP-S-COPIED, copied DRA2:[COLLINS]ALPHA.DAT;4
%BACKUP-S-COPIED, copied DRA2:[COLLINS]EDTINI.EDT;5
.
.
.
%BACKUP-I-RESUME, resuming operation on volume 2
%BACKUP-I-READYWRITE, mount volume 2 on _MTA0: for writing
Press return when ready: 
%BACKUP-S-COPIED, copied DRA2:[LANE]MAIL.MAI;1
%BACKUP-S-COPIED, copied DRA2:[LANE]MEMO.RNO;5
.
.
.
$ BACKUP/JOURNAL/LIST
Listing of BACKUP journal
Journal file_DB2:[SYSMGR]BACKUP.BJL;1 on 3-OCT-2002 00:40:56.36
Save set 3OCT.FUL created on 3-OCT-2002 00:40:56.36
Volume number 1, volume label 3OCT01

      [COLLINS]ALPHA.DAT;4
      [COLLINS]EDTINI.EDT;5
      [COLLINS]LOGIN.COM;46
      [COLLINS]LOGIN.COM;45
      [COLLINS]MAIL.MAI;1
      [COLLINS.MAR]GETJPI.EXE;9
      [COLLINS.MAR]GETJPI.LIS;14
      .
      .
      .
      [LANE]LES.MAI;1
      .
      .
      .
Save set 3OCT.FUL created on 3-OCT-2002 00:40:56.36
Volume number 2, volume label 3OCT02

      [LANE]MAIL.MAI;1
      [LANE]MEMO.RNO;5
      [LANE]MEMO.RNO;4
      .
      .
      .
      [WALTERS.VI]KD.RNO;52
```

End of BACKUP journal

This example shows how to create a BACKUP journal file and list the contents of the BACKUP journal file.

# BACKUP /LABEL

---

## /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 has fewer than six characters, BACKUP pads the volume label with the blank character to six characters.

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 the sixth character of the volume label is in the range 0 to 9, 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

Label Specified in the Command Line	Matching Volume Labels
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 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. You can also use the /EXACT\_ORDER qualifier to specify the exact order of tape volume labels that you want to use in a BACKUP operation.

## Examples

1. \$ 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.

2. \$ BACKUP DDA1: MTA0:PLAYS.BCK,MTA1,MTA2/REWIND/LABEL=(ACT1,ACT2,ACT3)

This example assumes that the three tapes have no volume labels. This command saves all files on the disk named DDA1 to the save set PLAYS.BCK. The first tape in the save set is labeled ACT1, the second is labeled ACT2, and the third is labeled ACT3.

# BACKUP /LIST

---

## /LIST

### Command Qualifier

Lists information about a BACKUP save set and about the files in a save set. You can display the list on your terminal or write it 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 the list information can be written.

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.

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]23MAR02.BCK/SAVE_SET
Listing of save set(s)
Save set:          23MAR02.BCK
Written by:       MOROCI
UIC:              [000200,000200]
Date:             23-MAR-2002 14:18:16.00
Command:          BACKUP [SAVE] DBA2:[SAVE]23MAR00.BCK/SAVE_SET
Operating system: OpenVMS Alpha Version V7.3-1
BACKUP version:   V7.3-1
CPU ID register:  08000000
Node name:        _SUZI::
Written on:       _DBA2:
Block size:       32,256
Group size:       10
Buffer count:     3
```

## BACKUP /LIST

[SAVE]LAST.DAT;1	1	18-JAN-2002	14:11
[SAVE]INFO.TXT;4	5	4-FEB-2002	13:12
[SAVE]WORK.DAT;3	33	1-JAN-2002	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.

## BACKUP /LOG

---

### /LOG

#### Command Qualifier

Determines whether the file specification of each file processed is displayed on SYS\$OUTPUT during the operation. The default is /NOLOG.

#### Format

/[NO]LOG input-specifier output-specifier

#### Example

```
$ BACKUP/LOG [SAVE]23MAR02.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.



## **/MEDIA\_FORMAT=[NO]COMPACTION**

### **Output Save-Set Qualifier**

Controls whether data records are automatically compacted and blocked together. Data compaction and record blocking increase the amount of data that can be stored on a single tape cartridge.

The compaction ratio depends on the data and the tape drive you use. For more information, refer to the documentation supplied with your tape drive.

BACKUP allows you to specify different compaction settings on different save sets on a tape. However, not all tape drives support the use of more than one compaction setting on a tape. Whether mixed mode tapes are permitted depends on the model of the tape drive you use.

### **Format**

input-specifier output-save-set-spec /MEDIA\_FORMAT=[NO]COMPACTION

### **Description**

The /MEDIA\_FORMAT qualifier can only be used with tape drives that support data compaction.

On Alpha and I64 systems, you can use the /MEDIA\_FORMAT=COMPACTION qualifier for hardware data compaction of SCSI tape drives.

### **Example**

```
$ BACKUP WORK$:[TESTFILES...]*.*;* MUA0:TEST.SAV -  
_$/MEDIA_FORMAT=COMPACTION /REWIND
```

This command saves all files in the directory [TESTFILES] and its subdirectories in a save set named TEST.SAV using a TA90E tape drive. The /MEDIA\_FORMAT=COMPACTION qualifier specifies that the tape drive automatically compacts and blocks together data records on the tape.

## **BACKUP /MODIFIED**

---

### **/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 of the input file-selection qualifiers /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 equal to 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 **/VERIFY** qualifier will cause unpredictable results. HP recommends that you do not use the **/NEW\_VERSION** qualifier with the **/VERIFY** qualifier.

### **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.

## BACKUP /NOINCREMENTAL

---

### /NOINCREMENTAL

#### Command Qualifier

Beginning with OpenVMS Version 7.2, on a save operation, /NOINCREMENTAL allows you to control the amount of file data that is saved. Use this qualifier only if you are sure that you want to save specific files and do not want to save all data.

In recent versions of OpenVMS, the /SINCE=BACKUP incremental save operation has been refined so that files that are saved are accurate and not redundant. As a result, the /NOINCREMENTAL and /SINCE=BACKUP qualifiers are not allowed together. This ensures an accurate /INCREMENTAL restore.

---

#### Note

---

/NOINCREMENTAL is valid only in BACKUP save operations. It is not related to the /INCREMENTAL qualifier, which is valid only in restore operations.

---

#### Format

/NOINCREMENTAL input-specifier output-specifier

#### Description

In OpenVMS Version 6.2 and prior versions, the system, by default, did not save files and subdirectories that were under directories that had been modified. In OpenVMS Versions 7.0 and 7.1, to ensure a successful restore, the system saved all files and subdirectories under directories that had been modified. This behavior, however, sometimes resulted in saving files and subdirectories that were not needed for later restore operations.

#### Example

```
$ BACKUP/ FAST/ NOINCREMENTAL /SINCE="3-MAY-2002" -  
_ $ MAC_DISK:[000000...]*.*;* -  
_ $ TAPE:MCDSK000503.BCK/ SAVE/ REWIND
```

The command in this example executes an incremental save BACKUP operation for an input volume; the command avoids saving all files under recently modified directories.

---

## **/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.

**BACKUP  
/OWNER\_UIC**

---

**/OWNER\_UIC**

The /OWNER\_UIC qualifier has been superseded by /BY\_OWNER. HP 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 BACKUP is to ignore any volume structure on the input device and is to process the volume in terms of physical blocks. If you write a save set with the BACKUP/PHYSICAL command, you must also restore it with the BACKUP/PHYSICAL command.

**Format**

/PHYSICAL input-specifier output-specifier

**Description**

For physical copy operations between disks, the output device must be either the same size or a larger-capacity disk.

If the output device is larger than the input device, only disk blocks less than the size of the input device are written to the output device. Depending on the volume structure of the input device, the extra uninitialized blocks at the end of the output device might create an unusable disk volume.

If the input device contains a FILES-11 ODS-2 or ODS-5 volume, you can expand the volume size on the output device after the restore using the SET VOLUME/LIMIT/SIZE DCL command.

For all physical operations, the output disk cannot have a bad block in any location that corresponds to a good block on the input disk. (This restriction does not apply to RA or more recent disk architectures.)

---

**Note**

---

BACKUP/PHYSICAL does not copy the first track (track 0) of RX01 and RX02 diskettes; HP does not support track 0.

---

**Examples**

1. \$ MOUNT/FOREIGN DYA0:  
\$ MOUNT/FOREIGN DYA1:  
\$ BACKUP/PHYSICAL DYA0: DYA1:

This example mounts RX02 diskettes in DYA0 and DYA1 as foreign devices and copies the contents of the diskette mounted in DYA0 to the diskette mounted in DYA1.

2. \$ MOUNT/FOREIGN DBA1:  
\$ BACKUP/PHYSICAL MTA0:28SEP.BCK DBA1:

This command restores a physical save set named 28SEP.BCK to DBA1.

## BACKUP /PROTECTION

---

### /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.)

#### 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). For more information about specifying protection codes, refer to the *OpenVMS User's Manual*.

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. If you specify /PROTECTION, any protection categories that you do not specify default to your default process protection.

To initialize a magnetic tape with the correct protection, specify the output save-set qualifier /REWIND with the /PROTECTION qualifier. If you do not specify /REWIND with /PROTECTION, the protection information, if any, in the volume header record is not changed. However, specifying /PROTECTION without /REWIND ensures that continuation volumes receive the correct protection.

If the save set is written to magnetic tape and /PROTECTION is not specified, BACKUP applies **no** protection to the tape.

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.

#### Examples

1. 

```
$ 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



## BACKUP /PROTECTION

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; and WORLD users are assigned no access.

```
2. $ BACKUP/IMAGE
   _From: DUA0:
   _To: MFA2:DAILY.BCK/REWIND/LABEL=TAPE1-
   _$ /PROTECTION=(S:RWED,O:RWED,G,W)
   $ BACKUP/IMAGE DUA2: MFA2:DAILY2.BCK/PROTECTION=(S:RWED,O:RWED,G,W)
   %BACKUP-I-RESUME, resuming operation on volume 2
   %BACKUP-I-READYWRITE, mount volume 2 on _MFA2: for writing
   Press return when ready: 
```

This first BACKUP command creates an image backup of the disk DUA0 in a save set named DAILY.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 /PROTECTION qualifier assigns the owner of the magnetic tape and SYSTEM users read, write, execute, and delete access; GROUP and WORLD users are assigned no access.

The second BACKUP command uses the same tape for an image backup of the disk DUA2. When the tape is full, BACKUP requests another volume. Because the /PROTECTION qualifier was specified with second BACKUP command, the continuation volume receives the desired protection.

## BACKUP /RECORD

---

### /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 in save or copy operations on Files-11 Structure Level 2 or 5 volumes. To use the /RECORD qualifier on files, the user privilege SYSPRV is required.

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 makes 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 qualifiers /DELETE, /COMPARE, or /PHYSICAL.

#### Example

```
$ BACKUP/RECORD DBA1:[000000...]/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.

---

## **/RELEASE\_TAPE**

### **Command Qualifier**

Dismounts and unloads a tape after a BACKUP save operation writes a save set to the tape (and optionally verifies the saveset information on that tape).

### **Format**

/RELEASE\_TAPE input-specifier output-specifier

### **Description**

By using the /RELEASE\_TAPE qualifier in conjunction with either the /DELETE or /RECORD qualifiers, you can make a tape drive available for other operations before the BACKUP command completes. You can also use the /RELEASE\_TAPE qualifier without the /DELETE or /RECORD qualifiers, in which case the /RELEASE\_TAPE qualifier dismounts and unloads the tape in the drive after the BACKUP command completes.

You cannot use the /RECORD and /DELETE qualifiers in the same BACKUP command.

### **Examples**

1. `$ BACKUP/IMAGE/RECORD/RELEASE_TAPE DUA1: MUA0:BACK.BCK`

The command in this example backs up the disk DUA1 to the save set BACK.BCK. By using the /RELEASE\_TAPE and /RECORD qualifiers, BACKUP dismounts and unloads the tape in MUA0 (making it available for other operations) before it performs the action of the /RECORD qualifier.

2. `$ ALLOCATE MUA0: TAPE  
$ BACKUP/DELETE/RELEASE_TAPE/LOG DUA1:[MAIN...] MUA0:MAIN.BCK  
.  
.  
.  
$ DEALLOCATE TAPE`

The commands in this example back up some directories on a disk named DUA1, and then delete the files that have been backed up. The /RELEASE\_TAPE qualifier dismounts and unloads the tape (making it available for other operations) before the /DELETE qualifier performs its action. The tape remains allocated until you enter the DEALLOCATE command.

## BACKUP /REPLACE

---

### /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 specifiers, BACKUP performs the following tasks:

- 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. The output file qualifier /REPLACE instructs BACKUP to first create a new version of an input file if the output medium has a file 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.

The /REWIND qualifier 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.

The /NOREWIND qualifier 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.

**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 fewer 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 in the range 0 to 9, 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:

<b>Label Specified in the Command Line</b>	<b>Matching Volume Labels</b>
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.

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 /REWIND

```
%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 /NOWIND 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 /NOWIND 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 /NOWIND. You must specify /REWIND to rewind and initialize a magnetic tape volume.

### Example

```
$ BACKUP  
_From: *.PS  
_To:  
MTA0:DSRSAVE.BCK/REWIND/LABEL=DSR01/TAPE_EXPIRATION=29-JUN-2002
```

The command in this example initializes a new magnetic tape and writes the volume label DSR01 and a tape expiration date of June 29, 2002, to the tape's volume header record. Then this command saves all files in the current default directory with a file type of .PS 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.

### **Examples**

1. \$ BACKUP DBA2:[BACKUP]1212MAR3.BCK/SAVE\_SET DBA1:[\*...]

This command restores a save set named 1212MAR3.BCK from DBA2 to DBA1.

2. \$ BACKUP/LIST DBA2:[SAVE]23MAR02.BCK/SAVE\_SET

This command lists the BACKUP summary information and the file name, size, and creation date for each file in the save set named 23MAR02.BCK. The /SAVE\_SET qualifier is required to identify the input specifier as a save set on a disk.

3. \$ BACKUP/LOG DBA2:[SAVE]23MAR02.BCK/SAVE\_SET DBA3:[PLI.WORK]

This command restores the directory that was listed in Example 2. File specifications are logged to SYS\$OUTPUT as the files are restored.

## BACKUP /SAVE\_SET

---

### /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.

#### 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.

#### Examples

1. \$ 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.

2. \$ BACKUP DBA2:[PLI.WORK]\*.\*; [SAVE]23MAR02.BCK/SAVE\_SET

This command saves the highest numbered version of each file in directory [PLI.WORK] in a save set named 23MAR02.BCK on the same disk.

3. \$ BACKUP  
\_From: []  
\_To: MILO"FRANKIE THISISMINE":DUA0:[FRANKIE]MYDIR.BCK/SAVE\_SET

This command saves all files in the current default directory to a network save set named MYDIR.BCK on node MILO.

4. \$ MOUNT/FOREIGN DBA0:  
\$ BACKUP [SIMS] DBA0:SIMS.BCK/SAVE\_SET

This command saves all files in the directory [SIMS] to a sequential-disk save set named SIMS.BCK.

---

**/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 . . . ]\*.\*;.\*.

**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.

## BACKUP /SINCE

---

### /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.

Qualifier	Function
/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 Levels 2 and 5 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

Be sure to perform an image backup, using the BACKUP/IMAGE/RECORD command, before performing regular incremental backups. The image backup saves a copy of the entire disk and marks each file as having been saved. Regularly performed subsequent incremental backups assume an image backup was already performed and therefore will save new or modified files. If an image backup was not performed first, the incremental backups will save more files than may be necessary, in an attempt to ensure that an incremental restore will be successful.

## **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. Even though it is used in this example, the /MODIFIED qualifier is not required because its action is the default when the /SINCE qualifier is specified.

## BACKUP /TAPE\_EXPIRATION

---

### /TAPE\_EXPIRATION

#### Output Save-Set Qualifier

Writes a file expiration date other than the current date to the file header label of the save set.

#### 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 expiration date of the *first* file on the tape has expired before initializing the tape. Initializing the tape removes access to data previously stored on the tape.

HP 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 3-letter abbreviation of the month.

yyyy is the year.

BACKUP writes the expiration date into the file's HDR1 ANSI label on the tape. If you do not use the `/TAPE_EXPIRATION` qualifier, BACKUP uses today's date as the expiration date.

#### Example

```
$ BACKUP DBA1:  
_To: MTA0:13SEPBAK.BCK/REWIND/TAPE_EXPIRATION=20-SEP-2002/LABEL=SEPW02
```

In this example, the save set file `13SEPBAK.BCK` receives an expiration date of September 20, 2002. Because this command includes the `/REWIND` qualifier, `13NOVBAK.BCK` is the first file on the tape and its expiration date indicates that the tape expires after seven days.

---

**/TRUNCATE****Command Qualifier**

Controls whether a copy or restore operation truncates a sequential output file at the end-of-file (EOF) when creating 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 performs the following tasks:

- 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.

## BACKUP /VERIFY

---

### /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 you perform 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.

You can restore any image save set with the /VOLUME qualifier. 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.

## BACKUP

### 7.6 BACKUP Examples

---

## 7.6 BACKUP Examples

Table 7–3 shows BACKUP command formats for save operations and some of the qualifiers you can use with a save operation.

**Table 7–3 Save Operation Quick Reference**

Command Action	Command Format and Example
Saves a file to a save set on magnetic tape	BACKUP file-spec save-set-specifier/LABEL=label \$ BACKUP STRATDAT1.DAT MTA0:STRATDAT1.BCK/LABEL=TAPE01
Saves the most recent versions of files in a directory to magnetic tape	BACKUP [directory]*.*; save-set-specifier/LABEL=label \$ BACKUP [LYKINS...]*.*; MTA0:1409MAR17.BCK/LABEL=WKY102
Saves a disk volume to a save set on magnetic tape	BACKUP/IMAGE ddcu: save-set-specifier/LABEL=label \$ BACKUP/IMAGE DBA1: MTA0:000FEB4.BCK/LABEL=MTH101
Saves a disk volume to a multivolume save set on more than one magnetic tape drive	BACKUP/IMAGE ddcu: save-set-specifier,ddcu: . . . /LABEL=(label1, . . . ) \$ BACKUP/IMAGE DBA1: MTA0:17MAR.BCK,MTA1:/ - _ \$ LABEL=(WKY101,WKY102)
Saves a list of files to a save set on magnetic tape	BACKUP file-spec,file-spec,... save-set-specifier/LABEL=label \$ BACKUP DBA1:[LYKINS...]*.PAS,DMA0:[DAKOTA...]*.PAS - _ \$ MTA0:PAS17MAR.BCK/LABEL=TAPE01
Saves a disk volume for incremental backups for the first time	BACKUP/RECORD/IMAGE/LOG ddcu: save-set-specifier/LABEL=label \$ BACKUP/RECORD/IMAGE/LOG DBA1: MTA0:985FEB4.BCK/ - _ \$ LABEL=DLY101
Saves a disk volume for incremental backups (not the first time)	BACKUP/RECORD/FAST/LOG ddcu:[*...]/SINCE=BACKUP save-set-specifier/LABEL=label \$ BACKUP/RECORD/FAST/LOG DBA1:[*...]/SINCE=BACKUP - _ \$ MTA0:988FEB4.BCK/LABEL=DLY101
Saves an unstructured disk volume	BACKUP/PHYSICAL ddcu: save-set-specifier/LABEL=label \$ BACKUP/PHYSICAL DMA1: MTA0:985FEB4.BCK/LABEL=MTH101

(continued on next page)

**Table 7–3 (Cont.) Save Operation Quick Reference**

Command Action	Command Format and Example
Saves a directory to a save set on a Files–11 disk	BACKUP [directory] save-set-specifier/SAVE_SET \$ BACKUP [LYKINS] DBA2:[BACKUP]1609FEB3.BCK/SAVE_SET
Saves a directory tree to a save set on magnetic tape	BACKUP [directory...] save-set-specifier/LABEL=label \$ BACKUP [LYKINS...] MTA0:1612FEB3.BCK/LABEL=TAPE01
Saves a directory tree to a save set on magnetic tape and creates a listing file	BACKUP/LIST=file-spec [directory...] save-set-specifier/LABEL=label \$ BACKUP/LIST=8SEP.LOG [LYKINS...] MTA0:8SEP.BCK/LABEL=WKL101
Saves a directory tree to a save set on magnetic tape using data compaction to increase the amount of data stored on a tape cartridge	BACKUP [directory...] save-set-specifier/MEDIA_FORMAT=COMPACTION \$ BACKUP [TESTFILES...]*.*;* MUA0:TEST.SAV/MEDIA_FORMAT=COMPACTION/REWIND

Table 7–4 shows BACKUP command formats for restore operations and some of the qualifiers you can use with restore operations. In the examples in this table, it is assumed that save sets already exist on the magnetic tape and disk.

**Table 7–4 Restore Operation Quick Reference**

Command Action	Command Format and Example
Restores from save set on disk to Files–11 disk with original UICs	BACKUP save-set-specifier/SAVE_SET ddcu:[*...]/BY_OWNER=ORIGINAL \$ BACKUP DBA2:[BACKUP]FEB2.BCK/SAVE_SET DBA1:[*...]-_\$/BY_OWNER=ORIGINAL
Restores from a save set on magnetic tape to a Files–11 disk with original UICs	BACKUP save-set-specifier ddcu:[*...]/BY_OWNER=ORIGINAL \$ BACKUP MTA0:1618FEB2.BCK DBA1:[*...]/BY_OWNER=ORIGINAL
Restores a selected file in a save set on magnetic tape to a Files–11 disk	BACKUP save-set-specifier/SELECT=file-spec file-spec \$ BACKUP MTA0:FEB2.BCK/SELECT=[POUDRE]UPLIFT.PAS - _ \$ DBA1:[GEO.PAS]UPLIFT.PAS

(continued on next page)

## BACKUP

### 7.6 BACKUP Examples

**Table 7–4 (Cont.) Restore Operation Quick Reference**

Command Action	Command Format and Example
Restores files with a specific UIC to a Files–11 disk	BACKUP save-set-specifier/BY_OWNER=[uic] file-spec \$ BACKUP MTA0:1641FEB2.BCK/BY_OWNER=[360,052] - _ \$ DBA1:[LYKINS...]
Restores files to a Files–11 disk with a new UIC	BACKUP save-set-specifier file-spec/BY_OWNER=[uic] \$ BACKUP MTA0:1641FEB2.BCK - _ \$ DBA1:[TESTS...]/BY_OWNER=[100,150]
Restores files to a Files–11 disk; if file exists, creates new version	BACKUP save-set-specifier file-spec/NEW_VERSION \$ BACKUP MTA0:1641FEB2.BCK DBA1:[LYKINS...]/NEW_VERSION
Restores files to a Files–11 disk; if file exists, replaces with new version	BACKUP save-set-specifier file-spec/REPLACE \$ BACKUP MTA0:1641FEB2.BCK DBA1:[LYKINS...]/REPLACE
Restores files to a Files–11 disk selecting certain files	BACKUP save-set-specifier/SELECT=file-spec file-spec \$ BACKUP MTA0:1641FEB2.BCK/SELECT=[LYKINS.PAS] - _ \$ DBA1:[LYKINS...]
Restores a directory tree, placing files in a different subtree	BACKUP save-set-specifier/SELECT=[directory...] [directory2...] \$ BACKUP MTA0:1641FEB2.BCK/SELECT=[FIELD...] - _ \$ DBA1:[LYKINS.NEWDATA...]
Restores a Files–11 volume from a physical save set	BACKUP/PHYSICAL save-set-specifier ddcu: \$ BACKUP/PHYSICAL MTA0:26MAR.BCK DMA3:
Restores a Files–11 volume from an image save set	BACKUP/IMAGE save-set-specifier ddcu: \$ BACKUP/IMAGE MTA0:17AUG.BCK DRA3:
Restores a Files–11 volume, maintaining the initialization parameters specified in the DCL command INITIALIZE	INITIALIZE ddcu: volume-name/new-parameters MOUNT/FOREIGN ddcu: BACKUP/IMAGE save-set-specifier ddcu:/NOINITIALIZE \$ INITIALIZE DBA1: UTTLPACK/CLUSTER=5 \$ MOUNT/FOREIGN DBA1: \$ BACKUP/IMAGE MTA0:17AUG.BCK DBA1:/NOINITIALIZE

Table 7–5 shows BACKUP command formats for copy operations, including some of the qualifiers you can use with a copy operation.

**Table 7–5 Copy Operation Quick Reference**

<b>Command Action</b>	<b>Command Format and Example</b>
Copies a directory tree to another directory tree	BACKUP [directory...] [directory...] \$ BACKUP [DAKOTA...] [SUNDANCE...]
Copies a file to another file	BACKUP file-spec file-spec \$ BACKUP LOGIN.COM [SAVE]OLDLOGIN.COM
Copies a disk volume to another disk volume	BACKUP/IMAGE ddcu: ddcu: \$ BACKUP/IMAGE DBA1: DBA2:
Copies a disk volume to another disk volume using the /PHYSICAL qualifier	BACKUP/PHYSICAL ddcu: ddcu: \$ BACKUP/PHYSICAL DYA1: DYA2:
Copies two disk volume sets using the /IMAGE qualifier	BACKUP/IMAGE volume-set-name ddcu:,ddcu: \$ BACKUP/IMAGE USER\$: DBA1:,DBA2:

Table 7–6 shows BACKUP command formats for compare operations, including some of the qualifiers you can use with a compare operation.

**Table 7–6 Compare Operation Quick Reference**

<b>Command Action</b>	<b>Command Format and Example</b>
Compares two Files–11 files	BACKUP/COMPARE file-spec file-spec \$ BACKUP/COMPARE UPLIFT.EXE;3 UPLIFT.EXE;2
Compares a selected file from a save set and a Files–11 file	BACKUP/COMPARE save-set-specifier/select=file-spec file-spec \$ BACKUP/COMPARE MTA0:FEB2.BCK/SELECT=[POUDRE]UPLIFT.PAS - _ \$ UPLIFT.PAS
Compares an image save set and Files–11 files	BACKUP/COMPARE/IMAGE save-set-specifier ddcu: \$ BACKUP/COMPARE/IMAGE MTA0:12OCT.BCK DRA3:

Table 7–7 shows BACKUP command formats for a list operation, including some of the qualifiers you can use with a list operation.

**Table 7–7 List Operation Quick Reference**

<b>Command Action</b>	<b>Command Format and Example</b>
Lists the files in a save set at the terminal	BACKUP/LIST save-set-specifier \$ BACKUP/LIST MTA0:1618FEB2.BCK

(continued on next page)

# BACKUP

## 7.6 BACKUP Examples

**Table 7-7 (Cont.) List Operation Quick Reference**

Command Action	Command Format and Example
Lists the files in a save set, writes to a file	BACKUP/LIST=file-spec save-set-specifier \$ BACKUP/LIST=NEWLIST.LIS MTA0:1618FEB2.BCK
Lists the files in a save set in full format	BACKUP/LIST/FULL save-set-specifier \$ BACKUP/LIST/FULL MTA0:1618FEB2.BCK
Lists selected files in a journal file	BACKUP/LIST/JOURNAL=journal-name/selection-qualifiers \$ BACKUP/LIST/JOURNAL=SYS\$MANAGER:INCBACKUP - _\$/SELECT=[LYKINS.WORK...]/SINCE=1-JAN-2002

---

**Crash Log Utility Extractor (CLUE) (VAX Only)**





## 8.1 CLUE Description (VAX Only)

On VAX systems, you can use the Crash Log Utility Extractor (CLUE) to display the contents of a **crash history file**. The crash history file, which is created and updated by CLUE, contains key parameters from crash dump files. Unlike crash dumps, which are overwritten with each system failure (crash) and are therefore typically available only for the most recent failure, the crash history file is a permanent record of system failures. By examining the contents of the crash history file with the CLUE/DISPLAY command, you can understand and resolve the issues responsible for some crashes and you can also obtain other useful data.

On Alpha and I64 systems, CLUE features are provided through System Dump Analyzer (SDA) utility commands. Refer to the *OpenVMS Alpha System Dump Analyzer Utility Manual* for more information.

## 8.2 CLUE Usage Summary (VAX Only)

When a system fails, physical memory is copied to the crash dump file, and CLUE automatically appends the relevant parameters to a file named CLUE\$OUTPUT:CLUE\$HISTORY.DATA. You can use CLUE to display and examine the data in this file.

### Format

CLUE /DISPLAY

#### Parameters

None.

### Description

Before you run CLUE, define the CLUE command (either interactively or in a procedure such as a login command procedure), as follows:

```
$ CLUE ::= $CLUE
```

To invoke CLUE, enter the CLUE/DISPLAY command. To terminate CLUE, enter the EXIT command at the CLUE\_DISPLAY> prompt or press Ctrl/Z.

Note that you can also issue CLUE/DISPLAY commands directly from DCL, as in the following example:

```
$ CLUE/DISPLAY = DIR/TYPE=INVEXCEPTN/SINCE=21-NOV-2002/OUT=CLUE.LIS
```

## 8.3 CLUE Commands (VAX Only)

This section describes and provides examples of the CLUE /DISPLAY commands. 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.

The following table lists the commands described in this section:

## CLUE (VAX Only)

### 8.3 CLUE Commands (VAX Only)

Command	Description
DELETE	Deletes an entry from the crash history file
DIRECTORY	Displays a list of entries from the crash history file
EXIT	Exits from CLUE
EXTRACT	Extracts data from an entry in the crash history file to a file
HELP	Provides online help for CLUE/DISPLAY commands
SHOW	Displays specific information for an entry in the crash history file

## **DELETE (VAX Only)**

Deletes an entry from the list and writes the remaining entries to a binary file.

### **Format**

```
DELETE n
```

### **Parameter**

**n**

The number of the entry that you want to delete from the crash history file. The number of the entry corresponds to the entry number shown by the **DIRECTORY** command.

### **Description**

Entries in the **CLUE.BIN** file can be permanently removed using the **DELETE** command. This command deletes the specified entry and writes the remaining entries to an output file. When an entry is deleted from the crash history file, a new history file (with a higher version number) is created.

### **Example**

```
CLUE_DISPLAY> DELETE 4
```

Deletes the fourth entry in the list.

## CLUE (VAX Only) DIRECTORY (VAX Only)

---

### DIRECTORY (VAX Only)

Displays a brief summary of each entry in the crash history file.

#### Format

DIRECTORY

#### Qualifiers

##### **/MODULE=**

Lists all failures that correspond to the specified module (that is, failures that start with the specified string).

##### **/OFFSET=**

Lists all failures that correspond to the specified offset (that is, failures that start with the specified string).

##### **/SINCE=**

Lists all failures that have occurred since the specified date. If the **/SINCE** qualifier is used and no date is specified, all failures that have occurred on the current day are listed.

##### **/TYPE=**

Lists all failures that correspond to the specified type (that is, failures that start with the specified string).

#### Description

The **DIRECTORY** command lists the record entries in the crash history file, beginning with the most recent entry. Six headings are displayed for each entry:

- Entry number
- Date and time of crash
- Type
- Process
- Module
- Offset

You can use one or more qualifiers for the **DIRECTORY** command to limit the crash entries that are displayed. You can also use more than one qualifier in the same command line. For example, the command **DIRECTORY/SINCE=18-APR-2002/MODULE=NETACP** lists only those crash entries that have taken place since April 18, 2002, and for which the module is **NETACP**.

## Examples

1. CLUE\_DISPLAY> DIRECTORY

#	Time	Type	Process name	Module	Offset
====	=====	=====	=====	=====	=====
1	11-JUL-2002 09:07:45.78	INVEXCEPTN	batman	NETACP	14B9
2	01-JAN-2002 11:32:55.23	SSRVEXCEPT	startrek	SYSLOA	10A8
3	15-MAY-2002 07:26:12.34	BADFID	evolushun	NONE	NONE
4	22-APR-2002 10:45:20.60	INVEXCEPTN	aprocess	IOBUF	015D

This example lists all entries in the crash history file.

2. CLUE\_DISPLAY> DIRECTORY /MODULE=SYSLOA

#	Time	Type	Process name	Module	Offset
====	=====	=====	=====	=====	=====
2	01-JAN-2002 11:32:55.23	SSRVEXCEPT	startrek	SYSLOA	10A8

This example lists only those entries for which the module is SYSLOA.

**CLUE (VAX Only)**  
**EXIT (VAX Only)**

---

**EXIT (VAX Only)**

Exits from CLUE.

**Format**

EXIT

**Description**

This command exits from CLUE and returns you to the DCL level.

**Example**

```
CLUE_DISPLAY> EXIT  
$
```

This example exits from CLUE.

## EXTRACT (VAX Only)

Extracts all of the data from an entry in the crash history file to either an ASCII or a binary file.

### Format

```
EXTRACT n
```

### Parameter

**n**

The number of the entry that you want to extract to a file. The number of the entry corresponds to the entry number shown by the DIRECTORY command.

### Qualifiers

**/BINARY=filename.bin**

The /BINARY qualifier writes the output to a binary file.

**/OUTPUT=filename.txt**

The /OUTPUT qualifier writes the output to an ASCII file. The /OUTPUT qualifier is the default.

### Description

Use the EXTRACT command to save record entries from the crash history file to an ASCII or a binary file. If no qualifier is used, entry is written to a text file with the name CLUE\$HISTORY.TXT.

### Example

```
CLUE_DISPLAY> EXTRACT 3 /OUTPUT=15MAYCRASH.TXT
```

This command writes the data from entry number 3 in the crash history file to an ASCII file named 15MAYCRASH.TXT.

**CLUE (VAX Only)**  
**HELP (VAX Only)**

---

**HELP (VAX Only)**

Provides online help for CLUE commands.

**Format**

HELP [command]

**Parameter**

**command**

The command for which you want help.

**Example**

```
CLUE_DISPLAY> HELP DIRECTORY
```

This command provides online help for the DIRECTORY command in CLUE.



## **SHOW (VAX Only)**

Displays specific information for an entry in the crash history file.

### **Format**

```
SHOW info_type n
```

### **Parameters**

#### **info\_type**

You must choose one of the following information types:

- **CRASH**—displays information similar to the **SHOW CRASH** command in the System Dump Analyzer (SDA) utility.
- **STACK**—displays information similar to the **SHOW STACK** command in SDA.
- **EXEC**—displays information similar to the **SHOW EXEC** and **SHOW DEVICE** commands in SDA.
- **ISTREAM**—displays the instruction stream around the failing PC.
- **SUMMARY**—displays the active processes on the system, similar to the **SHOW SUMMARY/IMAGE** command in SDA.
- **ALL**—displays all the information from the preceding types.

For more information about SDA commands, refer to the *OpenVMS VAX System Dump Analyzer Utility Manual*.

#### **n**

The number of the entry for which you want to display information. The number of the entry corresponds to the entry number shown by the **DIRECTORY** command.

### **Description**

This command lets you see all the data associated with a particular failure. This information is split into several sections; you can choose to see either individual sections or all of the information.

If the crash number is not specified, information for the most recent entry in the crash history file is displayed. You must, however, specify one of the keywords for *info\_type*.

If you request a specific entry from the crash history list (using the **SHOW n** command), the display will include some parameters that are useful only to HP service representatives.

## CLUE (VAX Only) SHOW (VAX Only)

### Examples

1. CLUE\_DISPLAY> SHOW ISTREAM 4

Instructions around the failing PC:

```
80A9F841 RSB
80A9F842 BUG_CHECK #019C
80A9F846 BUG_CHECK #019C
80A9F84A PUSHL R4
80A9F84C MOVL R5,R4
80A9F84F BEQL 00002C88

80A9F851 JSB @#-7FFCC48
80A9F857 MOVL (SP)+,R4
80A9F85A RSB
80A9F85B BUG_CHECK #019C
PC->80A9F85F BUG_CHECK #019C => CLUSTRLOA + 09B6F
80A9F863 MOVL #00,R1
80A9F866 MOVB #01,R0
80A9F869 RSB
80A9F86A INCL 00002301
80A9F86E TSTW 000022FF
80A9F872 BLSS 00002CA5
```

This example displays instruction stream information around the failing PC from the crash history file for entry number 4.

2. CLUE\_DISPLAY> SHOW CRASH 4

```
Time of system crash:          21-MAR-2002 15:21:33.72
Version of system:             VAX/VMS VERSION V7.3-1
System Version Major ID/Minor ID: 1/0
VAXcluster node:              HERMES, a VAX 6000-420
Crash CPU ID/Primary CPU ID:  03/01
Bitmask of CPUs active/available: 0000000A/0000000A
CPU bugcheck codes:           CPU 03      INCONSTATE
                               1 other    CPUEXIT
Current Process name:         OPCOM
Current IPL:                   8
CPU database address:         801AA000

General registers:
R0 = 80A9F85B   R1 = 00000002   R2 = 80A15B08   R3 = 00010008
R4 = 80A15AD0   R5 = 00000000   R6 = 00000001   R7 = 00000042
R8 = 00022520   R9 = 00020F18   R10 = 00021000  R11 = 00020EC0
.
.
.
```

This example displays information about the state of the system at the time of the failure. The actual output from this command may contain significantly more information than is shown in this example.

---

# EFI Utilities for OpenVMS



## 9.1 EFI Utilities Description

The following OpenVMS EFI (Extensible Firmware Interface) utilities provide device management functions at the EFI console on I64 systems:

- **VMS\_BCFG**  
Adds an entry to the EFI Boot Manager using a specified OpenVMS device name.
- **VMS\_SET**  
Sets the dump device and the debug device to the specified OpenVMS device name.
- **VMS\_SHOW**  
Displays the equivalent OpenVMS device name for devices mapped by the EFI console.

A full description of each EFI utility follows.

## VMS\_BCFG

Adds an entry to the EFI Boot Manager using a specified OpenVMS device name.

VMS\_BCFG is an extension of the EFI bcfg to support OpenVMS device names and additional features required for FC boot devices. In addition to using the OpenVMS device name as a parameter, this utility allows users to set the optional VMS boot flags to the boot options entry using a familiar OpenVMS syntax.

For multipath devices, this utility can add all active paths to the specified OpenVMS device automatically. It also appends the WWID on the description so users will know which FC path is selected.

This utility is compatible with all disk devices and network devices.

### Format

```
VMS_BCFG driver | boot [dump][add # device-name -fl x,y "desc"] [rm #] [mv # #] [-v]
```

### Parameter

**driver**

Selects a boot driver list.

**boot**

Selects a boot option list from the options below.

**add | set**

Adds an entry to the EFI Boot Manager.

**# (default=1)**

Boot order position of the new boot option list. Defaults to 1 if not specified, which corresponds to the first entry on the EFI Boot Manager.

**device-name**

The specified OpenVMS device name. For FC devices, specifying \$1\$ in the name is optional.

**-flags x,y (default=none)**

The specified OpenVMS flags to be passed on the Boot Option. Unless specified, no flags are passed.

**desc**

Unicode/ASCII description of the Boot Option. For multipath FC devices, it appends the WWID of the device on the description.

**dump | show**

Displays the boot option list.

**rm | delete #**

Removes an entry corresponding to the boot position number.

**mv | rename # #**

Changes the boot position number specified from the first # to the second #.

## Examples

1. `fs1:\efi\vms> vms_bcfg boot show`  
The boot option list is:  
01. VenHw(D65A6B8C-71E5-4DF0-A909-F0D2992B5AA9) "EFI Shell [Built-in]" OPT

This example shows the boot option list.

2. `fs1:\efi\vms> vms_bcfg boot add 2 $!$dga3730 -fl 1,0 "DGA3730 Root 1"`  
VMS: DGA3730 Fibre Device  
EFI: fs1: Acpi(000222F0,200)/Pci(1|1)/Fibre(50001FE10011B15D),Lun(D)  
vms\_bcfg: Add boot option as 2  
vms\_bcfg: Add the next available VMS path? (Yes/No) [YES]  
VMS: DGA3730 Fibre Device  
EFI: fs9: Acpi(000222F0,300)/Pci(1|0)/Fibre(50001FE10011B15C),Lun(D)  
vms\_bcfg: Add boot option as 3  
vms\_bcfg: Add the next available VMS path? (Yes/No) [YES]  
EFI Boot Manager ver 1.10 [14.61] Firmware ver 2.21 [4334]

Please select a boot option

```
EFI Shell [Built-in]
DGA3730 Root 1 Fibre(50001FE10011B15D)
DGA3730 Root 1 Fibre(50001FE10011B15C)
Boot Option Maintenance Menu
System Configuration Menu
```

Use ^ and v to change option(s). Use Enter to select an option.

This example adds a multipath FC disk to the EFI Boot Manager's boot options list.

## VMS\_SET

Sets the dump device and the debug device to the specified OpenVMS device name.

### Format

```
VMS_SET dump_dev | debug_dev [device-name]
```

### Parameter

#### **dump\_dev [device-name]**

Sets the NVRAM variable DUMP\_DEV to the specified OpenVMS dump device for the dump-off-system-disk (DOSD) function. For a multipath FC device, it adds other FC device paths automatically.

#### **debug\_dev [device-name]**

Sets the NVRAM variable DEBUG\_DEV to the specified OpenVMS debug device for use by the System Code Debugger (SCD).

### Examples

1. 

```
fs1:\efi\vms> vms_set dump_dev dga3730
VMS: DGA3730      Fibre Device
EFI: fs1: Acpi(000222F0,200)/Pci(1|1)/Fibre(50001FE10011B15D), Lun(D)

VMS: DGA3730      Fibre Device
EFI: fs9: Acpi(000222F0,300)/Pci(1|0)/Fibre(50001FE10011B15C), Lun(D)
```

This example sets the dump device to the specified OpenVMS device name.

2. 

```
fs1:\efi\vms> vms_set debug_dev eia0
VMS: EIA0         0-30-6E-39-E7-55
EFI: Acpi(000222F0,0)/Pci(3|0)/Mac(00306E39E755)
```

This example sets the debug device to the network device eia0.



---

## VMS\_SHOW

Displays the equivalent OpenVMS device name for devices mapped by the EFI console.

### Format

```
vms_show device | dump_dev | debug_dev [device-name] [-fs]
```

### Parameter

#### **device**

Displays all bootable devices mapped by the EFI console and their corresponding OpenVMS device names.

The first output line shows the OpenVMS device name and additional information about the device. The additional information consists of vendor identification and product identification if the device is a disk; or a MAC address if the device is a network device.

The second output line shows the file system designation (fsx>:) and its corresponding EFI device path.

#### **[device-name]**

If specified, the utility matches the specified OpenVMS device name to the EFI console mapping. For multipath FC devices, it displays all paths associated with the given OpenVMS device name.

#### **dump\_dev**

Displays the selected OpenVMS dump device for the dump-off-system-disk (DOSD) function.

#### **debug\_dev**

Displays the selected OpenVMS debug device.

#### **-fs**

Displays the OpenVMS device names of those devices that have only the system disk.

### Examples

```
1. fsl:\efi\vms> vms_show dev dkb0
VMS:DKB0          HP 18.2GST318406LC  HP05
EFI: fs0: Acpi(000222F0,100)/Pci(1|1)/Scsi(Pun0,Lun0)
```

This example shows the EFI device path for a specified OpenVMS device name.

```
2. fsl:\efi\vms> vms_show dump
VMS: DGA3730      COMPAQ HSV110 (C)COMPAQ3014
EFI: fsl: Acpi(000222F0,200)/Pci(1|1)/Fibre(50001FE10011B15D),Lun(D)
```

This example shows the settings for the OpenVMS DUMP\_DEV device.



---

## Error Log Viewer Utility (ELV)



## 10.1 ELV Description

The Error Log Viewer (ELV) utility allows you to quickly examine, from the command line, an error log file in a user-readable format before deciding whether the data warrants a more comprehensive analysis with a tool such as the System Event Analyzer (SEA).

ELV is particularly useful with error logs created on systems with newer devices or on newer systems such as most of the DS, ES, and GS series. ELV can also be used with error logs created on some older systems or on systems equipped with older devices.

ELV provides detailed information for all error log events of the following types:

- Bugchecks
- Volume mounts and dismounts
- Correctable error throttling notifications
- Software parameters
- Control entries:
  - System service messages
  - Network messages
  - Operator messages
  - ERRLOG.SYS created
  - Time stamps
  - System startup messages

ELV provides detailed information for some error log events of the following types:

- Machine checks
- Device errors
- Device timeouts
- Asynchronous device attentions
- Logged messages
- Logged MSCP messages

This chapter describes the ELV commands, along with their parameters and qualifiers, that support the Error Log Viewer utility. It also contains a section of sample reports produced using the ELV TRANSLATE command.

## ELV

### 10.2 ELV Usage Summary

## 10.2 ELV Usage Summary

You can execute ELV commands directly from the DCL command level or from ELV's interactive shell mode.

### Format

```
ANALYZE/ERROR_LOG/ELV [command]
```

#### Parameter

##### command

Specifies an ELV operation to be performed. If you do not specify an operation, the utility enters interactive shell mode, displays the ELV> prompt, and waits for command input.

To invoke ELV, enter the following DCL command:

```
$ ANALYZE/ERROR_LOG/ELV
```

If you do not enter an ELV command, the utility enters interactive shell mode and displays the ELV prompt:

```
ELV>
```

You can then enter an ELV command. After ELV executes the command, it again displays the ELV> prompt.

To return directly to DCL after executing an ELV command from the ELV prompt, use the /NOINTERACTIVE qualifier.

You can also enter an ELV command directly from DCL; for example:

```
$ ANALYZE/ERROR_LOG/ELV TRANSLATE ERRLOG.SYS;42
```

After ELV executes the command, you are returned to the DCL prompt by default.

To enter interactive shell mode after executing an ELV command directly from DCL, use the /INTERACTIVE qualifier.

## 10.3 Understanding Categories of Events

ELV recognizes several categories of events for inclusion in (or exclusion from) various operations. The first major distinction is between valid and invalid events. Within the category of valid events are selected and rejected events. Explanations of these categories follow.

- Valid

An event is considered valid if it can be read into a buffer and its header is translatable. The body of the event, however, does not need to be translatable; in this case, the event is valid but untranslatable. With the exception of the DUMP/INVALID command, all ELV operations are performed on valid events.

- Selected

By default, all valid events are considered to be selected events. Except when you use the /REJECTED qualifier, ELV operations are performed on selected events.

- Rejected

A valid event is considered to be rejected if it does not fall into an interval specified with the /BEFORE, /SINCE, or /ENTRY qualifier, or is filtered out with the /EXCLUDE, /INCLUDE, or /[NO]NODE qualifier.

You can combine certain intervals and filters to further refine event selection. In addition, you can use the /REJECTED qualifier to force ELV operations to be performed on rejected (rather than selected) events.

For more information, refer to the qualifiers for individual commands.

- Invalid

An event is considered to be invalid if it cannot be read into a buffer or its header is not translatable. You can use the ELV command DUMP/INVALID to output invalid events to an output dump file for further examination.

## 10.4 ELV Commands

This section describes and provides examples of the following ELV commands:

- CONVERT
- DUMP
- EXIT
- HELP
- TRANSLATE
- WRITE

---

## CONVERT

Converts and writes events from one or more binary error log files written in the newer format to a single new error log file written in the older format. The new file can then be read by ANALYZE/ERROR\_LOG.

This command is primarily used to enable translation of older error log events whose translation is not supported by ELV.

### Format

CONVERT [input-file,...]

### Parameter

#### input-file

Supplies one or more names of binary error log files to be converted to the older format.

If you do not specify an input file, the default input file is SYS\$ERRORLOG:ERRLOG.SYS. If you do not specify a device and directory, your current device and directory are used. If you do not specify a file name, the default file name is ERRLOG. If you do not specify a file type, the default file type is .SYS.

### Qualifiers

#### /BEFORE[=date-time]

Specifies that only those events dated earlier than the stated date and time are to be selected.

For *date-time*, you can specify an absolute time, a delta time, or a combination of absolute and delta times. See the *OpenVMS User's Manual* for details on specifying times.

If you omit a date and time, TODAY is used.

#### /ENTRY[=keyword,...]

Specifies the range of entries to be selected.

You can specify one or both of the following keywords:

Keyword	Description
START[:decimal-value]	Indicates the start of a range of entries to be selected.
END[:decimal-value]	Indicates the end of a range of entries to be selected.

#### Usage Notes:

- You can specify one or both of these keywords. If you specify both keywords, you must enclose them in parentheses.
- If you specify /ENTRY without an entry range, the entry range defaults to START:1,END:end-of-file.



- If you specify the START or END keyword without a value, the keyword is ignored.

**/EXCLUDE=event-class[,...]**

Specifies an event class or classes to be rejected. Do not use /EXCLUDE with /INCLUDE.

For *event-class*, specify one or more of the keywords shown in the following table. If you specify more than one keyword, use a comma-separated list of values enclosed in parentheses.

<b>Keyword</b>	<b>Description</b>
ATTENTIONS	Reject device attention entries.
BUGCHECKS	Reject all types of bugcheck entries.
CONFIGURATION	Reject system configuration entries.
CONTROL_ENTRIES	Reject control entries. Control entries include the following entry types: <ul style="list-style-type: none"> <li>• System power failure 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	Reject CPU-related entries. 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	Reject device error entries.
ENVIRONMENTAL_ENTRIES	Reject environmental entries.
MACHINE_CHECKS	Reject machine check entries.
MEMORY	Reject memory errors.
SNAPSHOT_ENTRIES	Reject snapshot entries.
SYNDROME	Reject firmware-generated entries that describe a symptom set used by HP support personnel to identify problems.
TIMEOUTS	Reject device timeout entries.

## ELV CONVERT

Keyword	Description
UNKNOWN_ENTRIES	Reject any entry that has an unknown entry class.
UNSOLICITED_MSCP	Reject unsolicited MSCP entries.
VOLUME_CHANGES	Reject volume mount and dismount entries.

### **/INCLUDE=event-class[,...]**

Specifies an event class or classes to be selected. All other event classes are implicitly rejected. Do not use /INCLUDE with /EXCLUDE.

For *event-class*, specify one or more of the keywords shown in the following table. If you specify more than one keyword, use a comma-separated list of values enclosed in parentheses.

Keyword	Description
ATTENTIONS	Select device attention entries.
BUGCHECKS	Select all types of bugcheck entries.
CONFIGURATION	Select system configuration entries.
CONTROL_ENTRIES	Select control entries. Control entries include the following entry types: <ul style="list-style-type: none"><li>• System power failure 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	Select CPU-related entries. 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	Select device error entries.
ENVIRONMENTAL_ENTRIES	Select environmental entries.
MACHINE_CHECKS	Select machine check entries.
MEMORY	Select memory errors.
SNAPSHOT_ENTRIES	Select snapshot entries.

Keyword	Description
SYNDROME	Select firmware-generated entries that describe a symptom set used by HP support personnel to identify problems.
TIMEOUTS	Select device timeout entries.
UNKNOWN_ENTRIES	Select any entry that has an unknown entry class.
UNSOLICITED_MSCP	Select unsolicited MSCP entries.
VOLUME_CHANGES	Select volume mount and dismount entries.

**/INTERACTIVE**

**/NOINTERACTIVE**

Specifies whether or not ELV is to run in interactive shell mode after you execute the ELV command. By default, interactive shell mode results from the way the current ELV command is entered.

For more information, refer to Section 10.2.

**/LOG**

**/NOLOG**

Specifies whether or not ELV is to output control and informational messages to the terminal. The default /NOLOG does not output these messages to the terminal.

**/NODE[=node-name,...]**

**/NONODE[=node-name,...]**

Specifies that events occurring on a node or nodes are to be selected or rejected.

If you enter /NODE without a value, only events that occur on the node on which you are running ELV are selected.

If you enter /NONODE without a value, events occurring on all nodes that are represented in the error log file are selected.

**/OUTPUT[=output-file]**

Specifies the output file that is to contain converted copies of events.

If you do not specify an output file name, the input file name is used. If you do not specify a device and directory, your current device and directory are used. If you do not specify a file type, the default file type is .CVT.

**/REJECTED**

Specifies that rejected (rather than selected) events are to be converted. For more information, refer to Section 10.3.

**/SINCE[=date-time]**

Specifies that only those events dated later than the stated date and time are to be selected.

For *date-time*, you can specify an absolute time, a delta time, or a combination of absolute and delta times. See the *OpenVMS User's Manual* for details on specifying times.

If you omit a date and time, TODAY is used.

# ELV CONVERT

## Examples

1. `$ ANALYZE/ERROR_LOG/ELV CONVERT /INTERACTIVE /NONODE=(BEAVIS,BUTTHD)`

The command in this example, executed from DCL command level, converts selected events in the default `SYS$ERRORLOG:ERRLOG.SYS` file to the older format and writes these events to the `ERRLOG.CVT` file. After the command executes, the system displays the `ELV>` prompt.

The selected events are from all nodes except `BEAVIS` and `BUTTHD`.

2. `ELV> CONVERT /LOG /OUTPUT=OUTFILE`

The command in this example converts events in the default `SYS$ERRORLOG:ERRLOG.SYS` file to the older format and writes the converted events to `OUTFILE.CVT`.

In addition, control and informational messages are output to the terminal.

3. `$ ANALYZE/ERROR_LOG/ELV CONVERT /NODE /BEFORE=YESTERDAY ERROR_LOG.SYS`

The command in this example converts selected events from `ERROR_LOG.SYS` to the older format and writes the converted events to `ERROR_LOG.CVT`.

The selected events are those that occurred before `YESTERDAY` on the current node.

4. `ELV> CONVERT /ENTRY=START:5 /EXCLUDE=BUGCHECKS`

The command in this example converts events in the default `SYS$ERRORLOG:ERRLOG.SYS` file to the older format and writes the converted events to `ERRLOG.CVT`.

The selected events, which start with entry 5, do not include the `BUGCHECKS` event class.

---

## DUMP

Writes events from one or more binary error log files to a single new ASCII output file in an OpenVMS dump-style format.

### Format

DUMP [input-file,...]

### Parameter

#### input-file

Supplies one or more names of binary error log files to be used to produce an output dump file.

If you do not specify an input file, the default input file is SYS\$ERRORLOG:ERRLOG.SYS. If you do not specify a device and directory, your current device and directory are used. If you do not specify a file name, the default file name is ERRLOG. If you do not specify a file type, the default file type is .SYS.

### Qualifiers

#### /BEFORE[=date-time]

Specifies that only those events dated earlier than the stated date and time are to be selected. Do not use /BEFORE with /INVALID.

For *date-time*, you can specify an absolute time, a delta time, or a combination of absolute and delta times. See the *OpenVMS User's Manual* for details on specifying times.

If you omit a date and time, TODAY is used.

#### /ENTRY[=keyword,...]

Specifies the range of entries to be selected. Do not use /ENTRY with /INVALID.

You can specify one or both of the following keywords:

Keyword	Description
START[:decimal-value]	Indicates the start of a range of entries to be selected.
END[:decimal-value]	Indicates the end of a range of entries to be selected.

#### Usage Notes:

- You can specify one or both of these keywords. If you specify both keywords, you must enclose them in parentheses.
- If you specify /ENTRY without an entry range, the entry range defaults to START:1,END:end-of-file.
- If you specify the START or END keyword without a value, the keyword is ignored.

# ELV DUMP

## **/EXCLUDE=event-class[,...]**

Specifies an event class or classes to be rejected. Do not use /EXCLUDE with /INCLUDE.

For *event-class*, specify one or more of the keywords shown in the following table. If you specify more than one keyword, use a comma-separated list of values enclosed in parentheses.

<b>Keyword</b>	<b>Description</b>
ATTENTIONS	Reject device attention entries.
BUGCHECKS	Reject all types of bugcheck entries.
CONFIGURATION	Reject system configuration entries.
CONTROL_ENTRIES	Reject control entries. Control entries include the following entry types: <ul style="list-style-type: none"><li>• System power failure 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	Reject CPU-related entries. 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	Reject device error entries.
ENVIRONMENTAL_ENTRIES	Reject environmental entries.
MACHINE_CHECKS	Reject machine check entries.
MEMORY	Reject memory errors.
SNAPSHOT_ENTRIES	Reject snapshot entries.
SYNDROME	Reject firmware-generated entries that describe a symptom set used by HP support personnel to identify problems.
TIMEOUTS	Reject device timeout entries.

<b>Keyword</b>	<b>Description</b>
UNKNOWN_ENTRIES	Reject any entry that has an unknown entry class.
UNSOLICITED_MSCP	Reject unsolicited MSCP entries.
VOLUME_CHANGES	Reject volume mount and dismount entries.

**/INCLUDE=event-class[,...]**

Specifies an event class or classes to be selected. All other event classes are implicitly rejected. Do not use /INCLUDE with /EXCLUDE.

For *event-class*, specify one or more of the keywords shown in the following table. If you specify more than one keyword, use a comma-separated list of values enclosed in parentheses.

<b>Keyword</b>	<b>Description</b>
ATTENTIONS	Select device attention entries.
BUGCHECKS	Select all types of bugcheck entries.
CONFIGURATION	Select system configuration entries.
CONTROL_ENTRIES	Select control entries. Control entries include the following entry types: <ul style="list-style-type: none"> <li>• System power failure 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	Select CPU-related entries. 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	Select device error entries.
ENVIRONMENTAL_ENTRIES	Select environmental entries.
MACHINE_CHECKS	Select machine check entries.
MEMORY	Select memory errors.
SNAPSHOT_ENTRIES	Select snapshot entries.

## ELV DUMP

Keyword	Description
SYNDROME	Select firmware-generated entries that describe a symptom set used by HP support personnel to identify problems.
TIMEOUTS	Select device timeout entries.
UNKNOWN_ENTRIES	Select any entry that has an unknown entry class.
UNSOLICITED_MSCP	Select unsolicited MSCP entries.
VOLUME_CHANGES	Select volume mount and dismount entries.

### **/INTERACTIVE**

### **/NOINTERACTIVE**

Specifies whether or not ELV is to run in interactive shell mode after you execute the ELV command. By default, interactive shell mode results from the way the current ELV command was entered.

For more information, refer to Section 10.2.

### **/INVALID**

Specifies that invalid (rather than valid) events are to be written to an output dump file. Do not use /INVALID with /BEFORE, /ENTRY, /EXCLUDE, /INCLUDE, /[NO]NODE, /REJECTED, or /SINCE.

For more information, refer to Section 10.3.

### **/LOG**

### **/NOLOG**

Specifies whether or not ELV is to output control and informational messages to the terminal. The default /NOLOG does not output these messages to the terminal.

### **/NODE[=node-name,...]**

### **/NONODE[=node-name,...]**

Specifies that events occurring on a node or nodes are to be selected or rejected. Do not use /NODE or /NONODE with /INVALID.

If you enter /NODE without a value, only events that occur on the node on which you are running ELV are selected.

If you enter /NONODE without a value, events occurring on all nodes that are represented in the error log file are selected.

### **/OUTPUT[=output-file]**

Specifies the output file that is to contain OpenVMS dump-style records for events.

If you do not specify an output file name, the input file name is used. If you do not specify a device and directory, your current device and directory are used. If you do not specify a file type, the default file type is .DMP.

### **/REJECTED**

Specifies that rejected (rather than selected) events are to be written to an output dump file. Do not use /REJECTED with /INVALID.

For more information, refer to Section 10.3.



**/SINCE[=date-time]**

Specifies that only those events dated later than the stated date and time are to be selected. Do not use /SINCE with /INVALID.

For *date-time*, you can specify an absolute time, a delta time, or a combination of absolute and delta times. See the *OpenVMS User's Manual* for details on specifying times.

If you omit a date and time, TODAY is used.

**Examples**

1. ELV> DUMP /NODE=DUMBO /ENTRY=END:10 ERROR\_FILE.SYS,ERRLOG.SYS

The command in this example writes OpenVMS dump-style records created from selected events in two input files, ERROR\_FILE.SYS and ERRLOG.SYS, to an output dump file named ERROR\_FILE.DMP. The name of the output file comes from the name of the first input file in the list.

The selected events, which end with entry 10, are those that occurred on node DUMBO.

2. \$ ANALYZE/ERROR\_LOG/ELV DUMP /INCLUDE=(DEVICE\_ERRORS,TIMEOUTS) /LOG

The command in this example writes OpenVMS dump-style records created from selected events in the default SYS\$ERRORLOG:ERRLOG.SYS file to an output dump file named ERRLOG.DMP.

The selected events include only the DEVICE\_ERRORS and TIMEOUTS event classes. In addition, control and informational messages are output to the terminal.

3. ELV> DUMP /SINCE=22-MAY-2003:01:00:00.00 /BEFORE=24-MAY-2003:04:51:33.87

The command in this example writes OpenVMS dump-style records created from selected events in the default SYS\$ERRORLOG:ERRLOG.SYS file to an output dump file named ERRLOG.DMP.

The selected events fall in the interval between the two specified dates.

4. ELV> DUMP /SINCE=22-MAY-2003:01:00:00.00 /BEFORE=24-MAY-2003:04:51:33.87  
/REJECTED

This example is identical to the previous example except that rejected events are written rather than selected events.

The rejected events are those that fall outside the interval between the two specified dates.

5. \$ ANALYZE/ERROR\_LOG/ELV DUMP /INVALID /OUTPUT=OUTFILE.OUT

The command in this example writes OpenVMS dump-style records created from invalid events in the default SYS\$ERRORLOG:ERRLOG.SYS file to an output dump file named OUTFILE.OUT.

## ELV EXIT

---

### EXIT

Stops the execution of ELV and returns control to the DCL command level. You can also enter Ctrl/Z to perform the same function.

### Format

EXIT

### Example

```
ELV> EXIT  
$
```

The command in this example terminates the ELV session and returns control to the DCL command level.

## HELP

Provides online help information for using ELV.

## Format

```
HELP [help-topic]
```

## Parameter

### **help-topic**

Specifies the topic for which you want help.

## Example

```
ELV> HELP CONVERT  
$ ANALYZE/ERROR_LOG/ELV HELP CONVERT
```

The commands in this example show how to get help information for the ELV CONVERT command from both the interactive shell mode and DCL command level.

# ELV TRANSLATE

---

## TRANSLATE

Performs a bit-to-text translation of events from one or more binary error log files and writes the resulting reports to the terminal or to a single new ASCII output file.

Section 10.5 contains examples of reports.

### Format

```
TRANSLATE [input-file,...]
```

### Parameter

#### **input-file**

Supplies one or more names of binary error log files to be translated.

If you do not specify an input file, the default input file is SYS\$ERRORLOG:ERRLOG.SYS. If you do not specify a device and directory, your current device and directory are used. If you do not specify a file name, the default file name is ERRLOG. If you do not specify a file type, the default file type is .SYS.

### Description

When you use the TRANSLATE command, ELV can generate reports in the formats shown in Table 10-1.

---

#### **Note**

---

When you use the ELV TRANSLATE command to examine error log files created on systems running OpenVMS Version 7.2 through 7.3-1, you might see the following set of messages displayed after the translation of certain types of events:

```
%ELV-E-B2TNOTFND, valid bit-to-text translation data not found  
-ELV-W-NODNOTFND, bit-to-text node not found
```

These messages are the result of a minor change in the error log file format between OpenVMS Versions 7.3-1 and 7.3-2; they can be disregarded. The events affected should be translated correctly.

---

**Table 10–1 Report Formats**

Format	Description
<b>Standard</b>	<p>A report that contains detailed bit-to-text translation information for each event.</p> <p>To specify the detail level of a standard report, use the <code>/BRIEF</code>, <code>/FULL</code>, or <code>/ONE_LINE</code> qualifier with the <code>TRANSLATE</code> command. In addition to these qualifiers, you can use the <code>/TERSE</code> qualifier to obtain a report that contains less interpretation of the data, regardless of detail level.</p> <p>At a minimum, the following header information is included for each event in a standard report:</p> <ul style="list-style-type: none"> <li>• Event Number</li> <li>• Event Type</li> <li>• Timestamp</li> <li>• Node</li> <li>• Event Class</li> </ul> <p>With the exception of Event Type, the header information is useful when using the following event filtering and interval qualifiers: <code>/BEFORE</code>, <code>/ENTRY</code>, <code>/EXCLUDE</code>, <code>/INCLUDE</code>, <code>/[NO]NODE</code>, <code>/SINCE</code>. For Event Class, refer to the keywords listed for the <code>/EXCLUDE</code> or <code>/INCLUDE</code> qualifier.</p> <p><b>Standard Report Detail Levels:</b></p> <ul style="list-style-type: none"> <li>• If you specify <code>/ONE_LINE</code>, the header information is the only event information that is included in a standard report.</li> <li>• If you specify <code>/BRIEF</code>, only the most essential event information is included with the header information.</li> <li>• If you do not specify a standard report detail level qualifier, only the most commonly useful event information is included with the header information.</li> <li>• If you specify <code>/FULL</code>, all event information is included with the header information.</li> </ul> <p><b>Summary</b> Summary information is broken down by node where applicable, followed by summary information for the entire file. See the example following this table.</p>

# ELV TRANSLATE

```
Output for SYS$SYSROOT:[SYSERR]ERRLOG.SYS;1
Total number of events:          19
Number of the first event:      1
Number of the last event:      19
Earliest event occurred:       19-JUN-2003 08:52:46.00
Latest event occurred:         19-JUN-2003 13:08:23.62
Number of events by event class:
    BUGCHECKS                   1
    CONFIGURATION                2
    CONTROL_ENTRIES              4
    UNKNOWN_ENTRIES              8
    VOLUME_CHANGES              4
```

## Qualifiers

### **/BEFORE[=date-time]**

Specifies that only those events dated earlier than the stated date and time are to be selected.

For *date-time*, you can specify an absolute time, a delta time, or a combination of absolute and delta times. See the *OpenVMS User's Manual* for details on specifying times.

If you omit a date and time, TODAY is used.

### **/BRIEF**

Specifies that ELV is to generate a brief standard report. Do not use /BRIEF with /FULL or /ONE\_LINE.

### **/ENTRY[=keyword,...]**

Specifies the range of entries to be selected.

You can specify one or both of the following keywords:

Keyword	Description
START[:decimal-value]	Indicates the start of a range of entries to be selected.
END[:decimal-value]	Indicates the end of a range of entries to be selected.

Usage Notes:

- You can specify one or both of these keywords. If you specify both keywords, you must enclose them in parentheses.
- If you specify /ENTRY without an entry range, the entry range defaults to START:1,END:end-of-file.
- If you specify the START or END keyword without a value, the keyword is ignored.

### **/EXCLUDE=event-class[,...]**

Specifies an event class or classes to be rejected. Do not use /EXCLUDE with /INCLUDE.

For *event-class*, specify one or more of the keywords shown in the following table. If you specify more than one keyword, use a comma-separated list of values enclosed in parentheses.

<b>Keyword</b>	<b>Description</b>
ATTENTIONS	Reject device attention entries.
BUGCHECKS	Reject all types of bugcheck entries.
CONFIGURATION	Reject system configuration entries.
CONTROL_ENTRIES	Reject control entries. Control entries include the following entry types: <ul style="list-style-type: none"> <li>• System power failure 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	Reject CPU-related entries. 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	Reject device error entries.
ENVIRONMENTAL_ENTRIES	Reject environmental entries.
MACHINE_CHECKS	Reject machine check entries.
MEMORY	Reject memory errors.
SNAPSHOT_ENTRIES	Reject snapshot entries.
SYNDROME	Reject firmware-generated entries that describe a symptom set used by HP support personnel to identify problems.
TIMEOUTS	Reject device timeout entries.
UNKNOWN_ENTRIES	Reject any entry that has an unknown entry class.
UNSOLICITED_MSCP	Reject unsolicited MSCP entries.
VOLUME_CHANGES	Reject volume mount and dismount entries.

**/FULL**

Specifies that ELV is to generate a full standard report. Do not use /FULL with /BRIEF or /ONE\_LINE.

**/INCLUDE=event-class[,...]**

Specifies an event class or classes to be selected. All other event classes are implicitly rejected. Do not use /INCLUDE with /EXCLUDE.

## ELV TRANSLATE

For *event-class*, specify one or more of the keywords shown in the following table. If you specify more than one keyword, use a comma-separated list of values enclosed in parentheses.

Keyword	Description
ATTENTIONS	Select device attention entries.
BUGCHECKS	Select all types of bugcheck entries.
CONFIGURATION	Select system configuration entries.
CONTROL_ENTRIES	Select control entries. Control entries include the following entry types: <ul style="list-style-type: none"><li>• System power failure 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	Select CPU-related entries. 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	Select device error entries.
ENVIRONMENTAL_ENTRIES	Select environmental entries.
MACHINE_CHECKS	Select machine check entries.
MEMORY	Select memory errors.
SNAPSHOT_ENTRIES	Select snapshot entries.
SYNDROME	Select firmware-generated entries that describe a symptom set used by HP support personnel to identify problems.
TIMEOUTS	Select device timeout entries.
UNKNOWN_ENTRIES	Select any entry that has an unknown entry class.
UNSOLICITED_MSCP	Select unsolicited MSCP entries.
VOLUME_CHANGES	Select volume mount and dismount entries.



**/INTERACTIVE  
/NOINTERACTIVE**

Specifies whether or not ELV is to run in interactive shell mode after you execute the ELV command. By default, interactive shell mode results from the way the current ELV command was entered.

For more information, refer to Section 10.2.

**/LOG  
/NOLOG**

Specifies whether or not ELV is to output control and informational messages to the terminal. The default /NOLOG does not output these messages to the terminal.

**/NODE[=node-name,...]  
/NONODE[=node-name,...]**

Specifies that events occurring on a node or nodes are to be selected or rejected.

If you enter /NODE without a value, only events that occur on the node on which you are running ELV are selected.

If you enter /NONODE without a value, events occurring on all nodes that are represented in the error log file are selected.

**/ONE\_LINE**

Specifies that ELV is to generate a one-line-per-event standard report. Do not use /ONE\_LINE with /BRIEF or /FULL.

**/OUTPUT[=output-file]**

Specifies the output file that is to contain bit-to-text translation reports.

By default, output is written to SYS\$OUTPUT. If you do not specify an output file name, the input file name is used. If you do not specify a device and directory, your current device and directory are used. If you do not specify a file type, the default file type is .LIS.

**/PAGE  
/NOPAGE**

Specifies whether or not to enable paged output of a report. The default /NOPAGE disables paged output of reports.

**/REJECTED**

Specifies that rejected (rather than selected) events are to be translated.

For more information, refer to Section 10.3.

**/SINCE[=date-time]**

Specifies that only those events dated later than the stated date and time are to be selected. If you omit a date and time, TODAY is used.

For *date-time*, you can specify an absolute time, a delta time, or a combination of absolute and delta times. See the *OpenVMS User's Manual* for details on specifying times.

**/SUMMARY  
/NOSUMMARY**

Specifies that a summary report or a standard report is to be generated:

- If you enter /SUMMARY, only a summary report is generated.

## ELV TRANSLATE

- If you enter `/NOSUMMARY`, only a standard report is generated.
- If you omit the qualifier altogether, the default is to display both a standard and a summary report.

### **/TERSE**

Specifies that the data in a standard report is to be displayed in a less interpreted format, regardless of detail level. `/TERSE` has no effect on a summary report.

See the `/TERSE` example in Section 10.5.5.

## Examples

1. `$ ANALYZE/ERROR_LOG/ELV TRANSLATE /ONE_LINE /PAGE`

The command in this example translates selected events from the default `SY$ERRORLOG:ERRLOG.SYS` file and writes the resulting standard and summary reports to the terminal in paged output. The standard report is a one-line-per-event report.

2. `ELV> TRANSLATE /BRIEF /NOSUMMARY`

The command in this example translates selected events from the default `SY$ERRORLOG:ERRLOG.SYS` file and writes the resulting standard report to the terminal. The standard report is a brief report.

3. `ELV> TRANSLATE /SINCE=TODAY /OUTPUT=REPORT.TXT`

The command in this example translates selected events from the default `SY$ERRORLOG:ERRLOG.SYS` file and writes the resulting standard report to an ASCII output file named `REPORT.TXT`.

The selected events are those that occurred since `TODAY`.

4. `$ ANALYZE/ERROR_LOG/ELV TRANSLATE /FULL /TERSE`

The command in this example translates selected events from the default `SY$ERRORLOG:ERRLOG.SYS` file and writes the resulting standard and summary reports to the terminal. The standard report is a full report in terse format.

5. `ELV> TRANSLATE /INCLUDE=VOLUME_CHANGES /SUMMARY`

The command in this example translates selected events from the default `SY$ERRORLOG:ERRLOG.SYS` file and writes the resulting summary report to the terminal.

The selected events include only the `VOLUME_CHANGES` event class.

---

## WRITE

Performs an image copy of events from one or more binary error log files to a single new binary error log file.

### Format

WRITE [input-file,...]

### Parameter

#### input-file

Supplies one or more names of binary error log files to be used to produce a new binary error log file.

If you do not specify an input file, the default input file is SYS\$ERRORLOG:ERRLOG.SYS. If you do not specify a device and directory, your current device and directory are used. If you do not specify a file name, the default file name is ERRLOG. If you do not specify a file type, the default file type is .SYS.

### Qualifiers

#### /BEFORE[=date-time]

Specifies that only those events dated earlier than the stated date and time are to be selected.

For *date-time*, you can specify an absolute time, a delta time, or a combination of absolute and delta times. See the *OpenVMS User's Manual* for details on specifying times.

If you omit a date and time, TODAY is used.

#### /ENTRY[=keyword,...]

Specifies the range of entries to be selected.

You can specify one or both of the following keywords:

Keyword	Description
START[:decimal-value]	Indicates the start of a range of entries to be selected.
END[:decimal-value]	Indicates the end of a range of entries to be selected.

#### Usage Notes:

- You can specify one or both of these keywords. If you specify both keywords, you must enclose them in parentheses.
- If you specify /ENTRY without an entry range, the entry range defaults to START:1,END:end-of-file.
- If you specify the START or END keyword without a value, the keyword is ignored.

## ELV WRITE

### **/EXCLUDE=event-class[,...]**

Specifies an event class or classes to be rejected. Do not use /EXCLUDE with /INCLUDE.

For *event-class*, specify one or more of the keywords shown in the following table. If you specify more than one keyword, use a comma-separated list of values enclosed in parentheses.

<b>Keyword</b>	<b>Description</b>
ATTENTIONS	Reject device attention entries.
BUGCHECKS	Reject all types of bugcheck entries.
CONFIGURATION	Reject system configuration entries.
CONTROL_ENTRIES	Reject control entries. Control entries include the following entry types: <ul style="list-style-type: none"><li>• System power failure 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	Reject CPU-related entries. 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	Reject device error entries.
ENVIRONMENTAL_ENTRIES	Reject environmental entries.
MACHINE_CHECKS	Reject machine check entries.
MEMORY	Reject memory errors.
SNAPSHOT_ENTRIES	Reject snapshot entries.
SYNDROME	Reject firmware-generated entries that describe a symptom set used by HP support personnel to identify problems.
TIMEOUTS	Reject device timeout entries.

Keyword	Description
UNKNOWN_ENTRIES	Reject any entry that has an unknown entry class.
UNSOLICITED_MSCP	Reject unsolicited MSCP entries.
VOLUME_CHANGES	Reject volume mount and dismount entries.

**/INCLUDE=event-class[,...]**

Specifies an event class or classes to be selected. All other event classes are implicitly rejected. Do not use /INCLUDE with /EXCLUDE.

For *event-class*, specify one or more of the keywords shown in the following table. If you specify more than one keyword, use a comma-separated list of values enclosed in parentheses.

Keyword	Description
ATTENTIONS	Select device attention entries.
BUGCHECKS	Select all types of bugcheck entries.
CONFIGURATION	Select system configuration entries.
CONTROL_ENTRIES	Select control entries. Control entries include the following entry types: <ul style="list-style-type: none"> <li>• System power failure 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	Select CPU-related entries. 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	Select device error entries.
ENVIRONMENTAL_ENTRIES	Select environmental entries.
MACHINE_CHECKS	Select machine check entries.
MEMORY	Select memory errors.
SNAPSHOT_ENTRIES	Select snapshot entries.

## ELV WRITE

Keyword	Description
SYNDROME	Select firmware-generated entries that describe a symptom set used by HP support personnel to identify problems.
TIMEOUTS	Select device timeout entries.
UNKNOWN_ENTRIES	Select any entry that has an unknown entry class.
UNSOLICITED_MSCP	Select unsolicited MSCP entries.
VOLUME_CHANGES	Select volume mount and dismount entries.

### **/INTERACTIVE**

### **/NOINTERACTIVE**

Specifies whether or not ELV is to run in interactive shell mode after you execute the ELV command. By default, interactive shell mode results from the way the current ELV command was entered.

For more information, refer to Section 10.2.

### **/LOG**

### **/NOLOG**

Specifies whether or not ELV is to output control and informational messages to the terminal. The default **/NOLOG** does not output these messages to the terminal.

### **/NODE[=node-name,...]**

### **/NONODE[=node-name,...]**

Specifies that events occurring on a node or nodes are to be selected or rejected.

If you **/NODE** without a value, only events that occur on the node on which you are running ELV are selected.

If you enter **/NONODE** without a value, events occurring on all nodes that are represented in the error log file are selected.

### **/OUTPUT[=output-file]**

Specifies the output file that is to contain image copies of events.

If you do not specify an output file name, the input file name is used. If you do not specify a device and directory, your current device and directory are used. If you do not specify a file type, the default file type is **.DAT**.

### **/REJECTED**

Specifies that rejected (rather than selected) events are to be copied to a new binary error log file.

For more information, refer to Section 10.3.

### **/SINCE[=date-time]**

Specifies that only those events dated later than the stated date and time are to be selected.

For *date-time*, you can specify an absolute time, a delta time, or a combination of absolute and delta times. See the *OpenVMS User's Manual* for details on specifying times.

If you omit a date and time, **TODAY** is used.

## Examples

1. ELV> WRITE /EXCLUDE=CONTROL\_ENTRIES /OUTPUT=NEW\_ERROR\_LOG.SYS

The command in this example copies selected events from the default SYS\$ERRORLOG:ERRLOG.SYS file to a new error log file named NEW\_ERROR\_LOG.SYS.

The selected events do not include the CONTROL\_ENTRIES event class.

2. ELV> WRITE /NONODE=PANDA /SINCE="20-AUG-2003 04:51:33.87"

The command in this example copies selected events from the default SYS\$ERRORLOG:ERRLOG.SYS file to a new error log file named ERRLOG.DAT.

The selected events are those that occurred since the specified date on all nodes except PANDA.

3. \$ ANALYZE/ERROR\_LOG/ELV WRITE /ENTRY=(START:5,END:10)

The command in this example copies selected events from the default SYS\$ERRORLOG:ERRLOG.SYS file to a new error log file named ERRLOG.DAT.

The selected events start with entry 5 and end with entry 10.

4. \$ ANALYZE/ERROR\_LOG/ELV WRITE /ENTRY=(START:5,END:10) /REJECTED

This example is identical to the previous example except that rejected events are copied rather than selected events.

The rejected events are those that fall outside the specified entry range.

# Error Log Viewer Utility (ELV)

## 10.5 ELV Sample Reports

### 10.5 ELV Sample Reports

You can use the following TRANSLATE qualifiers to produce variations of the standard report:

- /ONE\_LINE
- /BRIEF
- /FULL
- /TERSE

See Table 10–1 for more information about report formats.

The following sections contain examples of the reports that result from using these qualifiers. In addition, an example is shown of using no qualifiers to produce a standard report of the default detail level.

#### 10.5.1 /ONE\_LINE Example

The report produced when you use the /ONE\_LINE qualifier includes only the header information for each event.

The example that follows contains multiple summary reports, one for each node represented in this error log file, and a summary report for the entire file.

Output for SYS\$COMMON:[SYSEXE.ERRLOGS]EXAMPLE.DAT;2

EVENT	EVENT_TYPE	TIMESTAMP	NODE	EVENT_CLASS
1	Volume Mount	14-AUG-2003 13:31:39.12	FRANZ	VOLUME_CHANGES
2	Volume Mount	8-FEB-2001 10:15:16.12	WF35P0	VOLUME_CHANGES
3	Volume Dismount	8-FEB-2001 10:15:16.15	WF35P0	VOLUME_CHANGES
4	Volume Mount	8-FEB-2001 10:15:16.28	WF35P0	VOLUME_CHANGES

#### ERROR\_LOG\_SUMMARY

```
Summary for node:                FRANZ
Total number of events:          1
Number of the first event:       1
Number of the last event:        1
Earliest event occurred:         14-AUG-2003 13:31:39.12
Latest event occurred:           14-AUG-2003 13:31:39.12
Number of events by event class:
    VOLUME_CHANGES                1

Summary for node:                WF35P0
Total number of events:          3
Number of the first event:       2
Number of the last event:        4
Earliest event occurred:         8-FEB-2001 10:15:16.12
Latest event occurred:           8-FEB-2001 10:15:16.28
Number of events by event class:
    VOLUME_CHANGES                3

Summary for node:                <ALL NODES>
Total number of events:          4
Number of the first event:       1
Number of the last event:        4
Earliest event occurred:         8-FEB-2001 10:15:16.12
Latest event occurred:           14-AUG-2003 13:31:39.12
Number of events by event class:
    VOLUME_CHANGES                4
```



# Error Log Viewer Utility (ELV)

## 10.5 ELV Sample Reports

### 10.5.2 /BRIEF Example

The report produced when you use the /BRIEF qualifier includes only the most essential event information along with the header information for each event.

Output for SYS\$COMMON:[SYSEXE.ERRLOGS]EXAMPLE.DAT;1

EVENT	EVENT_TYPE	TIMESTAMP	NODE	EVENT_CLASS
1	Volume Mount	14-AUG-2003 13:31:39.12	FRANZ	VOLUME_CHANGES
DESCRIPTION				
	Logging CPU	RANGE	VALUE	TRANSLATED_VALUE
	Number of CPU's in Active Set		3	
	System Marketing Model		4	
	Error Mask	<31:00>	1968	COMPAQ AlphaServer GS160
	Operating System Version		0x00000003	
			X9WY-SSB	
	Device Unit Number		200	
	Device Generic Name		FRANZ\$DKB	
	Volume Label		OPAL_X9WY	

#### ERROR\_LOG\_SUMMARY

Total number of events:	1
Number of the first event:	1
Number of the last event:	1
Earliest event occurred:	14-AUG-2003 13:31:39.12
Latest event occurred:	14-AUG-2003 13:31:39.12
Number of events by event class:	
VOLUME_CHANGES	1

### 10.5.3 No Qualifiers Example

The report produced when you do not use any qualifiers includes only the most commonly useful event information along with the header information for each event.

Output for SYS\$COMMON:[SYSEXE.ERRLOGS]EXAMPLE.DAT;1

EVENT	EVENT_TYPE	TIMESTAMP	NODE	EVENT_CLASS
1	Volume Mount	14-AUG-2003 13:31:39.12	FRANZ	VOLUME_CHANGES
DESCRIPTION				
	Hardware Architecture	RANGE	VALUE	TRANSLATED_VALUE
	Hardware System Type		4	Alpha
	Logging CPU		35	Wildfire
	Number of CPU's in Active Set		3	
	System Marketing Model		4	
	Error Mask	<31:00>	1968	COMPAQ AlphaServer GS160
	Seconds Since Boot		0x00000003	
	Error Sequence Number		17	
	DSR String		46	
	Operating System Version		AlphaServer GS160 6/731	
			X9WY-SSB	
	Owner UIC of the Volume		65537	
	Unit Operation Count		378	
	Device Unit Number		200	
	Device Generic Name		FRANZ\$DKB	
	Volume Number within Set		0	
	Number of Volumes within Set		0	
	Volume Label		OPAL_X9WY	

#### ERROR\_LOG\_SUMMARY

Total number of events:	1
Number of the first event:	1
Number of the last event:	1
Earliest event occurred:	14-AUG-2003 13:31:39.12
Latest event occurred:	14-AUG-2003 13:31:39.12
Number of events by event class:	
VOLUME_CHANGES	1



## Error Log Viewer Utility (ELV) 10.5 ELV Sample Reports

Output for SYS\$COMMON:[SYSEXE.ERRLOGS]EXAMPLE.DAT;1

EVENT	EVENT_TYPE	TIMESTAMP	NODE	EVENT_CLASS
1	Volume Mount	14-AUG-2003 13:31:39.12	FRANZ	VOLUME_CHANGES
DESCRIPTION		RANGE	VALUE	
Operating System Type			0x0002	
Hardware Architecture			0x0004	
Vendor ID			0x00000DEC	
Hardware System Type			0x0000000000000023	
Logging CPU			0x00000003	
Number of CPU's in Active Set			0x00000004	
Device Class			0x0000	
System Marketing Model			0x000007B0	
Device Type			0x0000	
OS Flags		<15:00>	0x0000	
Error Mask		<31:00>	0x00000003	
Seconds Since Boot			0x00000011	
Chip Type			0x0000000B	
Error Sequence Number			0x0000002E	
DSR String			265767265536168706C41 372F36203036315347207 000000000000000003133 0x0	
DDR String			00000000000000000000 00000000000000000000 00000000000000000000 0x0	
System Serial Number			00000000000000000000 0x00000000000000000000	
Time - ISO 8601 Format			333313431383033303032 3034302D32312C3933313 0x30	
Operating System Version			0x4253532D59573958	
Computer Name			000002020205A4E415246 0x000000000000	
Owner UIC of the Volume			0x00010001	
Unit Error Count			0x00000000	
Unit Operation Count			0x0000017A	
Device Unit Number			0x00C8	
Device Generic Name			000424B44245A4E415246 00000000000000000000 00000000000000000000 0x	
Volume Number within Set			0x0000	
Number of Volumes within Set			0x0000	
Volume Label			020595739585F4C41504F 0x202	

ERROR\_LOG\_SUMMARY

Total number of events:	1
Number of the first event:	1
Number of the last event:	1
Earliest event occurred:	14-AUG-2003 13:31:39.12
Latest event occurred:	14-AUG-2003 13:31:39.12
Number of events by event class:	
VOLUME_CHANGES	1







## 11.1 INSTALL Description

The Install utility (INSTALL) creates **known file entries** to improve the performance of executable and shareable images, especially those that run frequently, run concurrently with several processes, or allow images to run in a privileged context.

Known file entries last only while the system is operating. If the system is shut down or fails for any reason, you must reinstall all known images after the system is rebooted. For this reason, you are encouraged to include additional INSTALL commands for selected images in the site-specific command procedure SYS\$MANAGER:SYSTARTUP\_VMS.COM.

## 11.2 INSTALL Usage Summary

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.

### Description

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.

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.

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
```

## 11.3 INSTALL Qualifiers

This section describes the /TRANSLATE qualifier and provides examples of its use.

# INSTALL /TRANSLATE

---

## /TRANSLATE

The /TRANSLATE qualifier forces INSTALL to attempt a logical name translation of a file that you specify.

---

### Note

Before you use the /TRANSLATE qualifier, you must invoke INSTALL as a foreign command:

```
$ INSTALL = "$INSTALL"
```

---

## Format

/TRANSLATE

## Description

Within OpenVMS, when you specify a file name without a device, directory, or file type, OpenVMS usually attempts a logical name translation of the file name before it applies device, directory, or file type defaults. However, if you specify a device, directory, or file type, OpenVMS does *not* attempt a logical name translation.

Using the /TRANSLATE qualifier forces INSTALL to attempt a logical name translation even if a device, directory, or file type is specified.

## Examples

1. DEFINE FILE1 FILE1\_EV6  
\$ INSTALL = "\$INSTALL"  
\$ INSTALL  
INSTALL> ADD SYS\$SHARE:FILE1.EXE ! SYS\$SHARE:FILE1 is added as a known image

The first command in this example defines FILE1 as a logical name with an equivalence name of FILE1\_EV6. However, because a device (SYS\$SHARE:) and file type (.EXE) are supplied, INSTALL treats FILE1 as part of a file specification rather than as a logical name.

2. DEFINE FILE1 FILE1\_EV6  
\$ INSTALL = "\$INSTALL"  
\$ INSTALL/TRANSLATE  
INSTALL> ADD SYS\$SHARE:FILE1.EXE ! SYS\$SHARE:FILE1\_EV6 is added as a known image

The first command in this example defines FILE1 as a logical name with an equivalence name of FILE1\_EV6. Because /TRANSLATE is specified, INSTALL treats FILE1 as a logical name even though a device (SYS\$SHARE:) and file type (.EXE) are supplied.



---

## 11.4 INSTALL Commands

This section describes and provides examples of the `INSTALL` commands. The following table summarizes the `INSTALL` command functions:

---

<b>Command</b>	<b>Function</b>
<code>ADD</code>	Synonym for the <code>CREATE</code> command
<code>CREATE</code>	Installs the specified image as a known image
<code>DELETE</code>	Synonym for the <code>REMOVE</code> command
<code>EXIT</code>	Exits from <code>INSTALL</code>
<code>HELP</code>	Describes how to use <code>INSTALL</code>
<code>LIST</code>	Displays a description of each specified known image, global sections, and the addresses of known image data structures
<code>PURGE</code>	Deletes all known images installed without the <code>/NOPURGE</code> qualifier
<code>REMOVE</code>	Deletes a known image
<code>REPLACE</code>	Associates a known image with the latest version of the image file or modifies the attributes of an installed image

---

## **INSTALL ADD**

---

### **ADD**

Installs the specified image file as a known image. The ADD command is a synonym for the CREATE command.

Requires the CMKRNL privilege. Also requires the SYSGBL privilege to create system global sections and the PRMGBL privilege to create permanent global sections.

---

## CREATE

Installs the specified image file as a known image. The CREATE command is a synonym for the ADD command.

Requires the CMKRNL privilege. Also requires the SYSGBL privilege to create system global sections and the PRMGBL privilege to create permanent global sections.

### Format

CREATE 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, which must have been linked with the /NOTRACEBACK qualifier. 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

#### **/ACCOUNTING**

#### **/NOACCOUNTING (default)**

Enables image-level accounting for the specified image even if image accounting is disabled (by using the DCL command SET ACCOUNTING/DISABLE=IMAGE). When image accounting is enabled on the local node, it logs all images, and the /NOACCOUNTING qualifier has no effect.

#### **/ARB\_SUPPORT=keyword**

On Alpha and I64 systems, overrides the system parameter ARB\_SUPPORT for this installed image.

The following table shows the keywords you can use with the /ARB\_SUPPORT qualifier:

<b>Keyword</b>	<b>Behavior</b>
None	The obsolete kernel data cells are not maintained by the system. Fields are initialized to zero or set to invalid pointers at process creation.
Clear	The obsolete kernel data cells are cleared or set to invalid pointers when the code would have set up values for backward compatibility.
Read-only	The obsolete cells are updated with corresponding security information stored in the current Persona Security Block (PSB) when a \$PERSONA_ASSUME is issued.

## INSTALL CREATE

Keyword	Behavior
Full (default)	Data is moved from the obsolete cells to the currently active PSB on any security-based operation.

For more information about obsolete kernel cells, refer to the `ARB_SUPPORT` system parameter in an appendix to this manual or in online help.

**/AUTHPRIVILEGES[=(priv-name[,...])]**

**/NOAUTHPRIVILEGES**

Installs the file as a known image installed with the authorized privileges specified.

### Usage Notes

- If a privileged image is not located on the system volume, the image is implicitly installed `/OPEN`.
- The set of privileges for a privileged image can be empty. You must, however, list each privilege every time you define or redefine privileges.
- The `/AUTHPRIVILEGES` qualifier applies only to executable images.
- You cannot specify this qualifier for an executable image linked with the `/TRACEBACK` qualifier.
- You cannot assign privilege names with the `/NOAUTHPRIVILEGES` qualifier.

You can specify one or more of the privilege names described in detail in an appendix to the *HP OpenVMS Guide to System Security*. (`ALL` is the default.)

**/EXECUTE\_ONLY**

**/NOEXECUTE\_ONLY (default)**

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 no read access. All shareable images referenced by the program must be installed, and OpenVMS RMS uses trusted logical names (those created for use in executive or kernel mode).

You cannot specify this qualifier for an executable image linked with the `/TRACEBACK` qualifier.

**/HEADER\_RESIDENT**

**/NOHEADER\_RESIDENT**

Installs the file as a known image with a permanently resident header (native mode images only). An image installed header resident is implicitly installed open.

**/LOG**

**/NOLOG (default)**

Lists the newly created known file entry along with any associated global sections created by the installation.

**/OPEN**

**/NOOPEN**

Installs the file as a permanently open known image.

**/PRIVILEGED[=(priv-name[,...])]  
/NOPRIVILEGED**

Installs the file as a known image with active privileges specified. If a privileged image is not located on the system volume, the image is implicitly installed /OPEN.

#### **Usage Notes**

- The set of privileges for a privileged image can be empty.
- You must list each privilege every time you define or redefine privileges.
- The /PRIVILEGED qualifier applies only to executable images.
- You cannot specify this qualifier for an executable image linked with the /TRACEBACK qualifier.
- You cannot assign privilege names with the /NOPRIVILEGED qualifier.

#### **Installing Shareable Images**

Installing an image with privileges declares that the image is trusted to maintain system integrity and security properly. To maintain that trust, any routine called by the privileged image must also be trusted. For this reason, any shareable images activated for use by a privileged image must be installed. Only trusted logical names (names defined in executive and kernel mode) can be used in locating shareable images to be used by a privileged image.

#### **Interaction of /PRIVILEGED and /AUTHPRIVILEGES**

When you create a new entry, the privileges you assign are also assigned for Authorized Privileges if you do not assign specific authorized privileges with the /AUTHPRIVILEGED qualifier.

When you replace an image, any privileges assigned with the /PRIVILEGED qualifier are *not* repeated as Authorized Privileges. Also, if you use the REPLACE command with the /NOAUTHPRIVILEGES qualifier, the Authorized Privileges become the same as the Default Privileges (set using the /PRIVILEGED qualifier).

You can specify one or more of the privilege names described in detail in an appendix to the *HP OpenVMS Guide to System Security*. (ALL is the default.)

For examples of how to use CREATE commands with /PRIVILEGES qualifiers, see the Examples section at the end of this command.

**/PROTECTED  
/NOPROTECTED (default)**

Installs the file as a known image that is protected from user-mode and supervisor-mode write access. You can write into the image only from executive or kernel mode. The /PROTECTED qualifier together with the /SHARE qualifier are used to implement user-written services, which become privileged shareable images.

**/PURGE (default)  
/NOPURGE**

Specifies that the image can be removed by a purge operation; if you specify /NOPURGE, you can remove the image only by a remove operation.

## INSTALL CREATE

### **/RESIDENT=[([NO]CODE,[NO]DATA)]**

On Alpha and I64 systems, causes image code sections or read-only data sections to be placed in the granularity hint regions and compresses other image sections, which remain located in process space. If you do not specify the /RESIDENT qualifier, neither code nor data is installed resident. If you specify the /RESIDENT qualifier without keyword arguments, code is installed resident, and data is not installed resident.

The image must be linked using the /SECTION\_BINDING=(CODE,DATA) qualifier. An image installed with resident code or data is implicitly installed header resident and shared.

### **/SHARED=[([NO]ADDRESS\_DATA)**

#### **/NOSHARED**

Installs the file as a shared known image and creates global sections for the image sections that can be shared. An image installed shared is implicitly installed open.

When you use the ADDRESS\_DATA keyword with the /SHARED qualifier, P1 space addresses are assigned for shareable images. With the assigned addresses, the Install utility can determine the content of an address data section when the image is installed rather than when it is activated, reducing CPU and I/O time. A global section is created to allow shared access to address data image sections.

### **/SIGNAL**

#### **/NOSIGNAL**

Installs the file as an image capable of accepting signals (or exceptions) from other processes. This allows privileged images to be signaled.

### **/WRITABLE=[GALAXY=[IDENT]]**

#### **/NOWRITABLE**

Installs the file as a writable known image when you also specify the /SHARED qualifier. The /WRITABLE qualifier applies only to images with image sections that are shareable and writable. The /WRITABLE qualifier is automatically negated if the /NOSHARED qualifier is specified.

You can use the GALAXY keyword with the /WRITABLE qualifier to place write shared image sections in Galaxy global sections. You can also use the IDENT keyword with GALAXY to include the image ident in the name of the Galaxy global section, so that multiple versions of an image can be used simultaneously in a Galaxy system.

## Examples

1. `INSTALL> CREATE/OPEN/SHARED WRKD$:[MAIN]STATSHR`

The command in this example installs the image file STATSHR as a permanently open shared known image.

2. `INSTALL> CREATE/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.

3. `INSTALL> CREATE/LOG GRPCOMM`

The command in this example installs the image file GRPCOMM as a known image and then displays the newly created known file entry.

4. `INSTALL> CREATE/SHARED=ADDRESS_DATA WRKD$: [MAIN]INFOSHR`

The command in this example installs the INFOSHR file as a shared known image and creates shared global sections for code sections and read-only data sections. Because the command includes the ADDRESS\_DATA keyword, address data is also created as a shared global section.

5. `INSTALL> CREATE STATSHR/PRIV`

The command in this example creates the STATSHR image with all privileges.

6. `INSTALL> CREATE STATSHR/PRIV=(OPER,SYSPRV)`

The command in this example creates the STATSHR image with the OPER and SYSPRV privileges.

7. `INSTALL> CREATE STATSHR/PRIV=NOALL`

The command in this example creates the STATSHR image with an empty set of privileges.

8. `INSTALL> CREATE STATSHR/NOPRIV`

The command in this example creates the STATSHR image explicitly with no privileges.

**INSTALL  
DELETE**

---

**DELETE**

Deletes a known image. The DELETE command is a synonym for the REMOVE command.



---

**EXIT**

Terminates **INSTALL** and returns control to the DCL command level. You can also exit from **INSTALL** by pressing Ctrl/Z.

**Format**

EXIT

# INSTALL HELP

---

## 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.

## Examples

1. INSTALL> HELP

The command in this example displays a list of INSTALL topics and a Topic? prompt. Any topic from the list can be entered at the prompt.

2. INSTALL> HELP LIST

LIST

For display of a one-line description of the specified known image, or if no file is specified, then for all known images. Use with /FULL to obtain a multiline description.

Format: LIST [file-spec] /qualifier

QUALIFIER	COMBINATION	BEHAVIOR
LIST	[file-spec]	List the known image for file-spec
LIST		List all entries

Additional information available:

Qualifiers  
/FULL      /GLOBAL      /STRUCTURE      /SUMMARY

The command in this example displays help information about the LIST command.

---

## **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 images.

### **Description**

You can use the LIST command with the /FULL qualifier to display information that is useful in tuning the known image 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. For a description of global sections and global pages, including information about how to estimate the size of installed images, refer to the *HP OpenVMS System Services Reference Manual*.

### **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 about global sections, plus the current owner and protection codes and access control entries.

#### **/GLOBAL**

Lists global sections for any specified shared image, or if you omit the file specification, lists all global sections. If a global section is created by INSTALL to support a particular image, that image is also identified.

#### **/RESIDENT**

Displays a description of each resident image.

#### **/STRUCTURE**

Lists addresses of known image data structures.

#### **/SUMMARY**

Used with the /GLOBAL qualifier, displays the global section and global page usage on the system for local and shared memory global sections.

# INSTALL LIST

## Examples

### 1. INSTALL> LIST

The command in this example displays a single-line description of all known images. The description includes the file specification of the known image and its attributes.

```

SYS$DISK:<SYS0.SYSCOMMON.SYSEXE>.EXE
  ANALIMDMP;1 ①          Prv ②
  AUTHORIZE;1          Prv
  CDU;1              Open Hdr Prv
  DCL;1              Open Hdr Shar      Lnkbl
  FAL;1              Open Hdr Shar
  INSTALL;1          Prv
  LOGINOUT;1         Open Hdr Shar Prv
  MAIL;1             Open Hdr Shar
  MAIL_SERVER;1     Open Hdr Shar Prv
  REQUEST;1         Prv
  SET;1             Open Hdr Shar Prv
  SETAUDIT;1        Prv
  SETP0;1           Open Hdr Shar Prv
  SETRIGHTS;1       Prv
  SHOW;1            Open Hdr Shar Prv
  SHWCLSTR;1        Open Hdr Shar Prv
  SUBMIT;1          Open Hdr Shar Prv
  SYSMAN;1          Prv

SYS$DISK:<SYS0.SYSCOMMON.SYSLIB>.EXE
  ANALIMDMPSHR;1     Prv
  CONVSHR;1
  DCLTABLES;1       Open Hdr Shar      Lnkbl
  .
  .
  .
  LIBOTS;1          Open Hdr Shar      Lnkbl      Resid
  LIBRTL;1          Open Hdr Shar      Lnkbl      Resid
  MAILSHR;1         Open Hdr Shar      Lnkbl
  .
  .
  .

```

- ① File specification of the known image
- ② Attribute of known image, as follows:

Attribute	Meaning
ACNT	Image accounting is enabled for the image (/ACCOUNTING).
†CMODE	Image is in compatibility mode, set by the VAX-11 RSX Linker.
HDR	Image header is permanently resident (/HEADER_RESIDENT).
LNKBL	Image is not executable; it is a shareable (linkable) image set by the OpenVMS Linker.

†VAX specific

Attribute	Meaning
NOPURG	Image cannot be removed by a purge operation; it can only be removed by a delete or remove operation (/NOPURGE).
OPEN	Image is permanently open (/OPEN).
PROT	Image contains protected code (/PROTECTED).
PRV	Image has privileges (/PRIVILEGED).
‡RESID	Image is resident (/RESIDENT).
SAFE	Image is a privileged image registered as being compatible with the running version of OpenVMS. For more information about registered images, refer to the <i>HP OpenVMS System Manager's Manual, Volume 1: Essentials</i> .
SHAR	Image is shared (/SHARED).
WRT	Image is writable (/WRITABLE).
XONLY	Only execute access to image is allowed (/EXECUTE_ONLY).

---

‡Alpha and I64 specific

---

2. INSTALL> LIST/RESIDENT

The command in this Alpha example displays a single-line description of all resident images. The description includes the location in memory, the size of the code sections, and the type of section.

System Resident Sections

SYS\$DISK:<SYS0.SYSCOMMON.SYSLIB>.EXE  
CMA\$TIS\_SHR;1

Base VA	End VA	Length	Type
80490000	80490A00	00000A00	Resident Code
7FC04000	7FC04A00	00000A00	Linkage
7FC14000	7FC14200	00000200	Writeable data
7FC34000	7FC34200	00000200	Writeable data

DECC\$SHR;1

Base VA	End VA	Length	Type
80548000	805D1C00	00089C00	Resident Code
805D2000	805D2400	00000400	Resident Code
7FE34000	7FE4B800	00017800	Linkage
7FE54000	7FE59A00	00005A00	Writeable data
7FE64000	7FE64800	00000800	Read-only data
7FE84000	7FE84200	00000200	Writeable data
7FE94000	7FE94200	00000200	Demand-zero
7FEA4000	7FEA7000	00003000	Demand-zero
7FEB4000	7FEB5800	00001800	Writeable data

DPML\$SHR;1

Base VA	End VA	Length	Type
80492000	80547600	000B5600	Resident Code

## INSTALL LIST

```

7FC44000 7FC6FA00 0002BA00 Read-only data
7FC74000 7FC88200 00014200 Linkage
7FC94000 7FC94400 00000400 Writeable data
7FCA4000 7FCCE600 0002A600 Read-only data
7FD94000 7FD95000 00001000 Writeable data

```

LIBOTS;1

Base VA	End VA	Length	Type
80482000	8048F600	0000D600	Resident Code
7FBC4000	7FBC6600	00002600	Read-only data
7FBD4000	7FBD5A00	00001A00	Linkage
7FBF4000	7FBF4200	00000200	Writeable data

LIBRTL;1

Base VA	End VA	Length	Type
80400000	80481A00	00081A00	Resident Code
7FB54000	7FB64800	00010800	Linkage
7FB74000	7FB75000	00001000	Writeable data
7FB84000	7FB8D600	00009600	Read-only data
7FB94000	7FB94200	00000200	Writeable data
7FBA4000	7FBA5000	00001000	Demand-zero
7FBB4000	7FBB5400	00001400	Writeable data

### 3. INSTALL> LIST/FULL LOGINOUT

The command in this example displays a multiline description of the known image LOGINOUT.

```
SYSDISK:<SYS0.SYSCOMMON.SYSEXEXE>.EXE
```

```

LOGINOUT;3      Open Hdr  Shar Priv
Entry access count = 44 ①
Current / Maximum shared = 3 / 5 ②
Global section count = 2 ③
Privileges = CMKRNLSYSNAMTMPMBXEXQUOTASYSPRV ④

```

- ① Number of times known file entry has been accessed by this node since it was installed.
- ② 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.

### 4. INSTALL> LIST/GLOBAL

The command in this example displays all global sections for shared images. Global sections created by INSTALL (prefix INS\$) for a specific image are listed following the name of that image.

## System Global Sections ❶

```

DSA1000:<SYS2.SYSCOMMON.SYSLIB>SL$USSSHR.EXE ❷
INS$86D9BFBO_003 ❸ (14000001) ❹ PRM SYS ❺ Pagcnt/Refcnt=1/1 ❻
INS$86D9BFBO_002 (14000001) PRM SYS Pagcnt/Refcnt=3/3
INS$86D9BFBO_001 (14000001) PRM SYS Pagcnt/Refcnt=1/1
.
.
.
RMS$87A63B00 (00000000) WRT DZRO TMP SYS Pagcnt/Refcnt=74/74
.
.
.
671 Global Sections Used, 102138/22862 Global Pages Used/Unused ❼

```

- ❶ Display of global sections in memory.
- ❷ Name of the image for which the following global sections were created by INSTALL.
- ❸ Name of global section. The prefix identifies the creator of the section; for example, INS means the global section was created by INSTALL. The number includes the address of the section.
- ❹ Version number (in hexadecimal) of global section; for shareable images only, the high-order byte (01 in CRFSHR\_003) contains major identification, and low-order bytes (0003E8 in CRFSHR\_003) contain minor identification determined by the programmer at link time. For executable images, the number is a known unique value determined by the system.
- ❺ Attributes of the global section:
  - DZRO Global section is demand-zero.
  - GRP Along with a group number indicates a groupwide section, which would be created by a program other than INSTALL.
  - PRM Global section is permanent.
  - SYS Global section is systemwide.
  - TMP Indicates a temporary global section, which would be created by a program other than INSTALL.
  - WRT Global section is writable.
- ❻ Number of pages (VAX) or pagelets (Alpha and I64) in the section and number of page table entries currently mapped to this global section. For a more detailed discussion of mapping global sections, refer to the *HP OpenVMS Programming Concepts Manual*.
- ❼ Number of global sections created, number of global pages used, and number of global pages unused in local memory. Note that, because of arithmetic rounding, the number of global sections created will sometimes be greater than the SYSGEN parameter GBLSECTIONS. When the size of the system header is being computed, the values of the GBLSECTIONS and SYSMWCNT parameters are combined with the size of the fixed part of the process header. The result is rounded up to the next page boundary. This rounding process sometimes adds space to the global section table, depending on the values of the two SYSGEN parameters and the amount of system paging that preceded the running of INSTALL to create all of the global sections.

# INSTALL LIST

5. INSTALL> LIST/GLOBAL/FULL

The command in this example displays a complete listing of global sections for shared images. The /FULL qualifier adds owner and protection codes to the display.

```
System Global Sections
NM_MAILSHR_003 (741A6919)      PRM SYS      Pagcnt/Refcnt=10/0
NM_MAILSHR_002 (741A6919)      PRM SYS      Pagcnt/Refcnt=1/0
NM_MAILSHR_001 (741A6919)      PRM SYS      Pagcnt/Refcnt=11/0
                                Owner:        [1,4] ❶
                                Protection:     S:RWED,O:RWED,G:RWED,W:RE ❷
```

.  
.  
.

❶ UIC of the owner of the global section

❷ Type of access allowed for the image



---

**PURGE**

Deletes all known file entries for images installed without the /NOPURGE qualifier.

Requires the CMKRNL privilege. Also requires the SYSGBL privilege to create system global sections and the PRMGBL privilege to create permanent global sections.

**Format**

PURGE

**Parameters**

None.

**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.

**Example**

```
INSTALL> PURGE
```

The command in this example deletes all images except those installed with the /NOPURGE qualifier. The image files remain unaffected. Writable global sections are written back to disk upon their removal as known images.

# INSTALL REMOVE

---

## REMOVE

Deletes a known image. The REMOVE command is identical to the DELETE command.

Requires the CMKRNL privilege. Also requires the SYSGBL privilege to create system global sections and the PRMGBL privilege to create permanent global sections.

### Format

```
REMOVE file-spec
```

### Parameter

**file-spec**

Names the file specification of a known image.

### Description

The REMOVE command deletes an entry from the known file list. The image's entry on the known file list and any global sections created for the image are deleted. 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 REMOVE 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> REMOVE GRPCOMM
```

The command in this example deletes the entry for the known image GRPCOMM from the known image file list.

---

## REPLACE

Replaces a known image entry with another version of the image, or with modified attributes. The REPLACE command is a synonym for the DELETE command.

Requires the CMKRNL privilege. Also requires the SYSGBL privilege to create system global sections and the PRMGBL privilege to create permanent global sections.

### Format

REPLACE file-spec

### Parameter

**file-spec**

Names the file specification of an image installed as a known image.

### Description

The REPLACE command updates a known file to the latest, or to a specified version found in the specified directory, or in another directory if the *file-spec* parameter uses a search list.

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 CREATE (or ADD) command. If you specify no qualifiers, the new image retains the same attributes as the old one. If the REPLACE command modifies neither the installed image file nor its attributes, the REPLACE command allows continued sharing of global sections.

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.

### Qualifiers

**/ACCOUNTING****/NOACCOUNTING (default)**

Enables image-level accounting for selected images even if image accounting is disabled on the local node (by using the DCL command SET ACCOUNTING/DISABLE=IMAGE). When image accounting is enabled on the local node, it logs all images, and the /NOACCOUNTING qualifier has no effect.

**/ARB\_SUPPORT=keyword**

On Alpha and I64 systems, overrides the system parameter ARB\_SUPPORT for this installed image.

The following table shows the keywords you can use with the /ARB\_SUPPORT qualifier:

## INSTALL REPLACE

Keyword	Behavior
None	The obsolete kernel data cells are not maintained by the system. Fields are initialized to zero (or set to invalid pointers) at process creation.
Clear	The obsolete kernel data cells are cleared (or set to invalid pointers) when the code would have set up values for backward compatibility.
Read-only	The obsolete cells are updated with corresponding security information stored in the current Persona Security Block (PSB) when a \$PERSONA_ASSUME is issued.
Full (default)	Data is moved from the obsolete cells to the currently active PSB on any security-based operation.

For more information about obsolete kernel cells, refer to the `ARB_SUPPORT` system parameter in an appendix to this manual or in online help.

**/AUTHPRIVILEGES[=(priv-name[,...])]**

**/NOAUTHPRIVILEGES**

Installs the file as a known image installed with the authorized privileges specified.

### Usage Notes

- If a privileged image is not located on the system volume, the image is implicitly installed `/OPEN`.
- The set of privileges for a privileged image can be empty. You must, however, list each privilege every time you define or redefine privileges.
- The `/AUTHPRIVILEGES` qualifier applies only to executable images.
- You cannot specify this qualifier for an executable image linked with the `/TRACEBACK` qualifier.
- You cannot assign privilege names with the `/NOAUTHPRIVILEGES` qualifier.

### Interaction of `/PRIVILEGED` and `/AUTHPRIVILEGES` Qualifiers

When you create a new entry, the privileges you assign are also assigned for Authorized Privileges if you do not assign specific authorized privileges with the `/AUTHPRIVILEGES` qualifier.

When you replace an image, any privileges assigned with the `/PRIVILEGED` qualifier are *not* repeated as Authorized Privileges. Also, if you use the `REPLACE` command with the `/NOAUTHPRIVILEGES` qualifier, the Authorized Privileges become the same as the Default Privileges (set using the `/PRIVILEGED` qualifier).

You can specify one or more of the privilege names described in detail in an appendix to the *HP OpenVMS Guide to System Security*. (`ALL` is the default.)

**/EXECUTE\_ONLY**

**/NOEXECUTE\_ONLY (default)**

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 no read access. All shareable images referenced by the program must be installed, and OpenVMS RMS uses trusted logical names, those created for use in executive or kernel mode.

You cannot specify this qualifier for an executable image linked with the /TRACEBACK qualifier.

**/HEADER\_RESIDENT**  
**/NOHEADER\_RESIDENT**

Installs the file as a known image with a permanently resident header (native mode images only). An image installed header resident is implicitly installed open.

**/LOG**  
**/NOLOG (default)**

Lists the newly created known file entry along with any associated global sections created by the installation.

**/OPEN**  
**/NOOPEN**

Installs the file as a permanently open known image.

**/PRIVILEGED[=(priv-name[,...])]**  
**/NOPRIVILEGED**

Installs the file as a known image installed with the working privileges specified.

**Usage Notes**

- If a privileged image is not located on the system volume, the image is implicitly installed /OPEN.
- The set of privileges for a privileged image can be empty. You must, however, list each privilege every time you define or redefine privileges.
- The /PRIVILEGED qualifier applies only to executable images.
- You cannot specify this qualifier for an executable image linked with the /TRACEBACK qualifier.
- You cannot assign privilege names with the /NOPRIVILEGED qualifier.

You can specify one or more of the privilege names described in detail in an appendix to the *HP OpenVMS Guide to System Security*. (ALL is the default.)

**/PROTECTED**  
**/NOPROTECTED (default)**

Installs the file as a known image that is protected from user-mode and supervisor-mode write access. You can write into the image only from executive or kernel mode. The /PROTECTED qualifier together with the /SHARE qualifier are used to implement user-written services, which become privileged shareable images.

**/PURGE (default)**  
**/NOPURGE**

Specifies that the image can be removed by a purge operation; if you specify /NOPURGE, you can remove the image only by a delete or remove operation.

**/RESIDENT[=(**[NO]CODE**,**[NO]DATA**)]**

On Alpha and I64 systems, causes image code sections or read-only data sections to be placed in the granularity hint regions and compresses other image sections, which remain located in process space. If you do not specify the /RESIDENT qualifier, neither code nor data is installed resident. If you specify

## INSTALL REPLACE

the /RESIDENT qualifier without keyword arguments, code is installed resident, and data is not installed resident.

The image must be linked using the /SECTION\_BINDING=(CODE,DATA) qualifier. An image installed with resident code or data is implicitly installed /HEADER\_RESIDENT and /SHARED.

### **/SHARED=[NO]ADDRESS\_DATA] /NOSHARED**

Installs the file as a shared known image and creates global sections for the image sections that can be shared. An image installed shared is implicitly installed open.

When you use the ADDRESS\_DATA keyword with the /SHARED qualifier, P1 space addresses are assigned for shareable images. With the assigned addresses, the Install utility can determine the content of an address data section when the image is installed rather than when it is activated, reducing CPU and I/O time. A global section is created to allow shared access to address data image sections.

### **/SIGNAL /NOSIGNAL**

Installs the file as an image capable of accepting signals (or exceptions) from other processes. This allows privileged images to be signaled.

### **/WRITABLE /NOWRITABLE**

Installs the file as a writable known image as long as you also specify the /SHARED qualifier. The /WRITABLE qualifier only applies to images with image sections that are shareable and writable. The /WRITABLE qualifier is automatically negated if the /NOSHARED qualifier is specified.

## 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 from this version.

The full name of the file specification is assumed to be SYS\$SYSTEM:GRPCOMM.EXE.

---

## LAN Control Program (LANCP) Utility





## 12.1 LANCP Description

The LAN Control Program (LANCP) utility allows you to configure and control the LAN software and hardware on OpenVMS systems.

You can use LANCP to:

- Obtain LAN device counters, revision, and configuration information
- Change the operational parameters of LAN devices on the system
- Maintain the LAN permanent and volatile device and node databases
- Control the LANACP LAN Server process (including Maintenance Operations Protocol (MOP) downline load server related functions)
- Initiate MOP console carrier and MOP trigger boot operations

On VAX systems, LANCP and LANACP are limited to the display of LAN counters and configuration data, maintenance of the node and device databases, and control of MOP functions.

On Alpha and I64 systems, LANCP and LANACP functionality is equivalent. On I64, however, FDDI, Token Ring, and ATM are not supported; therefore, the commands involving these devices are not relevant.

## 12.2 LANCP Usage Summary

You can use the LANCP utility to:

- Set LAN parameters to customize your LAN environment.
- Display LAN settings and counters.
- Provide MOP downline load support for devices such as terminal servers, x-terminals, and LAN-based printers, and for booting satellites in an OpenVMS Cluster environment.

### Format

LANCP [command]

#### **Parameter command**

Specifies a LANCP command. This parameter is optional. If no command is specified, the utility displays its prompt and waits for command input.

### Description

To invoke LANCP, enter the following command at the DCL command prompt:

```
$ RUN SYS$SYSTEM:LANCP
```

The LANCP utility responds by displaying the LANCP> prompt, at which you can enter any LANCP command described in this chapter.

To define LANCP as a foreign command, either at the DCL prompt or in a startup or login command file, enter:

```
$ LANCP ::= $SYS$SYSTEM:LANCP
```

Then you can enter the LANCP command at the DCL prompt to invoke the utility and enter LANCP commands.

## LANCP

### 12.2 LANCP Usage Summary

When you enter the LANCP command:

- Without specifying any command qualifiers, the LANCP utility displays the LANCP> prompt, at which you can enter commands.
- With command qualifiers, the LANCP utility terminates after it executes the command and returns you to the DCL command prompt.

---

**Note**

---

Some LANCP commands require special privileges.

---

To exit from the LANCP utility, enter the EXIT command at the LANCP> prompt or press Ctrl/Z.

For information about the LANCP utility, enter the HELP command at the LANCP> prompt.

### 12.3 LANCP Commands

This section describes and provides examples of the LANCP commands. The following table summarizes the LANCP commands:

Command	Function
@ (Execute Procedure)	Executes a command procedure.
CLEAR DEVICE	Deletes device data from the LAN volatile device database.
CLEAR DLL or CLEAR MOPDLL	Clears MOP downline load counters for all nodes and devices.
CLEAR NODE	Deletes a node from the LAN volatile node database.
CONNECT NODE	Connects to a LAN device, such as a terminal server, that implements a management interface using the MOP console carrier protocol.
CONVERT DEVICE_DATABASE	Converts the device database to the format required by the current version of LANCP.
CONVERT NODE_DATABASE	Converts the node database to the format required by the current version of LANCP.
DEFINE DEVICE	Enters a device into the LAN permanent device database or modifies an existing entry.
DEFINE NODE	Enters a node into the LAN permanent node database or modifies an existing entry.
EXIT	Stops execution of LANCP and returns control to the DCL command level.
HELP	Provides online help information about the LANCP utility.
LIST DEVICE	Displays information in the LAN permanent device database.
LIST NODE	Displays information in the LAN permanent node database.
PURGE DEVICE	Deletes device data from the LAN permanent device database.

## LANCP 12.3 LANCP Commands

Command	Function
PURGE NODE	Deletes a node from the LAN permanent node database.
SET ACP	Modifies the operation of the LANACP LAN Server process.
SET DEVICE	Enters a device into the LAN volatile device database or modifies an existing entry and sets device parameters.
SET NODE	Enters a node into the LAN volatile node database or modifies an existing entry.
SHOW ACP	Displays currently set LANCP and LANACP options.
SHOW CONFIGURATION	Displays a list of LAN devices on the system.
SHOW DEVICE	Displays information in the LAN volatile device database and displays device data.
SHOW DLL or SHOW MOPDLL	Displays the current state of MOP downline load services.
SHOW LOG	Displays recent downline load activity.
SHOW NODE	Displays information in the LAN volatile node database.
SPAWN	Creates a subprocess of the current process.
TRIGGER NODE	Issues a request to reboot to a remote node.
UPDATE DEVICE	Updates a device, primarily to issue a reset to it.

## LANCP @ (Execute Procedure)

---

### @ (Execute Procedure)

Executes a command procedure or requests the command interpreter to read subsequent command input from a specific file or device.

### Format

@ file-spec

### Parameter

#### file-spec

Specifies either the input device or the file for the preceding command, or the command procedure to be executed.

### Qualifiers

None.

### Example

```
$ CREATE COUNT.COM
SHOW DEVICE/COUNTERS EIA
SPAWN WAIT 00:01:00
@COUNT
Ctr/Z
$ RUN SYSSYSTEM:LANCP
LANCP> @COUNT

Device Counters EIA0:
  Value Counter
  -----
    62030 Seconds since last zeroed
19910982 Bytes received
  239192 Bytes sent
  155812 Packets received
    4348 Packets sent
19910982 Multicast bytes received
  239140 Multicast bytes sent
  155812 Multicast packets received
    4346 Multicast packets sent
    0 Unrecognized unicast destination packets
134311 Unrecognized multicast destination packets
    0 Unavailable station buffers
    0 Unavailable user buffers
    0 Alignment errors
    0 Frame check errors
    0 Frame size errors
    0 Frame status errors
    0 Frame length errors
    0 Frame too long errors
    0 Data overruns
    0 Send data length errors
    0 Receive data length errors
    0 Transmit underrun errors
    0 Transmit failures
    0 Carrier check failures
    0 Station failures
    0 Initially deferred packets sent
    0 Single collision packets sent
    0 Multiple collision packets sent
```

**LANCP**  
**@ (Execute Procedure)**

```
0 Excessive collisions
0 Late collisions
0 Collision detect check failures
1 Link up transitions (19-OCT-2004 13:47:50.86)
0 Link down transitions
None Time of last generic transmit error
None Time of last generic receive error
```

This example creates and runs a command procedure, COUNT.COM, that displays device counters once every minute.

## LANCP CLEAR DEVICE

---

### CLEAR DEVICE

Deletes device data from the LAN volatile device database. Requires SYSPRV privilege.

#### Format

CLEAR DEVICE device-name

#### Parameter

##### **device-name**

Supplies the LAN controller device name. The device name has the form *ddcu* where *dd* is the device code, *c* is the controller designation, and *u* is the unit number.

LAN devices are specified as the name of the template device, which is unit 0. For example, the first PCI Ethernet device is specified as EWA0, the second as EWB0., and so on. You can, for example, specify a DEMNA controller as EXA, EXA0, or EXA0:.

#### Qualifier

##### **/ALL**

Deletes data for all LAN devices in the LAN volatile device database. If you specify a device name, all matching LAN devices are selected. For example: E to select all Ethernet devices, F for FDDI, I for Token Ring, EW to select all PCI Ethernet PCI devices.

##### **/CHARACTERISTICS or /PARAMETERS**

Deletes device characteristics settings for the LAN device, such as speed, duplex mode, and other device parameters.

##### **/DLL or /MOPDLL**

Deletes MOP downline load settings for the LAN device.

##### **/TRACE**

Deletes driver tracing settings for the LAN device.

If no qualifier is present to select the type of data to delete, the entire device entry is deleted.

#### Example

```
LANCP> CLEAR DEVICE EXA0
```

This command deletes device EXA0 from the LAN volatile device database.

## **CLEAR DLL**

This command clears MOP downline load counters from the LAN volatile device and node databases for all nodes and devices. This command requires SYSPRV privilege.

### **Format**

CLEAR DLL

### **Parameters**

None.

### **Qualifiers**

None.

### **Example**

```
LANCP> CLEAR DLL
```

This command clears MOP downline load counters from the LAN volatile device and node databases for all nodes and devices.

## LANCP CLEAR NODE

---

### CLEAR NODE

Deletes a node from the LAN volatile node database. Requires SYSPRV privilege.

#### Format

```
CLEAR NODE node-name
```

#### Parameter

**node-name**

Supplies the name of a node in the LAN volatile node database.

#### Qualifier

**/ALL**

Deletes all LAN nodes in the LAN volatile node database. If you specify a node name, all matching nodes are selected; for example, A/ALL deletes all nodes whose name begins with A.

#### Example

```
LANCP> CLEAR NODE VAXSYS
```

This command deletes the node VAXSYS from the LAN volatile node database.



---

## CONNECT NODE

Opens a MOP console carrier connection to the specified node. This allows a local terminal to act as the console for a remote system.

### Format

CONNECT NODE node-specification

### Parameter

#### **node-specification**

Supplies either the node name or the node address of the target node. If you supply the node name, the node address is obtained by looking up the node name in the LAN volatile node database. If you supply the node address, the corresponding node need not be defined in the LAN volatile node database. The canonical form of the address consists of 6 hexadecimal byte characters separated by hyphens. Use a colon as the separator character to indicate the bit-reversed form of the address.

### Qualifiers

#### ***/DEVICE=device-name***

Specifies the LAN controller device name to be used for the connection. For example, you can specify a DEMNA controller as EXA, EXA0, or EXA0:.

#### ***/DISCONNECT=disconnect-character***

Specifies a character that you can use to terminate the connection to the remote node. To terminate a connection, press `Ctrl/disconnect-character`. You can select any ASCII character from @ through Z, except C, M, Q, S, Y; the default disconnect character is D.

#### ***/PASSWORD=16hexdigits***

Supplies the password to be used when the connection is initiated, in hexadecimal (for example, `/PASSWORD=0123456789ABCDEF`). The default password is zero. You can omit leading zeros.

#### ***/V3 or /V4***

Indicates that MOP Version 3 or Version 4 formatted messages, respectively, are to be used to make the connection. By default, LANCP determines the format by sending MOP Request ID messages to the remote node first in MOP Version 4 format, then in Version 3 format, repeating this process until a response is received or timeout occurs.

You can specify the format:

- To allow connection to nodes that do not support Request ID messages
- As a means of getting around implementation problems with one of the formats

## LANCP CONNECT NODE

### Examples

1. LANCP> CONNECT NODE GALAXY/DEVICE=EWA0

This command attempts a console-carrier connection to node GALAXY using the Ethernet device EWA0.

2. LANCP> CONNECT NODE 08-00-2B-11-22-33/DEVICE=EWA0/PASSWORD=0123456789AB

This command attempts a console-carrier connection to the given node address using the Ethernet device EWA0, with the specified password.

## **CONVERT DEVICE\_DATABASE**

Converts the device database to the format required by LANCP. If the database is not updated, LANCP can read the database but not update entries in it. The conversion is necessary when the contents of the device entry changes from one OpenVMS release to the next. Requires SYSPRV privilege.

### **Format**

CONVERT DEVICE\_DATABASE device\_database

### **Parameters**

None.

### **Qualifiers**

None.

## LANCP CONVERT NODE\_DATABASE

---

### CONVERT NODE\_DATABASE

Converts the node database to the format required by LANCP. If the database is not updated, LANCP can read the database but not update entries in it. The conversion is necessary when the contents of the node entry changes from one OpenVMS release to the next. Requires SYSPRV privilege.

#### Format

```
CONVERT NODE_DATABASE node_database
```

#### Parameters

None.

#### Qualifiers

None.

---

## DEFINE DEVICE

Enters a device into the LAN permanent device database or modifies an existing entry. Requires SYSPRV privilege.

---

### QUALIFIERS

---

See the SET DEVICE command for a list of qualifiers and the description of each. Except where noted, the only difference is that DEFINE DEVICE applies to entries in the LAN permanent device database rather than the volatile device database.

---

## Format

DEFINE DEVICE device-name

## Parameter

### device-name

Supplies the name of a device to be added to the LAN permanent device database or an entry to be modified. The device name has the form *ddcu* where *dd* is the device code, *c* is the controller designation, and *u* is the unit number. LAN devices are specified as the name of the template device, which is unit 0; for example, the first PCI Ethernet device is specified as EWA0, the second as EWBO.

## Examples

1. LANCP> DEFINE DEVICE EXA0/MOPDLL=(ENABLE,EXCLUSIVE)

This command defines LAN device EXA0 to enable LANACP MOP downline load service in exclusive mode. The setting of the KNOWNCLIENTSONLY and SIZE characteristics are not changed. If the device entry does not currently exist in the LAN permanent device database, these settings are set to the defaults.

2. LANCP> DEFINE DEVICE/ALL/MOPDLL=NOEXCLUSIVE

This command sets all LAN devices defined in the LAN permanent device database to nonexclusive mode for LANACP MOP downline load service.

3. LANCP> DEFINE DEVICE/ALL/UPDATE/VOLATILE\_DATABASE

This command enters all Ethernet devices into the LAN permanent device database and updates the entry to include the current parameter values.

## LANCP DEFINE NODE

---

### DEFINE NODE

Enters a node into the LAN permanent node database or modifies an existing entry. Requires SYSPRV privilege.

---

#### QUALIFIERS

---

See the SET NODE command for a list of qualifiers and the description of each. Except where noted, the only difference is that DEFINE NODE applies to entries in the LAN permanent node database rather than the volatile node database.

---

### Format

```
DEFINE NODE node-name
```

### Parameter

#### node-name

Supplies the name of a node to be added to the LAN permanent node database or an entry to be modified. Typically, the node name is the same as that given in the system parameter SCSNODE, but it does not need to be. The node name is limited to 63 characters in length.

### Examples

1. LANCP> DEFINE NODE GALAXY/ADDRESS=08-00-2B-11-22-33 -  
/FILE=NISCS\_LOAD.EXE -  
/ROOT=\$64\$DIA14:<SYS10.> -  
/BOOT\_TYPE=VAX\_SATELLITE

This command sets up node GALAXY in the LAN permanent node database for booting as a VAX satellite into an OpenVMS Cluster.

The NISCS\_LOAD.EXE file is actually located on \$64\$DIA14:<SYS10.SYSCOMMON.SYSLIB>. The <SYSCOMMON.SYSLIB> is supplied by the LANACP LAN Server process and is not included in the root definition.

2. LANCP> DEFINE NODE ZAPNOT/ADDRESS=08-00-2B-11-22-33 -  
/FILE=APB.EXE -  
/ROOT=\$64\$DIA14:<SYS10.> -  
/BOOT\_TYPE=ALPHA\_SATELLITE

This command sets up node ZAPNOT for booting as an Alpha satellite into an OpenVMS Cluster.

The APB.EXE file is actually located on \$64\$DIA14:<SYS10.SYSCOMMON.SYSEXEXE>. Note that the <SYSCOMMON.SYSEXEXE> is supplied by the LANACP LAN Server process and is not included in the root definition.

3. LANCP> DEFINE NODE CALPAL/ADDRESS=08-00-2B-11-22-33 -  
/FILE=APB\_061.EXE

This command sets up node CALPAL for booting an InfoServer image. It defines the file that should be loaded when a load request without a file name is received from node CALPAL.

Because the file does not include a directory specification, the logical name LAN\$DLL defines where to locate the file. You could give directory specification using the file name or by using the /ROOT qualifier.

Note that specifying the file name explicitly in the boot command overrides the file name specified in the node database entry.

## LANCP EXIT

---

### EXIT

Stops execution of LANCP and returns control to the DCL command level. You can also enter Ctrl/Z at any time to exit.

### Format

EXIT

### Parameters

None.

### Qualifiers

None.

### Example

```
LANCP> EXIT  
$
```

This command stops execution of LANCP and returns control to the DCL command level.



---

## HELP

Provides online help information about the LANCP utility.

### Format

HELP [topic]

### Parameter

**topic**

Specifies a subject for which you want information—a LANCP command or LANCP command and command keyword. If you enter the HELP command with a command name only, such as HELP SET, LANCP displays a list of all of the command keywords used with the SET command.

### Qualifiers

None.

### Example

```
LANCP> HELP DEFINE DEVICE
```

```
DEFINE
```

```
    DEVICE
```

```
        DEVICE device-name/qualifiers
```

```
        DEVICE/ALL/qualifiers
```

```
        Sets device specific parameters for the specified LAN devices.
```

```
        Requires SYSPRV privilege.
```

```
        Additional information available:
```

```
        Parameters Qualifiers
```

```
        /ALL /ATMADDRESS /DLL /ELAN /MOPDLL /VOLATILE_DATABASE /UPDATE
```

```
        Examples
```

```
DEFINE DEVICE Subtopic?
```

This command provides online help for the LANCP command DEFINE DEVICE.

## LANCP LIST DEVICE

---

### LIST DEVICE

Displays information in the LAN permanent device database.

#### Format

LIST DEVICE device-name

#### Parameter

##### **device-name**

Supplies the LAN controller device name. The device name has the form *ddcu* where *dd* is the device code, *c* is the controller designation, and *u* is the unit number. LAN devices are specified as the name of the template device which is unit 0. For example, the first DE435 Ethernet device is specified as EWA0, the second as EWB0.

For example, you can specify a DEMNA controller as EXA, EXA0, or EXA0:. This refers to the LAN template device, for which is maintained most of the device parameters and counters. Also, the device name can refer to a device unit representing an actual user or protocol. For example, the cluster protocol can be started on a device as EWA1. You can specify a device unit to view unit-specific parameter information.

If you do not specify a device name, all devices are displayed.

If you specify a device name, all matching LAN devices are displayed, for example: E to select all Ethernet devices, F for FDDI, I for Token Ring, EW to select all Ethernet PCI Tulip devices.

---

#### **Note**

---

If you do not specify a qualifier, the utility displays the matching devices without additional information.

---

#### Qualifiers

##### **/ALL**

Lists all devices in the LAN permanent device database.

##### **/CHARACTERISTICS or /PARAMETERS**

Lists status and related information about the device.

##### **/DLL or /MOPDLL**

Displays MOP downline load characteristics.

##### **/MAP**

Lists the current configuration of the functional address mapping table for a Token Ring device.

##### **/OUTPUT=file name**

Creates the specified file and directs output to it.

##### **/SR\_ENTRY**

Lists the contents of the current source routing cache table.

**/TRACE**

Lists driver trace parameters set for the device.

**Examples**

1. LANCP> LIST DEVICE/MOPDLL

Device Listing, permanent database:

```
--- MOP Downline Load Service Characteristics ---
Device      State   Access Mode      Clients           Data Size
-----
EXA0        Enabled Exclusive      KnownClientsOnly 1400 bytes
FXA0        Disabled NoExclusive    NoKnownClientsOnly 246 bytes
```

This command displays MOP downline load information in the LAN permanent device database for all known devices.

# LANCP LIST NODE

---

## LIST NODE

Displays information in the LAN permanent node database, especially MOP downline load information.

### Format

LIST NODE node-name

### Parameter

#### **node-name**

Specifies the node name. The name can include up to 63 characters associated with the node address. If no node name is given, all nodes are displayed.

### Qualifiers

#### **/ALL**

Displays data for all nodes in the LAN permanent node database. If you specify a node name, all matching nodes are selected; for example, A/ALL selects all nodes beginning with A.

#### **/OUTPUT=file-name**

Creates the specified file and directs output to that file. If the file extension is .com, the output is in the form of a list of DEFINE NODE or SET NODE commands. The resulting command file can be used to create the LAN node databases.

### Example

```
LANCP> LIST NODE
Node Listing:
GALAXY (08-00-2B-2C-51-28):
  MOP DLL: Load file:  APB.EXE
             Load root:  $64$DIA24:<SYS11.>
             Boot type:  Alpha satellite
ZAPNOT (08-00-2B-18-7E-33):
  MOP DLL: Load file:  NISCS_LOAD.EXE
             Load root:  LAVC$SYSDEVICE:<SYS10.>
             Boot type:  VAX satellite
CALPAL (08-00-2B-08-9F-4C):
  MOP DLL: Load file:  READ_ADDR.SYS
             Last file:  LAN$DLL:APB_X5WN.SYS
             Boot type:  Other
             2 loads requested, 1 volunteered
             1 succeeded, 0 failed
             Last request was for a system image, in MOP V4 format
             Last load initiated 10-JUN-1998 09:11:17 on EXA0 for 00:00:06.65
             527665 bytes, 4161 packets, 0 transmit failures
Unnamed (00-00-00-00-00-00):
```

```
Totals:
Requests received      2
Requests volunteered  1
Successful loads      1
Failed loads          0
Packets sent          2080
Packets received      2081
Bytes sent            523481
Bytes received        4184
Last load             CALPAL at 10-JUN-1998 09:11:17.29
```

This example shows output from a LIST NODE command issued on a local node on which there are three nodes defined (GALAXY, ZAPNOT, and CALPAL). CALPAL has issued two load requests:

- The first request is the multicast request from CALPAL that the local node volunteered to accept.
- The second request is the load request sent directly to the local node by CALPAL for the actual load data. The elapsed time from the second load request to completion of the load was 6.65 seconds.

# LANCP PURGE DEVICE

---

## PURGE DEVICE

Deletes device data from the LAN permanent device database. Requires SYSPRV privilege.

### Format

PURGE DEVICE device-name

### Parameter

#### **device-name**

Supplies the LAN controller device name. The device name has the form *ddcu* where *dd* is the device code, *c* is the controller designation, and *u* is the unit number. LAN devices are specified as the name of the template device which is unit 0. For example, the first DE435 Ethernet device is specified as EWA0, the second as EWB0.

For example, you can specify a DEMNA controller as EXA, EXA0, or EXA0:. To select all LAN devices, omit the device name and include the /ALL qualifier.

### Qualifier

#### **/ALL**

Deletes data for all LAN devices in the LAN permanent device database. If you specify a device name, all matching LAN devices are selected, for example: E to select all Ethernet devices, F for FDDI, I for Token Ring, EW to select all Ethernet PCI Tulip devices.

#### **/CHARACTERISTICS or /PARAMETERS**

Deletes device characteristics settings for the LAN device, such as speed, duplex mode, and other device parameters.

#### **/DLL or /MOPDLL**

Deletes MOP downline load settings for the LAN device.

#### **/TRACE**

Deletes driver tracing settings for the LAN device.

If no qualifier is present to select the type of data to delete, the entire device entry is deleted.

### Example

```
LANCP> PURGE DEVICE/ALL
```

This command deletes all devices from the LAN permanent device database.

## PURGE NODE

Deletes a node from the LAN permanent node database. Requires SYSPRV privilege.

### Format

PURGE NODE node-name

### Parameter

**node-name**

Supplies the name of a node in the LAN permanent node database.

### Qualifier

**/ALL**

Deletes all LAN nodes in the LAN permanent node database. If you specify a node name, all matching nodes are selected; for example, A/ALL deletes all nodes whose name begins with A.

### Example

```
LANCP> PURGE NODE/ALL
```

This command deletes all nodes from the LAN permanent node database.

## SET ACP

Modifies the operation of the LANACP LAN Server process. Requires SYSPRV privilege.

### Format

SET ACP

### Parameters

None.

### Qualifiers

#### **/ECHO**

#### **/NOECHO (default)**

Enables partial tracing of received and transmitted downline load messages (the first 32 bytes of the data portion of each message). Note that the last one or two MOP messages are displayed in full: the memory load message with cluster parameters, and the parameter load with transfer address message, where present in the load.

The data is written to a log file SYS\$MANAGER:LAN\$*node-name*.LOG.

To obtain the entire contents of each message, use the /FULL qualifier as follows:

SET ACP/ECHO/FULL

#### **/FULL**

#### **/NOFULL (default)**

When /ECHO is enabled, displays the entire contents of received and transmitted downline load messages.

#### **/OPCOM (default)**

#### **/NOOPCOM**

Enables OPCOM messages from LANACP LAN Server process.

Messages are generated by the LANACP LAN Server process when a device status changes, load requests are received, and loads complete. These messages are displayed on the operator's console and included in the log file written by LANACP, SYS\$MANAGER:LAN\$ACP.LOG.

#### **/STOP**

Stops the LANACP process. It can be restarted by reexecuting the SYS\$STARTUP:LAN\$STARTUP command file.

### Example

```
LANCP> SET ACP/ECHO/FULL
```

This command enables tracing of received and transmitted downline load messages. The /FULL qualifier displays the entire contents of received and transmitted downline load messages.



---

## SET DEVICE

Sets or modifies LAN device parameters. Requires PHY\_IO privilege.

### Format

SET DEVICE device-name

### Parameter

#### **device-name**

Specifies a device to be entered in the LAN volatile device database, or whose parameters are to be modified. The device name has the form *ddcu* where *dd* is the device code, *c* is the controller designation, and *u* is the unit number. LAN devices are specified as the name of the template device which is unit 0. For example, the first DE435 Ethernet device is specified as EWA0, the second as EWB0.

### Qualifiers (General)

#### **/ALL**

Sets data for all LAN devices. If you specify a device name, all matching LAN devices are selected, for example: E to select all Ethernet devices, F for FDDI, I for Token Ring, EW to select all Ethernet PCI Tulip devices.

#### **/DEVICE\_SPECIFIC=(FUNCTION="xxxx",VALUE=(n1,n2,n3...n16))**

#### **/NODEVICE\_SPECIFIC**

Allows some device-specific parameters to be adjusted. These are useful for debug purposes or for experiments while doing performance measurements. Like other LANCP commands which affect devices, this command requires SYSPRV privilege.

Specify the negated qualifier to clear all device-specific parameter data.

These functions are described in a text file on the system, SYS\$HELP:LAN\_COUNTERS\_AND\_FUNCTIONS.TXT.

#### **/DLL or MOPDLL=(enable-option, exclusive-option, size-option, knownclientsonly-option)**

Provides the MOP downline load service settings for the device.

Note that defaults apply to creation of an entry in the device database. If an existing entry is being modified, fields not specified remain unchanged.

You can specify the following keywords with this qualifier:

- *enable-option*

ENABLE  
DISABLE (default)

Specify ENABLE or DISABLE to indicate that MOP downline load service should be enabled or disabled for the device.

- *exclusive-option*

EXCLUSIVE  
NOEXCLUSIVE (default)

## LANCP SET DEVICE

Specify **EXCLUSIVE** to indicate that no other provider of MOP downline load service is allowed on the specified LAN device at the same time as LANACP. Specify **NOEXCLUSIVE** to indicate that the LAN MOP downline load service can coexist with other implementations (in particular, the DECnet Phase IV implementation that operates the MOP protocol in shared mode).

- *size-option*

**SIZE=value**

Use **SIZE=value** to specify the size in bytes of the file data portion of each downline load message. The permitted range is 246 to 1482 bytes. The default value is 246 bytes, which should allow any client to load properly. Note that some clients may not support the larger size.

The recommended size for better load performance and less server overhead is the largest size that results in successful loads of all clients. The 1482 value is derived from the maximum packet size for CSMA/CD (Ethernet) of 1518 bytes less the 802e header and CRC (26 bytes) and MOP protocol overhead (10 bytes).

You can override the size on a per-node basis. See the SET NODE commands for details.

- *knownclientonly-option*

**KNOWNCLIENTSONLY**  
**NOKNOWNCLIENTSONLY** (default)

Specify **KNOWNCLIENTSONLY** to indicate that MOP downline load requests should be serviced only for clients defined in the LAN permanent node database. When **NOKNOWNCLIENTSONLY** is selected, LANACP searches the LAN\$DLL directory for any images requested by clients that are not defined in the LAN permanent node database.

### **/MAX\_BUFFERS=value**

Sets the maximum number of receive buffers to be allocated and used by the LAN driver for the LAN device.

### **/MIN\_BUFFERS=value**

Sets the minimum number of receive buffers to be allocated and used by the LAN driver for the LAN device.

### **/PERMANENT (SET DEVICE only)**

Updates the device entries in the LAN volatile device database with any data currently set in the permanent database. This allows you to update the volatile database after changing data in the permanent database, rather than repeating the commands for each updated entry to apply the changes to the volatile database.

### **/TRACE=size-option, mask-option, stop-option**

#### **/NOTRACE**

Provides the LAN driver trace settings for the device. By default, most LAN drivers do tracing of error conditions and state changes. Tracing is controlled by an event mask that selects the events to trace, a stop mask that specifies when to stop tracing, and the size of the trace buffer.

You can change tracing settings at any time. The LAN driver allocates the trace buffer from non-paged pool. You can calculate the amount of pool needed by multiplying the number of entries by the size of each entry, 32-bytes.

The impact of tracing on the system is negligible for error and state change events, more significant when all events are selected, and very significant when full packet tracing is enabled.

The command SHOW DEVICE/TRACE displays trace results as well as the trace mask.

You can specify the following keywords with this qualifier:

- *size-option*

SIZE=*value*

Use SIZE=*value* to specify the size of the trace buffer in entries, each 32 bytes. The default is 2048 entries. The permitted range is 256 to 1000000.

The size MOD 3 encodes the amount of packet data to include, either 35 bytes, 66 bytes, 97 bytes, or the entire packet.

- *mask-option*

MASK=(*value1,value2*)

Use MASK=(*value1,value2*) to specify the trace mask to select which type of entries should be collected in the trace buffer. The first 32 bits consist of events common to most LAN drivers. The second 32 bits consist of events specific to the LAN driver for the specified device.

- *stop-option*

STOP=(*value1,value2*)

Use STOP=(*value1,value2*) to specify the trace mask to select which type of entries should stop tracing. When a trace entry is made that matches one of the bits in the stop mask, the trace mask is cleared so that you can look at the trace data accumulated so far.

#### **/UPDATE (DEFINE DEVICE only)**

Adds LAN devices that are not currently in the LAN permanent device database to that database. The initial entry for the device uses default values for all parameters. To update the permanent database with current information from the volatile database, use the DEFINE DEVICE command with the /VOLATILE qualifier. You can combine the /UPDATE and /VOLATILE qualifiers in a single DEFINE DEVICE command.

#### **/UPDATE (SET DEVICE only)**

Adds LAN devices that are not currently in the LAN volatile device database to that database. The initial entry for the device uses default values for all parameters. To update the volatile database with current information from the permanent database, use the SET DEVICE command with the /PERMANENT qualifier. You can combine the /UPDATE and /PERMANENT qualifiers in a single SET DEVICE command.

#### **/VOLATILE (DEFINE DEVICE only)**

Updates the device entries in the LAN permanent device database with any data currently set in the volatile database. This allows you to update the permanent database after changing data in the volatile database, rather than repeating the commands for each updated entry to apply the changes to the permanent database.

## LANCP SET DEVICE

### Qualifiers (Ethernet Devices)

**/AUTONEGOTIATE (default)**

**/NOAUTONEGOTIATE**

Enables or disables the use of auto-negotiation to determine the link settings. You may need to disable link auto-negotiation when connected to a switch or device that does support auto negotiation.

**/FLOW\_CONTROL (default)**

**/NOFLOW\_CONTROL**

Enables flow control on a LAN device.

**/FULL\_DUPLEX**

**/NOFULL\_DUPLEX (default)**

Enables full-duplex operation of a LAN device. Before full-duplex operation results from the use of this qualifier, additional device or network hardware setup may be required. Some devices may be enabled for full-duplex operation by default. Some devices may not allow the setting to be changed.

The **/NOFULL\_DUPLEX** qualifier disables full-duplex operation.

**/JUMBO**

**/NOJUMBO (default)**

Enables the use of jumbo frames on a LAN device. Only Gigabit Ethernet NICs support jumbo frames.

**/MEDIA=*value***

Selects the cable connection. Normally, the selection is made during device initialization using a limited autosensing algorithm that selects twisted pair, but fails over to AUI (Attachment Unit Interface) if twisted pair does not appear to be functional. Thereafter, a cabling change would require a reboot of the system to take effect. This command allows you to change the selection without rebooting.

Acceptable values are AUI (10Base2, 10Base5), TWISTEDPAIR (10BaseT), and AUTOSENSE (reperform the limited autosense algorithm). The default value is AUTOSENSE.

Some devices, such as the DE435, require a jumper change on the Ethernet card to switch between 10Base2 and 10Base5 (thinwire and thickwire). Other devices, such as the DE434, DE436, and DE500, have only twisted pair connections.

**/SPEED=*value***

Sets the speed of the LAN.

Valid values are either 10, 100, 1000, or autonegotiate, which selects the 10 megabits per second for Ethernet, 100 for Fast Ethernet, and 1000 for Gigabit Ethernet. If autonegotiate is selected, the LAN driver will redo auto-negotiation.

### Qualifiers (LAN Failover Devices)

**/DISABLE**

Disables the devices in a LAN Failover set. When disabled, LAN devices can be added to or deleted from the Failover set.

**/ENABLE**

Enables a Failover set, which activates the Logical LAN device. The LAN Failover driver selects a LAN device from the LAN Failover set as the active device and then allows I/O to the Logical LAN device.

**/FAILOVER\_SET=(device-name[,...])**

**/[NO]FAILOVER\_SET=(device-name[,...])**

Specifies the members of a LAN Failover set.

**/PRIORITY=*value***

Sets the Failover priority of a LAN device. Priority is given to the LAN Failover device with the highest priority when choosing the active LAN device.

**/SIZE=*value***

Sets the packet size of the LAN failover device.

Valid values are STANDARD (the default) or JUMBO:

- STANDARD is the Ethernet maximum packet size of 1518 bytes.
- JUMBO is the oversize packet size available with Gigabit Ethernet devices. JUMBO is allowed only if all the LAN devices in the LAN Failover set are Gigabit devices.

Note that the size specified for the LAN Failover device overrides the size set on the members of the LAN Failover set; that is, the JUMBO frame setting of a LAN device is of no consequence when the size is determined for the LAN Failover device.

**/SWITCH (SET DEVICE only)**

Forces a LAN failover to another member of the LAN Failover set.

You can test LAN failover operation by using this command to switch from one device to another.

### **Qualifiers (FDDI Devices)**

**/RING\_PURGER**

Enables the ring purger process of the FDDI device.

**/TOKEN\_ROTATION**

Sets the requested token rotation time for the FDDI ring.

**/TOKEN\_TIMEOUT**

Sets the restricted token timeout time for the FDDI ring.

**/TRANSMIT\_TIMEOUT**

Sets the valid transmission time for the FDDI device.

### **Qualifiers (Token Ring Devices)**

**/AGING\_TIMER=*value***

Sets the amount of time in seconds to age Token Ring source routing cache entries before marking them stale. This timer expires when no traffic is sent to or received from the remote node in this amount of time. The default value is 60 seconds.

Increase this value when idle connections bounce between the stale and known states. Setting this value too low may cause unnecessary explorer traffic to traverse the LAN.

**/CACHE\_ENTRIES=*value***

Sets the number of entries to reserve for caching Token Ring source routing address entries. The default value is 200 entries.

## LANCP SET DEVICE

If your system directly communicates to a large number of systems, you may want to increase this number.

### **/CONTENDER**

#### **/NOCONTENDER (default)**

Specifies that the Token Ring device is to participate in the Monitor Contention process when it joins the ring. The **/NOCONTENDER** qualifier, directs the device not to challenge the current ring server.

#### **/DISCOVERY\_TIMER=*value***

Sets the number of seconds to wait for a reply from a remote node when performing the source Token Ring routing route discovery process. The default value is 2 seconds.

If you have nodes that respond slowly on your extended LAN, you may need to increase this number to reduce the amount of explorer traffic that traverses your LAN.

#### **/EARLY (default)**

#### **/NOEARLY**

Enables Early Token Release on the device. The **/NOEARLY** qualifier, disables Early Token Release.

#### **/MAP=(MULTICAST\_ADDRESS=*address*, FUNCTIONAL\_ADDRESS=*address*)**

#### **/NOMAP=(MULTICAST\_ADDRESS=*address*)**

Maps a standard multicast address to a functional address. Token ring devices do not support IEEE 802 standard globally defined group addresses. They do support functional addresses. A functional address is a locally administered group address that has 31 possible values. Each functional address sets one bit in the third through sixth byte of the address and bytes 1 and 2 are 03-00 (C0:00 in bit reversed format).

The **/NOMAP=(MULTICAST\_ADDRESS=*address*)** qualifier, clears the mapping established for the specified address.

Specify the functional address as follows:

- The **MULTICAST\_ADDRESS** argument requires a standard 6-byte multicast address.
- The **FUNCTIONAL\_ADDRESS** argument requires only the last 4 bytes of the functional address (the preceding 03-00 bytes are automatically prefixed).
- The *address* variable, given as hexadecimal byte characters separated by hyphens, specifies the canonical form of the address. Use a colon as the separator character to indicate the bit-reversed form of the address.

For example, to map the multicast address CB-00-01-02-03-04 to the functional address 03-00-00-80-00-00 on the Token Ring device IRA0, enter the following command:

```
SET DEVICE IRA0/MAP=(MULTI=CB-00-01-02-03-04,FUNCT=00:01:00:00)
```

For the default address mapping, see Table 12-1 or issue the command **SHOW DEVICE/MAP *device-name***.

**Table 12–1 Default Functional Address Mapping for Token Ring Devices**

<b>Multicast Address</b>	<b>Functional Address</b>	<b>Description</b>
09-00-2B-00-00-04	03-00-00-00-02-00	ISO ALL ES
09-00-2B-00-00-05	03-00-00-00-01-00	ISO ALL IS
CF-00-00-00-00-00	03-00-00-08-00-00	Loopback Assistant
AB-00-00-01-00-00	03-00-02-00-00-00	DNA MOP Dump/Load
AB-00-00-02-00-00	03-00-04-00-00-00	DNA MOP Remote Console
AB-00-00-03-00-00	03-00-08-00-00-00	DNA L1 Routers
09-00-2B-02-00-00	03-00-08-00-00-00	DNA L2 Routers
09-00-2B-02-01-0A	03-00-08-00-00-00	DNA Phase IV Primary Router
AB-00-00-04-00-00	03-00-10-00-00-00	DNA Endnodes
09-00-2B-02-01-0B	03-00-10-00-00-00	DNA Phase IV Prime Unknown Destination
09-00-2B-00-00-07	03-00-20-00-00-00	PCSA NETBIOS Emulation
09-00-2B-00-00-0F	03-00-40-00-00-00	LAT Service Advertisement
09-00-2B-02-01-04	03-00-80-00-00-00	LAT Service Solicit
09-00-2B-02-01-07	03-00-00-02-00-00	LAT Xwindow Service Solicit
09-00-2B-04-00-00	03-00-00-04-00-00	LAST
09-00-2B-02-01-00	03-00-00-00-08-00	DNA Name Service Advertisement
09-00-2B-02-01-01	03-00-00-00-10-00	DNA Name Service Solicit
09-00-2B-02-01-02	03-00-00-00-20-00	DNA Time Service
03-00-00-00-00-01	03-00-00-00-00-01	NETBUI Emulation
03-00-02-00-00-00	03-00-02-00-00-00	RIPL

***/MEDIA=value***

Selects the type of cable media used to connect the adapter to the Token Ring Media Access Unit (MAU) for devices that do not automatically detect this. Acceptable values for this are either unshielded twisted pair (UTP) or shielded twisted pair (STP). The default value is STP.

***/SOURCE\_ROUTING (default)***

***/NOSOURCE\_ROUTING***

Enables source routing on the Token Ring device. If you only have one ring in your LAN or you use transparent bridging, use the */NOSOURCE\_ROUTING* qualifier to turn off source routing.

***/SPEED=value***

Sets the speed of the Token Ring LAN.

Valid values are either 4 or 16, indicating 4 megabits per second or 16 megabits per second. The default value for Token Ring is 16, unless the LAN adapter supports a nonvolatile mechanism for setting this parameter.

***/SR\_ENTRY=(LAN\_ADDRESS=address, RI=routing-information)***

***/NOSR\_ENTRY=(LAN\_ADDRESS=address)***

Statically defines a specific source-routed route for a specific node. The default value is no routes specified. This caching remains valid while used or until the aging timer expires.

## LANCP SET DEVICE

Use this qualifier only as a last resort when isolating communication failures on extended LAN topologies.

The `/NOSR_ENTRY=(LAN_ADDRESS=address)` qualifier, clears the previously defined static source routed route.

The address is a standard 6-byte LAN address (given as hexadecimal byte characters separated by hyphens), which specifies the canonical form of the address. Using a colon as the separator character indicates the bit-reversed form of the address.

The routing-information is the source routing field, specified as a series of two-byte hexadecimal characters (each byte separated by a hyphen). The field consists of a two-byte routing control field followed by up to 14 two-byte segment identifiers, each containing the ring number and the bridge number used in the hop.

### Qualifiers (ATM Devices)

#### **/ATMADDRESS=LES**

Sets the LAN emulation server (LES) address for asynchronous transfer mode (ATM). Usually the address is not user specified, and this qualifier is used only if you want a specific address. By default the address is determined by software from the configuration server for the LES.

The `/ATMADDRESS=LES` qualifier's syntax is as follows:

```
SET DEVICE/ATMADDRESS = ([NO]LES=the ATM server)
```

#### **/ATMADDRESS=ARP**

Sets the address resolution protocol (ARP) server address for Classical IP over ATM. This qualifier is required before a LIS is enabled if the local host is not the ARP server.

The `/ATMADDRESS=ARP` qualifier's syntax is as follows:

```
SET DEVICE/ATMADDRESS = (ARP=atm_arp_server)
```

#### **/CLIP**

Sets the Classical Internet Protocol (CLIP) over ATM (RFC1577). The CLIP qualifier implements a data-link level device as a client and/or a server in a logical IP subnet (LIS). This allows the IP protocol to transmit Ethernet frames over the ATM network. The `/CLIP = ENABLE` command causes the system to join the LIS. The `/CLIP = DISABLE` command causes the client to leave the logical IP subnet.

Note that a LIS requires a server, and there must be only one server for each subnet. Communication between subnets can only be performed by a router. There can only be one client for each ATM adapter.

The `/CLIP` qualifier's syntax with standard Internet dotted notation is as follows:

```
SET DEVICE/CLIP =(ip_subnet=a.b.c.d,  
                  ip_address=a.b.c.d,  
                  parent=devnam,  
                  name="ip subnet name",  
                  enable, disable  
                  type = client|server)
```



The meanings for the syntax for /CLIP are as follows:

Option	Meaning
ip_address	Specifies the IP address of the CLIP client.
ip_subnet	Specifies the subnet mask of the CLIP client.
parent=devnam	Specifies the parent device name.
name	Specifies a name for the LIS to aid in operations and diagnostics.
type=client	Starts up a classical IP client only. This is the default.
type=server	Starts up a classical IP server. Only one server for each LIS is allowed, and the server needs to be started first.
type=(server,client)	Starts up a classical IP server and client.

Keywords and their meaning for /CLIP are as follows:

Keyword	Meaning
Enable	Joins the logical IP subnet.
Disable	Causes a client to leave the logical IP subnet.

### **/ELAN**

The /ELAN qualifier has two values: enable and disable. With /ELAN=ENABLE along with the keyword STARTUP, the LAN emulation is loaded when LANACP starts. With /ELAN=DISABLE, the same parameters used with ENABLE can be invoked.

The /ELAN qualifier's syntax is as follows:

```
DEFINE DEVICE/ELAN =(parent=parent device,
                    name="ELAN NAME to join",
                    size=1516
                    type=CSMACD
                    Enable,
                    Disable,
                    description = "description string,")
```

The meaning of the syntax for /ELAN is as follows:

Option	Meaning
parent	The ATM adapter device name. An example of the parent device for DAPCA is: HWn0, where n is the controller number. An example of the parent device for DGLTA is: HCn0, where n is the controller number.
name	Optionally specified if you want to join a specific ELAN. The default is null.
size	Maximum frame size of the LAN you want to join. Valid sizes are 1516, 4544, or 9234 bytes. The default is 1516.
type	Support currently only for CSMACD, which is the default.
description	A method of describing the ELAN for display purposes only.

Keywords and their meanings for /ELAN are as follows:

## LANCP SET DEVICE

Keyword	Meaning
Enable	Begins a join on a specified emulated LAN. It also loads the driver, if not already loaded.
Disable	Causes a client to leave the emulated LAN.

**/PVC=(vci[,...])**

**/[NO]PVC=(vci[,...])**

Sets the permanent virtual circuit (PVC) to be used by a Classical IP over ATM client. This is an optional qualifier.

A list of PVCs is defined for use by CLIP clients. This command should be used before enabling the CLIP client. The PVC has to be set up manually in the ATM switch.

The **vci** is the VCI (Virtual Circuit ID) of the PVC to be used.

### Examples

1. LANCP> SET DEVICE/CONTENDER/MEDIA=UTP/NOEARLY/SOURCE ICA0

Enables monitor contention, UTP cable media, source routing and disables early token release for Token Ring device ICA0.

2. LANCP> SET DEVICE/MEDIA=TWI EWB0

Sets the media type to twisted pair for the second Tulip Ethernet device, EWB0.

3. LANCP> SET DEVICE EXA0/MOPDLL=ENABLE

Enables MOP downline load service for device EXA0, leaving the remaining MOPDLL parameters unchanged.

4. LANCP> SET DEVICE EXA0/MOPDLL=(ENABLE,EXCLUSIVE,SIZE=1482)

Enables MOP downline load service for device EXA0, in exclusive mode with the data transfer size of 1482 bytes, leaving the remaining MOPDLL parameters unchanged.

5. LANCP> SET DEVICE EXA0/MOPDLL=(ENABLE,NOEXCLUSIVE)

LANCP> SET DEVICE FXA0/MOPDLL=(ENABLE,EXCL,KNOWN)

These commands enable LANACP MOP downline load service for:

- LAN device EXA0 in nonexclusive mode
- LAN device FXB0 in exclusive mode for only known clients

---

**SET NODE**

Enters a node into the LAN volatile node database or modifies an existing entry. Requires SYSPRV privilege.

**Format**

SET NODE node-name

**Parameter****node-name**

Supplies the name of a node to be added to the LAN volatile node database or an entry to be modified. Typically, the node name is the same as that given in the system parameter SCSNODE, but it does not need to be. The node name is limited to 63 characters in length.

**Qualifiers*****/ADDRESS=node-address******/NOADDRESS (default)***

Associates a LAN address with the node name. Specify the address as 6 bytes in hexadecimal notation, separated by hyphens. The address does not have to be unique (as might be the case when the address is not known, so a nonexistent address is specified).

If multiple node addresses are to be associated with a node name, each combination may be given as a node name with an extension, for example, VAXSYS.EXA for the EXA device on node VAXSYS, or VAXSYS\_1 for the first LAN device on node VAXSYS.

If you do not specify the */ADDRESS* qualifier, the setting remains unchanged. The */NOADDRESS* qualifier clears the field.

***/ALL***

Defines data for all nodes in the LAN volatile node database. If you specify a node name, all matching nodes are selected; for example, *A/ALL* selects all nodes beginning with A.

***/BOOT\_TYPE=boot-option******/NOBOOT\_TYPE***

Indicates the type of processing required for downline load requests. You can specify one of the following keywords with this qualifier:

- VAX\_SATELLITE—a VAX satellite cluster boot
- ALPHA\_SATELLITE—an Alpha satellite cluster boot
- I64\_SATELLITE—an OpenVMS Cluster I64 satellite boot
- OTHER—the specified image; noncluster satellite loads that do not require additional data

The distinction is necessary, because OpenVMS Cluster satellite loads require additional cluster-related data be appended to the load image given by the */FILE* qualifier. The default value is OTHER.

If you do not specify the */BOOT\_TYPE* qualifier, the setting remains unchanged. The */NOBOOT\_TYPE* qualifier clears the field.

## LANCP SET NODE

Note that I64 satellites boot using TFTP rather than MOP services. This boot type is included so that the information in the node database can be maintained, not because it is needed for booting purposes.

### ***/DECNET\_ADDRESS=value***

#### **/NODECNET\_ADDRESS**

Associates a DECnet address with the node name. Specify the address in DECnet notation, *xx.xxxx*.

If you do not specify the */DECNET\_ADDRESS=value* qualifier, then the setting remains unchanged. The */NODECNET\_ADDRESS* qualifier clears the field.

### ***/FILE=file-spec***

#### **/NOFILE**

Supplies the file name of a boot file to be used when the downline load request does not include a file name (for example, OpenVMS Cluster satellite booting). The file specification is limited to 127 characters.

If no file name is specified, OpenVMS Cluster satellite loads default to APB.EXE where the boot type is set to ALPHA and NISCS\_LOAD.EXE where the boot type is set to VAX.

If you do not specify the */FILE* qualifier, the setting remains unchanged. The */NOFILE* qualifier clears the field.

### ***/IP\_ADDRESS=value***

#### **/NOIP\_ADDRESS**

Associates an IP address with the node name. Specify the address in the standard dotted notation, *xxx.xxx.xxx.xxx*.

If you do not specify the */IP\_ADDRESS=value* qualifier, the setting remains unchanged. The */NOIP\_ADDRESS* qualifier clears the field.

### ***/PERMANENT\_DATABASE (SET NODE only)***

Updates the node entries in the LAN volatile node database with any data currently set in the permanent database. This allows you to update the volatile database after changing data in the permanent database, rather than repeating the commands for each updated entry to apply the changes to the volatile database.

### ***/ROOT=directory-specification***

#### **/NOROOT**

Supplies the directory specification to be associated with the file name. For cluster satellite service, the */ROOT* qualifier specifies the satellite root directory. For noncluster service, this qualifier specifies the location of the file. If the file specification or the file name given in the boot request includes the directory name, this qualifier is ignored. The directory specification is limited to 127 characters.

If you do not specify the */ROOT* qualifier, the setting remains unchanged. The */NOROOT* qualifier clears the field.

### ***/SIZE=value***

#### **/NOSIZE**

Specifies the size in bytes of the file data portion of each downline load message. The default is the load data size specified for the device. The permitted range is 246 to 1482 bytes. Use a larger size for better load performance and less server overhead.

If you do not specify the /SIZE qualifier, the setting remains unchanged. The /NOSIZE qualifier clears the setting.

**/V3  
/NOV3**

Indicates that only MOP Version 3 formatted messages are to be used for downline load purposes, regardless of the requested format. This allows systems to load that have a problem with MOP Version 4 booting. This qualifier causes the requesting node to fail over from MOP Version 4 to MOP Version 3 when no response has been made to a MOP Version 4 load request.

If you do not specify the /V3 qualifier, the setting remains unchanged. The /NOV3 qualifier clears the setting.

**/VOLATILE\_DATABASE (DEFINE NODE only)**

Updates the node entries in the LAN permanent node database with any data currently set in the volatile database. This allows you to update the permanent database after changing data in the volatile database, rather than repeating the commands for each updated entry to apply the changes to the permanent database.

## Examples

1. LANCP> SET NODE VAXSYS/ADDRESS=08-00-2B-11-22-33 -  
/FILE=APB.EXE -  
/ROOT=\$64\$DIA14:<SYS10.> -  
/BOOT\_TYPE=ALPHA\_SATELLITE

This command sets up node VAXSYS for booting as an Alpha satellite into the cluster.

The APB.EXE file is actually located on \$64\$DIA14:<SYS10.SYSCOMMON.SYSEXE>. Note that the <SYSCOMMON.SYSEXE> is supplied by the LANACP LAN Server process and is not included in the root definition.

2. LANCP> SET NODE VAXSYS/ADDRESS=08-00-2B-11-22-33 -  
/FILE=NISCS\_LOAD.EXE -  
/ROOT=\$64\$DIA14:<SYS10.> -  
/BOOT\_TYPE=VAX\_SATELLITE

This command sets up node VAXSYS for booting as a VAX satellite into the OpenVMS Cluster.

The NISCS\_LOAD.EXE file is actually located on \$64\$DIA14:<SYS10.SYSCOMMON.SYSLIB>. The <SYSCOMMON.SYSLIB> is supplied by the LANACP LAN Server process and is not included in the root definition.

3. LANCP> SET NODE VAXSYS/ADDRESS=08-00-2B-11-22-33/NOROOT

This command changes the LAN address associated with node VAXSYS and clears the current root specification.

## LANCP SET NODE

4. SET NODE CALPAL/ADDRESS=08-00-2B-11-22-33/FILE=APB\_061.EXE

This command sets up node CALPAL for booting an InfoServer image. It defines the file that should be loaded when a load request without file name is received from node CALPAL.

Because the file does not include a directory specification, the logical name LAN\$DLL defines where to locate the file. You could give directory specification using the file name or by using the /ROOT qualifier.

Note that specifying the file name explicitly in the boot command overrides the file name specified in the node database entry.

## SHOW ACP

Displays the current settings for LANCP and LANACP.

### Format

SHOW ACP

### Parameters

None.

### Qualifiers

***/OUTPUT=file-name***

Creates the specified file and directs output to it.

### Example

```
LANCP> SHOW ACP
LANCP/LANACP options:
  Verify is OFF
  OPCOM messages are ENABLED
  DLL packet tracing is DISABLED
```

This example shows how LANCP and LANACP are currently configured.

# LANCP SHOW CONFIGURATION

---

## SHOW CONFIGURATION

Displays a list of LAN devices and characteristics on the system.

### Format

SHOW CONFIGURATION

### Parameters

None.

### Qualifiers

**/OUTPUT=***file-name*

Creates the specified file and directs output to it.

**/USERS**

Shows which protocols are using which template device.

### Example

LANCP> SHOW CONFIGURATION/USERS

LAN Configuration:

Device	Parent	Medium/User	Version	Speed	Duplex	Size	MAC Address	Current Address	Type
EIA0		CSMA/C	X-43	100	Full	1500	00-02-A5-20-00-A3	AA-00-04-00-CD-4C	82559
EIA3	60-03	DNA Routing					1498		
EIA31	08-00-2B-60-01	MOP Dump/Load					1492		
EIA32	60-01	MOP Dump/Load					1498		
EWA0		CSMA/CD	X-36	1000	Full	1500	00-10-18-04-30-26	00-10-18-04-30-26	BCM5703
EWB0		CSMA/CD	X-110	100	Full	1500	00-30-6E-4A-37-64	00-30-6E-4A-37-64	DE500
EWCO		CSMA/CD	X-46	1000	Full	1500	00-60-CF-21-82-08	00-60-CF-21-82-08	DEGPA
EWDO		CSMA/CD	X-36	100	Full	1500	00-02-A5-20-7F-AB	00-02-A5-20-7F-AB	BCM5703
EWEO		CSMA/CD	X-110	10	Full	1500	00-30-6E-46-4C-30	00-30-6E-46-4C-30	DE500
EWFO		CSMA/CD	X-110	10	Full	1500	00-30-6E-46-4C-31	00-30-6E-46-4C-31	DE500
EWGO		CSMA/CD	X-110	10	Full	1500	00-30-6E-46-4C-32	00-30-6E-46-4C-32	DE500
EWHO		CSMA/CD	X-110	10	Full	1500	00-30-6E-46-4C-33	00-30-6E-46-4C-33	DE500
EIB0		CSMA/CD	X-9	1000	Full	1500	00-30-6E-5D-A3-4E	00-30-6E-5D-A3-4E	Intel182540
EIC0		CSMA/CD	X-9	1000	Full	1500	00-30-6E-5D-A3-4F	00-30-6E-5D-A3-4F	Intel182540

This example shows the output from a SHOW CONFIGURATION command that was entered on a node that has eleven LAN devices, one of which is actively in use.



---

## SHOW DEVICE

Displays information in the volatile device database. If the LANACP process is not running, displays a list of current LAN devices.

### Format

SHOW DEVICE device-name

### Parameter

#### **device-name**

Supplies the LAN controller device name. The device name has the form *ddcu* where *dd* is the device code, *c* is the controller designation, and *u* is the unit number. LAN devices are specified as the name of the template device which is unit 0. For example, the first DE435 Ethernet device is specified as EWA0, the second as EWB0.

For example, you can specify a DEMNA controller as EXA, EXA0, or EXA0:. This refers to the LAN template device, for which is maintained most of the device parameters and counters. Also, the device name can refer to a device unit representing an actual user or protocol. For example, the cluster protocol can be started on a device as EWA1. You can specify specific device units to view unit-specific parameter information.

If you do not specify a device name, all devices are displayed.

If you specify a device name, all matching LAN devices are displayed, for example: E to select all Ethernet devices, F for FDDI, I for Token Ring, EW to select all Ethernet PCI Tulip devices.

---

#### **Note**

---

If you do not specify a qualifier, the utility displays the matching devices without additional information.

---

### Qualifiers

#### **/ALL**

Shows all devices that match device name.

#### **/CHARACTERISTICS or PARAMETERS**

Displays status and related information about the device.

#### **/COUNTERS**

Displays device counters.

#### **/DLL or MOPDLL**

Displays LAN volatile device database information related to MOP downline load for the device.

#### **/INTERNAL\_COUNTERS**

Displays internal counters. By default, it does not display zero counters. To see all counters, including zero, use the additional qualifier /ZERO. To see the debug counters, use the additional qualifier /DEBUG.

## LANCP SHOW DEVICE

### **/MAP**

Displays the current configuration of the functional address mapping table.

### **/MESSAGES**

Displays the console messages displayed by the LAN driver as part of the LAN driver internal counters. This qualifier is included for convenience, to avoid scanning the internal counters to get to the message data.

### **/OUTPUT=*file-name***

Creates the specified file and directs output to it.

### **/REVISION**

Displays the current LAN driver and device revision information, if available or applicable. Not all LAN drivers maintain revision information.

### **/SR\_ENTRY**

Displays the contents of the current Token Ring source routing cache table.

### **/TRACE**

Displays LAN driver trace data.

## Examples

1. LANCP> SHOW DEVICE/COUNTERS EXA0

```
Device Counters EXA0:
      Value Counter
      ---- -
      259225 Seconds since last zeroed
      5890496 Data blocks received
      4801439 Multicast blocks received
      131074 Receive failure
      764348985 Bytes received
      543019961 Multicast bytes received
      3 Data overrun
      1533610 Data blocks sent
      115568 Multicast packets transmitted
      122578 Blocks sent, multiple collisions
      86000 Blocks sent, single collision
      189039 Blocks sent, initially deferred
      198120720 Bytes sent
      13232578 Multicast bytes transmitted
      7274529 Send failure
      0 Collision detect check failure
      0 Unrecognized frame destination
      0 System buffer unavailable
      0 User buffer unavailable
```

This command displays counters for Ethernet device EXA0.

2. LANCP> SHOW DEVICE/MAP ICA0

Multicast to Functional Address Mapping ICA0:

Multicast address	Functional Address	Bit-Reversed
09-00-2B-00-00-04	03-00-00-00-02-00	C0:00:00:00:40:00
09-00-2B-00-00-05	03-00-00-00-01-00	C0:00:00:00:80:00
CF-00-00-00-00-00	03-00-00-08-00-00	C0:00:00:10:00:00
AB-00-00-01-00-00	03-00-02-00-00-00	C0:00:40:00:00:00
AB-00-00-02-00-00	03-00-04-00-00-00	C0:00:20:00:00:00
AB-00-00-03-00-00	03-00-08-00-00-00	C0:00:10:00:00:00
09-00-2B-02-00-00	03-00-08-00-00-00	C0:00:10:00:00:00
09-00-2B-02-01-0A	03-00-08-00-00-00	C0:00:10:00:00:00
AB-00-00-04-00-00	03-00-10-00-00-00	C0:00:08:00:00:00
09-00-2B-02-01-0B	03-00-10-00-00-00	C0:00:08:00:00:00
09-00-2B-00-00-07	03-00-20-00-00-00	C0:00:04:00:00:00
09-00-2B-00-00-0F	03-00-40-00-00-00	C0:00:02:00:00:00
09-00-2B-02-01-04	03-00-80-00-00-00	C0:00:01:00:00:00
09-00-2B-02-01-07	03-00-00-02-00-00	C0:00:00:40:00:00
09-00-2B-04-00-00	03-00-00-04-00-00	C0:00:00:20:00:00
09-00-2B-02-01-00	03-00-00-00-08-00	C0:00:00:00:10:00
09-00-2B-02-01-01	03-00-00-00-10-00	C0:00:00:00:08:00
09-00-2B-02-01-02	03-00-00-00-20-00	C0:00:00:00:04:00
03-00-00-00-00-01	03-00-00-00-00-01	C0:00:00:00:00:80
03-00-02-00-00-00	03-00-02-00-00-00	C0:00:40:00:00:00

This command displays mapping information for Token Ring device ICA0.

3. LANCP> SHOW DEVICE/MOPDLL

Device Listing, volatile database:

--- MOP Downline Load Service Characteristics ---				
Device	State	Access Mode	Clients	Data Size
EXA0	Enabled	Exclusive	KnownClientsOnly	1400 bytes
FXA0	Disabled	NoExclusive	NoKnownClientsOnly	246 bytes

This command displays MOP downline load information in the LAN volatile device database for all known devices.

4. LANCP> SHOW DEVICE/MOPDLL EXA0

Device Listing, volatile database:

--- MOP Downline Load Service Characteristics ---				
Device	State	Access Mode	Clients	Data Size
EXA0	Enabled	Exclusive	KnownClientsOnly	1400 bytes

This command displays MOP downline load information in the LAN volatile device database for device EXA0.

## LANCP SHOW DEVICE

5. LANCP> SHOW DEVICE/PARAMETERS IRA0

```
Device Parameters IRA0:
      Value  Parameter
      ----  -
      Normal  Controller mode
      External Internal loopback mode
00-00-93-58-5D-32 Hardware LAN address
      Token Ring Communication medium
      Enabled  Functional address mode
      No       Full duplex enable
      No       Full duplex operational
      16      Line speed (megabits/second)
      16 Mbps Ring speed
      STP     Line media
      Enabled  Early token release
      Disabled Monitor contender
      200     SR cache entries
      2       SR discovery timer
      60     SR Aging Timer
      Enabled  Source routing
      3       Authorized access priority
AA-00-04-00-92-FF Upstream neighbor
      0       Ring number
```

This command displays status and parameters information for Token ring device IRA0.

6. LANCP> SHOW DEVICE/REVISION EWF0

```
Device Revisions EWF0:
      Value  Component
      ----  -
      02000041 Device hardware revision
08020110 00000004 Port driver revision
08020172 00000001 LAN common routines revision
```

This command displays revision information for Ethernet device EWF0:.

7. LANCP> SHOW DEVICE/SR\_ENTRY ICA0

```
Source Routing Cache Table ICA0:
      LAN address      State      XmtTmo      RcvTmo      StaleTmo      DiscvTmo
      -----
AA-00-04-00-92-FF  LOCAL      00000028    00000028    00000245    00000000
```

This command displays source routing entry information for Token Ring device ICA0.

## SHOW DLL or MOPDLL

Displays the current state of MOP downline load services for the system, including devices for which MOP loading is enabled and counters information.

### Format

SHOW DLL or MOPDLL

### Parameters

None.

### Qualifier

***/OUTPUT=file-name***

Creates the specified file and directs output to it.

### Example

```
LANCP>SHOW DLL
LAN DLL Status:
EXA enabled in exclusive mode for known nodes only,
    data size 1482 bytes
FXA disabled
```

	#Loads	Packets	Bytes	Last load time	Last loaded
	-----	-----	-----	-----	-----
EXA	5	1675	4400620	22-SEP-2002 10:27.51	GALAXY
FXA	0	0	0		

On this node, there are two LAN devices, EXA (DEMNA) and FXA (DEMFA). MOP downline load service is enabled on EXA in exclusive mode.

Requests are answered only for nodes that are defined in the LANACP node database. The image data size in the load messages is 1482 bytes. There have been five downline loads, the last one occurring on node GALAXY at 10:27. Finally, there are no recorded downline loads for FXA, which is currently disabled for downline load service.

# LANCP SHOW LOG

---

## SHOW LOG

Displays recent downline load activity (the last 2048 bytes of log data written to the log file SYS\$MANAGER:LAN\$ACP.LOG).

### Format

SHOW LOG

### Parameters

None.

### Qualifier

***/OUTPUT=file-name***

Creates the specified file and directs output to it.

### Example

```
LANCP> SHOW LOG
SYS$MANAGER:LAN$ACP.LOG latest contents:
17-MAR-2001 07:29:51.71 Volunteered to load request on EXA0 from HELENA
  Requested file: LAVC$SYSDEVICE:<SYS1A.>[SYSCOMMON.SYSLIB]NISCS_LOAD.EXE
17-MAR-2001 07:29:53.00 Load succeeded for HELENA on EXA0
  MOP V3 format, System image,
LAVC$SYSDEVICE:<SYS1A.>[SYSCOMMON.SYSLIB]NISCS_LOAD.EXE
  Packets: 84 sent, 84 received
  Bytes: 121492 sent, 168 received, 120988 loaded
  Elapsed time: 00:00:01.09, 110998 bytes/second
17-MAR-2001 07:29:53.60 Could not respond to load request on EXA0 from AJAX,
file not found
  Requested file: LAN$DLL:READ_ADDR.SYS
17-MAR-2001 07:29:54.46 Could not respond to load request on EXA0 from AJAX,
file not found
  Requested file: LAN$DLL:READ_ADDR.SYS
17-MAR-2001 07:29:57.36 Volunteered to load request on EXA0 from HELENA
  Requested file: LAVC$SYSDEVICE:<SYS1A.>[SYSCOMMON.SYSLIB]NISCS_LOAD.EXE
17-MAR-2001 07:29:58.49 Volunteered to load request on EXA0 from AJAX
  Requested file: LAVC$SYSDEVICE:<SYS10.>[SYSCOMMON.SYSLIB]NISCS_LOAD.EXE
17-MAR-2001 07:29:59.49 Load succeeded for HELENA on EXA0
  MOP V3 format, System image,
LAVC$SYSDEVICE:<SYS1A.>[SYSCOMMON.SYSLIB]NISCS_LOAD.EXE
  Packets: 84 sent, 84 received
  Bytes: 121492 sent, 168 received, 120988 loaded
  Elapsed time: 00:00:01.73, 69935 bytes/second
17-MAR-2001 07:30:03.66 Volunteered to load request on EXA0 from AJAX
  Requested file: LAN$DLL:ONE.SYS
17-MAR-2001 07:30:04.05 Load succeeded for AJAX on EXA0
  MOP V3 format, System image, LAN$DLL:ONE.SYS
  Packets: 9 sent, 9 received
  Bytes: 11354 sent, 18 received, 11300 loaded
  Elapsed time: 00:00:00.04, 282500 bytes/second
17-MAR-2001 Requested file: LAN$DLL:ONE.SYS
17-MAR-2001 07:30:05.18 Load succeeded for AJAX on EXA0
  MOP V3 format, System image, LAN$DLL:ONE.SYS
  Packets: 9 sent, 9 received
  Bytes: 11354 sent, 18 received, 11300 loaded
  Elapsed time: 00:00:00.04, 282500 bytes/second
```

## **LANCP SHOW LOG**

This command displays the last 2048 bytes of log data written to the log file  
SYS\$MANAGER:LAN\$ACP.LOG.

# LANCP SHOW NODE

---

## SHOW NODE

Displays information in the LAN volatile node database.

### Format

SHOW NODE node-name

### Parameter

#### **node-name**

Specifies the name of a node in the LAN volatile node database. The name can include up to 63 characters associated with the node address. If you do not specify a node name, all nodes are displayed.

### Qualifiers

#### **/ALL**

Displays information for all nodes in the LAN volatile node database. If you specify a node name, all matching nodes are selected; for example, A/ALL selects all nodes beginning with A.

#### **/OUTPUT=file-name**

Creates the specified file and directs output to that file. If the file extension is .com, the output is in the form of a list of DEFINE NODE or SET NODE commands. The resulting command file can be used to create the LAN node databases.

#### **/TOTAL**

Display counter totals only, for the nodes selected.

### Examples

- LANCP> SHOW NODE  
Node Listing:  
GALAXY (08-00-2B-2C-51-28):  
MOP DLL: Load file: APB.EXE  
Load root: \$64\$DIA24:<SYS11.>  
Boot type: Alpha satellite  
  
ZAPNOT (08-00-2B-18-7E-33):  
MOP DLL: Load file: NISCS\_LOAD.EXE  
Load root: LAVC\$\$SYSDEVICE:<SYS10.>  
Boot type: VAX satellite  
  
CALPAL (08-00-2B-08-9F-4C):  
MOP DLL: Load file: READ\_ADDR.SYS  
Last file: LAN\$DLL:APB\_X5WN.SYS  
Boot type: Other  
2 loads requested, 1 volunteered  
1 succeeded, 0 failed  
Last request was for a system image, in MOP V4 format  
Last load initiated 12-JUN-2002 09:11:17 on EXA0 for 00:00:06.65  
527665 bytes, 4161 packets, 0 transmit failures  
  
Unnamed (00-00-00-00-00-00):



## LANCP SHOW NODE

```
Totals:
Requests received      2
Requests volunteered  1
Successful loads      1
Failed loads          0
Packets sent          2080
Packets received      2081
Bytes sent            523481
Bytes received        4184
Last load             CALPAL at 12-JUN-2002 09:11:17.29
```

This example shows output from a command issued on a local node on which there are three nodes defined (GALAXY, ZAPNOT, and CALPAL). CALPAL has issued two load requests:

- The first request is the multicast request from CALPAL that the local node volunteered to accept.
- The second request is the load request sent directly to the local node by CALPAL for the actual load data. The elapsed time from the second load request to completion of the load was 6.65 seconds.

2. LANCP> SHOW NODE VAXSYS

Displays node characteristics and counters information from the LAN volatile node database for node VAXSYS.

3. LANCP> SHOW NODE/ALL VAX

Displays node characteristics and counters information from the LAN volatile node database for all nodes whose name begins with VAX.

4. LANCP> SHOW NODE/ALL

Displays node characteristics and counters information from the LAN volatile node database for all nodes.

5. LANCP> SHOW NODE/ALL/OUTPUT=TMP.INI

Writes a list of all nodes to the file TMP.INI.

# LANCP SPAWN

---

## SPAWN

Creates a subprocess of the current process. The SPAWN command copies the context of the subprocess from the current process.

### Format

SPAWN [command-string]

### Parameter

#### **command-string**

A string of commands to be executed in the context of the created subprocess. After the command string is executed, control returns to LANCP.

### Qualifiers

None.

### Example

```
LANCP> SPAWN
$ MC LANCP
LANCP> DEFINE NODE BOOM/ROOT=LAVC$SYSDEVICE:<SYS22.>
LANCP> SPAWN SEARCH LAVC$SYSDEVICE:[*.SYSEXE]MOD*.DAT BOOM
*****
LAVC$SYSDEVICE:[SYS1A.SYSEXE]MODPARAMS.DAT;1
SCSNODE="BOOM    "
LANCP> DEFINE NODE BOOM/ROOT=LAVC$SYSDEVICE:<SYS1A.>
```

In this example, you enter the node information for a node, but are unsure of the root, so you spawn to search MODPARAMS.DAT for the node name and then correct the root.

---

## TRIGGER NODE

Issues a request to reboot to a remote node.

Rather than specify the format to send MOP Version 3 or 4, the LANCP utility sends one message in each format to the target node.

### Format

TRIGGER NODE node-specification

### Parameter

#### **node-specification**

Supplies either the node name or the node address of the target node. If you supply the node name, the node address is obtained by looking up the node name in the LAN volatile node database. If you supply the node address, the corresponding node need not be defined in the LAN volatile node database. The canonical form of the address consists of 6 hexadecimal byte characters separated by hyphens. Use a colon as the separator character to indicate the bit-reversed form of the address.

### Qualifiers

#### ***/DEVICE=device-name***

Specifies the LAN controller device name to be used for sending the trigger boot messages. For example, you can specify a DEMNA controller as EXA, EXA0 or EXA0:.

#### ***/PASSWORD=16hexdigits***

Supplies the password to be used when the connection is initiated, in hexadecimal (for example, */PASSWORD=0123456789ABCDEF*). The default password is zero. You can omit leading zeros.

### Examples

1. LANCP> TRIGGER NODE GALAXY/DEVICE=EWA0

This command sends MOP trigger boot messages to node GALAXY using Ethernet device EWA0.

2. LANCP> TRIGGER NODE 08-00-2B-11-22-33/DEVICE=EWA0/PASSWORD=0123456789AB

This command sends MOP trigger boot messages to the given node address using the Ethernet device EWA0, with indicated password.

## LANCP UPDATE DEVICE

---

### UPDATE DEVICE

Resets a LAN device. Requires SYSPRV privilege.

#### Format

UPDATE DEVICE device-name

#### Parameter

##### **device-name**

Specifies a device to be reset. The device name has the form *ddcu* where *dd* is the device code, *c* is the controller designation, and *u* is the unit number. LAN devices are specified as the name of the template device which is unit 0. For example, the first DE435 Ethernet device is specified as EWA0, the second as EWB0.

#### Qualifiers

##### **/RESET**

Specifies that the device is to be reset.

#### Example

```
LANCP> UPDATE DEVICE EWA0/RESET
```

Resets Ethernet device EWA0.

---

**LAT Control Program (LATCP) Utility**



## 13.1 LATCP Description

The LAT Control Program (LATCP) utility is used to configure and control the LAT software on OpenVMS systems. You can use LATCP to:

- Specify operational characteristics for your node and its services
- Turn the state of the LAT port driver (LTDRIVER) on and off
- Display the status of LAT services and service nodes in the network
- Display the status of links created on your LAT node
- Display the status of your LAT node
- Show and zero LAT counters
- Create, delete, and manage LAT ports
- Recall previously entered LATCP commands so that you can execute them again without having to retype them
- Create subprocesses so that you can execute DCL commands without exiting from LATCP

## 13.2 LATCP Usage Summary

LATCP allows you to control the LAT software on a node and to obtain information from it. For example, you can use LATCP to create services on the local node, to associate a port on the local node with a service or device on a remote terminal server, and to display information about services offered on the local node or on other nodes in the network.

When you use LATCP commands to change LAT characteristics (such as creating a service and associating a port with a service), the changes take effect immediately. However, when the LAT port driver stops, these characteristics are lost. If you want these characteristics to be present the next time you start the LAT port driver, edit LAT\$SYSTARTUP.COM by modifying or adding commands to set these characteristics. Then, invoke LAT\$STARTUP.COM to start the LAT port driver. (Refer to the *HP OpenVMS System Manager's Manual* for more information.)

### Format

```
RUN SYS$SYSTEM:LATCP
```

### Description

To invoke LATCP, enter RUN SYS\$SYSTEM:LATCP at the DCL command prompt. At the LATCP> prompt, you can enter the LATCP commands described in the following section.

To exit from LATCP, enter the EXIT command at the LATCP> prompt or press Ctrl/Z.

You can also execute a single LATCP command by using a DCL string assignment statement, as shown in the following example:

```
$ LCP := $LATCP  
$ LCP SET NODE/STATE=ON
```

LATCP executes the SET NODE command and returns control to DCL.

## LATCP

### 13.3 LATCP Commands

### 13.3 LATCP Commands

The following table summarizes the LATCP commands:

Command	Function
ATTACH	Transfers control from your current process to the specified process.
CREATE LINK	Creates LAT data links.
CREATE PORT	Creates a logical port on the local node.
CREATE SERVICE	Creates a service on a service node.
DEFINE/KEY	Assigns a command string to a function key on your keypad.
DELETE LINK	Deletes a LAT data link from a node.
DELETE PORT	Deletes an application port or dedicated port.
DELETE QUEUE_ENTRY	Deletes an incoming queued request from the local node.
DELETE SERVICE	Deletes a service on a service node.
EXIT	Returns the user to DCL command level.
HELP	Displays help text for LATCP commands.
RECALL	Recalls LATCP commands that you entered previously so that you can execute them again.
REFRESH	Refreshes your display screen, for example, after your display has been overwritten by output from some other source.
SCROLL	Allows you to retrieve information that has scrolled off the screen.
SET LINK	Modifies characteristics of LAT data links.
SET NODE	Specifies LAT characteristics for a node.
SET PORT	Maps a logical port on a node to either a remote device on a terminal server or a special application service on a remote LAT service node.
SET SERVICE	Changes service characteristics.
SHOW LINK	Displays the characteristics of links on your node.
SHOW NODE	Displays the characteristics of nodes.
SHOW PORT	Displays port characteristics.
SHOW QUEUE_ENTRY	Displays information about requests, or entries, queued on the local node.
SHOW SERVICE	Displays characteristics of LAT services known to your node.
SPAWN	Creates a subprocess.
ZERO COUNTERS	Resets the node counters, service counters, and link counters maintained by your node.



---

## ATTACH

Transfers control from your current process to the specified process. The LATCP command ATTACH is similar to the DCL command ATTACH. For example, from the DCL command level you can enter the DCL command SPAWN to create a LATCP subprocess without ending your DCL session, execute several LATCP commands at the LATCP prompt, then use the ATTACH command to return to DCL.

### Format

ATTACH [process-name]

### Parameter

#### **process-name**

Specifies the name of a parent process or spawned subprocess to which control passes. The process must already exist, be part of your current job, and share the same input stream as your current process.

Process names can contain from 1 to 15 alphanumeric characters. If a connection to the specified process cannot be made, LATCP displays an error message.

If you specify the /PID qualifier, do not use the process name parameter. If you omit the /PID qualifier, you must use the process name parameter.

To display processes, use the DCL command SHOW SYSTEM.

### Qualifier

#### **/PID=*pid***

Specifies the process identifier (PID) of the process that will have terminal control. When you specify a PID, you can omit the leading zeros. If you specify a PID, do not use the process name parameter. If you omit the /PID qualifier, you must use the **process-name** parameter.

### Description

The ATTACH command allows you to connect your input stream to another process. You can use ATTACH to change control from one process to another. For example, you can use ATTACH to change control from LATCP to the DCL command level (see the following example). While you are at the DCL command level, LATCP remains in a hibernation state until you use ATTACH to return to it.

You cannot use this command if you are logged in to a captive account. (A captive account is an account set up to restrict user access to the system. You cannot access the DCL command level from a captive account.) You cannot specify both a process name and the /PID qualifier.

## LATCP ATTACH

### Example

```
$ SET PROCESS/NAME="TOP_LEVEL"  
$ SPAWN RUN SYSS$SYSTEM:~LATCP  
LATCP> SHOW NODE/ALL  
.  
.  
.  
LATCP> ATTACH "TOP_LEVEL"  
$
```

In this example, the user enters the DCL command SPAWN to create a LATCP subprocess and uses LATCP to display the status of all nodes known to the local node. After using LATCP, the user enters the ATTACH command to return to the DCL command level.

---

## CREATE LINK

Creates the LAT data links, which are connections to LAN devices, such as Ethernet or FDDI (Fiber Distributed Data Interface) controllers, that you want your node to use. You must have OPER privilege to use this command.

### Format

```
CREATE LINK link-name
```

### Parameter

#### **link-name**

Specifies a name for a LAT data link. A link name can have up to 16 ASCII characters. The characters allowed are as follows:

- Alphanumeric characters: A–Z, a–z, 0–9
- A subset of the international character set: ASCII codes 192–253
- Punctuation characters: dollar sign (\$), hyphen (-), period (.), and underscore (\_)

You can create a maximum of eight links on your local node. Use the SHOW LINK command for a list of the link names that are defined for your node.

### Qualifiers

#### **/DECNET (default)**

#### **/NODECNET**

Directs LAT protocol to use the DECnet data link address (/DECNET) or the hardware address (/NODECNET) when starting the LAN controller. If you do not specify the /DECNET or /NODECNET qualifier, the default is that the LAT protocol will use the DECnet data link address.

Note that if you enter the CREATE LINK command with the /DECNET qualifier and receive an error message indicating a “bad parameter value,” it means the SCSSYSTEMID system parameter is set to an illegal value. To change the value of this parameter, use the following formula:

$$(1024 * a) + n$$

In the formula, *a* is the DECnet area and *n* is the DECnet computer number. If the value is outside the range of 1025 to 65535, the LAT protocol cannot start.

When you use the /NODECNET qualifier, the LAN device driver code determines which address to use. For example:

- If SCSSYSTEMID is set to 0 but DECnet is already running on an Ethernet controller, the LAN device code allows LAT to use the same address as DECnet (AA-00-04-00-xx-xx).
- If SCSSYSTEMID is set to 0 and DECnet is not running, the 08-00-2B-xx-xx-xx address is used (a different address format is used if your LAN controller is supplied by a vendor other than HP).
- If the setting for SCSSYSTEMID is the same as the DECnet node number and DECnet is not running, the LAN device code forces LAT to use the AA-00-04-00-xx-xx address.

## LATCP CREATE LINK

If DECnet is configured on the system (or if the system is part of a cluster), SCSSYSTEMID may contain a nonzero value. This is a problem only when the system has 2 or more LAN controllers connected to the same logical LAN.

For example, if your system has an FDDI controller and an Ethernet controller, your site may be configured so that the FDDI ring attached to the FDDI controller and the Ethernet segment attached to the Ethernet controller are bridged by a 10/100 LAN bridge (FDDI-to-Ethernet). In this configuration, it is impossible to run LAT over both controllers.

In such a configuration, you must run LAT and DECnet over the same controller if SCSSYSTEMID is not 0. If you fail to do so, DECnet starts first, which in turn causes the LAT startup on the other controller to fail. This failure occurs because LAT startup tries to use the AA-00-04-00-xx-xx address (the DECnet LAN address) but is prevented from doing so by the data link layer. The LAT startup fails because DECnet is already using this address on a different controller. (In a single logical LAN, all data link addresses must be unique. In this setup, both controllers try to use the same address, which is then not unique.)

The following command (which creates the LAT link) also fails because the LAN driver tries to use the address based on SCSSYSTEMID:

```
LATCP> CREATE LINK LAT$LINK_2 /NODECNET
```

If SCSSYSTEMID is set to 0, configuring LAT and DECnet on different controllers is possible. However, in a cluster environment, SCSSYSTEMID cannot be set to 0.

### ***/DEVICE=device-name***

Specifies the LAN controller device name for a LAT data link (for example, XEB0:). Only one LAT data link can be associated with a LAN controller. If you enter the CREATE LINK command without the /DEVICE qualifier, LATCP attempts to find an available controller by using a list of possible LAT data link device names. HP recommends that you specify a default device name by defining the LAT\$DEVICE logical name.

### ***/LOG***

#### ***/NOLOG (default)***

Specifies whether LATCP displays a message confirming that the link was created. If you do not specify the /LOG or /NOLOG qualifier, the default is that no message will be displayed.

### ***/STATE=option***

Specifies whether the link will be available for use. STATE can have two options:

- ON        Specifies that the created link will be available for use with the LAT protocol running.
- OFF      Specifies that the created link will not be available for use.

If you do not specify the /STATE=*option* qualifier, the default is that the created link will be available for use (ON).

## **Description**

The `CREATE LINK` command creates a link, or connection, for an OpenVMS node and a local area network (LAN) device (for example, an Ethernet or FDDI controller) and assigns a name to that link. An OpenVMS node can have eight LAN links. Each link must operate on a separate LAN controller and have its own LAN hardware.

If you do not explicitly create a link with this command before entering the `SET NODE/STATE=ON` command, LATCP automatically creates a link for you. LATCP names the link `LAT$LINK` and assigns it to the first available LAN controller or `LAT$DEVICE`, if defined. To establish additional links, use the `CREATE LINK` command.

Whenever you create a link, specify the LAN controller device name.

Use the `SET LINK` command to modify link characteristics.

## **Example**

```
LATCP> CREATE LINK NETWORK_A /DEVICE=XEB0: /STATE=ON
```

This command creates an Ethernet link named `NETWORK_A`. It specifies the Ethernet controller device `XEB0` for that link. The link will be available for use.

# LATCP CREATE PORT

---

## CREATE PORT

Creates a logical port on your local node that connects with a remote device on a terminal server. Alternatively, this command creates a logical port on your local node that connects with a specific service. The service can be offered by a terminal server or associated with one or more dedicated ports on a remote LAT service node.

You must have OPER privilege to use this command.

### Format

```
CREATE PORT [port-name]
```

### Parameter

#### **port-name**

Specifies the port name in the form `LTA $n$ :`, where  $n$  is a unique number from 1 to 9999. If the port you specify already exists, LATCP returns the following error message:

```
%LAT-W-CMDERROR, error reported by command executor  
-SYSTEM-F-DUPLNAM, duplicate name
```

If you do not specify the port name, you must specify the `/LOGICAL` qualifier.

---

#### Notes

---

When creating a port, note the following points:

- HP recommends that you assign a logical name when creating a port, instead of specifying a specific LTA device.
- You cannot use the `CREATE PORT` and `SET PORT` commands, along with the `DCL` command `SET TERMINAL`, to change the characteristics of a DECserver port unless there is an existing LAT connection to that DECserver.

### Qualifiers

#### **/APPLICATION**

Specifies that a logical port on your node is an application port. It can be used to connect to a remote device (typically a printer) on a terminal server or to a dedicated port on another LAT service node.

If you do not specify a port type, the default port type is `APPLICATION`.

---

#### Note

---

By default, LATCP creates application LAT devices with the `HANGUP` terminal characteristic. However, if you want to apply the `NOHANGUP` characteristic to application LAT devices, you can do so by entering specific LATCP and `DCL` commands. For example:

```
$ LCP ::= $LATCP
$ LCP CREATE PORT LTA1234
$ LCP SET PORT LTA1234 /APPLICATION /NODE=terminal-server /PORT=server-port
$ SET TERMINAL LTA1234 /PERMANENT /NOHANGUP
```

Note that you can insert the SET TERMINAL command in the SYS\$MANAGER:LAT\$SYSTARTUP.COM file (enter the command for each LAT device that requires the NOHANGUP characteristic).

---

### **/DEDICATED**

Specifies that a logical port on your local node is dedicated to an application service. When users on a terminal server (or on another node that supports outgoing connections) request a connection to this service name, they are connected to the dedicated port. Refer to the *HP OpenVMS I/O User's Reference Manual* for a description of programming an application service.

After creating a dedicated port on a node, use the SET PORT /SERVICE command to map this port to a service.

### **/LIMITED**

Specifies that a logical port on your local node is limited to a service in the same way a port created using the /DEDICATED qualifier is dedicated to an application service. The difference is that ports created using the /LIMITED qualifier are under the control of the system login image (LOGINOUT.EXE) instead of an application program (a user who connects to a limited service and is assigned to a limited port receives the Username: prompt).

Using the /LIMITED qualifier, you can create a limited number of ports and map them to a specific service offered by the host system. If users are logged in to all of the limited ports for the service, no more connections are allowed to that service (terminal server users receive a "service in use" message).

### **/LOG**

#### **/NOLOG (default)**

Specifies whether LATCP displays a message confirming that the port was created. If you do not specify the /LOG or /NOLOG qualifier, the default is that no message will be displayed.

#### **/LOGICAL=(NAME=logical-name[,TABLE=table][,MODE=mode])**

Specifies a logical name to be associated with the actual name of the created port. You must specify a logical name if you do not specify a port name.

---

### **Note**

---

If you have sufficient privileges to create a port, but lack the privilege to assign a logical name, the port will still be created.

---

You can specify one of the following options for the TABLE keyword:

- |       |  |
|-------|--|
| GROUP | Places the logical name in the group logical name table. You must have GRPNAM or SYSPRV privilege to place the logical name in the group logical name table. |
| JOB   | Places the logical name in the jobwide logical name table.   |

## LATCP CREATE PORT

PROCESS	Places the logical name in the process logical name table. This is the default.
SYSTEM	Places the logical name in the system logical name table. You must have SYSNAM or SYSPRV privilege to place a name in the system logical name table.

You can also specify the name of a specific table. For example, you could specify LNM\$PROCESS, which would be the equivalent of specifying PROCESS.

Options for the MODE keyword are:

EXECUTIVE	Creates an executive mode logical name. You must have SYSNAM privilege to create an executive mode logical name.
SUPERVISOR	Creates a supervisor mode logical name.
USER	Creates a user mode logical name.

The access mode associated with the logical name is determined by maximizing the access mode of the caller with the access mode specified by the MODE keyword: the mode with the lower privilege is used.

You cannot specify an access mode with a privilege higher than that of the table containing the logical name. However, if your process has SYSNAM privilege, then the specified access mode is associated with the logical name regardless of the access mode of the caller.

If you omit the MODE keyword, the access mode of the caller is associated with the logical name.

### Description

The CREATE PORT command creates a logical LAT port for your local node. You can set up the port as an application port that is later mapped to a remote printer (or other device) on a server, or you can set up the port to be mapped to a dedicated port on a remote LAT service node. See Example 1.

Alternatively, you can set up the port as a dedicated port for a special service on a LAT service node. See Example 2.

You can also create the port as a limited port, using the /LIMITED qualifier.

After creating a port, use the SET PORT command to associate (map) the port with a queue or a service. (See the discussion that follows Example 1.) Ordinarily, you create and set ports in the LAT site-specific startup procedure, LAT\$SYSTARTUP.COM. Refer to the *HP OpenVMS System Manager's Manual* for more details.

---

#### Note

---

When using the CREATE PORT command to create an application port (for example, CREATE PORT LTA5001: /APPLICATION), you might receive an error message similar to the following one:

```
%LAT-W-CMDERROR, error reported by command executor
-SYSTEM-F-DUPLNAM, duplicate name
```

This error occurs because the LAT application port that you are trying to create has already been created by some other application. That other application could be LATCP itself because LATCP's port, LATCP\$MGMT\_PORT, is used to communicate with LTDRIVER.



You can avoid creating duplicate ports in two ways:

- Use the SET NODE/DEVICE\_SEED command to move the lower boundary of the device unit number range beyond the LTA devices that you are intending to use as application ports. (By default, LTA device units that originate from the \$ASSIGN system service to LTA0: have unit numbers that fall within a range from 1 through 9999.) For example, if you know that all LTA devices from LTA7000: onward are not used as application ports, you could enter the following commands:

```
LATCP> SET NODE/DEVICE SEED=7000
LATCP> CREATE PORT LTA5001:/APPLICATION
      .
      .
      .
LATCP> CREATE PORT LTA5010:/APPLICATION
```

For more information, see the description of the /DEVICE\_SEED qualifier in the SET NODE reference section.

- Execute the LATCP command SET NODE/STATE=ON (either interactively or in a program) before any LTA application or dedicated ports are created. Because every LATCP management port (LATCP\$MGMT\_PORT) created by the previous LATCP invocation is deleted, no conflict exists with LAT application ports or newly created dedicated ports.

For more information, see the description of the /STATE qualifier in the SET NODE reference section.

---

## Examples

1. LATCP> CREATE PORT LTA22: /APPLICATION

This command creates an application port named LTA22: on a service node. You can associate the port with a specific printer on a terminal server (use the SET PORT /NODE /PORT command) or with a set of printers on a terminal server (use the SET PORT /NODE /SERVICE command). Or, you can associate the port with a dedicated port on a remote service node. In this case, use the SET PORT /NODE /SERVICE command, where the /SERVICE qualifier specifies an application service associated with a dedicated port on the remote node. See the examples for the SET PORT command.

2. LATCP> CREATE PORT LTA21: /DEDICATED

This command creates the LTA21: port. It will be used as a dedicated port that offers a specific service rather than a general timesharing service.

3. LATCP> CREATE PORT /LOG /APPLICATION -  
\_LATCP> /LOGICAL=(NAME=MAIL\_PORT, TABLE=PROCESS, MODE=SUPERVISOR)

This command creates an application port. It assigns the name of the new port to the specified logical name (MAIL\_PORT). The logical is created as a supervisor mode logical name in the LNM\$PROCESS\_TABLE logical name table. LATCP displays a confirmation message.

## LATCP CREATE PORT

```
4. $ LCP := $LATCP
   $ LCP CREATE SERVICE/LIMITED ONLY_ONE
   $ LCP CREATE PORT/LIMITED LTA1234:
   $ LCP SET PORT LTA1234: /SERVICE=ONLY_ONE
```

This series of commands creates a limited service that allows only one user to log in to the system through that service. When a user connects to service ONLY\_ONE by responding to the terminal server prompt (Local>), the user is assigned port LTA1234 and then prompted for the user name. Any user who attempts to connect to the same service while LTA1234 has a user logged in receives the “service in use” message.

---

## CREATE SERVICE

Creates a service on a service node. You must have OPER privilege to use this command.

### Format

```
CREATE SERVICE [service-name]
```

### Parameter

#### **service-name**

Specifies a LAT service name. By default, a service name is the name of the local node you defined with the SET NODE command.

The service name can be from 1 to 16 ASCII characters in length. The characters allowed are as follows:

- Alphanumeric characters: A–Z, a–z, 0–9
- A subset of the international character set: ASCII codes 192–253
- Punctuation characters: dollar sign (\$), hyphen (-), period (.), and underscore (\_)

### Qualifiers

#### **/APPLICATION**

Specifies that the created service is an application service. An application service offers a specific application on the service node rather than a general interactive service. You can define a dedicated port for the service by using the CREATE PORT and SET PORT commands.

#### **/IDENTIFICATION["identification-string"]**

Describes and identifies a service. Service nodes include the identification string in service announcements. A service node announces its services at regular intervals established with the SET NODE command. Entering the LATCP SHOW NODE command or the DECserver SHOW NODE command generates a display that includes this identification string. By default, the identification string is a translation of SYS\$ANNOUNCE.

You cannot specify more than 64 ASCII characters in an identification string (a SYS\$ANNOUNCE longer than that will be truncated to the first 64 characters). Enclose the string in quotation marks (").

#### **/LIMITED**

Specifies that the service is a limited service, using devices assigned the limited characteristic and associated with (mapped to) this limited service. This qualifier is used in conjunction with the SET PORT /LIMITED command.

#### **/LOG**

#### **/NOLOG (default)**

Specifies whether LATCP displays a message confirming that the service was created. If you do not specify the /LOG or /NOLOG qualifier, the default is that no message will be displayed.

## LATCP CREATE SERVICE

***/STATIC\_RATING=*rating****

***/NOSTATIC\_RATING***

Enables or disables dynamic service ratings. A dynamic service rating means that a LAT algorithm calculates the availability of a service dynamically, based on the overall level of activity of the node that offers the service and the amount of memory. When a terminal server or node requests a connection to a service that is offered on two or more service nodes, the requesting node selects the service node with the highest (most favorable) service rating. This selection process is called **load balancing**.

The dynamic service rating, which is the default, is usually adequate for efficient load balancing on the LAT network. However, when necessary, you can use the */STATIC\_RATING* qualifier to disable dynamic service ratings so that you can specify a static (fixed) rating. That static rating value does not change until the dynamic service rating is reenabled.

Use the static rating to direct users away from or toward your node temporarily. Static ratings range from 0 to 255. Specify a low value to make the local service node less likely to be used; specify a high value to make the local service node more likely to be used.

If you do not specify either the */STATIC\_RATING* or */NOSTATIC\_RATING* qualifier, the default is that the LAT software uses the dynamic service rating.

Limited and application services do not rely exclusively on the dynamically calculated service rating. Instead, they use a portion of the dynamic rating based on how many ports are available for the service. For example, if a limited service has 50 percent of its ports available, the dynamic service rating will be scaled, halved, and then added to 105. When ports are available, the rating will always be above the value 105.

When all ports for a limited or application service are in use, the rating will be based on the scaled dynamic rating and the number of free queue slots on the local node. The rating will always be less than 90.

This rating procedure for limited and application services follows the terminal server rating algorithm for services and available ports that the service offers, while at the same time taking into account the availability of the node (which is the factor used to calculate the dynamic rating).

If your system is licensed for a specific number of units (where only a fixed number of users can log in to the system regardless of how the login limit is set), then all dynamic ratings become 0 when all OpenVMS license units have been consumed. (This forces all node service ratings to the lowest possible value when logins are not possible because all OpenVMS license units have been consumed.)

Note as well that the LAT software transmits a service announcement message when a user logs in to or out of the system. This allows the system to more quickly provide information about service rating changes that result from a login or logout operation.

## Description

The `CREATE SERVICE` command creates a service that a service node offers to terminal servers (and nodes that support outgoing connections) on the LAT network. The service can be a general timesharing service that offers all the resources of the service node, or it can be an application service that offers a specific application on the service node. The number of services that you can create with the `CREATE SERVICE` command depends on the availability and capability of specific resources.

The following table lists the maximum number of services your node can offer and still be recognized by the DECserver terminal server, depending on the model number:

DECserver Terminal Server	Maximum Number of Services Offered by Node
Model 100	8
Model 200	64
Model 300	64
Model 90TL	64
Model 700	64
Model 500	127

**Note**

If you create more than the maximum number of services supported by a specific DECserver model, that server will not recognize your node.

To create an application service, use the `/APPLICATION` qualifier. In addition, define a dedicated port by using the `CREATE PORT` and `SET PORT` commands. Most often, a system manager creates services in `LAT$SYSTARTUP.COM`, the site-specific LAT configuration procedure. (Refer to the *HP OpenVMS System Manager's Manual* for further information about creating an application service. The *HP OpenVMS I/O User's Reference Manual* shows how to program an application service.)

Several service nodes can share one service name. A shared service name is especially useful in a cluster environment because it allows the cluster to be known by a single cluster name. When a user logs in, the terminal server connects to the least busy node offering that service.

You can modify the service characteristics with the `SET SERVICE` command.

## Examples

1. `LATCP> CREATE SERVICE/STATIC_RATING=195 SALES`

This command creates the service `SALES` on a service node. This command assigns a static rating of 195 so terminal servers (and nodes that support outgoing connections) can assess the availability of services on the node.

## LATCP CREATE SERVICE

2. LATCP> CREATE SERVICE/APPLICATION GRAPHICS

This command creates the service GRAPHICS on the local node. Use the CREATE PORT/DEDICATED and SET PORT/SERVICE=GRAPHICS commands to create a port that is dedicated to this service.

3. \$ LCP := \$LATCP  
\$ LCP CREATE SERVICE/LIMITED ONLY\_ONE  
\$ LCP CREATE PORT/LIMITED LTA1234:  
\$ LCP SET PORT LTA1234: /SERVICE=ONLY\_ONE

This series of commands creates a limited service that allows only one user to log in to the system through that service. When a user connects to service ONLY\_ONE by responding to the terminal server prompt (Local>), the user is assigned port LTA1234 and then prompted for the user name. Any user who attempts to connect to the same service while LTA1234 has a user logged in receives the “service in use” message.

---

## DEFINE/KEY

Assigns a command string to a function key. For example, you can assign the LATCP command SHOW NODE to a function key.

### Format

DEFINE/KEY key-name equivalence-string

### Parameters

**key-name**

Specifies the name of the function key that you want to define. Valid key names are as follows:

Key Name	LK201/LK401 Keyboards	VT100-Type	VT52-Type
PF1	PF1	PF1	Blue
PF2	PF2	PF2	Red
PF3	PF3	PF3	Black
PF4	PF4	PF4	
KP0-KP9	Keypad 0-9	Keypad 0-9	Keypad 0-9
PERIOD	Keypad period (.)	Keypad period (.)	
COMMA	Keypad comma (,)	Keypad comma (,)	
MINUS	Keypad minus (-)	Keypad minus (-)	
Enter	Enter	Enter	Enter
FIND	Find	–	–
INSERT_HERE	Insert Here	–	–
REMOVE	Remove	–	–
SELECT	Select	–	–
PREV_SCREEN	Prev Screen (LK201) Prev (LK401)	–	–
NEXT_SCREEN	Next Screen (LK201) Next (LK401)	–	–
HELP	Help	–	–
DO	Do	–	–
F6-F20	F6-F20	–	–

**equivalence-string**

Specifies the command string that you want assigned to the function key. To preserve spaces and lowercase characters, enclose the string in quotation marks (" ").

### Qualifiers

**/ECHO**

**/NOECHO**

Specifies whether LATCP displays the command string on your screen when you press the key. If you do not specify the /ECHO or /NOECHO qualifier, the default

## LATCP DEFINE/KEY

is that the command string will be displayed. You cannot use /NOECHO with the /NOTERMINATE qualifier.

### ***/IF\_STATE=state-name***

Specifies the state that must be set (for example, the GOLD state) for the key definition to work. Lets you assign alternative meanings to keys when the specified state is set. See the discussion of the /SET\_STATE qualifier. If you omit the /IF\_STATE qualifier, LATCP uses the current state. The state name is an alphanumeric string. States are established with the /SET\_STATE qualifier.

### ***/LOCK\_STATE***

#### ***/NOLOCK\_STATE***

Specifies that the state set by the /SET\_STATE qualifier remain in effect until explicitly changed. If you use the /NOLOCK\_STATE qualifier, the state set by /SET\_STATE remains in effect only for the next definable key that you press or for the next read-terminating character (such as Return or Ctrl/Z) that you type.

You can specify the /LOCK\_STATE qualifier only with the /SET\_STATE qualifier. If you do not specify the /LOCK\_STATE or /NOLOCK\_STATE qualifier, the default is that the state set by the /SET\_STATE qualifier remains in effect until explicitly changed.

### ***/LOG***

#### ***/NOLOG (default)***

Specifies whether LATCP displays a message confirming that the command was executed. If you do not specify the /LOG or /NOLOG qualifier, the default is that no message will be displayed.

### ***/SET\_STATE=state-name***

Causes the specified state to be set when you press the defined key. The state name can be any alphanumeric string (for example, GOLD). Use the DEFINE/KEY/IF\_STATE=*state-name* command to associate new meanings for keys when the specified state is set. See the example for the DEFINE/KEY command.

If you omit the /SET\_STATE qualifier, the current state that was locked remains in effect.

### ***/TERMINATE***

#### ***/NOTERMINATE***

Specifies whether the command string will be terminated (processed) when you press the function key. The default is /NOTERMINATE, which allows you to press other keys before the command string is processed. Pressing Return has the same effect as using /TERMINATE.

The /NOTERMINATE qualifier allows you to create key definitions that insert text into command lines, after prompts, or into other text that you are typing.

## Description

The DEFINE/KEY command assigns a command string to a function key so that when you press that key, the command is executed.



## **Example**

```
LATCP> DEFINE/KEY PF4 "SHOW NODE " /NOTERMINATE/SET STATE=GOLD  
LATCP> DEFINE/KEY PF4 "/ALL"/IF_STATE=GOLD/TERMINATE
```

The first DEFINE/KEY command in this example assigns the SHOW NODE command to function key PF4. To process the SHOW NODE command, you must press Return after pressing PF4. Note the space after the word NODE in the first DEFINE/KEY command. This space allows you to enter a node name after pressing PF4. When you press Return, the SHOW NODE command is processed. If the space is omitted, LATCP does not recognize the command (SHOW NODE). The state is set to GOLD; that state will be in effect for the next key that you press.

The second DEFINE/KEY command defines the use of the PF4 key when the keypad is in the GOLD state. When you press PF4 twice, the SHOW NODE/ALL command is processed.

## LATCP DELETE LINK

---

### DELETE LINK

Deletes a logical link from a node. You must have OPER privilege to use this command.

#### Format

```
DELETE LINK link-name
```

#### Parameter

**link-name**

Specifies the name of the link that you want to delete.

Use the SHOW LINK command for a list of the links that are defined for your node.

#### Qualifiers

**/LOG**

**/NOLOG (default)**

Specifies whether LATCP displays a message confirming that the link was deleted. If you do not specify the /LOG or /NOLOG qualifier, the default is that no message will be displayed.

#### Description

The DELETE LINK command stops any active sessions on the link and then deletes the link from your node.

#### Example

```
LATCP> DELETE LINK NETWORK_A /LOG
```

This command deletes the link NETWORK\_A. The link was created with the CREATE LINK command.

## DELETE PORT

Deletes a logical port from a node. You must have OPER privilege to use this command.

### Format

```
DELETE PORT port-name
```

### Parameter

**port-name**

Specifies the name of the application port or the dedicated port that you want to delete. An application port connects to a remote device on a terminal server, whereas a dedicated port connects to a special service.

Use the SHOW PORT command for a list of the application ports and the dedicated ports that are defined for your service node. You cannot use the DELETE PORT command to delete an interactive or forward LAT port.

### Qualifiers

**/LOG**

**/NOLOG (default)**

Specifies whether LATCP displays a message confirming that the port was deleted. If you do not specify the /LOG or /NOLOG qualifier, the default is that no message will be displayed.

### Description

The DELETE PORT command stops any active session on the port and then deletes the port from your service node.

### Example

```
LATCP> DELETE PORT LTA27:
```

This command deletes the LTA27: application port. The port was created with the CREATE PORT command.

## LATCP DELETE QUEUE\_ENTRY

---

### DELETE QUEUE\_ENTRY

Deletes an incoming queued request, or entry, from the local node.

#### Format

```
DELETE QUEUE_ENTRY queue-entry-id
```

#### Parameter

**queue-entry-id**

Specifies the identification number (ID) of the queued entry that you want to delete.

#### Description

The `DELETE QUEUE_ENTRY` deletes an incoming queued request, or entry, from the local node. Use the `SHOW QUEUE_ENTRY` command to view the list of queued entries and their IDs.

#### Example

```
LATCP> DELETE QUEUE_ENTRY 0056
```

This command deletes the queued request with an ID of 0056.

## DELETE SERVICE

Deletes a service that your service node currently offers. You must have OPER privilege to use this command.

### Format

```
DELETE SERVICE service-name
```

### Parameter

**service-name**

Specifies the name of the service, as displayed by the SHOW SERVICE command.

### Qualifiers

**/LOG**

**/NOLOG (default)**

Specifies whether LATCP displays a message confirming that the service was deleted. If you do not specify the /LOG or /NOLOG qualifier, the default is that no message will be displayed.

### Description

The DELETE SERVICE command removes a service from a service node. The service is no longer available to terminal server users and is no longer multicast in the configuration messages sent by your service node. Existing connections to the service node are not affected.

### Example

```
LATCP> DELETE SERVICE SALES
```

This command removes the service SALES from your service node. The service is no longer available to server users.

## LATCP EXIT

---

### EXIT

Stops execution of LATCP and returns control to the DCL command level. You can also enter Ctrl/Z at any time to exit.

### Format

EXIT

### Parameters

None.

### Example

```
LATCP> EXIT
```

This command exits the LATCP program and returns control to the DCL command level.

## HELP

Provides online help information for using the LATCP commands.

### Format

HELP [command-name...]

### Parameter

#### **command-name**

The name of a LATCP command or LATCP command and command keyword. If you enter the HELP command with a command name only, such as HELP SET, LATCP displays a list of all of the command keywords used with the SET command.

### Description

The HELP command is an online reference for LATCP commands. After you view an initial help display, press Return. The help display stops and the LATCP prompt is displayed. If you do not specify a command name, the HELP command displays general information about the commands for which help is available. Supplying a command name obtains syntax information for that command.

### Example

```
LATCP> HELP SET PORT
```

This command produces a description of the SET PORT command and shows the command format.

## RECALL

Displays previously entered LATCP commands on the screen so that you can execute them again.

### Format

RECALL [command-specifier]

### Parameter

#### **command-specifier**

Specifies the number or the first several characters of the LATCP command you want to recall. Command numbers can range from 1 to 20. The most recently entered command is number 1.

Use the /ALL qualifier to display all the commands in the RECALL buffer, along with their command number so that you can determine the number of the command that you want to recall.

If you do not include the command specifier or the /ALL qualifier when entering the RECALL command, LATCP displays the last command.

### Qualifiers

#### **/ALL**

Specifies that LATCP display all the commands in the RECALL buffer. LATCP displays the number of each command.

### Description

When you enter a LATCP command, LATCP stores it in a RECALL buffer for later use with the RECALL command. The RECALL command itself is never stored in the RECALL buffer.

When you use the RECALL command, LATCP displays the recalled command but does not process it. If you want the command processed as it appears, press Return. You can use the command line editing facility to make minor changes in the command line and then press Return to process the revised version of the command.

### Examples

1. LATCP> RECALL 2

This command recalls the second-to-last command you entered.

2. LATCP> RECALL SET

This command recalls the last SET command you entered.



## REFRESH

Refreshes the display screen so that any output from some other source (such as a broadcast message) is erased from the screen.

### Format

REFRESH

### Parameters

None.

### Description

Use the REFRESH command to refresh your display screen after output from other sources has overwritten the display screen. For example, if a broadcast message from a terminal server user is displayed on your screen, use the REFRESH screen to erase the broadcast message from the display. By default, you can refresh your screen by pressing Ctrl/W at the LATCP prompt.

### Example

```
LATCP> REFRESH
```

This command refreshes the display on your screen.

## SCROLL

Retrieves information that has scrolled off the screen, either up or down.

### Format

SCROLL

### Parameters

None.

### Qualifiers

#### ***/DOWN[=value]***

Scrolls the LATCP screen display down the number of lines indicated by the specified value. For convenience, you can also use the Next (or Next Screen) key on your keyboard to scroll down 15 lines (instead of entering the SCROLL/DOWN=15 command).

If you do not specify a value, the default value is 1.

#### ***/UP[=value]***

Scrolls the LATCP screen display up the number of lines indicated by the specified value. For convenience, you can also use the Prev (or Prev Screen) key on your keyboard to scroll up 15 lines (instead of entering the SCROLL/UP=15 command).

If you do not specify a value, the default value is 1.

### Description

The SCROLL command allows you to retrieve information that has scrolled off the screen (either up or down). The command works only after a LATCP SHOW command has produced output that scrolled off the screen display area. Each subsequent SHOW command erases the previous output area such that the SCROLL command retrieves the screen display produced by the last executed SHOW command.

### Example

```
LATCP> SCROLL /UP=5
```

This command scrolls up to view five lines of screen display that has previously scrolled off the viewing area.

---

## SET LINK

Changes the characteristics of LAT data links. You must have OPER privilege to use this command.

### Format

SET LINK link-name

### Parameter

#### link-name

Specifies the name for a LAT data link. A link name can have up to 16 ASCII characters. The characters allowed are as follows:

- Alphanumeric characters: A–Z, a–z, 0–9
- A subset of the international character set: ASCII codes 192–253
- Punctuation characters: dollar sign (\$), hyphen (-), period (.), and underscore (\_)

The SHOW LINK command displays the names of the links defined for a node.

### Qualifiers

#### /LOG

#### /NOLOG (default)

Specifies whether LATCP displays a message confirming that the link's characteristics were modified. If you do not specify the /LOG or /NOLOG qualifier, the default is that no message will be displayed.

#### /STATE=*option*

Specifies availability of the link for use. The two options for STATE are:

- |     |  |
|-----|--|
| ON  | Specifies that the link will be available for use with the LAT protocol running. |
| OFF | Specifies that the link will not be available for use.                           |

If you do not specify the /STATE=*option* qualifier, the default is that the link will be available (ON).

### Description

The SET LINK command changes the characteristics for a LAT data link, which must have been created previously in one of the following ways:

- Interactively entering the CREATE LINK command
- Using the SET NODE/STATE=ON command to create a default link named LAT\$LINK (if no other links are created when that command executes)
- Running a program that creates links

## LATCP SET LINK

### Example

```
LATCP> SET LINK NETWORK_A /LOG /STATE=ON
```

This command directs LATCP to start the controller for link NETWORK\_A and then to display a confirmation message.

---

## SET NODE

Specifies the LAT characteristics of your local node. You must have OPER privilege to use this command.

### Format

SET NODE [node-name]

### Parameter

#### **node-name**

Specifies a node name for your local node. By default, the node name is the translation of SYS\$NODE. A LAT node name should be the same as the DECnet node name. If the node is not running DECnet but will be in the future, HP recommends that you define SYS\$NODE and use it for both DECnet and LAT node names.

A LAT node name can be from 1 to 16 ASCII characters. The characters allowed are as follows:

- Alphanumeric characters: A–Z, a–z, 0–9
- A subset of the international character set: ASCII codes 192–253
- Punctuation characters: dollar sign (\$), hyphen (-), period (.), and underscore (\_)

### Qualifiers

#### **/ANNOUNCEMENTS**

#### **/NOANNOUNCEMENTS**

Controls whether your OpenVMS system multicasts information to the network.

If you specify /NOANNOUNCEMENTS, LAT service announcements are disabled on the local node. Remote nodes must rely on the LAT service responder feature in the LAT protocol Version 5.2 or higher to connect to the local node. Therefore, HP recommends that you use this qualifier only in a networking environment where newer model terminal servers and hosts are present (all LAT hosts, terminal servers, and PCs are running LAT protocol Version 5.2 or higher).

If you specify /NOANNOUNCEMENTS in an environment where LAT protocol Version 5.1 is present, those LAT protocol Version 5.1 systems (for example, DECserver 100, 200, and 500 systems) will be unable to connect to any of the systems that have LAT service announcements disabled.

#### **/CIRCUIT\_TIMER[=*msecs*]**

Controls the interval in milliseconds (msecs) between messages sent from the local node to other service nodes or terminal servers while connections to those nodes are active. Use this qualifier only if your node allows outgoing connections (/CONNECTIONS=OUTGOING\_ONLY or /CONNECTIONS=BOTH).

A low value for the interval decreases the response time for the port but increases the demand on service nodes. Set the circuit timer in the range of 10 to 1000 msecs.

## LATCP SET NODE

The default value of 80 msec gives a generally acceptable response time while creating a moderately low overhead on the service nodes. You cannot change this parameter when active or pending LAT connections exist.

### ***/CONNECTIONS=option***

Specifies the type of connections permissible on the local node. The four options for CONNECTIONS are:

INCOMING_ONLY	Specifies that the local node permit incoming connections only.
OUTGOING_ONLY	Specifies that the local node permit outgoing connections only. Specify this on systems that can tolerate the overhead associated with outgoing connections, such as standalone systems.
BOTH	Specifies that the local node permit both incoming and outgoing connections. Specify this on systems that can tolerate the overhead associated with outgoing connections, such as standalone systems.
NONE	Specifies that the local node disallow both incoming and outgoing connections.

If you do not specify the */CONNECTIONS=option* qualifier, the default is that the node will permit incoming connections only.

### ***/CPU\_RATING=cpu-power***

#### ***/NOCPU\_RATING***

The */CPU\_RATING* qualifier assigns your local node a rating that represents the power of your node's CPU (central processing unit) relative to other CPUs in the LAN. The value of *cpu-power* can range from 1 (for a CPU with the lowest power) to 100 (for a CPU with the highest power).

When a terminal server or node requests a connection to a service that is offered on the local node and one or more other service nodes, the requesting node selects the service node with the highest (most favorable) service rating, based on the overall level of activity of the node that offers the service and the amount of memory. This selection process is called **load balancing**.

You can influence the rating for services on your node by specifying a value for the */CPU\_RATING* qualifier. If you specify a high value for *cpu-power*, the LAT driver will calculate a relatively high service rating for services on your node (service ratings as high as 255 are possible). If you specify a low value, the LAT driver will calculate relatively low service ratings; connections will most likely be made to the same service that is offered on other nodes. In either case, the LAT driver can calculate a greater range of values for dynamic service ratings (the entire range from 0 to 255). Consequently, the ratings will more accurately reflect the availability of the service node.

If you do not specify either the */CPU\_RATING=cpu-power* or */NOCPU\_RATING* qualifier, the default is that no CPU rating will be used. A value of 0 indicates no CPU rating.

### ***/DEVICE\_SEED[=value]***

Sets the default starting number (within a range from 1 to 9999) for the unit numbers that will be assigned to new LTA devices. Note that when ports are created by assigning a channel to LTA0: with the \$ASSIGN system service, the channel numbers fall in this same range.

The default device seed value is approximately half of the maximum unit number (which you set by using the `/UNIT_NUMBER_MAXIMUM` qualifier). Interactive LAT ports, and those created with the `CREATE PORT/LOGICAL` command, are assigned unit numbers beginning with the specified device seed value and continuing up to the maximum unit number. When the maximum unit number is reached, the port is assigned the next available unit number beginning at the bottom of the range (LTA1:).

Note that each time you specify the `/UNIT_NUMBER_MAXIMUM` qualifier, the device seed value is reset to approximately half of the newly specified maximum unit number.

### **`/FORWARD_SESSION_LIMIT[=value]`**

Controls the number of sessions (a value within a range from 16 to 255) allowed on each outgoing connection. By default, 16 sessions are allowed on an outgoing connection, which means that 16 individual processes can direct the DCL command `SET HOST/LAT` to the same remote node.

You must increase the value for the `/FORWARD_SESSION_LIMIT` qualifier if a user on your system enters the command `SET HOST/LAT` and receives an error message indicating that the session limit for the LAT circuit has been reached (`%LAT-F-VCSESLIM`). Note, however, that you can change this value only when no connections exist.

### **`/GROUPS=option[,...]`**

Gives the listed groups access to services offered on your local node or prevents the listed groups from accessing services offered on your local node, depending on the options used.

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 service node.

As many as 256 groups, numbered 0 to 255, can be in the LAT network. By default, all terminal server nodes and nodes supporting outgoing connections belong to group 0. If you enter one group code, you can omit the parentheses. Use the `SHOW NODE` command for a list of the groups enabled for your service node.

The `/GROUPS` qualifier has several options. For each option described, you can specify more than one group by:

- Listing them separated by commas
- Specifying a range

The available options are:

`ENABLE=group-code[,...]`

Gives the listed groups access to your service node.

`DISABLE=group-code[,...]`

Prevents the listed groups from accessing your service node. The listed groups had been enabled previously for access to your node.

## LATCP SET NODE

ENABLE=*group-code*[,...],  
DISABLE=*group-code*[,...]

This option lets you enable certain groups and disable other groups in one command line: gives access to the groups listed with the ENABLE option and removes access from the groups listed with the DISABLE option. Enclose both ENABLE and DISABLE in parentheses; for example, /GROUP=(ENABLE=(10,12), DISABLE=(1-30)).

Example 2 shows how to specify the /GROUPS qualifier with the SET NODE command.

### **/IDENTIFICATION[="identification-string"]**

Describes and identifies a node. Service nodes include the identification string in service announcements. A service node announces its services at regular intervals established with the SET NODE command. Entering the LATCP command SHOW NODE or the DECserver command SHOW NODE generates a display that includes this identification string. By default, the identification string is the translation of SYS\$ANNOUNCE.

You cannot specify more than 64 ASCII characters in an identification string (a SYS\$ANNOUNCE longer than that will be truncated to the first 64 characters). Enclose the string in quotation marks (" ").

### **/KEEPALIVE\_TIMER[=*secs*]**

Controls the maximum interval, in seconds, between idle run messages sent by your local node to another service node to which it has a LAT connection. Your node sends these messages when no other traffic is being generated over the virtual circuit. If the service node acknowledges these messages, your node will continue to monitor the status of the circuit. If your node does not receive acknowledgment, it responds as if the circuit is down.

Use this qualifier only if your node allows outgoing connections (/CONNECTIONS=OUTGOING\_ONLY or /CONNECTIONS=BOTH).

The default value is 20. HP recommends this value for normal LAN environments. For a heavily loaded LAN, consider using a higher value. Set the timer in the range of 10 to 255. For applications that require quick notification and possible failover of a service node failure, use a lower value. You cannot change this value if active or pending connections exist.

### **/LARGE\_BUFFER /NOLARGE\_BUFFER**

Controls whether the LAT software uses large buffers while managing communications between OpenVMS systems (the default).

If you must use the /NOLARGE\_BUFFER qualifier (for example, to limit packet sizes to be no larger than the Ethernet maximum), HP recommends that you specify this command after all logical LAT links have been created and before the LAT node has been turned on. For example, note the following commands in LAT\$SYSTARTUP.COM:



```
$!  
$! Create each logical LAT link with a unique name and  
$! unique LAN address (forced with /NODECNENET).  
$!  
$ LCP CREATE LINK FDDI_1 /DEVICE=FCA0 /NODECNENET  
$ LCP CREATE LINK FDDI_2 /DEVICE=FCB0 /NODECNENET  
$!  
$! Don't use large buffer support (force packet  
$! sizes to be no larger than what Ethernet can  
$! support).  
$!  
$ LCP SET NODE /NOLARGE_BUFFER  
$!  
$! Turn on the LAT protocol.  
$!  
$ LCP SET NODE /STATE=ON
```

### **/LOG**

#### **/NOLOG (default)**

Specifies whether LATCP displays a message confirming that the node's characteristics were modified. If you do not specify the /LOG or /NOLOG qualifier, the default is that no message will be displayed.

#### **/MULTICAST\_TIMER[=*secs*]**

Specifies the time, in seconds, between multicast messages sent by a service node. A multicast message announces the services offered by a service node. The minimum value is 10 seconds; the maximum is 180 seconds. The default value is 60.

#### **/NODE\_LIMIT=*value***

#### **/NONODE\_LIMIT**

Specifies the maximum number of service nodes that your local node can store in its service and node database. Use this qualifier only if your node allows outgoing connections (/CONNECTIONS=OUTGOING\_ONLY or /CONNECTIONS=BOTH).

When the database reaches the node limit, no more nodes are added to the database when your local node receives service announcement messages. You can ensure that the node limit is not reached by using the /USER\_GROUPS qualifier to restrict access from the local node to other service nodes on the network.

If you do not specify either the /NODE\_LIMIT=*value* or /NONODE\_LIMIT qualifier, the default is no limit. A value of 0 indicates no limit.

#### **/QUEUE\_LIMIT=*value***

Sets a limit on the number of entries (incoming LAT connections only, not outgoing printer connections) that are queued on the system. The queue limit value can range from 0 to 200, with a default of 24. A value of 0 indicates that no queuing is allowed.

#### **/RETRANSMIT\_LIMIT[=*count*]**

Specifies the number of times your local node repeats transmission of a message to a service node after a transmission fails. If the transmission is still unsuccessful after these attempts, the virtual circuit between your local node and the service node terminates, along with all sessions associated with the virtual circuit.

Use this qualifier only if your node allows outgoing connections (/CONNECTIONS=OUTGOING\_ONLY or /CONNECTIONS=BOTH).

## LATCP SET NODE

Specify a value in the range of 4 to 120. The default is 8. The value you choose depends on the type of physical link used for your network, as well as the amount of traffic on the network. See your network manager for a suggested value. You cannot change this value if active or pending connections exist.

### **/SERVICE\_RESPONDER**

### **/NOSERVICE\_RESPONDER**

Specifies whether your system responds to special LAT multicast messages that request service information. Some terminal servers do not have their own service and node database. When a user on such a terminal server requests a connection to a service, the server sends a LAT multicast message requesting names of nodes that offer the requested service. **Service responder** nodes reply with the requested information.

If you specify `/SERVICE_RESPONDER`, your system responds to the special LAT multicast messages. (If you specify `/NOSERVICE_RESPONDER`, your system does not respond to those messages.) HP recommends that you set up only one or two nodes in the LAN as service responder nodes. The nodes should have the largest databases in the LAN. Use this option only if your node allows outgoing connections (`/CONNECTIONS=OUTGOING_ONLY` or `/CONNECTIONS=BOTH`).

If you do not specify either the `/SERVICE_RESPONDER` or `/NOSERVICE_RESPONDER` qualifier, the default is that your system will not respond to the special LAT multicast messages.

### **/SESSION\_LIMIT=option**

Specifies the maximum number of simultaneous sessions across all local-access ports. This limit does not affect the use of dedicated and application ports. It affects interactive port creation only, limiting the amount of resources consumed by interactive users creating new sessions.

The options for the `/SESSION_LIMIT` qualifier are:

<code>INCOMING=value</code>	Sets the session limit for incoming connections only. The default is no limit (a value of 0).
<code>OUTGOING=value</code>	Sets the session limit for outgoing connections only. The default is no limit (a value of 0).
<code>INCOMING=value,OUTGOING=value</code>	Sets the limit for both outgoing and incoming connections. Enclose both options in parentheses; for example, <code>/SESSION_LIMIT=(INCOMING=20,OUTGOING=25)</code> .

- A high limit allows users to have more sessions but increases memory utilization on your local node.
- A low limit decreases memory utilization on your local node but limits user access to services on the network.

If the limit is reached, interactive users cannot create new sessions. In this case, increase the session limit or disconnect any connections that are no longer being used.

Specify a value in the range of 0 to 255. Specifying 0 leaves no limit on the number of sessions that can be created. To prevent sessions from being created, use the /CONNECTIONS qualifier.

Not specifying the /SESSION\_LIMIT qualifier causes no limit on the number of incoming and outgoing sessions. This is the default.

**/STATE=option**

Specifies whether LAT connections are allowed. The three options for STATE are:

- ON Starts the LAT port driver (and LAT protocol software) on your node.  
HP strongly recommends that the LATCP command SET NODE/STATE=ON be executed before any LTA application or dedicated ports are created (use the format provided in SYS\$MANAGER:LAT\$SYSTARTUP.TEMPLATE) for two reasons:
- It ensures that LTDRIVER will delete any leftover LTA devices that have a reference count of 0 and are explicitly marked for deletion (using the \$DASSGN system service or the LATCP command DELETE PORT, for example). Because every LATCP management port (LATCP\$MGMT\_PORT) that was created by the previous LATCP invocation is deleted, no conflicts result with the LAT application ports or newly created dedicated ports.
  - The deletion of leftover LTA devices with a reference count of 0 minimizes the use of nonpaged pool memory.
- OFF Stops the LAT port driver (and LAT protocol software) on your node. Any existing LAT connections are aborted. Any characteristics that you changed or set with LATCP are lost. To start the LAT protocol on your node again, invoke LAT\$STARTUP.COM. (Refer to the *HP OpenVMS System Manager's Manual* for more information.) The LAT characteristics defined in LAT\$SYSTARTUP.COM will take effect.
- SHUT Specifies that new LAT connections cannot be created on your local node, but existing connections may continue. The LAT protocol continues running only until the last active session disconnects, (after which LTDRIVER will stop). At that time, your node changes to the OFF state.

---

**Caution**

---

If you stop the LAT software by specifying either the SET NODE/STATE=OFF or SET NODE/STATE=SHUT command, the LAT print symbiont (LATSYSM) will shut down all print queues that it is processing. The system will then generate an OPCOM message indicating that the print queues are stopped. You must manually restart those print queues.

---

If you do not specify the /STATE=option qualifier, the default is that the LAT port driver and LAT protocol software on your node will be started (ON).

## LATCP SET NODE

### **/UNIT\_NUMBER\_MAXIMUM=*value***

Specifies the maximum unit number for a LAT device. For example, if you specify 140, then LTA140: will be the device with the highest unit number. Specify a value that is high enough to accommodate all devices that may be in use simultaneously. When the number of devices in use exceeds the value you specify, the system gives certain LAT devices unit numbers that exceed your maximum.

Also note the following points:

- When LATCP reaches the maximum unit number, it will continue to implicitly create LTA devices beginning with the lowest available unit number.
- You cannot use the System Generation utility (SYSGEN) to set the maximum unit number for a LAT device.

The range of maximum unit numbers is 99 to 9999. The default is 9999. Note that each time you specify the /UNIT\_NUMBER\_MAXIMUM qualifier, the LTA device seed value is reset to approximately half of the newly specified maximum unit number.

### **/USER\_GROUPS=*option*[,...]**

Restricts access (from the local node) to service nodes in the network that belong to the specified groups. Your local node can access only those service nodes associated with the user groups specified. The /USER\_GROUPS qualifier also serves to limit the number of nodes stored in your node's node database. (The local node only stores information about the nodes and services that belong to at least one of the specified user groups.) By default, all LAT service nodes belong to group 0.

This qualifier affects your local node when outgoing connections are enabled (/CONNECTIONS=OUTGOING\_ONLY or /CONNECTIONS=BOTH).

Use the SHOW NODE command for a list of the user groups (service groups) enabled for your node.

The /USER\_GROUPS qualifier has several options. For each option described here, you can use two ways to specify more than one group:

- List them separated by commas.
- Specify a range.

The available options are as follows:

<code>ENABLE=<i>group-code</i>[,...]</code>	Gives your node access to the listed user groups.
<code>DISABLE=<i>group-code</i>[,...]</code>	Prevents your node from accessing the listed groups. The listed groups were enabled previously.
<code>ENABLE=<i>group-code</i>[,...], DISABLE=<i>group-code</i>[,...]</code>	This option lets you enable certain groups and disable other groups in one command line: gives your node access to the groups listed with the ENABLE option and prevents your node from accessing the groups listed with the DISABLE option. Enclose both ENABLE and DISABLE in parentheses; for example, /GROUP=(ENABLE=(10,12), DISABLE=(1-30)).

## Description

The SET NODE command, which is typically executed in the site-specific LAT configuration command procedure, LAT\$SYSTARTUP.COM, allows you to specify such characteristics as:

- Node name
- Node identification
- Service and user groups
- Timing of service announcements
- The maximum number of LAT sessions allowed simultaneously on the node
- The maximum number of outgoing sessions and incoming interactive sessions

Because LATCP commands change characteristics dynamically (that is, the commands take effect immediately), you can use the SET NODE command any time the LAT port driver is active. These changes remain in effect until the LAT port driver stops. To make sure the changes take effect when you start the LAT port driver again, edit LAT\$SYSTARTUP.COM to include these changes. Start the LAT port driver by invoking LAT\$SYSTARTUP.COM.

The *HP OpenVMS System Manager's Manual* contains additional information about the LAT network in general and service nodes in particular.

---

### Note

---

The SET NODE command must be executed first (after LTDRIVER is loaded and the LATACP is started) to ensure that other management commands execute properly thereafter.

---

## Examples

1. LATCP> SET NODE DUKE /IDENT="NODE DUKE, SALES VMSCLUSTER"

This command specifies node name DUKE for your local node. The identification string "NODE DUKE, SALES VMSCLUSTER" is multicast from node DUKE.

2. LATCP> SET NODE /MULTICAST\_TIMER=50 /GROUPS=(ENABLE=(1-3,8,11),DISABLE=5)

This command causes your local node to send multicast messages every 50 seconds to announce DUKE's services to terminal servers. The command also enables groups 1, 2, 3, 8, and 11 for access to the local node, and it disables group 5 from accessing the local node. Group 5 had been previously enabled.

3. LATCP> SET NODE /CONNECTIONS=BOTH /USER\_GROUPS=(ENABLE=(24,121-127),DISABLE=0)

This command sets up your local node to allow both incoming and outgoing connections. Users on your local node can access those service nodes belonging to user groups 24 and 121 through 127. Users cannot access service nodes in user group 0.

## LATCP SET NODE

4. LATCP> SET NODE /CIRCUIT\_TIMER=80 /KEEPALIVE\_TIMER=20 -  
LATCP> /RETRANSMIT\_LIMIT=20 /CONNECTIONS=BOTH /MULTICAST\_TIMER=60-  
LATCP> /GROUPS=(DISABLE=0,ENABLE=73) /SESSION\_LIMIT=(OUTGOING=10,INCOMING=0)

This command sets many characteristics at once for node DUKE.

---

## SET PORT

Associates a logical port on the local node with a remote port on a terminal server that supports a device. Alternatively, it associates a logical port on the local node with a specific service. The service can be offered by a terminal server or associated with one or more dedicated ports on a remote LAT service node.

You must have OPER privilege to use this command.

### 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 to 9999.

---

#### Note

---

You cannot use the CREATE PORT and SET PORT commands, along with the DCL command SET TERMINAL, to change the characteristics of a DECserver port unless there is an existing LAT connection to that DECserver.

---

### Qualifiers

#### /APPLICATION

Specifies that a port on the local node is an application port, logically associated with a port on a terminal server or a dedicated port on another LAT service node. 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. Refer to the *HP OpenVMS System Manager's Manual* for a description of configuring remote printers on a terminal server.

If you do not specify a port type, the default port type is APPLICATION.

#### /DEDICATED

Specifies that a logical port on your local node is dedicated to an application service. The /DEDICATED qualifier requires the /SERVICE qualifier.

To set up an application service for a logical port on a LAT service node:

1. Create the service by specifying the CREATE SERVICE/APPLICATION command and then define the dedicated port by specifying the CREATE PORT/DEDICATED command. You can include these commands in LAT\$SYSTARTUP.COM.
2. Associate the dedicated ports with the service by specifying the SET PORT/DEDICATED/SERVICE command.
3. Start the application program. Within the program, allocate dedicated ports with the same name as those defined in LAT\$SYSTARTUP.COM.

## LATCP SET PORT

### **/LIMITED**

Specifies that a logical port on your local node is limited to a service in the same way a port created using the **/DEDICATED** qualifier is dedicated to an application service. The difference is that ports created using the **/LIMITED** qualifier are under the control of the system login image (LOGINOUT.EXE) instead of an application program (a user who connects to a limited service and is assigned to a limited port receives the Username: prompt).

Using the **/LIMITED** qualifier, you can create a limited number of ports and map them to a specific service offered by the host system. If users are logged in to all of the limited ports for the service, no more connections are allowed to that service (terminal server users receive a “service in use” message).

### **/LOG**

#### **/NOLOG (default)**

Specifies whether LATCP displays a message confirming that the port’s characteristics were modified. If you do not specify the **/LOG** or **/NOLOG** qualifier, the default is that no message will be displayed.

#### ***/NODE=remote-node-name***

Specifies the name of a terminal server (or a remote node that supports outgoing connections) to be logically associated with the specified application port on your node. The server supports a remote device. Note that you can set up an application port on your local node and associate the port with a dedicated port on a remote LAT service node. The remote port is dedicated to an application service.

#### ***/PASSWORD=remote-password***

Specifies the password required to access a remote service that is logically associated with the specified application port.

#### ***/PORT=remote-port-name***

Specifies the name of the remote port on a terminal server that supports a remote device, or specifies the name of a remote port dedicated to an application service on a remote LAT service node. In either case, the remote port is logically associated with the specified application port on your local node.

### **/QUEUED**

#### **/NOQUEUED**

Specifies queued or nonqueued access to the server port. A queued or nonqueued request is accepted by a terminal server if a remote port is free. If the remote port is busy and queuing is enabled on the terminal server, then the server queues the remote request. If you do not want your remote requests to be queued on the server, specify **/NOQUEUED**.

Not specifying either the **/QUEUED** or **/NOQUEUED** qualifier results in queued access to the server port. This is the default.

#### ***/SERVICE=service-name***

Specifies either of the following names:

- The name of the remote service offered at a terminal server port that will be associated with the specified application port (**/APPLICATION**) on the local node
- A service name for an application program being offered on a dedicated port (**/DEDICATED**) on a LAT service node



To specify the name of a remote service offered at a terminal server port, use the /NODE and /SERVICE qualifiers. To specify a particular port for a service, use the /NODE, /PORT, and /SERVICE qualifiers. Ask the terminal server manager for these names.

To name a service for a particular application program to be offered locally on a dedicated port, use the /DEDICATED and /SERVICE qualifiers. (The service must have been created with the CREATE SERVICE command.) Assign only one service to a dedicated port, but note that several ports can have the same service assigned.

## Description

The SET PORT command associates an application port on your local node with a port or service on a terminal server.

To create a port, use one of the following methods:

- Interactively enter the CREATE PORT command.
- Run a program that creates ports.

When you associate an application port with a service on a terminal server, you allow access to any of the ports (printers) represented by that service (see Examples 1 and 2). Note that the application port must have been created with the CREATE PORT/APPLICATION command.

The SET PORT command can also associate a dedicated port on the local node with an application service offered locally. The service must already exist (see Example 3). Note that you must use the /DEDICATED and /SERVICE qualifiers.

The SET PORT command can also associate an application port on your local node with an application service associated with one or more dedicated ports on a remote LAT service node. This service is offered to users on terminal servers or on nodes that support outgoing connections (see Example 4). Note that the dedicated port must have been created with the CREATE PORT/DEDICATED command.

You can also set up the port as a limited port, using the /LIMITED qualifier.

## Examples

See the examples for the SHOW PORT command for displays that reflect the changes made by the following SET PORT command examples.

1. LATCP> SET PORT LTA22: /APPLICATION /NODE=TS33EW /PORT=LN02

This command sets up port LTA22: as an application port to be associated with the port named LN02 on the terminal server named TS33EW. This command associates port LTA22: with a specific printer on the server. In the next example, the SET PORT command associates a port with a set of printers (designated by the service name PRINTER) on a terminal server.

## LATCP SET PORT

2. LATCP> SET PORT LTA19: /APPLICATION /NODE=TLAT1 /SERVICE=PRINTER /QUEUED

This command shows how to associate a local logical port with a service (several printers) on a terminal server. The command associates the application port LTA19: with the service PRINTER on terminal server TLAT1. The service PRINTER can be associated with one or more ports on TLAT1. The /QUEUED qualifier specifies that the server offering the service PRINTER can queue the remote connection request if all ports offering the service are in use. Refer to the description of print operations in the *HP OpenVMS System Manager's Manual* for information about setting up print queues.

3. LATCP> SET PORT LTA21: /DEDICATED /SERVICE=GRAPHICS

This command specifies that the application port LTA21: on the local service node offers the service GRAPHICS to users on terminal servers or on nodes that support outgoing connections. GRAPHICS is a particular utility or application program.

4. LATCP> SET PORT MAIL\_PORT /SERVICE=MAIL/NODE=RMNODE

This command associates the port whose logical name is MAIL\_PORT with the dedicated service MAIL on remote node RMNODE. The port logically named MAIL\_PORT was created with the CREATE PORT command (see Example 3 in the discussion of the CREATE PORT command). The logical name could also have been created with the DCL command ASSIGN or DEFINE. On node RMNODE, a port must be dedicated to the service MAIL by using the SET PORT port-name /DEDICATED/SERVICE=MAIL command.

5. \$ LCP ::= \$LATCP  
\$ LCP CREATE SERVICE/LIMITED ONLY ONE  
\$ (U>(LCP CREATE PORT/LIMITED LTA1234:)  
\$ (U>(LCP SET PORT LTA1234: /SERVICE=ONLY\_ONE)

This series of commands, which includes the SET PORT command, creates a limited service that allows only one user to log in to the system through that service. When a user connects to service ONLY\_ONE by responding to the terminal server prompt (Local>), the user is assigned port LTA1234 and then prompted for the user name. Any user who attempts to connect to the same service while LTA1234 has a user logged in receives the "service in use" message.

---

## SET SERVICE

Dynamically changes the characteristics of a locally offered service. You must have OPER privilege to use this command.

### 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 name of the local node you defined by using the SET NODE command.

### Qualifiers

#### **/APPLICATION**

Sets up the service as an application service. An application service offers a specific application on the service node rather than all of the resources on the service node. Define a dedicated port for the service by using the CREATE PORT and SET PORT commands.

#### **/CONNECTIONS**

#### **/NOCONNECTIONS**

Specifies whether a service offered by an OpenVMS system accepts incoming connections. If you use the /NOCONNECTIONS qualifier to disable incoming connections, users cannot connect to that service and receive instead the error message "service is disabled."

By default, a service accepts incoming connections (/CONNECTIONS).

#### **/IDENTIFICATION[="identification-string"]**

Describes and identifies a service. Service nodes include the identification string in service announcements. A service node announces its services at regular intervals established with the SET NODE command. Entering the LATCP command SHOW NODE or the DECserver command SHOW NODE generates a display that includes this identification string.

By default, the identification string is the translation of SYS\$ANNOUNCE. A service node announces its services at regular intervals established with the SET NODE command.

You cannot specify more than 64 ASCII characters in an identification string (a SYS\$ANNOUNCE longer than that will be truncated to the first 64 characters). Enclose the string in quotation marks (" ").

#### **/LIMITED**

Specifies that the service is a limited service, using devices assigned the limited characteristic and associated with (mapped to) this limited service. This qualifier is used in conjunction with the SET PORT /LIMITED command (see Example 2).

## LATCP SET SERVICE

### **/LOG**

#### **/NOLOG (default)**

Specifies whether or not LATCP displays a message confirming that the command was executed. If you do not specify the `/LOG` or `/NOLOG` qualifier, the default is that no message will be displayed.

### **/QUEUED**

#### **/NOQUEUED**

Specifies whether a locally offered limited (`/LIMITED`) or application (`/DEDICATED`) service is allowed to have queued connections when all ports are busy (the default). If you specify `/NOQUEUED`, incoming connections will be rejected if all ports are busy.

### **/STATIC\_RATING=*rating***

#### **/NOSTATIC\_RATING**

Enables or disables dynamic service ratings. A dynamic service rating means that a LAT algorithm calculates the availability of a service dynamically, based on the overall level of activity of the node that offers the service and the amount of memory. When a terminal server or node requests a connection to a service that is offered on two or more service nodes, the requesting node selects the service node with the highest (most favorable) service rating. This selection process is called **load balancing**.

The dynamic service rating, which is the default, is usually adequate for efficient load balancing on the LAT network. However, when necessary, you can use the `/STATIC_RATING` qualifier to disable dynamic service ratings so that you can specify a static (fixed) rating. That static rating value does not change until the dynamic service rating is reenabled.

Use the static rating to direct users away from or toward your node temporarily. Static ratings range from 0 to 255. Specify a low value to make the local service node less likely to be used; specify a high value to make the local service node more likely to be used.

If you do not specify either the `/STATIC_RATING` or `/NOSTATIC_RATING` qualifier, the default is that the LAT software uses the dynamic service rating.

Limited and application services do not rely exclusively on the dynamically calculated service rating. Instead, they use a portion of the dynamic rating based on how many ports are available for the service. For example, if a limited service has 50 percent of its ports available, the dynamic service rating will be scaled, halved, and then added to 105. When ports are available, the rating will always be above the value 105.

When all ports for a limited or application service are in use, the rating will be based on the scaled dynamic rating and the number of free queue slots on the local node. The rating will always be less than 90.

This rating procedure for limited and application services follows the terminal server rating algorithm for services and available ports that the service offers, while at the same time taking into account the availability of the node (which is the factor used to calculate the dynamic rating).

If your system is licensed for a specific number of units (where only a fixed number of users can log in to the system regardless of how the login limit is set), then all dynamic ratings become 0 when all OpenVMS license units have been consumed. (This forces all node service ratings to the lowest possible value when logins are not possible because all OpenVMS license units have been consumed.)

Note that the LAT software transmits a service announcement message when a user logs in to or out of the system. This allows the system to more quickly provide information about service rating changes that result from a login or logout operation.

## **Description**

The SET SERVICE command dynamically changes the characteristics of a service that you created previously (by interactively entering the CREATE SERVICE command or by running a program that created services).

## **Examples**

1. LATCP> SET SERVICE SALES /IDENT="SALES FORCE TIMESHARING SERVICES"

This command specifies a new identification string, "SALES FORCE TIMESHARING SERVICES", for the service SALES. This string is announced with the service SALES in the multicast messages sent by a service node.

2. \$ LCP := \$LATCP  
\$ LCP SET SERVICE/LIMITED ONLY ONE  
\$ LCP CREATE PORT/LIMITED LTA1234:  
\$ LCP SET PORT LTA1234: /SERVICE=ONLY\_ONE

This series of commands changes an existing service to a limited service that allows only one user to log in to the system through that service. When a user connects to service ONLY\_ONE by responding to the terminal server prompt (Local>), the user is assigned port LTA1234 and then prompted for the user name. Any user who attempts to connect to the same service while LTA1234 has a user logged in receives the "service in use" message.

# LATCP SHOW LINK

---

## SHOW LINK

Displays the status and LAT characteristics of links on the local node.

### Format

SHOW LINK [link-name]

### Parameter

#### link-name

Specifies the name for a LAT data link. A link name can have up to 16 ASCII characters.

If you do not specify a link name, LATCP displays information about all links currently defined for the node.

### Qualifiers

#### /BRIEF

Displays the device name and state of the link. This is the default display.

#### /COUNTERS

Displays the device counters kept for the link. The numbers displayed represent the values recorded since the last time the counters were reset (when the node first started or when the ZERO COUNTERS command was used).

Do not use the /BRIEF or /FULL qualifier with this qualifier.

The following table lists and describes counters common to both CSMA/CD (carrier sense, multiple access with collision detect) and FDDI (Fiber Distributed Data Interface) links:

Counter	Description
Messages received	The total number of messages received over the link.
Multicast messages received	The total number of multicast messages received over the link.
Bytes received	The total number of bytes of information received over the link.
Multicast bytes received	The total number of multicast bytes received over the link.
System buffer unavailable	The total number of times no system buffer was available for an incoming frame.
Unrecognized destination	The total number of times a frame was discarded because there was no portal with the protocol enabled. This count includes frames received for the physical address only.
Messages sent	The total number of messages sent over the link.
Multicast messages sent	The total number of multicast messages sent over the link.

<b>Counter</b>	<b>Description</b>
Bytes sent	The total number of bytes of information sent over the link.
Multicast bytes sent	The total number of bytes of multicast messages sent over the link.
User buffer unavailable	The total number of times no user buffer was available for an incoming frame that passed all filtering.
Data overrun	The total number of bytes lost on the link's device because the local node's input buffers were full. A nonzero value can indicate noisy lines, a bad device, a busy or poorly tuned system (not enough resources allocated), or a hardware problem with another device on the LAN connection.

The following table lists and describes receive errors common to both CSMA/CD and FDDI links. These errors, which are included in the display generated by the SHOW LINK/COUNTERS command, are represented by flags that indicate the error has occurred.

<b>Flag</b>	<b>Description</b>
Block check error	CRC error in packets received.
Framing error	Received frames ended incorrectly.
Frame too long	Frames received longer than length limits.
Frame status error	CRC error on ring noticed by local FDDI station (FDDI only).
Frame length error	Frame length too short (FDDI only).

The following table lists and describes transmit errors common to both CSMA/CD and FDDI links. These errors, which are included in the display generated by the SHOW LINK/COUNTERS command, are represented by flags that indicate the error has occurred.

<b>Flag</b>	<b>Description</b>
Excessive collisions	Frames failed to transmit because the collision limit of 16 was reached (CSMA/CD only).
Carrier check failures	Indicates transceiver problem or short circuit in cable.
Short circuit	Short circuit in cable.
Open circuit	Open circuit in cable.
Frame too long	Frames too long. Indicates a transmission problem in one of the portals using the link.
Remote failure to defer	A remote station failed to defer frames transmission. Could indicate a misconfigured network.

## LATCP SHOW LINK

Flag	Description
Transmit underrun	Transmission of a frame was too slow. Indicates a hardware controller error.
Transmit failure	Frames failed to transmit.

The following table lists and describes link counters specific to CSMA/CD only:

Counter	Description
Transmit CDC failure	The total number of carrier detect check errors, that is, the number of times the local node failed to detect that another Ethernet station was already transmitting when the local node began transmitting.
Messages transmitted:	<p><b>Single collision</b>—The total number of times a frame was successfully transmitted on the second attempt after a normal collision on the first attempt.</p> <p><b>Multiple collision</b>—The total number of times a frame was successfully transmitted on the third or later attempt after normal collisions on previous attempts.</p> <p><b>Initially deferred</b>—The total number of times a frame transmission was deferred on its first attempt. This counter is used to measure Ethernet contention with no collisions.</p>

The following table lists and describes link counters specific to FDDI only:

Counter	Description
Ring initializations initiated	The total number of times a ring reinitialization was initiated by the link.
Ring initializations received	The total number of times a ring reinitialization was initiated by some other link.
Directed beacons received	The number of times the link detected the directed beacon process. Each invocation of the directed beacon process is counted only once.
Connections completed	The number of times the station successfully connected to the concentrator.
Duplicate tokens detected	The number of times a duplicate token was detected on the link.
Ring purge errors	The number of times the ring purger received a token while still in the ring purge state.



Counter	Description
LCT rejects	Link Confidence Test rejects. Indicates a problem with communication between station and concentrator.
Elasticity buffer errors	Elasticity buffer function errors. Indicates a station on the ring with a transmit clock out of tolerance.
MAC error count	The number of times the Media Access Control (MAC) changed the E indicator in a frame from R to S.
Traces initiated	The number of times the PC-trace process was initiated by the link.
Traces received	The number of times the link was requested to perform the PC-trace process.
Ring beacons initiated	The number of times the ring beacon process was initiated by the link.
Link errors	The number of times the Link Error Monitor (LEM) detected an error in a received message. Slow counts are normal.
Duplicate address test failures	The number of times the link address was a duplicate.
FCI strip errors	The number of times a Frame Content Independent Strip operation was terminated by receipt of a token.
LEM rejects	The number of times excessive LEM errors were encountered.
MAC frame count	The total number of frames (other than tokens) seen by the link.
MAC lost count	The total number of times a frame (other than a token) was improperly terminated.

**/FULL**

Displays the device name, state, and datalink address of the link and indicates whether the DECnet address is enabled.

**Description**

Displays information about the specified link or all links if you do not specify a link. Depending on the qualifier you use with the SHOW LINK command, you can display a link's device name, state, LAT datalink address, DECnet address, or counters.

# LATCP SHOW LINK

## Examples

1. LATCP> SHOW LINK/FULL NETWORK\_A

This command produces the following display of information about link NETWORK\_A:

```
Link Name:      NETWORK_A          Datalink Address: 08-00-2B-10-12-E3
Device Name:    _ESA7:             DECnet Address:   Disabled
Link State:     On
```

The display in this example gives the device name of link NETWORK\_A and the device's hardware address. The link is in the On state.

2. LATCP> SHOW LINK LINK\_A/COUNTERS

This command produces the following display of counters for link LINK\_A:

```
Link Name:      LINK_A
Device Name:    _ETA6:

Seconds Since Zeroed:      65535
Messages Received:        18582254
Multicast Msgs Received:  15096805
Bytes Received:           1994694325
Multicast Bytes Received: 1528077909
System Buffer Unavailable:  8724
Unrecognized Destination:  0

Messages Sent:            3550507
Multicast Msgs Sent:     413178
Bytes Sent:               290838585
Multicast Bytes Sent:    32637472
User Buffer Unavailable:  6269
Data Overrun:            0

Receive Errors -
Block Check Error:       No
Framing Error:          No
Frame Too Long:         No
Frame Status Error:     No
Frame Length Error:     Yes

Transmit Errors -
Excessive Collisions:   No
Carrier Check Failure:  No
Short Circuit:         No
Open Circuit:          No
Frame Too Long:        No
Remote Failure To Defer: No
Transmit Underrun:     No
Transmit Failure:      No
```

### CSMACD Specific Counters

```
-----
Transmit CDC Failure:    0

Messages Transmitted -
Single Collision:       43731
Multiple Collisions:   73252
Initially Deferred:    164508
```

---

## SHOW NODE

Displays the status and LAT characteristics of a node.

### Format

SHOW NODE [node-name]

### Parameter

#### **node-name**

Specifies the name of the node for which information is displayed. If you do not specify a node name, LATCP displays information about the local node.

You can also specify any valid wildcard for this parameter. For example, the SHOW NODE A\* command displays the status and characteristics of all nodes that begin with the letter A.

### Qualifiers

#### **/ALL**

Displays information about all nodes known to your local node. When you use this qualifier, specify the /FULL or /BRIEF qualifier as well. If you do not specify either the /FULL or /BRIEF qualifier, the default display will contain the node status and identification string (the display generated by the /BRIEF qualifier).

#### **/BRIEF**

Displays the node status and identification string. This is the default display if you specify the /ALL qualifier.

#### **/COUNTERS**

Displays the counters kept for the node. Do not use the /BRIEF or /FULL qualifier with this qualifier. The following table lists and describes the counters displayed with SHOW NODE/COUNTERS:

Counter	Description
Messages received	The total number of LAT messages received by the local node. If you specify a remote node with the SHOW NODE command, the number of LAT messages received from that remote node.
Messages transmitted	The total number of LAT messages transmitted by the local node. If you specify a remote node with the SHOW NODE command, the number of LAT messages transmitted to that remote node.
Slots received	The total number of LAT slots received by the local node. If you specify a remote node with the SHOW NODE command, the number of slots received from that remote node. A slot is a message segment that contains information corresponding to a single session.

## LATCP SHOW NODE

Counter	Description
Slots transmitted	The total number of LAT slots transmitted by the local node. If you specify a remote node with the SHOW NODE command, the number of slots transmitted to that remote node.
Bytes received	The total number of bytes of LAT information received by the local node. If you specify a remote node with the SHOW NODE command, the number of bytes received from that remote node.
Bytes transmitted	The total number of bytes of LAT information transmitted by the local node. If you specify a remote node with the SHOW NODE command, the number of bytes transmitted to that remote node.
Multicast bytes received	The total number of LAT multicast bytes received by the local node.
Multicast bytes sent	The total number of LAT multicast bytes sent by the local node.
Multicast messages received	The total number of LAT multicast messages received by the local node.
Multicast messages sent	The total number of LAT multicast messages sent by the local node.
No transmit buffer	The total number of times no buffer was available on the local node for transmission.
Multicast messages lost	The total number of times LTDRIVER failed to process an inbound multicast message because of failed communication with the LATACP.
Multicast send failures	The total number of times LTDRIVER failed to send a multicast message because of failed communication with the LATACP.
Controller errors	The total number of times LTDRIVER failed to communicate with the data link controller driver.
Last controller error	The most recent controller error.
Multiple node addresses	The total number of times that a node announced itself with a physical address different from that in a previous announcement.
Duplicates received	The total number of duplicate messages received by the local node. If you specify a remote node with the SHOW NODE command, the number of duplicate messages received from that remote node. This counter can indicate a system slowdown.

Counter	Description
Messages retransmitted	The total number of LAT messages that the local node retransmitted because they were not acknowledged by terminal servers (or nodes that support outgoing connections). If you specify a remote node with the SHOW NODE command, the number of messages retransmitted to that remote node.
Illegal messages received	The total number of invalidly formatted LAT messages received by the local node. If you specify a remote node with the SHOW NODE command, the number of invalidly formatted messages the local node received from that remote node. Illegal messages are grouped into several types of protocol errors, which are listed at the end of this table.
Illegal slots received	The total number of invalidly formatted LAT slots received by the local node. If you specify a remote node with the SHOW NODE command, the number of invalidly formatted slots the local node received from that remote node.
Solicitations accepted	The total number of times a remote node accepted solicitations from the local node. If you specify a remote node with the SHOW NODE command, the number of accepted solicitations by that remote node.
Solicitations rejected	The total number of times a remote node rejected solicitation from the local node. If you specify a remote node with the SHOW NODE command, the number of rejected solicitations by that remote node.
Solicitation failures	The total number of times solicitations by the local node received no response.
Transmit errors	The total number of times the data link failed to transmit a LAT message.
Last transmit error	The most recent transmit error.
Virtual circuit timeouts	The total number of times a LAT circuit to another node timed out, indicating that the remote node failed to send a valid message in the required time span. If you specify a remote node with the SHOW NODE command, the number of times the local node timed out from a connection to that remote node.
Discarded output bytes	The total number of data bytes that were discarded because of an overflow of an internal buffer before the data could be output to an LTA device.

## LATCP SHOW NODE

Counter	Description
User data lost	The total number of times LTDRIVER failed to allocate resources to buffer session data. User data is lost and the session is stopped.
Resource errors	The number of times LTDRIVER was unable to allocate system resources.
Incoming solicits accepted	The total number of times the local node accepted solicitations from other nodes.
Incoming solicits rejected	The total number of times the local node rejected solicitations from other nodes.

The protocol errors that are counted as illegal messages are as follows. These protocol error messages are displayed if their associated counter is greater than zero:

- Invalid message type received
- Invalid start message received
- Invalid sequence number received in start message
- Zero-node index received
- Node circuit index out of range
- Node circuit sequence invalid
- Node circuit index no longer valid
- Circuit was forced to halt
- Invalid server slot index
- Invalid node slot index
- Invalid credit field or too many credits used
- Repeat creation of slot by server
- Repeat disconnection of slot by master

### **/FULL**

Displays the node's status, identification string, LAT protocol version, and the values of the node's characteristics. This is the default except when you specify the /ALL qualifier.

### **/STATUS**

Displays statistical information for parameters such as the number of active circuits, sessions, and incoming queue entries. For each parameter, the display shows the current value, the highest value recorded, and the maximum value allowed.

Note that you can specify the /STATUS qualifier with the SHOW NODE command to display information about the *local* node only (for example, the command SHOW NODE /STATUS FOREIGN\_NODE is not supported).

## Description

This command displays information about a specified node or, if you do not specify a node name, about your local node. With the /ALL qualifier, the SHOW NODE command displays information about all nodes known to your local node. Depending on the qualifiers you use, you can display node counters, node status, the node identification string, the LAT protocol version running on the node, and the values set for the node's characteristics.

## Examples

1. LATCP> SHOW NODE/FULL

This command produces the following display of information about the local node:

```

Node Name:   LTC                               LAT Protocol Version:  5.2
Node State:  On
Node Ident:  LTC - Engineering Development

Incoming Connections: Enabled                 Incoming Session Limit:  None
Outgoing Connections: Enabled                 Outgoing Session Limit:  None
Service Responder:   Disabled

Circuit Timer (msec):      80                 Keepalive Timer (sec):   20
Retransmit Limit (msg):    20                 Node Limit (nodes):     None
Multicast Timer (sec):     20                 CPU Rating:              8
Maximum Unit Number:      9999

User Groups:   43, 73
Service Groups: 7-9, 13, 23, 40, 43, 45, 66, 72-73, 89, 120-127, 248-255

Service Name  Status   Rating  Identification
LTVMS        Available  31 D   .

```

This display indicates that the local node LTC is in the On state, which means LAT connections can be created on the node. LTC is running Version 5.2. of the LAT protocol. The identification of the node is "LTC - Engineering Development". Because this is the local node, the display does not give the address of a LAN device. Use the SHOW LINK command to find addresses of devices on the local node. The display for the status of remote nodes, as shown in Example 2, gives the Ethernet address of that node.

Both incoming and outgoing connections can be made on node LTC, the number of sessions is unlimited. The display indicates the values of various timers and lists the groups that are enabled. Users on the local node can access service nodes belonging to user groups 43 and 73. Locally offered services can be accessed by nodes belonging to the service groups listed.

The display indicates that the CPU rating of the local node is 8. The display shows that the node offers a service named LTVMS. This service is available and its rating is 31 D (dynamic). (An S would indicate the rating is static.)

2. LATCP> SHOW NODE/FULL RWWUP

This command displays the following information about the remote node RWWUP:

# LATCP SHOW NODE

```
Node Name:   RWWUP                      LAT Protocol Version: 5.2
Node State:  Reachable                   Address:    AA-00-04-00-11-10
Node Ident:  .
```

Incoming Connections: Enabled

```
Circuit Timer (msec): 80
Multicast Timer (sec): 20
```

Service Groups: 7, 13, 42-43, 45, 66, 70-72, 75-82, 88-89

Service Name	Status	Rating	Identification
NAC	Available	28	.
SYSMGR	Available	28	.

This display indicates that remote node RWWUP is reachable and runs Version 5.2 of the LAT protocol. The display includes the Ethernet address of node RWWUP. Because incoming connections are enabled, you can connect to a service on node RWWUP, provided that your node belongs to one of the service groups listed in the display.

Node RWWUP offers two services: NAC and SYSMGR. Both are available.

### 3. LATCP> SHOW NODE/ALL/BRIEF

This command displays the following information about all nodes known to the local node:

Node Name	Status	Identification
ABLAN	Reachable	Unauthorized access is prohibited.
ASKWEN	Reachable	.
CHUNK	Reachable	A member of the MAIN VMScLuster
.	.	.
UTOO	On	Can be healthy at the Center
VULCUN	Reachable	Beam me up
ZENX	Reachable	ZENX

This command indicates the status (whether a node is reachable) and identification of all nodes known to the local node. The display includes the status of the local node UTOO. The status can be either On, Off, or Shut. Here it is On.

### 4. \$ LCP := \$LATCP \$ LCP SHOW NODE /STATUS

The SHOW NODE /STATUS produces the following display:

```
Node Name:   NODE1                      LAT Protocol Version: 5.2
Node State:  On
Node Ident:  Test system
```

	Current	Highest	Maximum
Active Circuits:	1	2	1023
Connected Sessions:	1	6	260865
Incoming Queue Entries:	0	0	24
Outgoing Queue Entries:	0	1	32767
Unprocessed Announcements:	0	7	500
Unprocessed Solicits:	0	2	250
Local Services:	1	2	255
Available Services:	188	194	N/A
Reachable Nodes:	166	172	N/A
Discarded Nodes:	0		



---

## SHOW PORT

Displays the status and LAT characteristics of ports on the local node.

### Format

SHOW PORT [port-name]

### Parameter

#### **port-name**

Specifies the name of the port for which information is displayed. If you do not specify a port name, the SHOW PORT command displays the characteristics for all LTA $n$ : ports on a node.

Do not use the /APPLICATION, /DEDICATED, /FORWARD, /INTERACTIVE, or /LIMITED qualifiers with a specific port name.

### Qualifiers

#### **/APPLICATION**

Generates a display of all application ports.

#### **/BRIEF**

Displays port type, port status, and the remote node name, port, and service associated with the port. This is the default if you do not specify a port name with the SHOW PORT command.

#### **/COUNTERS**

Displays the counters kept for the port. Do not use the /BRIEF or /FULL qualifiers with this qualifier.

#### **/DEDICATED**

Generates a display of all dedicated ports.

#### **/FORWARD**

Generates a display of all LAT ports used for either outgoing LAT connections or local LAT management functions.

#### **/FULL**

Displays the following information:

- Port type
- Port status
- Target port name, node name, and service name associated with the port
- Remote node name, port, and service associated with the port if a connection is currently active

#### **/INTERACTIVE**

Generates a display of all LAT ports used for incoming interactive connections.

#### **/LIMITED**

Generates a display of all limited LTA devices on the system (previously established with the CREATE PORT /LIMITED or SET PORT /LIMITED command).

# LATCP SHOW PORT

## Description

If a port is an application port, the display lists the remote node name, remote port name, and remote service name that you specified in the SET PORT command.

If the port is a dedicated port, the display lists the service name that you specified in the SET PORT command.

If LATCP shows the port as Interactive in the display, a user on a terminal server or on a node that supports outgoing LAT connections is currently using the port.

For all ports with active sessions, the remote node sends its node name and port name to your local node. These names are listed in the display.

## Examples

1. LATCP> SHOW PORT /FULL

This command produces the following type of display. The display reflects the characteristics set by the command examples given with the SET PORT command.

```
Local Port Name:  LTA16:           Local Port Type:  Forward
Local Port State: Inactive
Connected Link:
```

```
Target Port Name:           Actual Port Name:
Target Node Name:  LATCP$MGMT_PORT Actual Node Name:
Target Service Name:           Actual Service Name:
```

```
-----
Local Port Name:  LTA17:           Local Port Type:  Interactive
Local Port State: Active
Connected Link:  LAT$LINK
```

```
Target Port Name:           Actual Port Name:  PORT 1
Target Node Name:           Actual Node Name:  MY_DS200_SERVER
Target Service Name:           Actual Service Name:
```

```
-----
Local Port Name:  LTA19:           Local Port Type:  Application (Queued)
Local Port State: Active
Connected Link:  LAT$LINK
```

```
Target Port Name:           Actual Port Name:
Target Node Name:  TLAT1          Actual Node Name:  TLAT1
Target Service Name:  PRINTER     Actual Service Name:  PRINTER
```

```
-----
Local Port Name:  LTA21:           Local Port Type:  Dedicated
Local Port State: Inactive
Connected Link:
```

```
Target Port Name:           Actual Port Name:
Target Node Name:           Actual Node Name:
Target Service Name:  GRAPHICS    Actual Service Name:
```

```
-----
Local Port Name:  LTA22:           Local Port Type:  Application (Queued)
Local Port State: Active
Connected Link:  LAT$LINK
```

```
Target Port Name:  LN02           Actual Port Name:  LN02
Target Node Name:  TS33EW         Actual Node Name:  TS33EW
Target Service Name:           Actual Service Name:
```

## LATCP SHOW PORT

The display in this example shows information about all the ports on the local node. The display shows information for each of the four types of ports:

- **Forward:** a port used for outgoing LAT connections or for executing local management functions and LATCP commands. Port LTA16: is a forward port. The display shows that the port is currently inactive—no current LAT connection exists. The target node name of LATCP\$MGMT\_PORT indicates that LATCP is using this port to execute the LATCP commands entered by the user. If the display listed a node and service name, it would mean that the port is being used for an outgoing connection.
- **Interactive:** a port created as a result of an incoming LAT connection request from another node or terminal server. Port LTA17: is an interactive port connected with port PORT\_1 on the terminal server MY\_DS200\_SERVER.
- **Application:** a port used for solicited connections to devices on terminal servers or to application services on remote LAT service nodes. Port LTA22: is an application port. The port maps to port LN02 (a printer) on a terminal server node TS33EW. The display indicates that server TS33EW queues connection requests from the local node. Port LTA19: is also an application port. The port maps to the service PRINTER on terminal server TLAT1.
- **Dedicated:** a port dedicated to a local application service. Port LTA21: is dedicated to the service GRAPHICS.

The target port name, target node name, and target service name are the names specified with the SET PORT command. They are passed to the remote node or terminal server when the connection request is made.

The actual port name, actual node name, and actual service name are the names returned by the remote node when it accepts the connection request. They may differ from the corresponding target names (specified with the SET PORT command) if the remote node translates the names. For example, terminal servers that accept connections to LAT service names usually return the name of the port to which the connection was actually directed.

### 2. LATCP> SHOW PORT LTA1 /COUNTERS

This command produces a display that lists counter information for the LTA1 device:

```
Port Name:  _LTA1:
Seconds Since Zeroed:          66
Remote Accesses:               0   Framing Errors:                0
Local Accesses:               0   Parity Errors:              0
Bytes Transmitted:            0   Data Overruns:             0
Bytes Received:               0   Password Failures:         0
Solicitations Accepted:       1
Solicitations Rejected:       1
Incoming Solicits Accepted:    0
Incoming Solicits Rejected:    0
Last disconnect reason code:   18
(%LAT-F-LRJDELETED, queue entry deleted by server)
```

## LATCP SHOW QUEUE\_ENTRY

---

### SHOW QUEUE\_ENTRY

Displays information about requests, or entries, queued on the local node.

#### Format

```
SHOW QUEUE_ENTRY [queue-entry-id]
```

#### Parameter

##### **queue-entry-id**

Specifies the identification number (ID) of the queued entry for which information is displayed. If you do not specify a value for this parameter, information about all queued entries is displayed.

#### Qualifiers

##### **/BRIEF**

Displays the following information about the queued entries:

- Position
- Entry ID
- Source node
- Service
- Port name

This is the default display.

##### **/FULL**

In addition to the information displayed by the /BRIEF qualifier, the /FULL qualifier provides the following information for each node:

- Node queue position
- Service queue position
- Node address
- Soliciting Link

#### Description

The SHOW QUEUE\_ENTRY command displays information about requests, or entries, queued on the local node. You can display information about a specific entry by including the queue entry ID on the command line or you can display information about all entries (the default). Use the DELETE QUEUE\_ENTRY command to delete specific entries from the queue.

## Examples

1. LATCP> SHOW QUEUE\_ENTRY

This command produces the following type of display:

Position	Entry ID	Source Node	Service	Port Name
1	79EC	NODE1	LAT_LIMITED	
2	7AEC	NODE2	LAT_LIMITED	
3	7CEC	NODE3	LAT_LIMITED	

2. LATCP> SHOW QUEUE\_ENTRY/FULL

This command produces the following type of display:

Entry ID: 7AEC Remote Node: NODE1  
Node Queue Position: 1 Address: 08-00-2B-0A-A0-A0  
Service Queue Position: 1

Target Port:  
Target Service: LAT\_LIMITED  
Soliciting Link: LAT\$LINK

---

Entry ID: 7CEC Remote Node: NODE2  
Node Queue Position: 2 Address: AA-00-04-00-37-DD  
Service Queue Position: 2

Target Port:  
Target Service: LAT\_LIMITED  
Soliciting Link: LAT\$LINK

# LATCP SHOW SERVICE

---

## SHOW SERVICE

Displays the status and LAT characteristics of LAT services known to the local node.

### Format

SHOW SERVICE [service-name]

### Parameters

#### **service-name**

Specifies the name of the service for which information will be displayed. If you do not specify a service name, LATCP displays information about all services known to the node.

You can also specify any valid wildcard for this parameter. For example, the SHOW SERVICE LAT\_\* command displays the status and characteristics of all services that begin with the LAT\_ prefix.

### Qualifiers

#### **/BRIEF**

Displays the status and identification string of the service.

#### **/COUNTERS**

Displays the counters kept for the service. Do not use the /BRIEF or /FULL qualifier with this qualifier. The following table lists and describes the counters:

<b>Counter</b>	<b>Description</b>
<b>Remote Counters</b>	
Connections attempted	The total number of times the local node attempted to connect to the service offered on a remote node.
Connections completed	The total number of times the local node successfully connected to the service offered on a remote node.
<b>Local Counters</b>	
Connections accepted	The total number of times the local node accepted a connection request from a remote node to a locally offered service.
Connections rejected	The total number of times the local node rejected a connection request from a remote node to a locally offered service.
Password failures	The total number of connect requests to the service which were rejected due to password violation errors.

**/FULL**

Displays the status, identification string, and type of service, and the values set for service characteristics. This qualifier also displays the status of all service nodes offering the service.

**/LOCAL**

Displays information about services offered by the local node only. You can use this qualifier with the /BRIEF, /COUNTERS, or /FULL qualifier.

**Description**

This command displays information about services. If you do not specify a service name, the command displays information about all services known to your local node. If you do not specify a service name but specify the /LOCAL qualifier, the command displays information about all services offered by your local node.

Depending on whether you use the /BRIEF, /COUNTERS, or /FULL qualifier, you can display the status, identification string, and type of service, the status of all service nodes offering the service, the values set for service characteristics, and service counters.

**Examples**

1. LATCP> SHOW SERVICE NODE1 /FULL

This command produces the following display of information about service NODE1. This service is offered by the local node.

```
Service Name:  NODE1                Service Type:  General
Service Status: Available           Connections:  Enabled
Service Password: Enabled          Queueing:     N/A
Service Ident:  NODE1 - Test system
```

Node Name	Status	Rating	Identification
LAV	On	31 D	.
LATP	Reachable	48	.
LITTN	Reachable	37	.
LTDRV	Reachable	82	.

The display in this example indicates that the locally offered service NODE1 is available and its service type is general, meaning that it is a general timesharing service (in contrast to a dedicated application service). The display also lists the status of all the nodes that offer the service. The local node is LAV. The status of the local node can be either On, Off, or Shut. Here node LAV's status is On. The status of the other nodes indicates whether they are reachable. The display lists the ratings of each service node, indicating their relative capacity to accept new connections. The D next to the locally offered service indicates that node LAV computes its rating dynamically. An S would indicate that the node's rating was set permanently by the node's system manager.

## LATCP SHOW SERVICE

2. LATCP> SHOW SERVICE OFFICE/FULL

This command produces the following display of information about the service OFFICE, which is offered by a remote node:

```
Service Name: OFFICE
Service Status: Available
Service Ident: .
```

Node Name	Status	Rating	Identification
BURGIL	Reachable	121	.
DARWIN	Reachable	43	.

The display in this example indicates that the service is available. The display also indicates the status and other information about the nodes that offer the service, BURGIL and DARWIN.



---

**SPAWN**

Creates a subprocess, enabling you to execute DCL commands without terminating your LATCP session. The LATCP command SPAWN is similar to the DCL command SPAWN.

To return to your LATCP session, either log out of the subprocess by entering the DCL command LOGOUT, or use the DCL command ATTACH to attach your terminal to the process running LATCP.

**Format**

SPAWN [DCL-command]

**Parameter****DCL-command**

Specifies a DCL command. If you specify a DCL command, LATCP executes the command in a subprocess. Control returns to LATCP when the DCL command terminates.

If you do not specify a DCL command, LATCP creates a subprocess and you can then enter DCL commands. You can continue your LATCP session by logging out of the spawned subprocess or by attaching to the parent process with the DCL command ATTACH.

**Description**

The SPAWN command acts exactly like the DCL command SPAWN. You can enter DCL commands (such as to create print queues, change the protection of a device, answer mail, and so forth) without ending your LATCP session.

You cannot use this command to gain access to DCL if you are running LATCP from a captive account.

**Example**

```
LATCP> SPAWN  
$
```

This command creates a subprocess at DCL level. You can now enter DCL commands. Log out or enter the DCL command ATTACH to return to the LATCP prompt.

# LATCP ZERO COUNTERS

---

## ZERO COUNTERS

Resets the link, node, and service counters maintained by the local node. You must have OPER privilege to use this command.

### Format

ZERO COUNTERS

### Parameters

None.

### Qualifiers

**/LOG**

**/NOLOG (default)**

Specifies whether LATCP displays a message confirming that the counters were reset. If you do not specify the /LOG or /NOLOG qualifier, the default is that no message will be displayed.

**/LINK[=*link-name*]**

Specifies the link (on your local node) for which you want counters reset. If you do not specify a link name, LATCP zeroes counters for the link LAT\$LINK.

**/NODE[=*node-name*]**

Specifies the node for which you want counters reset. If you do not specify a node name, LATCP zeroes the counters for your local node.

**/PORT=*port-name***

Specifies the port (on your local node) for which you want counters reset.

**/SERVICE=*service-name***

Specifies the service (on your local node) for which you want counters reset.

### Description

This command resets counters. You can specify whether you want to reset link, node, or service counters. You must specify either /LINK, /NODE, or /SERVICE.

### Example

```
LATCP> ZERO COUNTERS/SERVICE=LTVM  
LATCP> SHOW SERVICE LTVM /COUNTERS
```

```
Service Name: LTVM
```

```
Seconds Since Zeroed:          9  
Connections Attempted:        0   Connections Accepted:          0  
Connections Completed:        0   Connections Rejected:          0  
Password Failures:            0
```

This command resets the counters kept for service LTVM. The display produced by the SHOW SERVICE command shows how the ZERO COUNTERS command reset the counters to zero.

---

**Log Manager Control Program (LMCP) Utility**



## 14.1 LMCP Description

The Log Manager Control Program (LMCP) utility creates and manages the transaction logs used by DECdtm services.

---

**Caution**

---

Some LMCP commands can corrupt data.

Refer to the *HP OpenVMS System Manager's Manual* to understand the reasons for using LMCP and how to use it safely.

---

## 14.2 LMCP Usage Summary

LMCP lets you create and manage the transaction logs used by DIGITAL's distributed transaction manager, DECdtm services.

### Format

RUN SYS\$SYSTEM:LMCP

#### Parameters

None

### Description

To invoke LMCP, enter RUN SYS\$SYSTEM:LMCP at the DCL command prompt. At the LMCP> prompt, you can enter any of the LMCP commands described in the following section.

To exit from LMCP, enter the EXIT command at the LMCP> prompt, or press Ctrl/Z.

## 14.3 LMCP Commands

The following table summarizes the LMCP commands:

Command	Description
CLOSE LOG	Closes the transaction log and stops the TP_SERVER process
CONVERT LOG	Creates a new transaction log and copies records from an existing transaction log to the new transaction log
CREATE LOG	Creates a new transaction log
DUMP	Displays the contents of a transaction log
EXIT	Exits LMCP
HELP	Gives help on LMCP commands
REPAIR	Changes the state of transactions
SHOW LOG	Displays information about transaction logs

## LMCP CLOSE LOG

---

### CLOSE LOG

Closes the transaction log and stops the TP\_SERVER process.  
Requires the SYSNAM privilege.

### Format

CLOSE LOG

### Description

Use the CLOSE LOG command to:

- Close the transaction log of the local node.
- Stop the TP\_SERVER process on the local node.

The CLOSE LOG command fails if the node is currently executing transactions.

---

## CONVERT LOG

Creates a new transaction log and copies records from an existing transaction log to the new one.

Use the CONVERT LOG command when you want to move a transaction log or change its size.

---

### Caution

---

If a node already has a transaction log, using the CONVERT LOG command to create a new one can corrupt data. Refer to the *HP OpenVMS System Manager's Manual* for information about how to use the CONVERT LOG command safely.

---

The CONVERT LOG command requires:

- The CMKRNL privilege
- Read access to the existing transaction log and the directory it is in
- Read and write access to the directory in which the new transaction log is to be created

## Format

CONVERT LOG old-filespec new-filespec

## Parameters

### old-filespec

The file specification of the transaction log whose records are to be copied.

The CONVERT LOG command uses the following defaults:

- If you omit the disk and directory, the CONVERT LOG command looks for the transaction log in the directories pointed to by the logical SYS\$JOURNAL, which must be defined in executive mode in the system logical name table.
- If you omit the file type, the CONVERT LOG command uses .LM\$JOURNAL.

### new-filespec

The file specification of the new transaction log to be created.

For DECdtm services to use the transaction log, the file must have a name of the form SYSTEM\$node.LM\$JOURNAL, where *node* is the name of the node.

The CONVERT LOG command uses the following defaults:

- If you omit the disk and directory, the CONVERT LOG command creates the new transaction log in the first accessible directory pointed to by the logical SYS\$JOURNAL, which must be defined in executive mode in the system logical name table.
- If you omit the file type, the CONVERT LOG command uses .LM\$JOURNAL.

## LMCP CONVERT LOG

### Qualifiers

#### **/OWNER=uc**

Specifies the owner of the new transaction log.

Specify the owner using the standard UIC format, as described in the *OpenVMS User's Manual*.

#### **/SIZE=size**

Specifies the size of the new transaction log in blocks.

The minimum size is 100 blocks. If you omit this qualifier, the new transaction log is created with the default size of 4000 blocks.

### Example

```
LMCP> CONVERT LOG/SIZE=6000 DISK$LOG2:[LOGFILES]SYSTEM$RED.LM$OLD -  
_LMCP> DISK$LOG2:[LOGFILES]SYSTEM$RED.LM$JOURNAL
```

This example creates a 6000-block transaction log called `SYSTEM$RED.LM$JOURNAL` in directory `DISK$LOG2:[LOGFILES]`. It then copies records from the existing transaction log, `SYSTEM$RED.LM$OLD` in directory `DISK$LOG2:[LOGFILES]`, into the new transaction log.



---

## CREATE LOG

Creates a new transaction log.

---

### Caution

---

If a node already has a transaction log, using the CREATE LOG command to create a new one can corrupt data.

---

Requires read and write access to the directory in which the transaction log is to be created.

## Format

CREATE LOG filespec

## Parameter

### filespec

The file specification of the transaction log to be created.

For DECdtm services to use the transaction log, the file must have a name of the form SYSTEM\$node.LM\$JOURNAL, where *node* is the name of the node.

The CREATE LOG command uses the following defaults:

- If you omit the disk and directory, the CREATE LOG command creates the transaction log in the first accessible directory pointed to by the logical SYS\$JOURNAL, which must be defined in executive mode in the system logical name table.
- If you omit the file type, the CREATE LOG command uses .LM\$JOURNAL.

If you specify a disk and directory not pointed to by SYS\$JOURNAL, a warning message is displayed. However, the transaction log is still created, but will not be used until either (a) SYS\$JOURNAL is modified to point to the disk and directory where the log was created, or (b) you move the new transaction log to a directory pointed to by SYS\$JOURNAL.

## Qualifiers

### /NEW\_VERSION

Forces the CREATE LOG command to create a new version of an existing transaction log.

---

### Caution

---

Creating a new version of an existing transaction log can lead to data corruption.

The data in the two transaction logs cannot be merged. Once it has started using the new transaction log, DECdtm services cannot access any transaction records in the old transaction log.

---

### /OWNER=uic

Specifies the owner of the transaction log.

## LMCP CREATE LOG

Specify the owner using the standard UIC format, as described in the *OpenVMS User's Manual*.

### **/SIZE=size**

Specifies the size of the transaction log in blocks.

The minimum size is 100 blocks. If you omit this qualifier, the transaction log is created with the default size of 4000 blocks.

## Example

```
LMCP> CREATE LOG/SIZE=5000 DISK$LOG1:[LOGFILES]SYSTEM$ORANGE.LM$JOURNAL
```

This example creates a 5000-block transaction log for node ORANGE in DISK\$LOG1:[LOGFILES].

---

## DUMP

Displays the contents of a transaction log.

Requires read access to the transaction log and the directory it is in.

### Format

DUMP filespec

### Parameter

#### filespec

The file specification of the transaction log whose contents you want to display.

The DUMP command uses the following defaults:

- If you omit the disk and directory, the DUMP command looks for the transaction log in the directories pointed to by the logical SYS\$JOURNAL, which must be defined in executive mode in the system logical name table.
- If you omit the file type, the DUMP command uses .LM\$JOURNAL.

### Qualifiers

#### /ACTIVE

Selects records only for transactions that have not yet been forgotten.

#### /FORMAT (default)

#### /NOFORMAT

Determines whether the contents of the transaction log are displayed as formatted records. Specify both the /NOFORMAT and the /HEX qualifiers to display the contents of the transaction log in hexadecimal only.

If the /NOFORMAT qualifier is specified without the /HEX qualifier, only the transaction log header is displayed.

#### /HEX

#### /NOHEX (default)

Specifies that the contents of the transaction log are displayed as both ASCII characters and hexadecimal longwords. Specify both the /NOFORMAT and /HEX qualifiers to display the contents of the transaction log in hexadecimal only.

#### /LOGID=logid

Selects records only for transactions that have participants whose **logid** field matches the specified value.

The **logid** is in the *Log ID* field, to the right of the *Type* field. The value you specify must be exactly as it appears in the display, including hyphens.

Note that you can use this qualifier only with the /RM qualifier.

#### /OUTPUT[=filespec]

Requires read and write access to the directory in which the output file is to be created.

## LMCP DUMP

Specifies where the output from the DUMP command is sent. If you omit this qualifier, output is sent to the current SYS\$OUTPUT device (usually your terminal). To send the output to a file, use the /OUTPUT qualifier. If you do not supply a file specification, the output is sent to the file LMCP\_DUMP.LIS in your default directory.

### **/RM=name**

Selects records only for transactions that have participants whose names begin with the specified value.

The participant name is shown in the *Name* field, and is output in both ASCII and hexadecimal.

If the participant name includes undisplayable characters, you can select records for that participant by using the hexadecimal form of its name. When specifying the hexadecimal form of the name, you must convert it by reversing the pairs in the hexadecimal number. For example, the participant name is:

```
Name (11): "SYSTEM$RED" (4445 52244D45 54535953)
```

The value you specify for the /RM qualifier is:

```
/RM=%X53595354454D24524544
```

### **/STATE=COMMITTED**

### **/STATE=PREPARED**

Selects records only for transactions in either the Committed or Prepared states.

### **/TID=transaction\_id**

Selects records only for the specified transaction.

The **transaction\_id** is shown in the *Transaction ID* field. The value you specify must be exactly as it appears in the display, including hyphens.

## Description

Use the DUMP command to display the contents of a transaction log. Example 14–1 is a sample of a transaction log, with the important fields identified.

### **Example 14–1 Sample Transaction Log**

```
Log Manager Control Program V1.1
```

```
Dump of transaction log DISK$LOGFILE:SYSTEM$BLUE.LM$JOURNAL;1
```

```
End of file block 4002 / Allocated 4002
```

```
Log Version 1.0
```

```
Transaction log UID: 647327A0-2674-11C9-8001-AA00040069F8 ❶
```

```
Penultimate Checkpoint: 000000000239 0039
```

```
Last Checkpoint: 00000000042E 002E
```

```
Dump of transaction log DISK$LOGFILE:SYSTEM$BLUE.LM$JOURNAL;1
```

```
Present Length: 134 (00000086) Last Length: 0 (00000000)
```

```
VBN Offset: 0 (00000000) Virtual Block: 2 (00000002) ❷
```

```
Section: 3 (00000003)
```

(continued on next page)

**Example 14–1 (Cont.) Sample Transaction Log**

```

Record number 1 (00000001), 114 (0072) bytes
Transaction state (1): PREPARED
Transaction ID: 1D017140-2676-11C9-9F34-08002B174360
(8-JUL-2002 14:08:29.14)
DECdtm Services Log Format V1.1
Type ( 2): CHILD
Name (13): "SYSTEM$WESTRN" (4E 52545345 57244D45 54535953)
Type ( 8): CHILD NODE
Name (6): "WESTRN" (4E52 54534557)
Type ( 3): LOCAL RM
Name (6): "ORANGE" (4547 4E41524F)

```

In this example, the significant fields are:

- ❶ Transaction log header — information about the transaction log's attributes.
- ❷ Section header — the section header of multiple transaction records.
- ❸ Record number — the record number, in both decimal and hexadecimal.
- ❹ Record size — the record size in both decimal and hexadecimal.
- ❺ Transaction state — the type of the record. This can be:
  - Prepared
 

This type of record is logged when the transaction enters the Prepared state. Note that this type of record is not logged at the node on which the transaction was started.
  - Committed
 

This type of record is logged when the transaction enters the Committed state.
  - Forgotten
 

This type of record is logged:

    - When the transaction is aborted, if a record of type Prepared was logged for the transaction.
    - For a transaction that commits, when no participants require the local DECdtm transaction manager to remember that the outcome of the transaction is commit.

Note that DECdtm uses the presumed abort logging protocol.
  - Checkpoint
 

Unlike the other types of record, this is not associated with a particular transaction. It is used internally by the DECdtm transaction manager to compress space in the transaction log.
- ❻ Transaction ID — the unique transaction identifier (TID) generated by the DECdtm transaction manager.
- ❼ DECdtm Services Log Format — the version number of the transaction log format.

## LMCP DUMP

- ⑧ Type — information about the participant in the transaction. This can be:
  - Child — an immediate child transaction manager. This transaction manager may query the local DECdtm transaction manager to determine the outcome of the transaction.
  - Child Node — the name of the node that an immediate child transaction manager is on.
  - Parent — the immediate parent transaction manager. The local DECdtm transaction manager may query this transaction manager to determine the outcome of the transaction.
  - Parent Node — the name of the node that an immediate parent transaction manager is on.
  - Local RM — a resource manager on the local node.
- ⑨ Log ID — the identifier of the participant's log. For type Child, Child Node, Parent, or Parent Node, this is the identifier of the DECdtm transaction log. For a local resource manager, this is the identifier of its private log.
- ⑩ Name — the name of the participant in the transaction, in both ASCII and hexadecimal.

### Example

```
LMCP> DUMP/RM="RMS$" DISK$LOGFILE:SYSTEM$BLUE.LM$JOURNAL
```

This example displays the contents of the transaction log for node BLUE, selecting only transactions in which RMS Journaling for OpenVMS is participating.

**LMCP  
DUMP**

Dump of transaction log DISK\$LOGFILE:SYSTEM\$BLUE.LM\$JOURNAL;1  
End of file block 4002 / Allocated 4002  
Log Version 1.0  
Transaction log UID: 6A034B20-6FCC-0095-D7E4-EAA500000000  
Penultimate Checkpoint: 00000000382E 002E  
Last Checkpoint: 000000003C2E 002E

Dump of transaction log DISK\$LOGFILE:SYSTEM\$BLUE.LM\$JOURNAL;1  
Present Length: 46 (0000002E) Last Length: 512 (00000200)  
VBN Offset: 30 (0000001E) Virtual Block: 32 (00000020)  
Section: 1 (00000001)

Record number 2 (00000002), 5 (0005) bytes  
Transaction state (3): CHECKPOINT  
Checkpoint record contains no active transactions.

Record number 1 (00000001), 21 (0015) bytes  
Transaction state (0): FORGOTTEN  
Transaction ID: 271D9FC0-7082-0095-98E7-EAA500000000

Dump of transaction log DISK\$LOGFILE:SYSTEM\$BLUE.LM\$JOURNAL;1  
Present Length: 113 (00000071) Last Length: 512 (00000200)  
VBN Offset: 29 (0000001D) Virtual Block: 31 (0000001F)  
Section: 2 (00000002)

Record number 1 (00000001), 93 (005D) bytes  
Transaction state (2): COMMITTED  
Transaction ID: 271D9FC0-7082-0095-98E7-EAA500000000 (3-MAR-2002 13:53:03.42)  
DECdtm Services Log Format V1.1  
Type ( 2): CHILD Log ID: EF006060-CF37-11C9-8001-AA000400DEFA  
Name (10): "SYSTEM\$ORANGE" (45 474E4152 4F244D45 54535953)  
Type ( 8): CHILD NODE Log ID: EF006060-CF37-11C9-8001-AA000400DEFA  
Name ( 6): "ORANGE" (4547 4E41524F)  
Type ( 3): LOCAL RM Log ID: 28C5D180-7082-0095-0000-000000000000  
Name (22): "RMS\$USER1.....'"....."  
(0000 00178B60 00000000 00000031 52455355 24534D52)

.  
.  
.

Total of 1 transactions active, 0 prepared and 1 committed

**LMCP**  
**EXIT**

---

**EXIT**

Exits LMCP.

**Format**

EXIT



## HELP

Provides help on LMCP commands.

### Format

HELP [help-topic [help-subtopic]]

### Parameter

**help-topic**

Specifies the command that you want help for.

**help-subtopic**

Specifies the parameter or qualifier that you want help for.

---

## REPAIR

Changes the state of transactions.

---

### Caution

---

The REPAIR command can corrupt data. Use it only if none of the resource managers participating in the transaction provides a means of changing transaction states.

---

The REPAIR command requires:

- The CMKRNL privilege
- Read and write access to the transaction log and the directory it is in

## Format

REPAIR filespec

## Parameter

### filespec

The file specification of the transaction log containing the transactions whose states you want to change.

The REPAIR command has the following requirements:

- The logical SYS\$JOURNAL must be defined in executive mode in the system logical name table.
- The transaction log must be in a directory pointed to by the logical SYS\$JOURNAL.
- The file type of the transaction log must be .LM\$JOURNAL.

The REPAIR command uses the following defaults:

- If you omit the disk and directory, the REPAIR command looks for the transaction log in the directories pointed to by the logical SYS\$JOURNAL.
- If you omit the file type, the REPAIR command uses .LM\$JOURNAL.

## Qualifiers

### /LOGID=logid

Selects records only for transactions that have participants whose **logid** field matches the specified value.

The **logid** is in the *Log ID* field, to the right of the *Type* field in the output from the DUMP command. The value you specify must be exactly as it appears in the display, including hyphens.

Note that you can use this qualifier only with the /RM qualifier.

### /RM=name

Selects records only for transactions that have participants whose names begin with the specified value.

The participant name is shown in the *Name* field in the output from DUMP, and is output in both ASCII and hexadecimal.

If the participant name includes undisplayable characters, you can select records for that participant by using the hexadecimal form of its name. When specifying the hexadecimal form of the name, you must convert it by reversing the pairs in the hexadecimal number. For example, the participant name is:

```
Name (11): "SYSTEM$RED" (4445 52244D45 54535953)
```

The value you specify for the /RM qualifier is:

```
/RM=%X53595354454D24524544
```

**/STATE=COMMITTED**

**/STATE=PREPARED**

Selects records only for transactions in either the Committed or Prepared states.

**/TID=transaction\_id**

Selects records only for the specified transaction.

The **transaction\_id** is shown in the *Transaction ID* field in the output from the DUMP command. The value you specify must be exactly as it appears in the display, including hyphens.

## Description

Use the REPAIR command to change the state of transactions.

---

### Caution

---

The REPAIR command can corrupt data. Use it only if none of the resource managers participating in the transaction provides a means of changing transaction states.

---

Use this command only if none of the resource managers participating in the transaction provides a means of changing the transaction state.

Change the transaction state only when you already know the outcome of the transaction and need to manually update the transaction log immediately. You might want to do this because, for example, you have lost the network link to a remote node.

When you use the REPAIR command you use qualifiers to specify which transactions you want to change. By default, the REPAIR command selects all transactions.

Once you have selected the transactions to change, enter the REPAIR subcommand mode. Within this mode, the prompt changes to REPAIR>, and you have an additional set of subcommands. Use these subcommands either to manually change the state of the transaction or to select the next transaction that matches your selection criteria. The subcommands are as follows:

## LMCP REPAIR

Subcommand	Action
ABORT	Specifies that a Prepared transaction is to be aborted by removing its record from the transaction log. This writes a record of type Forgotten for the transaction. Note that DECdtm services use the presumed abort logging protocol.
COMMIT	Specifies that a Prepared transaction is to be committed. This writes a record of type Committed for the transaction.
EXIT	Returns to the LMCP> prompt.
FORGET	Specifies that a Committed transaction can be removed from the transaction log. This writes a record of type Forgotten for the transaction.
NEXT	Displays the next transaction that matches your selection criteria.

LMCP displays each of the selected transactions in turn, so that you can change them. For each selected transaction, you can either use the ABORT, COMMIT, and FORGET subcommands to change the state of the transaction, or use the NEXT subcommand to select the next transaction.

To exit from the REPAIR subcommand mode, enter the EXIT subcommand or press Ctrl/Z.

### Example

```
LMCP> REPAIR/STATE=PREPARED DISK$JOURNALS:[LOGFILES]SYSTEM$ORANGE
```

In this example, transactions to be modified are selected from the transaction log for node ORANGE. The transactions selected are those in the Prepared state.

The first transaction is committed by manually changing its state from Prepared to Committed, then the NEXT subcommand is used to advance to the next selected transaction.

```
Dump of transaction log DISK$JOURNALS:[LOGFILES]SYSTEM$ORANGE;1
End of file block 4002 / Allocated 4002
Log Version 1.0
Transaction log UID: 98A43B80-81B7-11CC-A27A-08002B1744C3
Penultimate Checkpoint: 00000407B9AC 07AC
Last Checkpoint: 00000407C3B7 07B7

Transaction state (1): PREPARED
Transaction ID: 9F7DF804-CBC4-11CC-863D-08002B17450A (18-OCT-2002 16:11:03.67)
DECdtm Services Log Format V1.1
Type ( 3): LOCAL RM Log ID: 00000000-0000-0000-0000-000000000000
Name (1): "B" (42)
Type ( 4): PARENT Log ID: AEC2FB64-C617-11CC-B458-08002B17450A
Name (13): "SYSTEM$BLUE" (45554C 42244D45 54535953)
Type (16): PARENT NODE Log ID: AEC2FB64-C617-11CC-B458-08002B17450A
Name (6): "BLUE" (45554C42))

REPAIR> COMMIT
REPAIR> NEXT
.
.
.
```

---

## SHOW LOG

Displays information about transaction logs.

Requires read access to the transaction logs and the directories they are in.

### Format

SHOW LOG [filespec]

### Parameter

#### **filespec**

The file specification of the transaction logs you want to display information about. This can include the percent (%) and asterisk (\*) wildcard characters.

The SHOW LOG command uses the following defaults:

- If you omit the disk and directory, the SHOW LOG command looks for the transaction log in the directories pointed to by SYS\$JOURNAL, which must be defined in executive mode in the system logical name table.
- If you omit the file type, the SHOW LOG command uses .LM\$JOURNAL.

### Qualifiers

#### **/CURRENT**

Displays information about the local node's transaction log. This includes the number of checkpoints and stalls that have occurred since DECdtm services started on this node.

To use the /CURRENT qualifier:

- You must have the CMKRNL privilege.
- You must omit the parameter to the SHOW LOG command.

#### **/FULL**

Lists all attributes of the transaction logs. For each transaction log, both the full file specification of the transaction log and its size are displayed.

If you do not specify which transaction log you want to display, the SHOW LOG command lists all transaction logs of the form SYSTEM\$\*.LM\$JOURNAL, in all directories pointed to by the logical SYS\$JOURNAL, which must be defined in executive mode in the system logical name table.

#### **/OUTPUT[=filespec]**

Requires read and write access to the directory in which the output file is to be created.

Specifies where the output of the SHOW LOG command is sent. If you omit this qualifier, output is sent to the current SYS\$OUTPUT device (usually your terminal). To send the output to a file, use the /OUTPUT qualifier. If you do not supply a file specification, the output is sent to the file LMCP\_SHOW.LIS in your default directory.

# LMCP SHOW LOG

## Example

```
LMCP> SHOW LOG/FULL
```

This example displays full details about the transaction logs in all directories pointed to by the logical SYS\$JOURNAL. This logical is defined in executive mode in the system logical name table.

```
Directory of DISK$JOURNALS:[LOGFILES]
DISK$JOURNALS:[LOGFILES]SYSTEM$BLUE.LM$JOURNAL;1
End of file block 4002 / Allocated 4002
Log Version 1.0
Transaction log UID: 647327A0-2674-11C9-8001-AA00040069F8
Penultimate Checkpoint: 000000001A39 0039
Last Checkpoint: 000000001C8A 008A

Total of 1 file.

Directory of DISK$RED:[LOGFILES]
DISK$RED:[LOGFILES]SYSTEM$RED.LM$JOURNAL;1
End of file block 4002 / Allocated 4002
Log Version 1.0
Transaction log UID: 17BB9140-2674-11C9-8001-AA0004006AF8
Penultimate Checkpoint: 000000ECADE5 41E5
Last Checkpoint: 000000F105FC 41FC

Total of 1 file.

Directory of DISK$LOGFILES:[LOGS]
DISK$LOGFILES:[LOGS]SYSTEM$YELLOW.LM$JOURNAL;1
End of file block 1002 / Allocated 1002
Log Version 1.0
Transaction log UID: 590DAA40-2640-11C9-B77A-08002B14179F
Penultimate Checkpoint: 00000C8B4819 2019
Last Checkpoint: 00000C8BC15B 335B

Total of 1 file.

Total of 3 files in 3 directories.
```

---

## ACL Editor Keypad Editing Commands

By default, the access control list editor (ACL editor) prompts you for each access control entry (ACE) and provides values for some of the fields within an ACE. You can navigate the ACE fields by using keypad commands, such as FIELD and ITEM.

This appendix describes all the keypad editing commands supplied by the ACL editor. You can supplement or change these key definitions by modifying and recompiling the ACL editor section file `SYS$LIBRARY:ACLEDIT.TPU` (see Appendix B for more information). To get help on the ACL editor keypad commands, press PF2.

### A.1 ACL Editor Keypad Commands

Figure A-1 shows the default ACL editor keypad commands for LK201 keyboards. The numeric keypad on VT100-series terminals is identical to that of the LK201 keyboard shown in Figure A-1; VT100 terminals, however, do not have the supplemental editing keypad (keys E1 through E6).

# ACL Editor Keypad Editing Commands

## A.1 ACL Editor Keypad Commands

Figure A-1 Keypad for an LK201-Series Keyboard

VT200

<b>E1</b> FIND	<b>E2</b> INSERT	<b>E3</b> REMOVE COPY	<b>PF1</b> GOLD	<b>PF2</b> HELP HELP FMT	<b>PF3</b> FNDNXT FIND	<b>PF4</b> DEL ACE UND ACE
<b>E4</b> SELECT	<b>E5</b> PREV SCREEN	<b>E6</b> NEXT SCREEN	<b>7</b> FIELD ADVFIELD	<b>8</b> MOVE SCREEN	<b>9</b>	- DEL W UND W
	↑ UP		<b>4</b> ADVANCE BOTTOM	<b>5</b> BACKUP TOP	<b>6</b>	, DEL C UND C
← LEFT	↓ DOWN	→ RIGHT	<b>1</b> WORD	<b>2</b> EOL DEL EOL	<b>3</b>	ENTER
			<b>0</b> OVER ACE INSERT	.ITEM		

ZK-1758-GE

Table A-1 describes each of the keypad commands you can use with the ACL editor. In this table, KP $n$  refers to a keypad key labeled with the number  $n$ . For example, KP4 refers to the keypad key labeled with the number 4.

Table A-1 ACL Editor Keypad Commands

Command	Key or Key Sequence	Description
ADVANCE	KP4	Sets the current direction forward for the FIND, FNDNXT, MOVE SCREEN, OVER ACE, and WORD commands. Movement is toward the end of the ACL.
ADV FIELD	GOLD-KP7	Completes the current ACE field and moves the cursor to the next ACE field.
BACKUP	KP5	Reverses the current direction for the FIND, FNDNXT, MOVE SCREEN, OVER ACE, and WORD keys. Movement is toward the beginning of the ACL.
BOTTOM	GOLD-KP4	Positions the cursor after the last line of the last ACE. Any entries you add are placed at the end of the ACL.
DEL ACE	PF4	Deletes the entire ACE in which the cursor is positioned and stores it in the delete-ACE buffer.
DEL C	Comma	Deletes the character on which the cursor is positioned and stores it in the delete-character buffer.
DEL EOL	GOLD-KP2	Deletes text from the current cursor position to the end of the line and stores it in the delete-line buffer.

(continued on next page)



## ACL Editor Keypad Editing Commands

### A.1 ACL Editor Keypad Commands

**Table A–1 (Cont.) ACL Editor Keypad Commands**

Command	Key or Key Sequence	Description
DEL W	Minus	Deletes the text from the current cursor position to the beginning of the next word and stores it in the delete-word buffer.
ENTER	Enter	Indicates that the current ACE is complete. The ACL editor terminates the insertion and verifies that the syntax of the ACE is complete. You can press the Enter key while the cursor is located at any position within the ACE. (Pressing the Return key produces the same results.)
EOL	KP2	Moves the cursor to the end of the current line.
FIELD	KP7	Completes the current ACE field and moves the cursor to the next ACE field or subfield, inserting text as needed. If the ACL editor is not in prompt mode, the ACL editor advances to the next field in the current ACE.
FIND	GOLD-PF3	Searches for an occurrence of a string. Press the FIND key and then enter the string from the main keyboard. Press the ENTER key to search for the string in the current direction, or the ADVANCE or BACKUP key to change the search direction.
FNDNXT	PF3	Searches in the current direction for the next occurrence of the string previously entered with the FIND key.
GOLD	PF1	When pressed before another keypad key, specifies the second key's alternate function (the bottom function on the keypad diagram).
HELP	PF2	Displays information about using the editing keypad.
HELP FMT	GOLD-PF2	Displays information about ACE formats.
INSERT	GOLD-KP0	Moves all text from the current line down one line, leaving a blank line where an ACE is to be inserted.
ITEM	Period	Selects the next item for the current ACE field. If the ACL editor is not in prompt mode, this key is ignored.
MOVE SCREEN	KP8	Moves the cursor one screen in the current direction (see ADVANCE or BACKUP). A screen is defined as two-thirds the number of lines in the display.
OVER ACE	KP0	Moves the cursor to the beginning of the next ACE (if the direction is set to ADVANCE) or to the beginning of the previous ACE (if the direction is set to BACKUP).
TOP	GOLD-KP5	Moves the cursor position to the first character of the first ACE in the access control list.
UND ACE	GOLD-PF4	Inserts the contents of the delete-ACE buffer in front of the ACE in which the cursor is currently positioned.
UND C	GOLD-Comma	Inserts the contents of the delete-character buffer directly in front of the cursor.
UND W	GOLD-Hyphen	Inserts the contents of the delete-word buffer directly in front of the cursor.
WORD	KP1	Moves the cursor one word forward (if the direction is set to ADVANCE) or backward (if the direction is set to BACKUP).

# ACL Editor Keypad Editing Commands

## A.2 Additional ACL Editing Keys and Key Sequences

### A.2 Additional ACL Editing Keys and Key Sequences

In addition to keypad editing, the ACL editor lets you use other keyboard keys and key sequences to perform editing functions. Table A–2 describes these additional ACL editing keys and key sequences. Keys in parentheses indicate the equivalent key for an LK201-series keyboard.

**Table A–2 Additional ACL Editing Keys and Key Sequences**

Key or Sequence	Action Taken When Key or Sequence Is Pressed
DOWN ARROW KEY	Moves the cursor to the character directly in line below it. If the ACE in which the cursor is positioned is new, the ACL editor processes the ACE before moving the cursor. If the entry is incomplete or formatted incorrectly, an error occurs and the cursor does not move.
LEFT ARROW KEY	Moves the cursor one character to the left. If the cursor is at the left margin, moves it to the rightmost character in the line above.
RIGHT ARROW KEY	Moves the cursor one character to the right. If the cursor is at the right margin, moves it to the leftmost character in the line below.
UP ARROW KEY	Moves the cursor to the character directly in line above it. If the ACE in which the cursor is positioned is new, the ACL editor processes the ACE before moving the cursor. If the entry is incomplete or formatted incorrectly, an error occurs and the cursor does not move.
GOLD-←	Shifts the text in the display window 8 characters to the left.
GOLD-→	Shifts the text in the display window 8 characters to the right.
Backspace (F12)	Moves the cursor to the beginning of the current line.
Ctrl/A	Changes the current mode from insert mode to overstrike mode or from overstrike mode to insert mode. Insert mode (the default) inserts a character to the left of the current character. Overstrike mode replaces the current character.
Ctrl/D	Allows you to execute one TPU command.
Ctrl/H	Moves the cursor to the beginning of the line. (Performs the same function as the backspace key.)
Ctrl/J	Deletes the text from the cursor back to the beginning of the word. (Performs the same function as the linefeed key.)
Ctrl/R	Refreshes the screen display. Clears and redraws the screen, deleting any extraneous characters or messages that might have appeared on the screen but are not part of the ACL you are editing. (Performs the same function as Ctrl/W.)
GOLD-Ctrl/R	Returns the ACL to its original state before the ACL editor was invoked. (Performs the same function as GOLD-Ctrl/W.)
Ctrl/U	Deletes the text from the cursor to the beginning of the line.
GOLD-Ctrl/U	Inserts the contents of the deleted-line buffer into the line at the current position. The line might wrap automatically.
Ctrl/W	See Ctrl/R.
GOLD-Ctrl/W	See GOLD Ctrl/R.
Ctrl/Z	Ends the editing session and updates the ACL. (Unless otherwise specified, any recovery and journal files are deleted.)
GOLD-Ctrl/Z	Ends (quits) the editing session without saving any of the changes made to the ACL. (Unless otherwise specified, any recovery and journal files are deleted.)
DELETE KEY	Deletes the character to the left of the cursor.

(continued on next page)

## ACL Editor Keypad Editing Commands

### A.2 Additional ACL Editing Keys and Key Sequences

**Table A–2 (Cont.) Additional ACL Editing Keys and Key Sequences**

Key or Sequence	Action Taken When Key or Sequence Is Pressed
Linefeed (F13)	Deletes the text from the cursor back to the beginning of the word. If the cursor is positioned at the first character of the word, deletes to the beginning of the previous word.
Tab	Moves the text located to the right of the cursor to the next tab stop.

### A.3 ACL Editing Keys on the Supplemental Keypad (LK201-Series Keyboards)

You can use the supplemental keypad on an LK201-series keyboard to move sections of text from one part of an ACL to another. However, note that certain supplemental editing keys (Insert Here, Remove, and Select) require a PASTE buffer, which is not enabled by default. To enable the PASTE buffer for the current editing session, perform the following actions:

1. Press Ctrl/D.
2. At the TPU command: prompt, enter the following statement:

```
TPU command: ACLEDIT$X_PASTE_BUFFER:=1
```

3. Press Ctrl/D again, and enter the following statement:

```
TPU command: ACLEDIT$X_CHECK_MODIFY:=0
```

Setting the value of the ACLEDIT\$X\_CHECK\_MODIFY variable to 0 prevents the ACL editor from checking for a modifiable ACE. The two features (support for the PASTE buffer and the check for a modifiable ACE) are not compatible.

To enable the PASTE buffer for all ACL editing sessions, change the values of the variables ACLEDIT\$X\_PASTE\_BUFFER and ACLEDIT\$X\_CHECK\_MODIFY in the ACL editor section file and recompile the file (see Appendix B).

Table A–3 describes the supplemental keypad keys you can use with the ACL editor.

**Table A–3 ACL Editing Keys on the Supplemental Keypad**

Key or Key Sequence	Description
Find	Elicits the Search for: prompt as the first step in the FIND operation. Type the search string after the prompt; then, press either the Do key or the Enter key to process the search. Performs the same function as the FIND keypad command.
Insert Here	Indicates where an ACE is to be inserted or, if support for the PASTE buffer is enabled, indicates the line where the selected text in the PASTE buffer is to be inserted.
Remove	Removes the selected text to the PASTE buffer. Each time you press the Remove key, the ACL editor deletes the previous contents of the PASTE buffer.

(continued on next page)

## ACL Editor Keypad Editing Commands

### A.3 ACL Editing Keys on the Supplemental Keypad (LK201-Series Keyboards)

**Table A-3 (Cont.) ACL Editing Keys on the Supplemental Keypad**

<b>Key or Key Sequence</b>	<b>Description</b>
GOLD-Remove (COPY)	Copies the selected text to the PASTE buffer. Each time you use the COPY command, the ACL editor deletes the previous contents of the PASTE buffer.
Select	Marks the beginning of a range of text to be removed or copied to the PASTE buffer. Press the Select key. Then, move the cursor to include the desired amount of text to be removed or copied. Press either Remove or GOLD-Remove (COPY) to complete the operation.
Prev Screen	Moves the cursor to the previous screen. By default, a screen is defined as two-thirds the number of lines in the display.
Next Screen	Moves the cursor one screen forward. By default, a screen is defined as two-thirds the number of lines in the display.

---

## Customizing the ACL Editor

You can modify the access control list editor (ACL editor) by modifying and recompiling the ACL section file `SYS$LIBRARY:ACLEDIT.TPU` (the source file used to create the compiled ACL section file `SYS$LIBRARY:ACLEDT$SECTION.TPU$SECTION`). You can also create your own ACL section file.

Refer to the *DEC Text Processing Utility Reference Manual* for more information about writing and processing section files.

### B.1 Modifying Variables in the ACL Section File

Table B-1 lists the ACL section file variables and their defaults.

**Table B-1 ACL Section File Variables**

Variable	Meaning
<code>ACLEDIT\$X_CHECK_DUPLICATES</code>	Controls whether a check for duplicate ACEs is made. This variable can take the following values:  0 No duplicate ACE check is made.  1 A duplicate ACE check is made. If the ACE to be entered matches an existing ACE, an error message is returned. This is the default.
<code>ACLEDIT\$X_CHECK_MODIFY</code>	Allows or disallows modification of ACEs. This variable can take the following values:  0 The ACE can be modified.  1 The ACE cannot be modified. If an attempt is made to modify the ACE, it is replaced with the original ACE. This is the default.
<code>ACLEDIT\$X_DIRECTORY_FILE</code>	Indicates whether the object is a directory file. This variable can take the following values:  0 The object is not a directory file.  1 The object is a directory file.

(continued on next page)

## Customizing the ACL Editor

### B.1 Modifying Variables in the ACL Section File

**Table B–1 (Cont.) ACL Section File Variables**

Variable	Meaning
ACLEDIT\$X_PASTE_BUFFER	Controls whether PASTE buffer support is enabled for VT200 series terminals. This variable can take the following values: 0 PASTE buffer support is disabled. This is the default. 1 PASTE buffer support is enabled.
ACLEDIT\$X_PROMPT	Controls whether automatic text insertion (prompt mode) is enabled. This variable can take the following values: 0 Prompt mode is disabled. 1 Prompt mode is enabled. This is the default.
ACLEDIT\$X_USE_DEFAULT_OPT	Controls whether the DEFAULT option can be used with nondirectory ACEs. This variable can take the following values: 0 The DEFAULT option can only be used with ACEs of directory (.DIR) files. This is the default. 1 The DEFAULT option is available for use with ACEs of all object types.
ACLEDIT\$C_WINDOW_SHIFT	Specifies the number of columns to shift the edit window in the direction wanted, GOLD key and left arrow for a left shift and GOLD key and right arrow for a right shift. The default is 8 columns.

If you modify any of the variables in Table B–1 or change any other part of the ACL section file, recompile the section file with the following command:

```
$ EDIT/TPU/NOSECTION/COMMAND=SYS$LIBRARY:ACLEDIT
```

Use the preceding command if you make changes directly to the source code file (SYS\$LIBRARY:ACLEDIT) that creates the compiled ACL section file SYS\$LIBRARY:ACLEDT\$SECTION. If you add a private command file to the existing ACL section file, recompile the section file using the following command:

```
$ EDIT/TPU/SECTION=SYS$LIBRARY:ACLEDT$SECTION/COMMAND=CUSTOM_ACL.TPU
```

The compiled DECTpu ACL section file is placed in your current directory. To use the new section file, perform one of the following actions:

- Move the compiled section file, ACLEDT\$SECTION.TPU\$SECTION, to the SYS\$LIBRARY directory. This changes the default ACL editor section file for all users.
- Keep the compiled section file in your directory and define the logical name ACLEDT\$SECTION in your LOGIN.COM file to point to the file, as follows:

```
$ DEFINE ACLEDT$SECTION yourdisk:[yourdir]ACLEDT$SECTION
```

Note that the default file type for the section file before compiling (the source file) is TPU, and the default file type for the compiled section file is TPU\$SECTION.

For more information about writing and processing a DECTpu section file, refer to the *DEC Text Processing Utility Reference Manual*.

## B.2 Using the ACL Editor CALL\_USER Routine

The ACL editor CALL\_USER routine is part of the shareable image SYS\$LIBRARY:ACLEDTSHR.EXE. You can incorporate the ACL editor CALL\_USER routine with its existing function codes into your own ACL section file, or you can write your own CALL\_USER routine that recognizes a different set of function codes.

The ACL editor CALL\_USER routine recognizes only those functions used by the ACL editor DECTpu section file. All other function codes are passed to a user-supplied CALL\_USER routine; if the high-order word of the CALL\_USER function code contains the ACL editor facility code (277 in decimal or 115 in hexadecimal), it is handled by the ACL editor CALL\_USER routine. Otherwise, an attempt is made to locate a user-supplied CALL\_USER routine. Refer to the description of the CALL\_USER routine in the *DEC Text Processing Utility Reference Manual* for more information about creating your own CALL\_USER routine.

Table B-2 describes the CALL\_USER routine function codes supported by the ACL editor.

**Table B-2 CALL\_USER Function Codes**

Function Code	Mnemonic	Description
18153473	ACLEDIT\$C_PARSE_ACE	Parses the input string (ACE) and returns the parsed (binary) ACE if no errors are found. Otherwise, the returned string contains a zero as the first two characters, and the unparsed portion of the input ACE as the remainder of the string.
18153474	ACLEDIT\$C_CHECK_MODIFY	Returns the string "READ_WRITE" if the ACE can be modified by the user. Otherwise, returns the string "READ_ONLY."
18153475	ACLEDIT\$C_PROMPT_MODE	Returns the string "PROMPT_MODE" if the prompt mode option was specified. Otherwise, returns the string "NOPROMPT_MODE."
18153476	ACLEDIT\$C_CHECK_ACE	Parses the input string (ACE) and returns the parsed (binary) ACE if no errors are found. Otherwise, the ACE text is highlighted in reverse video and a DECTpu variable of the form ACLEDIT\$X_RANGE_x is created to identify the ACE in error. (The "x" is a sequential number starting with 1.)
18153477	ACLEDIT\$C_CHECK_DIR	Returns the string "DIRECTORY_FILE" if the object being edited is a directory file. Otherwise, returns the string "NODIRECTORY_FILE."

(continued on next page)

## Customizing the ACL Editor

### B.2 Using the ACL Editor CALL\_USER Routine

Table B-2 (Cont.) CALL\_USER Function Codes

Function Code	Mnemonic	Description
18153478	ACLEEDIT\$C_SET_CANDIDATE	Parses the input string (ACE) and returns the string "PARSE_OK" if no error was encountered. Otherwise, returns the string "PARSE_ERROR." If the parse was successful, a check is made for duplicate ACEs using the CALL_USER function ACLEEDIT\$C_CHECK_DUP.
18153479	ACLEEDIT\$C_CHECK_DUP	Parses the input string (ACE) and returns the string "PARSE_ERROR" if an error was encountered. Otherwise, the parsed (binary) ACE is compared with the candidate ACE set by the CALL_USER function ACLEEDIT\$C_SET_CANDIDATE. Returns the string "DUPLICATE_ACE" if the ACE is a duplicate, or "UNIQUE_ACE" if it is not a duplicate.
18153482	ACLEEDIT\$C_MESSAGE	Assumes the input string is a system error code and returns in the ACL editor message window the message text associated with the error code.



---

## Accounting Information for Programmers

Table C–1 gives a summary of the system services that relate to accounting. No system service reads accounting files; to do this you must use knowledge of the structure of accounting files.

**Table C–1 Summary of Accounting System Services**

System Service	Description
\$CREPRC	Creates a process in which accounting can be disabled.
\$SNDJBC	Controls what resources are logged in the current accounting file, or logs a user-defined record in the current accounting file.

This appendix describes the structure of an accounting file. It is for programmers who want to access accounting data directly.

---

**Note**

---

The formats described here are subject to change without notice in a future release.

---

The symbols and offsets described in this appendix are defined by the \$ACRDEF macro in the STARLET library.

### C.1 Format of an Accounting File Record

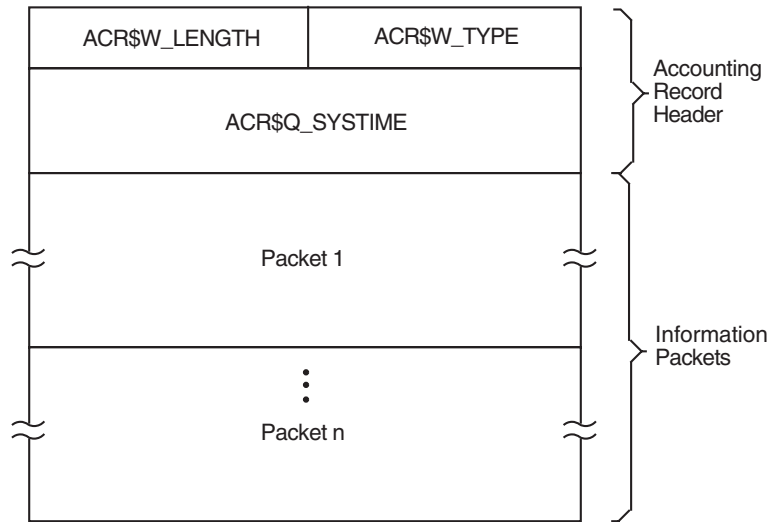
An accounting record consists of an accounting record header and a number of information packets. The number and type of information packets depend on the type of the record.

Figure C–1 illustrates the general format of an accounting record. Table C–2 describes the fields in the record header. The type field in the record header is subdivided into five fields, described in Table C–3.

# Accounting Information for Programmers

## C.1 Format of an Accounting File Record

**Figure C-1 Format of an Accounting Record**



ZK-0957-GE

**Table C-2 Fields in an Accounting Record Header**

Symbolic Offset	Description
ACR\$W_TYPE	Identifies the type of the record. This field is subdivided into five fields, described in Table C-3. (word)
ACR\$W_LENGTH	Total length of the record, in bytes. (word)
ACR\$Q_SYSTIME	System time (64-bit absolute time). (quadword)

**Table C-3 ACR\$W\_TYPE Fields in an Accounting Record Header**

Symbolic Offset	Description
ACR\$V_PACKET	Identifies this header as a record header. This bit must be 0. (1 bit)
ACR\$V_TYPE	Identifies the type of the record. The eight record types are described in Table C-4. (7 bits)

(continued on next page)

## Accounting Information for Programmers

### C.1 Format of an Accounting File Record

**Table C–3 (Cont.) ACR\$W\_TYPE Fields in an Accounting Record Header**

Symbolic Offset	Description												
ACR\$V_SUBTYPE	<p>Identifies the type of process with which the record is associated. The subtypes (4 bits) are:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Symbol</th> <th style="text-align: left;">Meaning</th> </tr> </thead> <tbody> <tr> <td>ACR\$K_BATCH</td> <td>Batch process</td> </tr> <tr> <td>ACR\$K_DETACHED</td> <td>Detached process</td> </tr> <tr> <td>ACR\$K_INTERACTIVE</td> <td>Interactive process</td> </tr> <tr> <td>ACR\$K_NETWORK</td> <td>Network process</td> </tr> <tr> <td>ACR\$K_SUBPROCESS</td> <td>Subprocess</td> </tr> </tbody> </table> <p>Note that this field is only meaningful for records of type ACR\$K_IMGDEL and ACR\$K_PRCDEL.</p>	Symbol	Meaning	ACR\$K_BATCH	Batch process	ACR\$K_DETACHED	Detached process	ACR\$K_INTERACTIVE	Interactive process	ACR\$K_NETWORK	Network process	ACR\$K_SUBPROCESS	Subprocess
Symbol	Meaning												
ACR\$K_BATCH	Batch process												
ACR\$K_DETACHED	Detached process												
ACR\$K_INTERACTIVE	Interactive process												
ACR\$K_NETWORK	Network process												
ACR\$K_SUBPROCESS	Subprocess												
ACR\$V_VERSION	<p>Identifies the version of the accounting file record structure. The versions (3 bits) are:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Symbol</th> <th style="text-align: left;">Meaning</th> </tr> </thead> <tbody> <tr> <td>ACR\$K_VERSION2</td> <td>VAX/VMS Version 2.0</td> </tr> <tr> <td>ACR\$K_VERSION3T</td> <td>VAX/VMS Version 3.0 field test</td> </tr> <tr> <td>ACR\$K_VERSION3</td> <td>OpenVMS Alpha Version 1.0 and VAX/VMS Version 3.0 and later versions of Alpha and VAX</td> </tr> </tbody> </table>	Symbol	Meaning	ACR\$K_VERSION2	VAX/VMS Version 2.0	ACR\$K_VERSION3T	VAX/VMS Version 3.0 field test	ACR\$K_VERSION3	OpenVMS Alpha Version 1.0 and VAX/VMS Version 3.0 and later versions of Alpha and VAX				
Symbol	Meaning												
ACR\$K_VERSION2	VAX/VMS Version 2.0												
ACR\$K_VERSION3T	VAX/VMS Version 3.0 field test												
ACR\$K_VERSION3	OpenVMS Alpha Version 1.0 and VAX/VMS Version 3.0 and later versions of Alpha and VAX												
ACR\$V_CUSTOMER	<p>Identifies whether the record was written by HP software or by customer software. If this bit is 0, the record was written by HP software. If this bit is 1, the record was written by customer software. (1 bit)</p>												

**Note**

ACR\$K\_CURVER = Current version. Set equal to ACR\$K\_VERSION3 in this release.

### C.1.1 Types of Accounting Record

The type of an accounting record identifies the type of event that caused the record to be logged. The eight types of accounting records are shown in Table C–4. This table shows the information packets contained in each type of record.

**Table C–4 Types of Accounting Record**

Symbol	Event	Information Packets
ACR\$K_FILE_BL	The accounting file was opened	ACR\$K_FILENAME
ACR\$K_FILE_FL	The accounting file was closed	ACR\$K_FILENAME

(continued on next page)

## Accounting Information for Programmers

### C.1 Format of an Accounting File Record

**Table C–4 (Cont.) Types of Accounting Record**

Symbol	Event	Information Packets
ACR\$K_IMGDEL	An image terminated	ACR\$K_ID ACR\$K_RESOURCE ACR\$K_IMAGENAME
ACR\$K_LOGFAIL	A login attempt failed	ACR\$K_ID ACR\$K_RESOURCE
ACR\$K_PRCDEL	A process terminated	ACR\$K_ID ACR\$K_RESOURCE
ACR\$K_PRINT	A print job finished	ACR\$K_ID ACR\$K_PRINT
ACR\$K_SYSINIT	The system was initialized	ACR\$K_ID ACR\$K_RESOURCE
ACR\$K_USER	An accounting message was sent by the \$SNDJBC system service	ACR\$K_ID ACR\$K_USER_DATA

#### C.1.2 Format of an Information Packet

The header, in each of the six types of information packets, defines the type of packet as follows:

- File name packet (ACR\$K\_FILENAME)
- Identification packet (ACR\$K\_ID)
- Image name packet (ACR\$K\_IMAGENAME)
- Print resource packet (ACR\$K\_PRINT)
- Resource packet (ACR\$K\_RESOURCE)
- User data packet (ACR\$K\_USER\_DATA)

Section C.1.2.1 describes the general format of an information packet.

Section C.1.2.2 to Section C.1.2.7 describe the format of each type of information packet.

##### C.1.2.1 General Format

Each information packet contains a packet header, followed by data fields. The data fields can contain fixed-length data, variable-length data, or offsets to variable-length data. Offsets contain the distance in bytes from the beginning of the packet to the variable-length data.

All variable-length data are represented as counted strings. Variable-length data follow the last fixed-length data field in the packet. Figure C–2 shows the general format of an information packet. An information packet may not have values in all of its data fields.

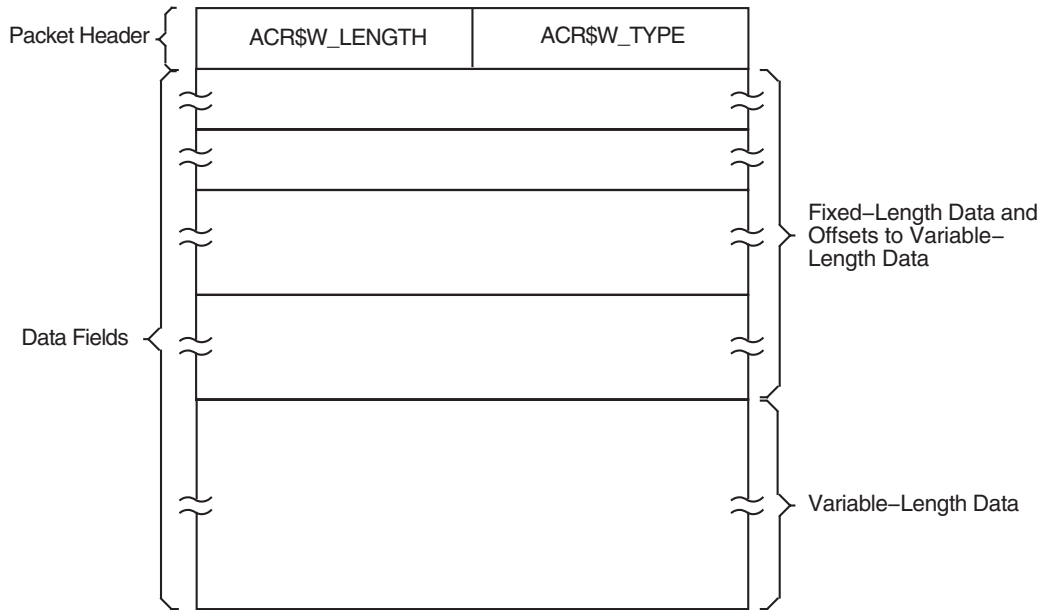
See Section C.1.2.2 to Section C.1.2.7 for complete descriptions of the data fields contained in each information packet.

All information packets start with a packet header that has ACR\$W\_LENGTH and ACR\$W\_TYPE fields (see Table C–5 and Table C–6).

# Accounting Information for Programmers

## C.1 Format of an Accounting File Record

**Figure C-2 Format of an Information Packet**



ZK-0958-GE

**Table C-5 Fields in an Information Packet Header**

Symbolic Offset	Description
ACR\$W_TYPE	Identifies the type of the packet. This field is subdivided into five fields, described in Table C-6. (word)
ACR\$W_LENGTH	Total length of the packet, in bytes. (word)

**Table C-6 ACR\$W\_TYPE Fields in an Information Packet Header**

Symbolic Offset	Description
ACR\$V_PACKET	Identifies this header as a packet header. This bit must be 1. (1 bit)
ACR\$V_TYPE	Identifies the type of the packet. The six packet types (7 bits) are:
<b>Symbol</b>	<b>Description</b>
ACR\$K_FILENAME	File name packet
ACR\$K_ID	Identification packet
ACR\$K_IMAGENAME	Image name packet
ACR\$K_PRINT	Print resource packet
ACR\$K_RESOURCE	Resource packet
ACR\$K_USER_DATA	User data packet

(continued on next page)

# Accounting Information for Programmers

## C.1 Format of an Accounting File Record

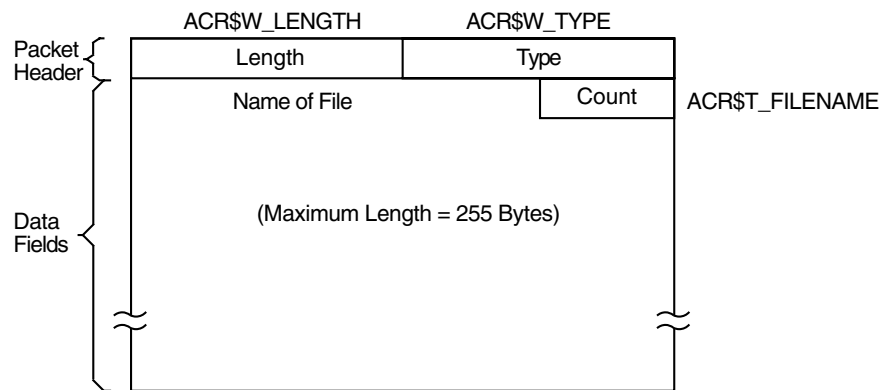
**Table C-6 (Cont.) ACR\$W\_TYPE Fields in an Information Packet Header**

Symbolic Offset	Description
ACR\$V_SUBTYPE	Identifies the packet subtype; reserved for future use. (4 bits)
ACR\$V_VERSION	See Table C-3.
ACR\$V_CUSTOMER	See Table C-3.

### C.1.2.2 File Name Packet (ACR\$K\_FILENAME)

The file name packet contains the name of the accounting file. Figure C-3 shows the format of the file name packet. Table C-7 describes the field contained in the packet. See Section C.1.2.1 for information on the packet header.

**Figure C-3 Format of a File Name Packet**



ZK-0964-GE

**Table C-7 Data Fields in a File Name Packet**

Symbolic Offset	Description
ACR\$T_FILENAME	Name of the file (counted ASCII string that gives full file specification).

### C.1.2.3 Identification Packet (ACR\$K\_ID)

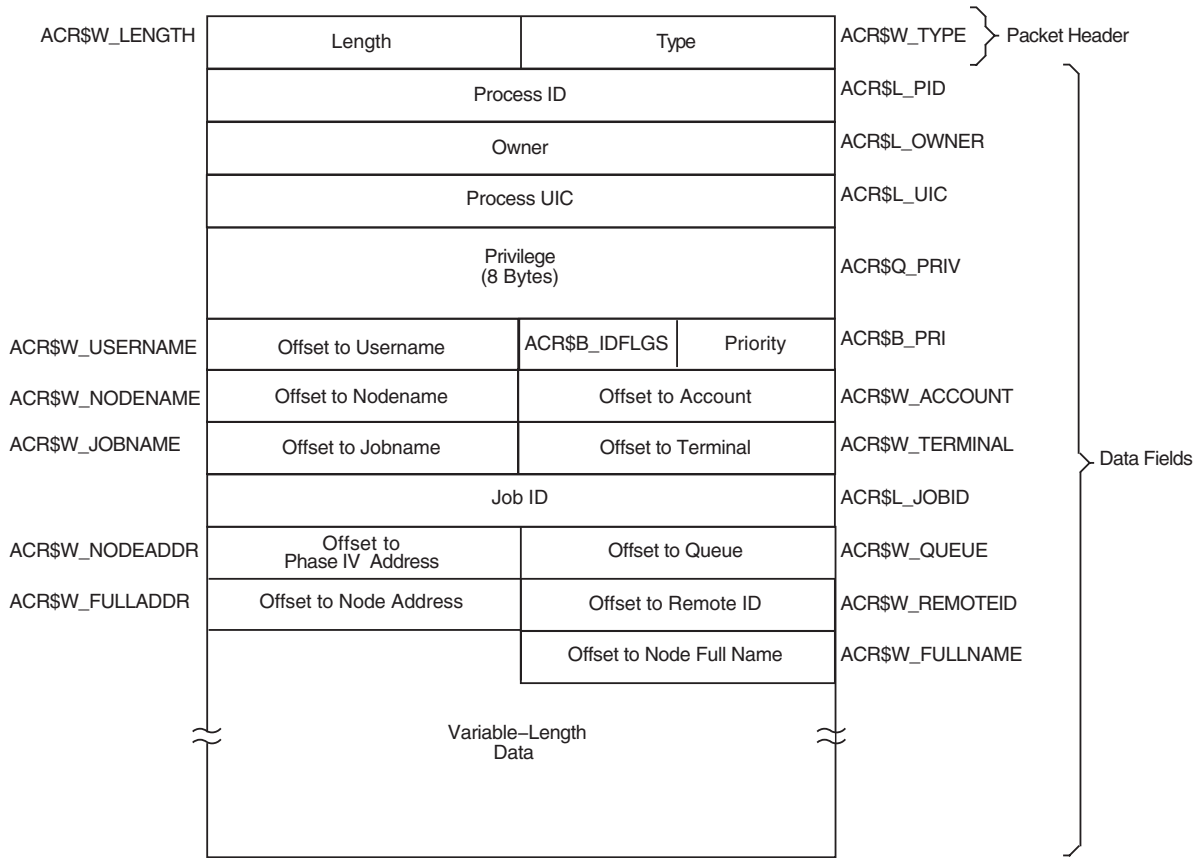
The identification packet identifies the process that caused the record to be logged.

Figure C-4 shows the format of the identification packet. Table C-8 describes the fields contained in the packet. See Section C.1.2.1 for information on the packet header.

# Accounting Information for Programmers

## C.1 Format of an Accounting File Record

**Figure C-4 Format of an Identification Packet**



ZK-0960-GE

**Table C-8 Data Fields in an Identification Packet**

Symbolic Offset	Description
ACR\$L_PID	Process identifier (PID) of the process. (longword)
ACR\$L_OWNER	PID of the parent process. (longword)
ACR\$L_UIC	UIC of the process. The UIC can be addressed as two separate words: ACR\$W_MEM for the member number, and ACR\$W_GRP for the group number. (longword)
ACR\$Q_PRIV	Privileges held by the process. (quadword)
ACR\$B_PRI	Base priority of the process. (byte)
ACR\$B_IDFLGS	Flags byte; full address and full name present if low bit is set.
ACR\$W_USERNAME	Offset to counted ASCII string containing the user name of the process. (word)
ACR\$W_ACCOUNT	Offset to counted ASCII string containing the account name of the process. (word)

(continued on next page)

# Accounting Information for Programmers

## C.1 Format of an Accounting File Record

**Table C-8 (Cont.) Data Fields in an Identification Packet**

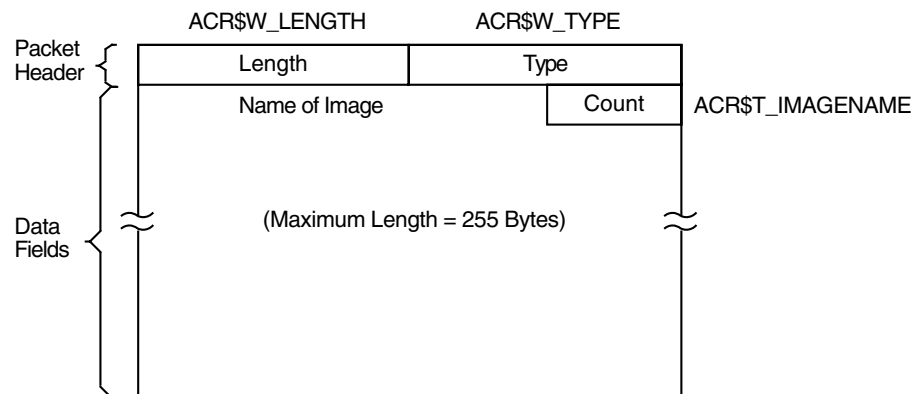
Symbolic Offset	Description
ACR\$W_NODENAME	Offset to counted ASCII string containing the Phase W node name of the remote process. (word)
ACR\$W_TERMINAL	Offset to counted ASCII string containing the terminal name. (word)
ACR\$W_JOBNAME	Offset to counted ASCII string containing the job name. (word)
ACR\$L_JOBID	Identification of the print or batch job (queue entry number). (longword)
ACR\$W_QUEUE	Offset to counted ASCII string containing the name of the queue with which a batch or print job is associated. (word)
ACR\$W_NODEADDR	Offset to a counted binary string containing the Phase W remote node address. (word)
ACR\$W_REMOTEID	Offset to counted ASCII string containing the remote ID of the remote process (varies with network implementation and use). (word)
ACR\$W_FULLADDR	Offset to a counted binary string containing the complete remote node network address. On a DECnet-Plus system, this is the remote node's NSAP address.
ACR\$W_FULLNAME	Offset to a counted ASCII string containing the complete remote node name. On a DECnet-Plus system, this is the remote node's full name.

### C.1.2.4 Image Name Packet (ACR\$K\_IMAGENAME)

The image name packet contains the name of the image executed by the identified process.

Figure C-5 shows the format of the image name packet. Table C-9 describes the field contained in the packet. See Section C.1.2.1 for information on the packet header.

**Figure C-5 Format of an Image Name Packet**



ZK-0962-GE



## Accounting Information for Programmers

### C.1 Format of an Accounting File Record

**Table C-9 Data Field in an Image Name Packet**

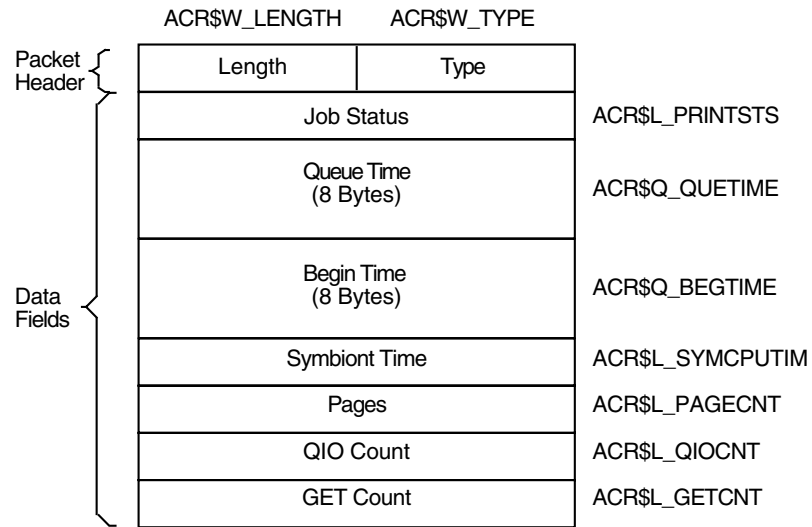
Symbolic Offset	Description
ACR\$T_IMAGENAME	Name of the image (counted ASCII string that gives full file specification).

#### C.1.2.5 Print Resource Packet (ACR\$K\_PRINT)

The print resource packet contains information about print jobs.

Figure C-6 shows the format of the print resource packet. Table C-10 describes the fields contained in the packet. See Section C.1.2.1 for information on the packet header.

**Figure C-6 Format of a Print Resource Packet**



ZK-0963-GE

# Accounting Information for Programmers

## C.1 Format of an Accounting File Record

**Table C–10 Data Fields in a Print Resource Packet**

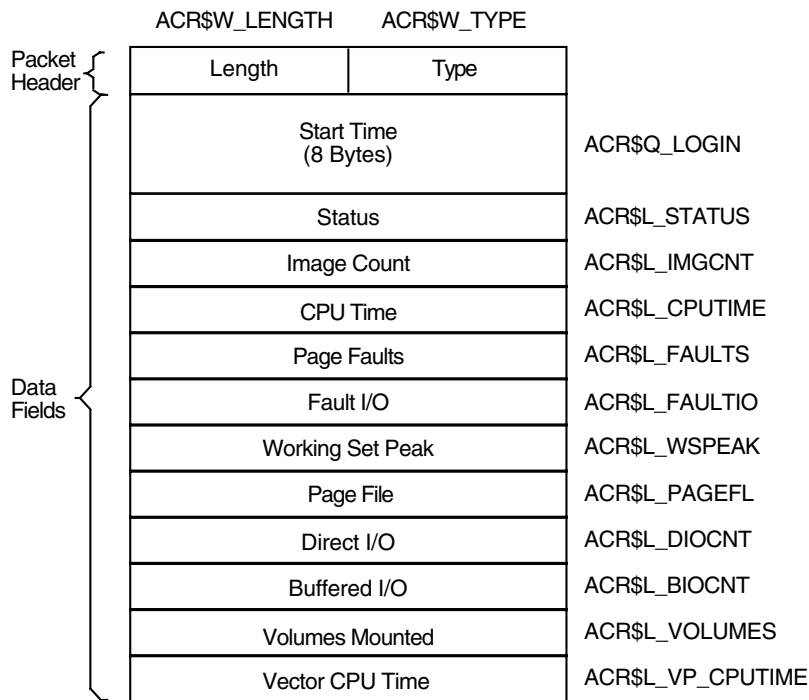
Symbolic Offset	Description
ACR\$L_PRINTSTS	Status of the print job. (longword)
ACR\$Q_QUEETIME	Time the job was queued. (64-bit absolute time)
ACR\$Q_BEGTIME	Time the job was started. (64-bit absolute time)
ACR\$L_SYMCPUTIM	Symbiont CPU time (always zero). (longword)
ACR\$L_PAGECNT	Number of pages printed. (longword)
ACR\$L_QIOCNT	Number of QIOs issued to the printer. (longword)
ACR\$L_GETCNT	Number of GETs from the file that was printed. (longword)

### C.1.2.6 Resource Packet (ACR\$K\_RESOURCE)

The resource packet contains information about the identified process.

Figure C–7 shows the format of a resource packet. Table C–11 describes the fields contained in the packet. See Section C.1.2.1 for information on the packet header.

**Figure C–7 Format of a Resource Packet**



ZK-0961-GE

## Accounting Information for Programmers

### C.1 Format of an Accounting File Record

**Table C–11 Data Fields in a Resource Packet**

Symbolic Offset	Description
ACR\$Q_LOGIN	64-bit absolute time at which the image was run or the process was created. (quadword)
ACR\$L_STATUS	Final exit status of the image, or for a process, the final status of the last image executed in the process. (longword)
ACR\$L_IMGCNT	Number of images run by the process. (longword)
ACR\$L_CPUTIME	Total CPU time used by the image or process, measured in units of 10 milliseconds. This includes any vector CPU time. (longword)
ACR\$L_FAULTS	Number of hard and soft page faults incurred by the image or process. (longword)
ACR\$L_FAULTIO	Number of hard page faults incurred by the image or process. (longword)
ACR\$L_WSPEAK	Maximum working set size used by the image or process. (longword)
ACR\$L_PAGEFL	Maximum page file usage. (longword)
ACR\$L_DIOCNT	Number of direct I/Os made by the image or process. (longword)
ACR\$L_BIOCNT	Number of buffered I/Os made by the image or process. (longword)
ACR\$L_VOLUMES	Number of volumes mounted by the image or process. (longword)
ACR\$L_VP_CPUTIME	Vector CPU time used by the image or process, measured in units of 10 milliseconds. (longword)

#### C.1.2.7 User Data Packet (ACR\$K\_USER\_DATA)

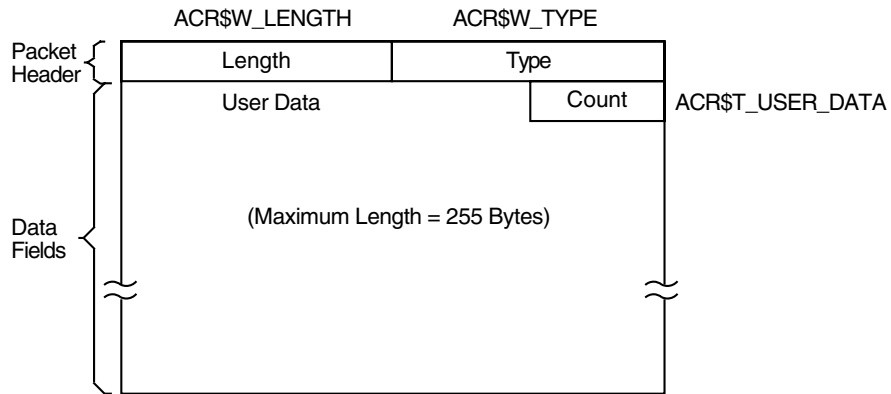
The user data packet contains an accounting message sent by the \$SNDJBC system service.

Figure C–8 shows the format of the user data packet. Table C–12 describes the fields contained in the packet. See Section C.1.2.1 for information on the packet header.

# Accounting Information for Programmers

## C.1 Format of an Accounting File Record

**Figure C-8 Format of a User Data Packet**



ZK-0965-GE

**Table C-12 Data Field in a User Data Packet**

Symbolic Offset	Description
ACR\$T_USER_DATA	Up to 255 bytes of data (counted string).

---

## ANALYZE/DISK\_STRUCTURE—Stage Checks

ANALYZE/DISK\_STRUCTURE performs the verification of a volume or volume set in eight distinct stages. During these stages, ANALYZE/DISK\_STRUCTURE compiles information that is used in reporting errors and performing repairs.

Before ANALYZE/DISK\_STRUCTURE can proceed with each stage, it must perform the following four initialization functions:

- Read the device name, validate access to the device, and save the device name
- Read the user-specified file names for the /LIST and /USAGE qualifiers, if specified, and open the files
- Assign all appropriate channels to the device being checked
- Write-lock the volume set to prevent simultaneous updates

The following sections describe the eight stages that ANALYZE/DISK\_STRUCTURE goes through while verifying a disk. These descriptions assume that you specified the /REPAIR qualifier in the command. An annotated ANALYZE/DISK\_STRUCTURE listing is included at the end of this appendix.

### D.1 Stage 1

In Stage 1, ANALYZE/DISK\_STRUCTURE gathers various volume information (such as cluster size, volume labels, and the number of volumes in the set) from several reserved files, verifies the information for accuracy, reports all discrepancies, and corrects problems discovered during this stage.

ANALYZE/DISK\_STRUCTURE identifies the volume and all the characteristics of that volume by using the parameters of the home block in INDEXF.SYS. When ANALYZE/DISK\_STRUCTURE confirms this information, it builds a current version of VOLSET.SYS in memory and reads and verifies the status control block (SCB) of BITMAP.SYS.

ANALYZE/DISK\_STRUCTURE then compares the volume-set attributes for the version of VOLSET.SYS in memory to the attributes listed in the version of VOLSET.SYS resident on the volume, reports discrepancies, and corrects errors.

### D.2 Stage 2

In Stage 2, ANALYZE/DISK\_STRUCTURE copies the current version of QUOTA.SYS into working memory, and establishes the structure on which another QUOTA.SYS file is built during subsequent stages. In Stage 7, these copies are compared with each other and inconsistencies are reported.

## ANALYZE/DISK\_STRUCTURE—Stage Checks

### D.3 Stage 3

#### D.3 Stage 3

Stage 3 checks consist of ANALYZE/DISK\_STRUCTURE operations that use the reserved file INDEXF.SYS. During Stage 3, ANALYZE/DISK\_STRUCTURE opens INDEXF.SYS, reads each file header, and completes the following steps:

- Validates each file's FID, and confirms that all files can be retrieved through the FID
- Validates the header and the revision date of each file
- Validates any extension headers of each file
- Confirms that each segment number reflects the proper sequence of extension headers

ANALYZE/DISK\_STRUCTURE also performs the following operations during Stage 3:

- Builds a map of header linkage so that ambiguities can be detected
- Determines the high block (HIBLK) and end-of-file block (EFBLK) record attributes and compares these values with the recorded values in INDEXF.SYS
- Checks the high-water mark (HIWATERMARK)

While performing these checks, ANALYZE/DISK\_STRUCTURE builds several maps that it uses in subsequent stages. Table D–1 briefly describes each map built in Stage 3.

**Table D–1 Stage 3 Maps**

<b>Bitmap</b>	<b>Function</b>
Valid file numbers	The current state of the bitmap for INDEXF.SYS
Lost file numbers	All the valid file numbers not yet found in a directory
Directory files	List of all directory files
Extension linkages	List of all valid extension headers
Multiply allocated clusters	List of all clusters that are referenced by more than one header
Allocated clusters	All allocated clusters on the volume (or volume set)
System map	The new storage bitmap
Valid file backlink	A map of all valid file backlinks
Invalid backlink	A map of all invalid backlinks

#### D.4 Stage 4

In Stage 4, ANALYZE/DISK\_STRUCTURE builds a current version of BITMAP.SYS using the maps built during Stage 3. In addition, ANALYZE/DISK\_STRUCTURE reports any discrepancies between the headers' maps and the storage bitmap. In Stage 4, ANALYZE/DISK\_STRUCTURE performs the following operations:

- Copies BITMAP.SYS into working memory
- Compares the corrected version of BITMAP.SYS with a map built from INDEXF.SYS

- Writes a corrected version of BITMAP.SYS to disk
- Reports multiply allocated clusters

## **D.5 Stage 5**

In this stage, ANALYZE/DISK\_STRUCTURE completes a pass of all entries in the invalid backlink map. ANALYZE/DISK\_STRUCTURE searches the directory hierarchy of the volume to confirm that all files included in INDEXF.SYS are retrievable through the directory structure. In addition, ANALYZE/DISK\_STRUCTURE identifies lost directories and attempts to reestablish valid backlinks to those directories.

In Stage 5, ANALYZE/DISK\_STRUCTURE performs the following operations:

- Confirms the locations of all directories listed in the directory map (compiled in Stage 3) and the subsequent files in those directories
- Enters all directories indicated as lost and locates a valid parent (if any)

## **D.6 Stage 6**

Stage 6 is essentially a cleanup operation for lost file headers. Following Stage 5, ANALYZE/DISK\_STRUCTURE is left with a list of files that are truly lost—files that have backlinks to nonexistent directories. These files were not traceable through the directory structure. ANALYZE/DISK\_STRUCTURE is also left with a list of files with bad backlinks; these files are traceable through the directory structure, but the backlinks of the files do not point back to the directory that contains them.

During Stage 6, ANALYZE/DISK\_STRUCTURE performs the following operations:

- Checks the backlink map to locate all files with invalid backlinks, then repairs backlinks
- Checks the lost file bitmap for lost files and places lost files in [SYSLOST] if you specified /REPAIR
- If you specified the /USAGE qualifier, creates an entry for each lost file

## **D.7 Stage 7**

In this stage, ANALYZE/DISK\_STRUCTURE compares the values stored in the quota file built during Stage 2 with those stored in the reserved file QUOTA.SYS. During Stage 7, ANALYZE/DISK\_STRUCTURE opens QUOTA.SYS and performs the following operations:

- Compares the block usage for each UIC listed in QUOTA.SYS to parallel statistics listed in the copy of QUOTA.SYS built in Stage 2
- Modifies QUOTA.SYS such that values in QUOTA.SYS match values in the copy built in Stage 2
- Closes QUOTA.SYS

## ANALYZE/DISK\_STRUCTURE—Stage Checks

### D.8 Stage 8

### D.8 Stage 8

Throughout the first seven stages, ANALYZE/DISK\_STRUCTURE places operations that cannot be performed during a particular stage on a deferred list. The list includes FIDs sorted by operation. In Stage 8, ANALYZE/DISK\_STRUCTURE performs all operations stored on the deferred list. In Stage 8, ANALYZE/DISK\_STRUCTURE performs the following operations:

- Removes an FID from the deferred list, renames the file, and adds the file to SYSLOST.DIR or to a user-specified directory
- Updates QUOTA.SYS to reflect all additional blocks used by the UIC that received the lost file
- Updates VOLSET.SYS to correct inconsistencies discovered during previous ANALYZE/DISK\_STRUCTURE stages

### D.9 Annotated Example

The following example is an annotated sample of an ANALYZE/DISK\_STRUCTURE session. The command used to generate this example did not include the /REPAIR qualifier.

```
%VERIFY-I-BADHEADER, file (487,173,1) MAIL$0004008EEAEE0572.MAI;1 ❶
  invalid file header
%VERIFY-I-BADHEADER, file (531,112,1) MAIL$0004008EEFBB198B.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (589,104,1) MAIL$0004008EEAF199B9.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (604,157,1) MAIL$0004008EF12C3B28.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (674,247,1) MAIL$0004008EF6053C9B.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (688,41,1) MAIL$0004008EF608AFF4.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (689,135,1) MAIL$0004008EEE445A31.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (750,71,1) MAIL$0004008EEED19ADF.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (753,217,1) MAIL$0004008EE7C4A017.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (780,236,1) MAIL$0004008EF777ACA8.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (852,57,1) MAIL$0004008EF06C15F6.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (856,44,1) MAIL$0004008EE7D2520D.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (1059,42,1) MAIL$0004008EEB045608.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (1134,76,1) MAIL$0004008EE9EC806D.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (1316,147,1) MAIL$0004008EEEDA734F.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (1350,74,1) MAIL$0004008EE89BA8B0.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (1351,64,1) MAIL$0004008EEB09B036.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (1490,104,1) MAIL$0004008EE8B448B0.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (1493,106,1) LASTNOTIC.NIL;1
  invalid file header
%VERIFY-I-BADHEADER, file (1548,204,1) MAIL$0004008EF7B4D1B8.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (1613,61,1) MAIL$0004008EECEE4BA5.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (1812,81,1) MAIL$0004008EE7DF05EC.MAI;1
  invalid file header
%VERIFY-I-BADHEADER, file (1848,26,1) MAIL$0004008EF78659B9.MAI;1
```



## ANALYZE/DISK\_STRUCTURE—Stage Checks D.9 Annotated Example

```
invalid file header
%VERIFY-I-BADHEADER, file (1983,34119,1) MAIL$0004008EE7E49C13.MAI;1
invalid file header
%VERIFY-I-BADHEADER, file (1987,33907,1) REMIND.CAL;9
invalid file header
%VERIFY-I-BADHEADER, file (2196,123,1) MAIL$0004008EE6FA2DC9.MAI;1
invalid file header
%VERIFY-I-BADHEADER, file (2372,125,1) MAIL$0004008EF06339F9.MAI;1
invalid file header
%VERIFY-I-BADHEADER, file (2569,67,1) MAIL$0004008EF2BF0C15.MAI;1
invalid file header
%VERIFY-I-BADHEADER, file (2605,72,1) MAIL$0004008EE856FC73.MAI;1
invalid file header
%VERIFY-I-BADHEADER, file (2616,70,1) MAIL$0004008EF063C04F.MAI;1
invalid file header
%VERIFY-I-BADHEADER, file (2774,29818,1) LASTNOTIC.NIL;1
invalid file header
%VERIFY-I-ALLOCCLR, blocks incorrectly marked allocated ②
LBN 442398 to 445538, RVN 1
%VERIFY-I-BADHEADER, file (487,0,1) MAIL$0004008EEAEE0572.MAI;1 ③
invalid file header
%VERIFY-I-LOSTEXTHDR, file (487,0,1)
lost extension file header
%VERIFY-I-BADHEADER, file (531,0,1) MAIL$0004008EEFBB198B.MAI;1
invalid file header
%VERIFY-I-LOSTEXTHDR, file (531,0,1)
lost extension file header
%VERIFY-I-BADHEADER, file (589,0,1) MAIL$0004008EEAF199B9.MAI;1
invalid file header
%VERIFY-I-LOSTEXTHDR, file (589,0,1)
lost extension file header
%VERIFY-I-BADHEADER, file (604,0,1) MAIL$0004008EF12C3B28.MAI;1
invalid file header
%VERIFY-I-LOSTEXTHDR, file (604,0,1)
lost extension file header
%VERIFY-I-BADHEADER, file (674,0,1) MAIL$0004008EF6053C9B.MAI;1
invalid file header
%VERIFY-I-LOSTEXTHDR, file (674,0,1)
lost extension file header
%VERIFY-I-BADHEADER, file (688,0,1) MAIL$0004008EF608AFF4.MAI;1
invalid file header
%VERIFY-I-LOSTEXTHDR, file (688,0,1)
lost extension file header
%VERIFY-I-BADHEADER, file (689,0,1) MAIL$0004008EEE445A31.MAI;1
invalid file header
%VERIFY-I-LOSTEXTHDR, file (689,0,1)
lost extension file header
%VERIFY-I-BADHEADER, file (750,0,1) MAIL$0004008EEED19ADF.MAI;1
invalid file header
%VERIFY-I-LOSTEXTHDR, file (750,0,1)
lost extension file header
%VERIFY-I-BADHEADER, file (753,0,1) MAIL$0004008EE7C4A017.MAI;1
invalid file header
%VERIFY-I-LOSTEXTHDR, file (753,0,1)
lost extension file header
%VERIFY-I-BADHEADER, file (780,0,1) MAIL$0004008EF777ACA8.MAI;1
invalid file header
%VERIFY-I-LOSTEXTHDR, file (780,0,1)
lost extension file header
%VERIFY-I-BADHEADER, file (852,0,1) MAIL$0004008EF06C15F6.MAI;1
invalid file header
%VERIFY-I-LOSTEXTHDR, file (852,0,1)
lost extension file header
%VERIFY-I-BADHEADER, file (856,0,1) MAIL$0004008EE7D2520D.MAI;1
invalid file header
%VERIFY-I-LOSTEXTHDR, file (856,0,1)
lost extension file header
%VERIFY-I-BADHEADER, file (1059,0,1) MAIL$0004008EEB045608.MAI;1
invalid file header
%VERIFY-I-LOSTEXTHDR, file (1059,0,1)
lost extension file header
%VERIFY-I-BADHEADER, file (1134,0,1) MAIL$0004008EE9EC806D.MAI;1
```

## ANALYZE/DISK\_STRUCTURE—Stage Checks

### D.9 Annotated Example

```
invalid file header
%VERIFY-I-LOSTEXTHDR, file (1134,0,1)
  lost extension file header
%VERIFY-I-BADHEADER, file (1316,0,1) MAIL$0004008EEEDA734F.MAI;1
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (1316,0,1)
  lost extension file header
%VERIFY-I-BADHEADER, file (1350,0,1) MAIL$0004008EE89BA8B0.MAI;1
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (1350,0,1)
  lost extension file header
%VERIFY-I-BADHEADER, file (1351,0,1) MAIL$0004008EEB09B036.MAI;1
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (1351,0,1)
  lost extension file header
%VERIFY-I-BADHEADER, file (1490,0,1) MAIL$0004008EE8B448B0.MAI;1
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (1490,0,1)
  lost extension file header
%VERIFY-I-BADHEADER, file (1493,0,1) LASTNOTIC.NIL;1
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (1493,0,1)
  lost extension file header
%VERIFY-I-BADHEADER, file (1548,0,1) MAIL$0004008EF7B4D1B8.MAI;1
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (1548,0,1)
  lost extension file header
%VERIFY-I-BADHEADER, file (1613,0,1) MAIL$0004008EECEE4BA5.MAI;1
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (1613,0,1)      lost extension file header
%VERIFY-I-BADHEADER, file (1812,0,1) MAIL$0004008EE7DF05EC.MAI;1
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (1812,0,1)
  lost extension file header
%VERIFY-I-BADHEADER, file (1848,0,1) MAIL$0004008EF78659B9.MAI;1
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (1848,0,1)
  lost extension file header
%VERIFY-I-BADHEADER, file (1983,0,1) MAIL$0004008EE7E49C13.MAI;1
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (1983,0,1)
  lost extension file header
%VERIFY-I-BADHEADER, file (1987,0,1) REMIND.CAL;9
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (1987,0,1)
  lost extension file header
%VERIFY-I-BADHEADER, file (2196,0,1) MAIL$0004008EE6FA2DC9.MAI;1
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (2196,0,1)
  lost extension file header
%VERIFY-I-BADHEADER, file (2372,0,1) MAIL$0004008EF06339F9.MAI;1
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (2372,0,1)
  lost extension file header
```

# ANALYZE/DISK\_STRUCTURE—Stage Checks D.9 Annotated Example

```
%VERIFY-I-BADHEADER, file (2569,0,1) MAIL$0004008EF2BF0C15.MAI;1
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (2569,0,1)
  lost extension file header
%VERIFY-I-BADHEADER, file (2605,0,1) MAIL$0004008EE856FC73.MAI;1
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (2605,0,1)
  lost extension file header
%VERIFY-I-BADHEADER, file (2616,0,1) MAIL$0004008EF063C04F.MAI;1
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (2616,0,1)
  lost extension file header
%VERIFY-I-BADHEADER, file (2774,0,1) LASTNOTIC.NIL;1
  invalid file header
%VERIFY-I-LOSTEXTHDR, file (2774,0,1)
  lost extension file header
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [ALLWAY]NOTES.LOG;25 ④
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [BLAIN.BOOTLS]LOADER.OBJ;1
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [BLAIN.BOOTLS]SYSGEN.OBJ;1
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [BLAIN]MAIL.2060841.TMP;1
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [BLAIN]NETSERVER.LOG;181
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [BLAIN]NETSERVER.LOG;180
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [BLAIN]NETSERVER.LOG;179
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [BLAIN]NETSERVER.LOG;178
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [BLAIN]NETSERVER.LOG;170
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [BOEMUS.MAIL]MAIL$0004008EF94A72A0.MAI;1
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [BOEMUS]NETSERVER.LOG;10
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [BOEMUS]UPDATE.LOG;1
%VERIFY-I-BACKLINK, incorrect directory back link [CALGON.GER]OBJ.DIR;1
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [CALGON]T.TMP;1
%VERIFY-I-BACKLINK, incorrect directory back link [CLABIN.BACKUP.TMP SRC]BACKDEF.SDL;1
%VERIFY-I-BACKLINK, incorrect directory back link [CLABIN.BACKUP.TMP SRC]COMMON.REQ;1
%VERIFY-I-BACKLINK, incorrect directory back link [CLABIN.BACKUP.TMP SRC]DUMMY.MSG;1
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [CLABIN.NMAIL]NMAIL.LOG;77
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [CLABIN.NMAIL]NMAIL.LOG;76
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [DESIN.8800]2840HT86.GNC;1
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [DESIN.8800]2840TP86.GNC;1
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [DOWNE.MAIL]MAIL$0004008EF94A79B3.MAI;1
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [DOWNE.PRO]MORT.OBJ;15
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [DOWNE.PRO]OUTPUT.LOG;36
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [DOWNE.PRO]OUTPUT.LOG;35
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [DOWNE.PRO]OUTPUT.LOG;34
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [DOWNE.PRO]OUTPUT.LOG;33
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [DOWNE.PRO]OUTPUT.LOG;32
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [DOWNE.PRO]OUTPUT.LOG;31
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [DOWNE.PRO]OUTPUT.LOG;30
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [GAMBLE]CONFLICTS.LIS;1
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [GAMBLE.DOC]SMP.LOCK;6
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [GAMBLE]NETSERVER.LOG;5
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [GAMBLE.NMAIL]NMAIL.LOG;22
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [GAMBLE.NMAIL]NMAIL.LOG;21
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [GILLEY.MAIL]MAIL$0004008EF94A7B70.MAI;1
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [GILLEY]NETSERVER.LOG;657
%VERIFY-I-BADDIRENT, invalid file identification in
directory entry [GILLEY]NETSERVER.LOG;656
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [HALL]2.LOG;33
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [HALL]2.LOG;32
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [HALL]2.LOG;31
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [HALL]2.LOG;30
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [HALL]2.LOG;29
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [HALL]2.LOG;28
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [HALL]2.LOG;27
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [HALL]2.LOG;26
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [HALL]2.LOG;25
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [HALL]2.LOG;24
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [NAMOLLY]NETSERVER.LOG;2
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [NAMOLLY]NETSERVER.LOG;1
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [RUSS]082654.LOG;1
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [SCHROEDER.LOGIN]NETSERVER.LOG;17
%VERIFY-I-BADDIR, directory [SYSL0ST.BOOTLS] has invalid format
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [THOEN]NETSERVER.LOG;374
```

## ANALYZE/DISK\_STRUCTURE—Stage Checks

### D.9 Annotated Example

```
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [THOEN]NETSERVER.LOG;373
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [THOEN]NETSERVER.LOG;367
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [THOMAS.MAIL]MAIL$0004008EF94D75EB.MAI;1
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [THOMAS.MAIL]MAIL$0004008EF955DDF3.MAI;1
%VERIFY-I-BADDIRENT, invalid file identification in directory entry [THOMAS.MAIL]MAIL$0004008EFD118B44.MAI;1
%VERIFY-I-LOSTSCAN, due to directory errors, lost files will not be entered ⑤
%VERIFY-I-INCQUOTA, QUOTA.SYS indicates 69663 blocks used, actual use is 69740 blocks for [11,402] ⑥
%VERIFY-I-INCQUOTA, QUOTA.SYS indicates 1764 blocks used, actual use is 1770 blocks for [12,12]
%VERIFY-I-INCQUOTA, QUOTA.SYS indicates 0 blocks used, actual use is 31 blocks for [11,720]
```

- ① ANALYZE/DISK\_STRUCTURE has completed the first two stages, and is beginning Stage 3. Stage 1 involves collection and verification of various volume information. ANALYZE/DISK\_STRUCTURE found no problems with volume information. In Stage 2, ANALYZE/DISK\_STRUCTURE copies the current version of QUOTA.SYS to working memory, and builds the structure on which a new copy is built during subsequent stages. The first error message is produced by Stage 3. Stage 3 uses the reserved file INDEXF.SYS to locate a variety of file problems. Here, Stage 3 detects a number of invalid file headers. Note that the error message includes the FID and the file name.
- ② This error message is produced during Stage 4, during which ANALYZE/DISK\_STRUCTURE builds a current version of BITMAP.SYS, resolves multiple references to extension headers, and corrects discrepancies in the map sections of headers. Here, ANALYZE/DISK\_STRUCTURE has found that the specified logical blocks on the specified relative volume were marked allocated in the storage bit map, but were not allocated to a file.
- ③ This message marks the beginning of Stage 5. Here, messages stating “lost extension file header” and “invalid file header” indicate that ANALYZE/DISK\_STRUCTURE is performing a pass of all entries placed on the invalid backlink map. This map was created in Stage 3.
- ④ This message marks the beginning of the second phase of Stage 5, in which ANALYZE/DISK\_STRUCTURE confirms that all files in INDEX.SYS are retrievable through the directory structure. Here, the series of “invalid file identification . . .” messages indicates those directory entries that did not contain a valid file identification.
- ⑤ This message is produced by Stage 6, which is essentially a cleanup phase for lost files. This message indicates that ANALYZE/DISK\_STRUCTURE encountered errors during the directory scan that were reported in previous messages. As a result, the file is not entered in directory [SYSLOST].
- ⑥ Here, ANALYZE/DISK\_STRUCTURE begins Stage 7, in which it compares values stored in the quota file built during Stage 2 with values in the reserved file QUOTA.SYS. The last three messages here indicate discrepancies between the two files.

Note that no messages were produced during Stage 8. During Stage 8, ANALYZE/DISK\_STRUCTURE executes all operations placed on the deferred list, and if you specified /REPAIR, updates QUOTA.SYS and VOLSET.SYS as necessary.

---

## ANALYZE/DISK\_STRUCTURE—Usage File

When you specify the /USAGE qualifier, ANALYZE/DISK\_STRUCTURE creates a disk usage accounting file. The first record of this file, the identification record, contains a summary of the disk and volume characteristics. The identification record is followed by many file summary records, one record for each file on the disk. Each file summary record contains the owner, size, and name of a file.

The identification record is characterized by the type code USG\$K\_IDENT in the USG\$B\_TYPE field of the record. Table E-1 contains a description of all the fields in this record.

**Table E-1 Identification Record Format (Length USG\$K\_IDENT\_LEN)**

Field	Meaning
USG\$L_SERIALNUM	Serial number of the volume. This is an octal longword value.
USG\$T_STRUCNAM	Volume set name (if the volume is part of a volume set). For a Files-11 Structure Level 1 volume, this field contains binary zeros; for a Files-11 Structure Level 2 or 5 volume that is not part of a volume set, this field contains spaces. The length of this field is USG\$\$S_STRUCNAME.
USG\$T_VOLNAME	Volume name of relative volume 1. The length of this field is USG\$\$S_VOLNAME.
USG\$T_OWNERNAME	Volume owner name. The length of this field is USG\$\$S_OWNERNAME.
USG\$T_FORMAT	Volume format type. For a Files-11 Structure Level 1 volume, this field contains "DECFILE11A"; for a Files-11 Structure Level 2 or 5 volume, this field contains "DECFILE11B". The length of this field is USG\$\$S_FORMAT.
USG\$Q_TIME	Quadword system time when this usage file was created. The length of this field is USG\$\$S_TIME.

Each file summary record is characterized by the type code USG\$K\_FILE in the USG\$B\_TYPE field of the record. Table E-2 contains a description of all the fields in these records.

## ANALYZE/DISK\_STRUCTURE—Usage File

**Table E-2 File Record Format (Length USG\$K\_FILE\_LEN)**

Field	Meaning
USG\$L_FILEOWNER	File owner UIC. This can be considered as a single longword value or as two word values (USG\$W_UICMEMBER and USG\$W_UICGROUP).
USG\$W_UICMEMBER	The member field of the file owner UIC. This is an octal word value.
USG\$W_UICGROUP	The group field of the file owner UIC. This is an octal word value.
USG\$L_ALLOCATED	Number of blocks allocated to the file, including file headers. This is a decimal longword value.
USG\$L_USED	Number of blocks used, up to and including the end-of-file block. This is a decimal longword value.
USG\$W_DIR_LEN	Length of the directory string portion of USG\$T_FILESPEC, including the brackets. This is a decimal word value.
USG\$W_SPEC_LEN	Length of the complete file specification in USG\$T_FILESPEC. This is a decimal word value.
USG\$T_FILESPEC	File specification, in the following format: [dir]nam.typ;ver  This field is of variable length. A file that has more than one directory entry is listed under the first file specification found. A lost file has an empty directory string “[ ]” and the file name is taken from the file header. In some cases this information does not exist; you must take this into consideration when you write application programs to process the usage file. The length of this field is USG\$S_FILESPEC.

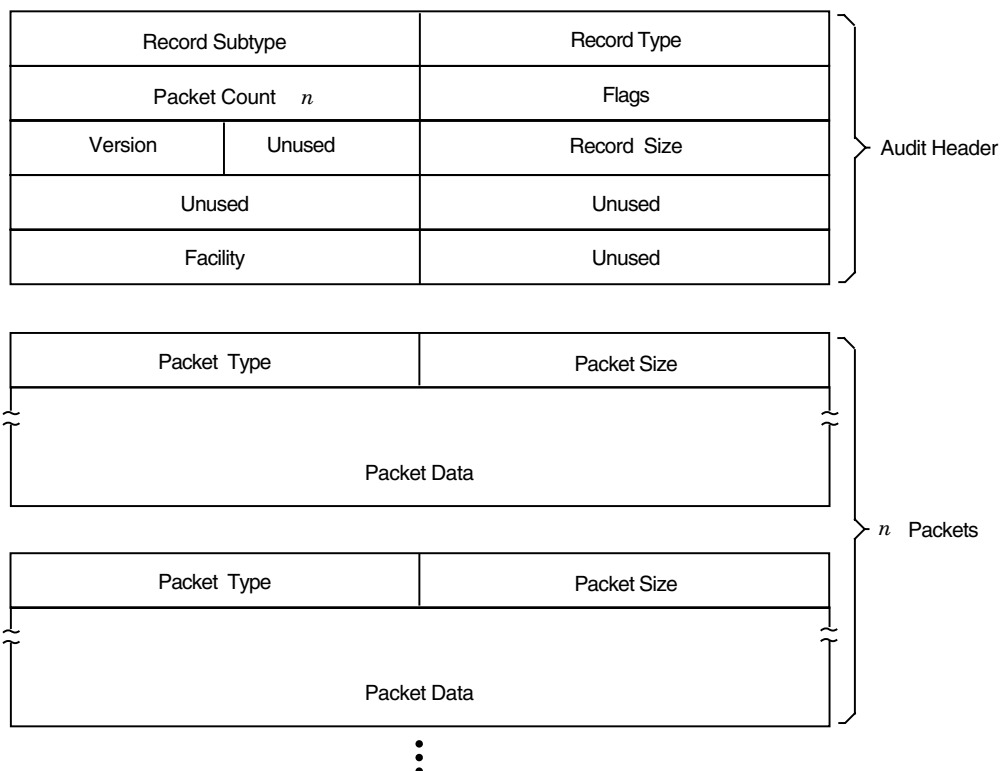
The symbolic names referenced in both the identification and the file summary records are defined in the system definition macro \$USGDEF. The length of the identification record is USG\$K\_IDENT\_LEN. The length of a file summary record is USG\$K\_FILE\_LEN.

## Security Audit Message Format

This appendix describes the format of the auditing messages written to the security auditing log file. The default audit log file SECURITY.AUDIT\$JOURNAL is created by default in the SYS\$COMMON:[SYSMGR] directory.

Each security audit record consists of a header packet followed by one or more data packets, as shown in Figure F-1. The number of data packets depends on the type of information being sent. This appendix describes the format of the audit header and its data packets as well as the contents of the data packets.

**Figure F-1 Format of a Security Audit Message**



ZK-3422A-GE

# Security Audit Message Format

## F.1 Audit Header Packet

### F.1 Audit Header Packet

Table F-1 describes the fields contained in Figure F-2.

**Figure F-2 Audit Header Packet Format**

NSA\$W_RECORD_SUBTYPE		NSA\$W_RECORD_TYPE
NSA\$W_PACKET_COUNT		NSA\$W_FLAGS
NSA\$C_VERSION_3	Unused	NSA\$W_RECORD_SIZE
Unused		Unused
NSA\$W_FACILITY		Unused

ZK-0825A-GE

**Table F-1 Description of the Audit Header Fields**

Field	Symbolic Offset	Contents
Type	NSA\$W_RECORD_TYPE	Indicates the type of event that has occurred. See Table F-2 for details.
Subtype	NSA\$W_RECORD_SUBTYPE	Further defines the type of event that has occurred. See Table F-2 for details.
Flags	NSA\$W_FLAGS	Identifies any flags associated with the audited event. See Table F-3 for details. Reserved to HP. (Word)
Packet count	NSA\$W_PACKET_COUNT	Number of data packets in the audit record. (Word)
Record size	NSA\$W_RECORD_SIZE	Total size of the audit message; the size represents the header packet plus all its data packets. (Word)
Version	NSA\$C_VERSION_3	Indicates the version of the security auditing facility. The symbol NSA\$C_VERSION_3 indicates the current version. (Byte)
Facility	NSA\$W_FACILITY	The facility code for the generated event. By default, this field is zero, indicating a system-generated event. (Word)

When you enter subtypes, do not include a prefix, as shown in Table F-2.

Symbols representing the types or subtypes of security events are listed in Table F-2. For each audit event record type defined by NSA\$W\_RECORD\_TYPE, there is a record subtype defined by the symbol NSA\$W\_RECORD\_SUBTYPE, which further defines the event.



**Table F-2 Description of Audit Event Types and Subtypes**

Symbols for Event Types and Subtypes	Meaning
NSA\$C_MSG_AUDIT	Systemwide change to auditing
ALARM_STATE	Events enabled as alarms
AUDIT_DISABLED	Audit events disabled
AUDIT_ENABLED	Audit events enabled
AUDIT_INITIATE	Audit server startup
AUDIT_LOG_FIRST	First entry in audit log (backward link)
AUDIT_LOG_FINAL	Final entry in audit log (forward link)
AUDIT_STATE	Events enabled as audits
AUDIT_TERMINATE	Audit server shutdown
SNAPSHOT_ABORT <sup>1</sup>	System snapshot attempt has aborted
SNAPSHOT_ACCESS <sup>1</sup>	Snapshot file access/deaccess
SNAPSHOT_SAVE <sup>1</sup>	System snapshot save in progress
SNAPSHOT_STARTUP <sup>1</sup>	System booted from a snapshot file
NSA\$C_MSG_BREAKIN	Break-in attempt detected
BATCH	Batch process
DETACHED	Detached process
DIALUP	Dialup interactive process
LOCAL	Local interactive process
NETWORK	Network server task
REMOTE	Interactive process from another network node
SUBPROCESS	Subprocess
NSA\$C_MSG_CONNECTION	Logical link connection or termination
CNX_ABORT	Connection aborted
CNX_ACCEPT	Connection accepted
CNX_DECNET_CREATE	DECnet logical link created
CNX_DECNET_DELETE	DECnet logical link disconnected
CNX_DISCONNECT	Connection disconnected
CNX_INC_ABORT	Incoming connection request aborted
CNX_INC_ACCEPT	Incoming connection request accepted
CNX_INC_DISCONNECT	Incoming connection disconnected
CNX_INC_REJECT	Incoming connection request rejected
CNX_INC_REQUEST	Incoming connection request
CNX_IPC_CLOSE	Interprocess communication association closed
CNX_IPC_OPEN	Interprocess communication association opened
CNX_REJECT	Connection rejected

<sup>1</sup>Obsolete as of OpenVMS Version 7.1

(continued on next page)

## Security Audit Message Format

### F.1 Audit Header Packet

**Table F-2 (Cont.) Description of Audit Event Types and Subtypes**

<b>Symbols for Event Types and Subtypes</b>	<b>Meaning</b>
CNX_REQUEST	Connection requested
NSA\$C_MSG_INSTALL	Use of the Install utility (INSTALL)
INSTALL_ADD	Known image installed
INSTALL_REMOVE	Known image deleted
NSA\$C_MSG_LOGFAIL	Login failure
See subtypes for NSA\$C_MSG_BREAKIN	
NSA\$C_MSG_LOGIN	Successful login
See subtypes for NSA\$C_MSG_BREAKIN	
NSA\$C_MSG_LOGOUT	Successful logout
See subtypes for NSA\$C_MSG_BREAKIN	
NSA\$C_MSG_MOUNT	Volume mount or dismount
VOL_DISMOUNT	Volume dismount
VOL_MOUNT	Volume mount
NSA\$C_MSG_NCP	Modification to network configuration database
NCP_COMMAND	Network Control Program (NCP) command issued
NSA\$C_MSG_NETPROXY	Modification to network proxy database
NETPROXY_ADD	Record added to network proxy authorization file
NETPROXY_DELETE	Record removed from network proxy authorization file
NETPROXY_MODIFY	Record modified in network proxy authorization file
NSA\$C_MSG_OBJ_ACCESS	Object access attempted
OBJ_ACCESS	Access attempted to create, delete, or deaccess an object
NSA\$C_MSG_OBJ_CREATE	Object creation attempted
OBJ_CREATE	Access attempted to create an object
NSA\$C_MSG_OBJ_DEACCESS	Object deaccessed
OBJ_DEACCESS	Attempt to complete access to an object

(continued on next page)

**Table F-2 (Cont.) Description of Audit Event Types and Subtypes**

Symbols for Event Types and Subtypes	Meaning
NSA\$C_MSG_OBJ_DELETE	Object deletion attempted
OBJ_DELETE	Object deletion attempted
NSA\$C_MSG_PROCESS	Process controlled through a system service
PRC_CANWAK	Process wakeup canceled
PRC_CREPRC	Process created
PRC_DELPRC	Process deleted
PRC_FORCEX	Process exit forced
PRC_GETJPI	Process information gathered
PRC_GRANTID	Process identifier granted
PRC_RESUME	Process resumed
PRC_REVOKID	Process identifier revoked
PRC_SCHDWK	Process wakeup scheduled
PRC_SETPRI	Process priority altered
PRC_SIGPRC	Process exception issued
PRC_SUSPND	Process suspended
PRC_TERM	Process termination notification requested
PRC_WAKE	Process wakeup issued
NSA\$C_MSG_PRVAUD	Use of privilege
PRVAUD_FAILURE	Unsuccessful use of privilege
PRVAUD_SUCCESS	Successful use of privilege
NSA\$C_MSG_RIGHTSDB	Modification to the rights database
RDB_ADD_ID	Identifier added to rights database
RDB_CREATE	Rights database created
RDB_GRANT_ID	Identifier granted to user
RDB_MOD HOLDER	List of identifier holders modified
RDB_MOD_ID	Identifier name or attributes modified
RDB_REM_ID	Identifier removed from rights database
RDB_REVOKE_ID	Identifier taken away from user
NSA\$C_MSG_SYSGEN	Use of the System Generation utility (SYSGEN)
SYSGEN_SET	System parameter modified
NSA\$C_MSG_SYSTIME	Modification to system time
SYSTIM_SET	System time set
SYSTIM_CAL	System time calibrated

(continued on next page)

## Security Audit Message Format

### F.1 Audit Header Packet

**Table F–2 (Cont.) Description of Audit Event Types and Subtypes**

Symbols for Event Types and Subtypes	Meaning
NSA\$C_MSG_SYSUAF	Modification to system user authorization file (SYSUAF)
SYSUAF_ADD	Record added to system user authorization file
SYSUAF_COPY	Record added to system user authorization file
SYSUAF_DELETE	Record deleted from system user authorization file
SYSUAF_MODIFY	Record modified in system user authorization file
SYSUAF_RENAME	Record renamed in system user authorization file

Table F–3 identifies any flags associated with the audited event.

The symbol NSA\$K\_MSG\_HDR\_LENGTH defines the current size of the message header (in bytes).

**Table F–3 Description of Audit Event Flags**

Symbol	Meaning
NSA\$M_ACL	Event generated by an alarm access control entry (ACE) or an audit ACE.
NSA\$M_ALARM	Event is a security alarm.
NSA\$M_AUDIT	Event is a security audit.
NSA\$M_FLUSH	Event forced the audit server to write all buffered event messages to the audit log file.
NSA\$M_FOREIGN	Event occurred outside of the system trusted computing base.
NSA\$M_MANDATORY	Event resulted from a mandatory process audit.

**Note**

All other flags besides those listed in the table are reserved by HP.

## F.2 Audit Data Packets

Figure F–3 illustrates the format of an audit data packet. NSA\$K\_PKT\_HDR\_LENGTH defines the current size of each packet header (in bytes).

Note that audit data packets do not appear in any predefined order within an event message, and packet types can appear more than once throughout the event message.

For examples of the types of data appearing in different event messages, refer to the appendix of alarm messages in the *HP OpenVMS Guide to System Security*.

**Figure F-3 Audit Data Packet Format**

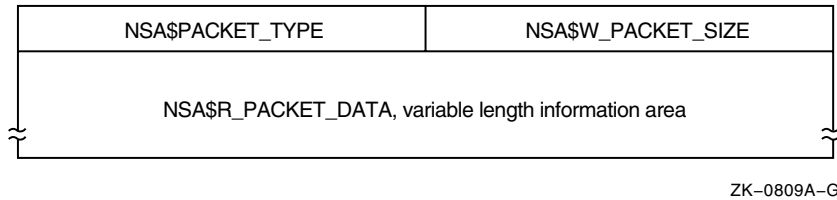


Table F-4 describes the fields contained in these packets.

**Table F-4 Description of the Audit Data Packet**

Field	Symbolic Offset	Contents
Packet size	NSA\$W_PACKET_SIZE	Indicates the size of the data packet. (Word)
Packet type	NSA\$W_PACKET_TYPE	Indicates the type of data in the packet, as described in Table F-5.
Packet data	NSA\$R_PACKET_DATA	Variable length field containing the packet data.

Table F-5 describes the types of data in audit packets.

**Table F-5 Types of Data in Audit Packets**

Symbol	Packet Contents
NSA\$_ACCESS_DESIRED	Access requested or granted to the object as defined by \$ARMDEF (Longword)
NSA\$_ACCESS_MODE	Access mode of the process (Byte)
NSA\$_ACCOUNT	Account name associated with the process (String of 1-32 characters)
NSA\$_ALARM_NAME	Name of the user (or the security class operators terminal) to receive the record (String of 1-32 characters)
NSA\$_ASSOCIATION_NAME	Interprocess communication (IPC) association name (String of 1-256 characters)
NSA\$_AUDIT_FLAGS	Bit mask of enabled or disabled events. This is reserved to HP. (40-byte record) (String of 1-65 characters)
NSA\$_AUDIT_NAME	Journal file to receive the audit record (String of 1-65 characters)
NSA\$_COMMAND_LINE	Command line the user entered (String of 1-2048 characters)
NSA\$_CONNECTION_ID	Interprocess communication (IPC) connection identification (Longword)
NSA\$_DECNET_LINK_ID	DECnet logical link identification (Longword)
NSA\$_DECNET_OBJECT_NAME	DECnet object name (String of 1-16 characters)
NSA\$_DECNET_OBJECT_NUMBER	DECnet object number (Longword)
NSA\$_DEFAULT_USERNAME	Default local user name for incoming network proxy requests (String of 1-32 characters)

(continued on next page)

## Security Audit Message Format

### F.2 Audit Data Packets

**Table F–5 (Cont.) Types of Data in Audit Packets**

<b>Symbol</b>	<b>Packet Contents</b>
NSA\$_DEVICE_NAME	Device name where the volume resides (String of 1-64 characters)
NSA\$_DIRECTORY_ENTRY	Directory entry associated with file system operation (Longword)
NSA\$_DIRECTORY_ID	Directory file identification (Array of 3 words)
NSA\$_DIRECTORY_NAME	Directory file name
NSA\$_DISMOUNT_FLAGS	The \$DMTDEF macro in STARLET defines the dismount flags; each flag is one quadword.
NSA\$_EFC_NAME	Event flag cluster name (String of 1-16 characters)
NSA\$_EVENT_FACILITY	Facility code for the generated event (Word)
NSA\$_FIELD_NAME	Name of the field being modified. This is used in combination with NSA\$_ORIGINAL_DATA and NSA\$_NEW_DATA. (String of 1-256 characters)
NSA\$_FILE_ID	File identification (Array of words)
NSA\$_FINAL_STATUS	Status (successful or unsuccessful) causing the auditing facility to be invoked (Longword)
NSA\$_HOLDER_NAME	Name of user holding the identifier (String of 1-32 characters)
NSA\$_HOLDER_OWNER	Owner (UIC) of holder (Longword)
NSA\$_ID_ATTRIBUTES	Attributes of the identifier, which are defined by the \$KGBDEF macro in STARLET (Longword)
NSA\$_IDENTIFIERS_USED	Identifiers (from the access control entry (ACE) granting access) used to gain access to the object (Array of longwords)
NSA\$_ID_NAME	Name of the identifier (String of 1-32 characters)
NSA\$_ID_NEW_ATTRIBUTES	New attributes of the identifier, which are defined by the \$KGBDEF macro in STARLET (Longword)
NSA\$_ID_NEW_NAME	New name of the identifier (String of 1-32 characters)
NSA\$_ID_NEW_VALUE	New value of the identifier (Longword)
NSA\$_ID_VALUE	Value of the identifier (Longword)
NSA\$_ID_VALUE_ASCII	Identification value provided by \$IDTOASC (Longword)
NSA\$_IMAGE_NAME	Name of the image being executed when the event took place (String of 1-1024 characters)
NSA\$_INSTALL_FILE	The name of the installed file (String of 1-255 characters)
NSA\$_INSTALL_FLAGS	The INSTALL flags correspond to qualifiers for the Install utility (for example, NSA\$_M_INS_EXECUTE_ONLY); each flag is one longword.
NSA\$_LNM_PARENT_NAME	Name of the parent logical name table (String of 1-31 characters)
NSA\$_LNM_TABLE_NAME	Name of the logical name table (String of 1-31 characters)
NSA\$_LOCAL_USERNAME	User name of the account available for incoming network proxy requests (String of 1-32 characters)

(continued on next page)

## Security Audit Message Format F.2 Audit Data Packets

**Table F-5 (Cont.) Types of Data in Audit Packets**

Symbol	Packet Contents
NSA\$_LOGICAL_NAME	Logical name associated with the device (String of 1-255 characters)
NSA\$_MAILBOX_UNIT	Mailbox unit number (Longword)
NSA\$_MATCHING_ACE	ACE granting or denying access (Array of bytes)
NSA\$_MESSAGE	Associated message code; see NSA\$_MSGFILNAM for translation (Longword)
NSA\$_MOUNT_FLAGS	The MOUNT flags defined by the \$MNTDEF macro in STARLET (Longword)
NSA\$_MSGFILNAM	Message file containing the translation for the message code in NSA\$_MESSAGE (String of 1-255 characters)
NSA\$_NEW_DATA	Contents of the field named in NSA\$_FIELD_NAME after the event occurred. NSA\$_ORIGINAL_DATA contains the field contents prior to the event. (String of 1-n characters)
NSA\$_NEW_IMAGE_NAME	Name of the new image (String of 1-1024 characters)
NSA\$_NEW_OWNER	New process owner (UIC) (Longword)
NSA\$_NEW_PRIORITY	New process priority (Longword)
NSA\$_NEW_PRIVILEGES	New privileges (Quadword)
NSA\$_NEW_PROCESS_ID	New identification of the process (Longword)
NSA\$_NEW_PROCESS_NAME	New name of the process (String of 1-15 characters)
NSA\$_NEW_PROCESS_OWNER	New owner (UIC) of the process (Longword)
NSA\$_NEW_USERNAME	New user name (String of 1-32 characters)
NSA\$_NOP	Packet in static event list to omit from processing
NSA\$_OBJECT_CLASS	Object class name, as defined by the system or by the user (String of 1-23 characters)
NSA\$_OBJECT_MAX_CLASS	The minimum access classification of the object (20-byte record)
NSA\$_OBJECT_MIN_CLASS	The minimum access classification of the object (20-byte record)
NSA\$_OBJECT_NAME	Object's name (String of 1-255 characters)
NSA\$_OBJECT_NAME_2	Alternate object name; currently applies to file-backed global sections where the alternate name of global section is the file name. (String of 1-255 characters)
NSA\$_OBJECT_OWNER	UIC or general identifier of the process causing the auditable event (Longword)
NSA\$_OBJECT_PROTECTION	UIC-based protection of the object (Vector of words or longwords)
NSA\$_OBJECT_TYPE	Object's type code, as listed in \$ACLDEF. (String of 1-23 characters)
NSA\$_OLD_PRIORITY	Former process priority (Longword)
NSA\$_OLD_PRIVILEGES	Former privileges (Quadword)

(continued on next page)

## Security Audit Message Format

### F.2 Audit Data Packets

**Table F–5 (Cont.) Types of Data in Audit Packets**

<b>Symbol</b>	<b>Packet Contents</b>
NSA\$_ORIGINAL_DATA	Contents of the field named in NSA\$_FIELD_NAME before the event occurred. NSA\$_NEW_DATA contains the field contents following the event. (String of 1-n characters)
NSA\$_PARAMS_INUSE	Set of parameter values given to the SYSGEN command USE (String of 1-255 characters)
NSA\$_PARAMS_WRITE	File name for the SYSGEN command WRITE (String of 1-255 characters)
NSA\$_PARENT_ID	Process identifier (PID) of the parent process; only used when auditing events pertaining to a subprocess (Longword)
NSA\$_PARENT_NAME	Parent's process name; only used when auditing events pertaining to a subprocess (String of 1-15 characters)
NSA\$_PARENT_OWNER	Owner (UIC) of the parent process (Longword)
NSA\$_PARENT_USERNAME	User name associated with the parent process (String of 1-32 characters)
NSA\$_PASSWORD	Password used in unsuccessful break-in attempt (String of 1-32 characters)
NSA\$_PRIVILEGES	Privilege mask (Quadword)
NSA\$_PRIVS_MISSING	Privileges that are lacking (Longword or quadword)
NSA\$_PRIVS_USED	Privileges used to gain access to the object (Longword or quadword)
NSA\$_PROCESS_ID	PID of the process causing the auditable event (Longword)
NSA\$_PROCESS_NAME	Process' name that caused the auditable event (String of 1-15 characters)
NSA\$_REM_ASSOCIATION_NAME	Interprocess communication (IPC) remote association name (String of 1-256 characters)
NSA\$_REMOTE_LINK_ID	Remote logical link identification number (Longword)
NSA\$_REMOTE_NODE_ID	DECnet address of the remote process (Longword)
NSA\$_REMOTE_NODENAME	DECnet node name of the remote process (String of 1-6 characters)
NSA\$_REMOTE_USERNAME	User name of the remote process (String of 1-32 characters)
NSA\$_REQUEST_NUMBER	Request number associated with the system service call (Longword)
NSA\$_RESOURCE_NAME	Lock resource name (String of 1-32 characters)
NSA\$_SECTION_NAME	Global section name (String of 1-42 characters)
NSA\$_SNAPSHOT_BOOTFILE	The name of the snapshot boot file, the saved system image file from which the system just booted (String of 1-255 characters)
NSA\$_SNAPSHOT_SAVE_FILNAM	The name of the snapshot save file, which is the original location of the snapshot file at the time that the system was saved (String of 1-255 characters)

(continued on next page)



## Security Audit Message Format F.2 Audit Data Packets

**Table F–5 (Cont.) Types of Data in Audit Packets**

Symbol	Packet Contents
NSA\$_SNAPSHOT_TIME	The time the picture of the configuration was taken and saved in the snapshot boot file (Quadword)
NSA\$_SOURCE_PROCESS_ID	Identification of process originating the request (Longword)
NSA\$_SUBJECT_CLASS	The current access class of the process causing the auditable event (A 20-byte record)
NSA\$_SUBJECT_OWNER	Owner (UIC) of the process causing the event (Longword)
NSA\$_SYSTEM_ID	SCS identification of the cluster node where the event took place (SYSGEN parameter SCSSYSTEMID) (Longword)
NSA\$_SYSTEM_NAME	System Communication Services (SCS) node name where the event took place (SYSGEN parameter SCSNODE) (String of 1-6 characters)
NSA\$_SYSTEM_SERVICE_NAME	Name of the system service associated with the event (String of 1-256 characters)
NSA\$_SYSTEM_NEW	New system time (Quadword)
NSA\$_SYSTEM_OLD	Old system time (Quadword)
NSA\$_TARGET_DEVICE_NAME	Target device name (String of 1-64 characters)
NSA\$_TARGET_PROCESS_CLASS	The target process classification. (A 20-byte vector)
NSA\$_TARGET_PROCESS_ID	Target process identifier (PID) (Longword)
NSA\$_TARGET_PROCESS_NAME	Target process name (String of 1-64 characters)
NSA\$_TARGET_PROCESS_OWNER	Target process owner (UIC) (Longword)
NSA\$_TARGET_USERNAME	Target user name (String of 1-32 characters)
NSA\$_TERMINAL	Name of the terminal to which the process was connected when the auditable event occurred (String of 1-256 characters)
NSA\$_TIME_STAMP	The time that the event occurred (Quadword)
NSA\$_TRANSPORT_NAME	Name of transport: interprocess communication (IPC), DECnet, or System Management Integrator (SMI), which handles requests from the SYSMAN utility (String of 1-256 characters)
NSA\$_UAF_ADD	Name of the authorization record being added (String of 1-32 characters)
NSA\$_UAF_COPY	Original and new names of the authorization record being copied (String of 1-32 characters)
NSA\$_UAF_DELETE	Name of the authorization record being removed (String of 1-32 characters)
NSA\$_UAF_FIELDS	Fields being changed in an authorization record and their new values. This is reserved to HP. (Quadword bit mask)
NSA\$_UAF_MODIFY	Name of the authorization record being modified (String of 1-32 characters)
NSA\$_UAF_RENAME	Name of the authorization record being renamed (String of 1-32 characters)

(continued on next page)

## Security Audit Message Format

### F.2 Audit Data Packets

**Table F-5 (Cont.) Types of Data in Audit Packets**

<b>Symbol</b>	<b>Packet Contents</b>
NSA\$_UAF_SOURCE	User name of the source record for an Authorize utility (AUTHORIZE) copy operation (String of 1-32 characters)
NSA\$_USERNAME	User name of process causing the auditable event (String of 1-32 characters)
NSA\$_VOLUME_NAME	Volume name (String of 1-15 characters)
NSA\$_VOLUME_SET_NAME	Volume set name (String of 1-15 characters)

---

## Valid Combinations of BACKUP Qualifiers

The following figures show the qualifiers that can be used in BACKUP save, restore, copy, compare and list operations. The figures also indicate valid combinations of BACKUP qualifiers.

- Figure G–1 shows command qualifiers used in save operations.
- Figure G–2 shows input file-selection qualifiers used in save operations.
- Figure G–3 shows output save-set qualifiers used in save operations.
- Figure G–4 shows command qualifiers used in restore operations.
- Figure G–5 shows input save-set qualifiers used in restore operations.
- Figure G–6 shows output file qualifiers used in restore operations.
- Figure G–7 shows command qualifiers used in copy operations.
- Figure G–8 shows input file-selection qualifiers used in copy operations.
- Figure G–9 shows output file qualifiers used in copy operations.
- Figure G–10 shows command qualifiers used in compare operations.
- Figure G–11 shows input file-selection qualifiers used in compare operations.
- Figure G–12 shows input save-set qualifiers used in compare operations.

# Valid Combinations of BACKUP Qualifiers

Figure G-1 Command Qualifiers Used in Save Operations

Command Qualifiers	/[NO]ASSIST	/BRIEF	/DELETE	/FAST	/FULL	/IGNORE	/IMAGE	/[NO]INITIALIZE	/INTERCHANGE	/JOURNAL	/LIST	/[NO]LOG	/PHYSICAL	/RECORD	/VERIFY	/VOLUME
/[NO]ASSIST <sup>1</sup>	-	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y
/BRIEF <sup>2</sup>	Y	-	Y	Y	N	Y	Y	N	Y	Y	Y	Y	N	Y	Y	Y
/DELETE	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y	Y
/FAST	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y
/FULL <sup>2</sup>	Y	N	Y	Y	-	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y
/IGNORE	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/IMAGE	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	N	Y	Y	Y	Y
/[NO]INITIALIZE	N	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	N	Y	Y	Y
/INTERCHANGE	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	N	Y	Y	Y	Y
/JOURNAL	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	N	Y	Y	Y	Y
/LIST	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y <sup>3</sup>	N	Y	Y	Y
/NOINCREMENTAL	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/[NO]LOG	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	N	Y	Y
/PHYSICAL	Y	N	N	N	N	Y	N	N	N	N	N	N	-	N	Y	N
/RECORD	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	-	Y	N
/VERIFY	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
/VOLUME <sup>4</sup>	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	-
Input File-Selection Qualifiers																
/BACKUP	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	N
/BEFORE	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	N
/BY_OWNER	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	N
/CONFIRM	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	N
/CREATED	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	N
/EXCLUDE	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	N
/EXPIRED	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	N
/MODIFIED	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	N
/SINCE	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	N
Output File Qualifiers																
/BLOCK_SIZE	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/BY_OWNER	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
/COMMENT	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/[NO]CRC	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/DENSITY <sup>1,6</sup>	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
/GROUP_SIZE	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/LABEL <sup>1</sup>	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
/PROTECTION	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/[NO]REWIND <sup>6</sup>	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
/SAVE_SET <sup>5</sup>	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/TAPE_EXPIRATION <sup>1</sup>	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y

<sup>1</sup> Use only when saving to magnetic tape save sets.  
<sup>2</sup> You must specify the /LIST qualifier with this qualifier.  
<sup>3</sup> When you specify /LIST with /LOG, direct the output from the list operation to a file.  
<sup>4</sup> You must specify /IMAGE with this qualifier.  
<sup>5</sup> Required when creating save sets on disk.  
<sup>6</sup> You must specify /REWIND with this qualifier.

## Valid Combinations of BACKUP Qualifiers

**Figure G-2 Input File-Selection Qualifiers Used in Save Operations**

Input File-Selection Qualifiers		/BACKUP	/BEFORE	/BY_OWNER	/CONFIRM	/CREATED	/EXCLUDE	/EXPIRED	/MODIFIED	/SINCE
/BACKUP	-	Y	Y	Y	N	Y	N	N	Y	
/BEFORE	Y	-	Y	Y	Y	Y	Y	Y	N	
/BY_OWNER	Y	Y	-	Y	Y	Y	Y	Y	Y	
/CONFIRM	Y	Y	Y	-	Y	Y	Y	Y	Y	
/CREATED	N	Y	Y	Y	-	Y	N	N	Y	
/EXCLUDE	Y	Y	Y	Y	Y	-	Y	Y	Y	
/EXPIRED	N	Y	Y	Y	N	Y	-	N	Y	
/MODIFIED	N	Y	Y	Y	N	Y	N	Y	Y	
/SINCE	Y	N	Y	Y	Y	Y	Y	Y	-	
<hr/>										
Output Save-Set Qualifiers										
/BLOCK_SIZE	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/BY_OWNER	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/COMMENT	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/[NO]CRC	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/DENSITY <sup>1,2</sup>	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/GROUP_SIZE	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/LABEL <sup>1</sup>	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/PROTECTION	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/[NO]REWIND <sup>1</sup>	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/SAVE_SET <sup>3</sup>	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/TAPE_EXPIRATION <sup>1</sup>	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

<sup>1</sup> Use only when saving to magnetic tape save sets.

<sup>2</sup> You must specify /REWIND with this qualifier.

<sup>3</sup> Required when creating save sets on disk.

# Valid Combinations of BACKUP Qualifiers

Figure G-3 Output Save-Set Qualifiers Used in Save Operations

Input File-Selection Qualifiers		/BLOCK_SIZE	/BY_OWNER	/COMMENT	/[NO]CRC	/DENSITY	/GROUP_SIZE	/LABEL	/MEDIA_FORMAT=[NO]COMPACTION	/PROTECTION	/[NO]REWIND	/SAVE_SET	/TAPE_EXPIRATION
/BACKUP	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/BEFORE	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/BY_OWNER	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/CONFIRM	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/CREATED	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/EXCLUDE	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/EXPIRED	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/MODIFIED	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/SINCE	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Output Save-Set Qualifiers													
/BLOCK_SIZE	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/BY_OWNER	Y	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/COMMENT	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/[NO]CRC	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	Y	Y
/DENSITY <sup>1, 2</sup>	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	Y
/GROUP_SIZE	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y
/LABEL <sup>1</sup>	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y
/MEDIA_FORMAT=[NO]COMPACTION <sup>1</sup>	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y
/PROTECTION	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y
/[NO]REWIND <sup>1</sup>	Y	Y	Y	Y	Y	Y <sup>2</sup>	Y	Y	Y	-	Y	Y	Y
/SAVE_SET <sup>3</sup>	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y
/TAPE_EXPIRATION <sup>1</sup>	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y

<sup>1</sup> Use only when saving to magnetic tape save sets.

<sup>2</sup> You must specify /REWIND with this qualifier.

<sup>3</sup> Required when creating save sets on disk.

## Valid Combinations of BACKUP Qualifiers

Figure G-4 Command Qualifiers Used in Restore Operations

Command Qualifiers	/[NO]ASSIST	/BRIEF	/FULL	/IMAGE	/INCREMENTAL	/[NO]INITIALIZE	/LIST	/[NO]LOG	/PHYSICAL	/[NO]TRUNCATE	/VERIFY	/VOLUME
/[NO]ASSIST <sup>1</sup>	-	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
/BRIEF <sup>2</sup>	Y	-	N	Y	Y	Y	Y	Y	N	Y	Y	Y
/FULL <sup>2</sup>	Y	N	-	Y	Y	Y	Y	Y	N	Y	Y	Y
/IMAGE	Y	Y	Y	-	N	Y	Y	Y	N	Y	Y	Y
/INCREMENTAL	Y	Y	Y	N	-	N	Y	Y	N	Y	Y	Y
/[NO]INITIALIZE	N	Y	Y	Y	N	-	Y	Y	N	Y	Y	Y
/LIST	Y	Y	Y	Y	Y	Y	-	Y <sup>3</sup>	N	Y	Y	Y
/[NO]LOG	Y	Y	Y	Y	Y	Y	Y	Y <sup>3</sup>	N	Y	Y	Y
/PHYSICAL	Y	N	N	N	N	N	N	N	-	N	Y	N
/[NO]TRUNCATE	Y	Y	Y	Y	Y	Y	Y	Y	N	-	Y	Y
/VERIFY	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
/VOLUME <sup>4</sup>	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	-

Input Save-Set Qualifiers	/[NO]ASSIST	/BRIEF	/FULL	/IMAGE	/INCREMENTAL	/[NO]INITIALIZE	/LIST	/[NO]LOG	/PHYSICAL	/[NO]TRUNCATE	/VERIFY	/VOLUME
/[NO]CRC	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
/LABEL <sup>1</sup>	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
/[NO]REWIND <sup>1</sup>	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
/SAVE_SET <sup>5</sup>	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/SELECT	Y	Y	Y	N	Y	N	Y	Y	N	Y	Y	N

Output File Qualifiers	/[NO]ASSIST	/BRIEF	/FULL	/IMAGE	/INCREMENTAL	/[NO]INITIALIZE	/LIST	/[NO]LOG	/PHYSICAL	/[NO]TRUNCATE	/VERIFY	/VOLUME
/BY_OWNER	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
/NEW_VERSION	Y	Y	Y	N	Y	Y	Y	Y	N	Y	N	Y
/OVERLAY	Y	Y	Y	N	Y	Y	Y	Y	N	Y	Y	Y
/REPLACE	Y	Y	Y	N	Y	Y	Y	Y	N	Y	Y	Y

<sup>1</sup> Use only when restoring from magnetic tape save sets.

<sup>2</sup> You must specify the /LIST qualifier with this qualifier.

<sup>3</sup> When you specify /LIST with /LOG, direct the output from the list operation to a file.

<sup>4</sup> You must specify /IMAGE with this qualifier.

<sup>5</sup> Required when restoring save sets from disk.

## Valid Combinations of BACKUP Qualifiers

**Figure G–5 Input Save-Set Qualifiers Used in Restore Operations**

Input SaveSet Qualifiers	/[NO]CRC	/[NO]REWIND	/SAVE_SET	/SELECT
/[NO]CRC	–	Y	Y	Y
/[NO]REWIND <sup>1</sup>	Y	–	Y	Y
/SAVE_SET <sup>2</sup>	Y	Y	–	Y
/SELECT	Y	Y	Y	–

---

Output File Qualifiers				
/BY_OWNER	Y	Y	Y	Y
/NEW_VERSION	Y	Y	Y	Y
/OVERLAY	Y	Y	Y	Y
/REPLACE	Y	Y	Y	Y

<sup>1</sup> Use only when restoring from magnetic tape save sets.

<sup>2</sup> Required when restoring save sets from disk.

VM-0949A-AI

**Figure G–6 Output File Qualifiers Used in Restore Operations**

Input Save-Set Qualifiers	/BY_OWNER	/NEW_VERSION	/OVERLAY	/REPLACE
/[NO]CRC	Y	Y	Y	Y
/[NO]REWIND <sup>1</sup>	Y	Y	Y	Y
/SAVE_SET <sup>2</sup>	Y	Y	Y	Y
/SELECT	Y	Y	Y	Y

---

Output File Qualifiers				
/BY_OWNER	–	Y	Y	Y
/NEW_VERSION	Y	–	N	N
/OVERLAY	Y	N	–	N
/REPLACE	Y	N	N	–

<sup>1</sup> Use only when restoring from magnetic tape save sets.

<sup>2</sup> Required when restoring save sets from disk.

ZK–0837A–GE



## Valid Combinations of BACKUP Qualifiers

**Figure G-7 Command Qualifiers Used in Copy Operations**

Command Qualifiers		/BRIEF	/DELETE	/FAST	/FULL	/IGNORE	/IMAGE	/[NO]INITIALIZE	/LIST	/[NO]LOG	/PHYSICAL	/RECORD	/[NO]TRUNCATE	/VERIFY	/VOLUME
/BRIEF <sup>1</sup>	-	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y
/DELETE	Y	-	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y	Y	Y
/FAST	Y	Y	-	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y
/FULL <sup>1</sup>	N	Y	Y	-	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y
/IGNORE	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
/IMAGE	Y	Y	Y	Y	Y	-	Y	Y	Y	N	Y	Y	Y	Y	Y
/[NO]INITIALIZE	Y	Y	Y	Y	Y	Y	-	Y	Y	N	Y	Y	Y	Y	Y
/LIST	Y	Y	Y	Y	Y	Y	Y	-	Y <sup>2</sup>	Y	Y	Y	Y	Y	Y
/[NO]LOG	Y	Y	Y	Y	Y	Y	Y	Y <sup>2</sup>	-	N	Y	Y	Y	Y	Y
/PHYSICAL	N	N	N	N	Y	N	N	N	N	-	N	N	Y	N	N
/RECORD	Y	N	Y	Y	Y	Y	Y	Y	Y	N	-	Y	Y	Y	Y
/[NO]TRUNCATE	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	-	Y	Y	Y
/VERIFY	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y
/VOLUME <sup>3</sup>	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	-

Input File-Selection Qualifiers															
/BACKUP	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	N	N
/BEFORE	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	N
/BY_OWNER	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	N
/CONFIRM	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	N
/CREATED	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	N
/EXCLUDE	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	N
/EXPIRED	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	N
/MODIFIED	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	N
/SINCE	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	N

Output File Qualifiers															
/BY_OWNER	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	N
/NEW_VERSION	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	N	N	N
/OVERLAY	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	N
/REPLACE	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	N

<sup>1</sup> You must specify the /LIST qualifier with this qualifier.

<sup>2</sup> When you specify /LIST with /LOG, direct the output from the list operation to a file.

<sup>3</sup> You must specify /IMAGE with this qualifier.

# Valid Combinations of BACKUP Qualifiers

Figure G-8 Input File-Selection Qualifiers Used in Copy Operations

Input File-Selection Qualifiers									
/BACKUP	-	Y	Y	Y	N	Y	N	N	Y
/BEFORE	Y	-	Y	Y	Y	Y	Y	Y	N
/BY_OWNER	Y	Y	-	Y	Y	Y	Y	Y	Y
/CONFIRM	Y	Y	Y	-	Y	Y	Y	Y	Y
/CREATED	N	Y	Y	Y	-	Y	N	N	Y
/EXCLUDE	Y	Y	Y	Y	Y	-	Y	Y	Y
/EXPIRED	N	Y	Y	Y	N	Y	-	N	Y
/MODIFIED	N	Y	Y	Y	N	Y	N	-	Y
/SINCE	Y	N	Y	Y	Y	Y	Y	Y	-
<hr/>									
Output File Qualifiers									
/BY_OWNER	Y	Y	Y	Y	Y	Y	Y	Y	Y
/NEW_VERSION	Y	Y	Y	Y	Y	Y	Y	Y	Y
/OVERLAY	Y	Y	Y	Y	Y	Y	Y	Y	Y
/REPLACE	Y	Y	Y	Y	Y	Y	Y	Y	Y

ZK-6532-GE

## Valid Combinations of BACKUP Qualifiers

**Figure G-9 Output File Qualifiers Used in Copy Operations**

Input File-Selection Qualifiers	/BY_OWNER	/NEW_VERSION	/OVERLAY	/REPLACE
/BACKUP	Y	Y	Y	Y
/BEFORE	Y	Y	Y	Y
/BY_OWNER	Y	Y	Y	Y
/CONFIRM	Y	Y	Y	Y
/CREATED	Y	Y	Y	Y
/EXCLUDE	Y	Y	Y	Y
/EXPIRED	Y	Y	Y	Y
/MODIFIED	Y	Y	Y	Y
/SINCE	Y	Y	Y	Y
<hr/>				
Output File Qualifiers				
/BY_OWNER	-	Y	Y	Y
/NEW_VERSION	Y	-	N	N
/OVERLAY	Y	N	-	N
/REPLACE	Y	N	N	-

ZK-0945A-GE

# Valid Combinations of BACKUP Qualifiers

Figure G-10 Command Qualifiers Used in Compare Operations

Command Qualifiers	/[NO]ASSIST	/BRIEF	/COMPARE	/FAST	/FULL	/IMAGE	/LIST	/[NO]LOG	/PHYSICAL	/VOLUME
/[NO]ASSIST <sup>1</sup>	-	Y	Y	Y	Y	Y	Y	Y	Y	Y
/BRIEF <sup>2</sup>	Y	-	Y	Y	N	Y	Y	Y	Y	Y
/COMPARE	Y	Y	-	Y	Y	Y	Y	Y	Y	Y
/FAST	Y	Y	Y	-	Y	Y	Y	Y	Y	Y
/FULL <sup>2</sup>	Y	N	Y	Y	-	Y	Y	Y	Y	Y
/IMAGE	Y	Y	Y	Y	Y	-	Y	Y	N	Y
/LIST	Y	Y	Y	Y	Y	Y	-	Y <sup>3</sup>	Y	Y
/[NO]LOG	Y	Y	Y	Y	Y	Y	Y <sup>3</sup>	-	Y	Y
/PHYSICAL	Y	Y	Y	Y	Y	N	Y	Y	-	Y
/VOLUME <sup>4</sup>	Y	Y	Y	Y	Y	Y	Y	Y	N	-

Input File-Selection Qualifiers	/[NO]ASSIST	/BRIEF	/COMPARE	/FAST	/FULL	/IMAGE	/LIST	/[NO]LOG	/PHYSICAL	/VOLUME
/BY_OWNER	Y	Y	Y	Y	Y	N	Y	Y	N	N
/BEFORE	Y	Y	Y	Y	Y	N	Y	Y	N	N
/EXCLUDE	Y	Y	Y	Y	Y	N	Y	Y	N	N
/SINCE	Y	Y	Y	Y	Y	N	Y	Y	N	N

<sup>1</sup> Use only when comparing magnetic tape save sets.  
<sup>2</sup> You must specify the /LIST qualifier with this qualifier.  
<sup>3</sup> When you specify /LIST with /LOG, direct the output from the list operation to a file.  
<sup>4</sup> You must specify /IMAGE with this qualifier.

## Valid Combinations of BACKUP Qualifiers

**Figure G–11 Input File-Selection Qualifiers Used in Compare Operations**

Input File–Selection Qualifiers	/BY_OWNER	/BEFORE	/EXCLUDE	/SINCE
/BY_OWNER	–	Y	Y	Y
/BEFORE	Y	–	Y	N
/EXCLUDE	Y	Y	–	Y
/SINCE	Y	N	Y	–

Input Save–Set Qualifiers	/BY_OWNER	/BEFORE	/EXCLUDE	/SINCE
/[NO]REWIND <sup>1</sup>	Y	Y	Y	Y
/SAVE_SET <sup>2</sup>	Y	Y	Y	Y
/SELECT	Y	Y	Y	Y

<sup>1</sup> Use only when comparing magnetic tape save sets.

<sup>2</sup> Required when comparing save sets on disk.

ZK–0946A–GE

**Figure G–12 Input Save-Set Qualifiers Used in Compare Operations**

Input File–Selection Qualifiers	/[NO]REWIND	/SAVE_SET	/SELECT
/BY_OWNER	Y	Y	Y
/BEFORE	Y	Y	Y
/EXCLUDE	Y	Y	Y
/SINCE	Y	Y	Y

Input Save–Set Qualifiers	/[NO]REWIND	/SAVE_SET	/SELECT
/[NO]REWIND <sup>1</sup>	–	Y	Y
/SAVE_SET <sup>2</sup>	Y	–	Y
/SELECT	Y	Y	–

<sup>1</sup> Use only when comparing magnetic tape save sets.

<sup>2</sup> Required when comparing save sets on disk.

ZK–0947A–GE



## A

---

### Access control

- required by ACL editor, *Vol 1*, 1–4
- with Identifier ACEs, *Vol 1*, 1–14
- with protection codes, *Vol 1*, 1–13

### Access control entries

- See ACEs

### Access control list editor

- See ACL editor

### Access control lists

- See ACLs

### Accounting

- enabling image-level accounting, *Vol 1*, 11–7, 11–23

### ACCOUNTING command, *Vol 1*, 2–3

### Accounting files

- copying records between, *Vol 1*, 2–8, 2–24
- processing to produce reports, *Vol 1*, 2–9, 2–12, 2–33
- record format of, *Vol 1*, C–1
- types of record in, *Vol 1*, 2–37

### Accounting reports

- brief reports, *Vol 1*, 2–9
- full reports, *Vol 1*, 2–12
- summary reports, *Vol 1*, 2–33

### Accounting utility (ACCOUNTING), *Vol 1*, 2–3

### ACEs (access control entries), *Vol 1*, 1–3, 1–5

- application-specific, *Vol 1*, 1–5
- Creator ACEs, *Vol 1*, 1–10
- Default Protection ACEs, *Vol 1*, 1–12
- Identifier ACEs, *Vol 1*, 1–14
- ordering entries in an ACL, *Vol 1*, 1–3
- relationship to ACLs, *Vol 1*, 1–3
- security alarm, *Vol 1*, 1–6
- security audit, *Vol 1*, 1–8
- Subsystem ACEs, *Vol 1*, 1–16

### ACL editor, *Vol 1*, 1–3

- ACL qualifiers, *Vol 1*, 1–18 to 1–23
- /CLASS qualifier, *Vol 1*, 1–19
- control access to objects, *Vol 1*, 1–4
- customizing, *Vol 1*, B–1
- exiting from, *Vol 1*, 1–4
- file format required, *Vol 1*, 1–4
- how to use, *Vol 1*, 1–3
- invoking, *Vol 1*, 1–3
- /JOURNAL qualifier, *Vol 1*, 1–20
- keypad editing, *Vol 1*, A–1

### ACL editor (cont'd)

- /MODE qualifier, *Vol 1*, 1–21
- object ownership, *Vol 1*, 1–4
- /OBJECT\_TYPE qualifier, *Vol 1*, 1–22
- privileges required, *Vol 1*, 1–4
- prompting by, *Vol 1*, A–1
- /RECOVER qualifier, *Vol 1*, 1–23
- specifying file type, *Vol 1*, 1–4
- specifying object class, *Vol 1*, 1–4, 1–19
- wildcard restriction, *Vol 1*, 1–4

### ACLs (access control lists)

- impact of length on performance, *Vol 1*, 1–3
- list of supported object classes, *Vol 1*, 1–3
- relationship to ACEs, *Vol 1*, 1–3
- rules for ordering entries, *Vol 1*, 1–3

### ACL section files, *Vol 1*, B–1

### ACP system parameters

- See also System parameters

- ACP\_BASEPRIO, *Vol 2*, C–3
- ACP\_DATACHECK, *Vol 2*, C–3
- ACP\_DINDXCACHE, *Vol 2*, C–4
- ACP\_DIRCACHE, *Vol 2*, C–4
- ACP\_EXTCACHE, *Vol 2*, C–4
- ACP\_EXTLIMIT, *Vol 2*, C–4
- ACP\_FIDCACHE, *Vol 2*, C–4
- ACP\_HDRCACHE, *Vol 2*, C–4
- ACP\_MAPCACHE, *Vol 2*, C–5
- ACP\_MAXREAD, *Vol 2*, C–5
- ACP\_MULTIPLE, *Vol 2*, C–5
- ACP\_QUOCACHE, *Vol 2*, C–5
- ACP\_REBLDSYSD, *Vol 2*, C–5
- ACP\_SHARE, *Vol 2*, C–5
- ACP\_SWAPFLGS, *Vol 2*, C–5
- ACP\_SYSACC, *Vol 2*, C–6
- ACP\_WINDOW, *Vol 2*, C–6
- ACP\_WORKSET, *Vol 2*, C–6
- ACP\_WRITEBACK, *Vol 2*, C–6
- ACP\_XQP\_RES, *Vol 2*, C–6

### Actual node name, *Vol 1*, 13–63

### Actual port name, *Vol 1*, 13–63

### Actual service name, *Vol 1*, 13–63

### Adapters

- See Bus adapters

### ADD (Field) command

- in Show Cluster utility, *Vol 2*, 5–32

### ADD/IDENTIFIER command

- in Authorize utility, *Vol 1*, 5–20

ADD/PROXY command  
     in Authorize utility, *Vol 1, 5–22*  
 ADD CIRCUITS command  
     in Show Cluster utility, *Vol 2, 5–23*  
 ADD CLUSTER command  
     in Show Cluster utility, *Vol 2, 5–25*  
 ADD command  
     in Authorize utility, *Vol 1, 5–7*  
     in Install utility, *Vol 1, 11–6*  
 ADD CONNECTIONS command  
     in Show Cluster utility, *Vol 2, 5–27*  
 ADD COUNTERS command  
     in Show Cluster utility, *Vol 2, 5–29*  
 ADD CREDITS command  
     in Show Cluster utility, *Vol 2, 5–31*  
 ADD DEVICE command  
     in UCM, *Vol 2, 8–12*  
 ADD ERRORS command  
     in Show Cluster utility, *Vol 2, 5–32*  
 ADD LOCAL\_PORTS command  
     in Show Cluster utility, *Vol 2, 5–35*  
 ADD MEMBERS command  
     in Show Cluster utility, *Vol 2, 5–37*  
 ADD SYSTEMS command  
     in Show Cluster utility, *Vol 2, 5–38*  
 AFFINITY\_SKIP system parameter, *Vol 2, C–7*  
 AFFINITY\_TIME system parameter, *Vol 2, C–7*  
 AGEN\$FEEDBACK.DAT file, *Vol 1, 6–7, 6–8*  
 AGEN\$FEEDBACK.EXE file, *Vol 1, 6–7*  
 AGEN\$PARAMS.REPORT file, *Vol 1, 6–9*  
 Alarm ACEs, *Vol 1, 1–6*  
     keywords, *Vol 1, 1–7*  
     specifying access  
         criteria for, *Vol 1, 1–7*  
         specifying attributes for, *Vol 1, 1–6*  
 Alarm messages  
     See Security alarm messages  
 ALF (automatic login facility)  
     adding records, *Vol 2, 7–12*  
     displaying records, *Vol 2, 7–16*  
     removing records, *Vol 2, 7–14*  
 ALF ADD command  
     in SYSMAN, *Vol 2, 7–12*  
 ALF REMOVE command  
     in SYSMAN, *Vol 2, 7–14*  
 ALF SHOW command  
     in SYSMAN, *Vol 2, 7–16*  
 /ALIAS qualifier  
     in Backup utility, *Vol 1, 7–15*  
 ALLOCLASS system parameter, *Vol 2, C–7*  
 ALPHAVMSSYS.OLD file, *Vol 1, 6–10*  
 ALPHAVMSSYS.PAR file, *Vol 1, 6–10*  
 ANALYZE/AUDIT command, *Vol 1, 4–4*  
     See also Audit Analysis utility  
     /BEFORE qualifier, *Vol 1, 4–5*  
     /BINARY qualifier, *Vol 1, 4–6*  
     /BRIEF qualifier, *Vol 1, 4–7*  
 ANALYZE/AUDIT command (cont'd)  
     /EVENT\_TYPE qualifier, *Vol 1, 4–8*  
     /FULL qualifier, *Vol 1, 4–10*  
     /IGNORE qualifier, *Vol 1, 4–11*  
     /INTERACTIVE qualifier, *Vol 1, 4–12*  
     /OUTPUT qualifier, *Vol 1, 4–4, 4–13*  
     /PAUSE qualifier, *Vol 1, 4–14*  
     /SELECT qualifier, *Vol 1, 4–15*  
     /SINCE qualifier, *Vol 1, 4–25*  
     /SUMMARY qualifier, *Vol 1, 4–26*  
 ANALYZE/DISK\_STRUCTURE  
     invoking, *Vol 1, 3–7*  
 ANALYZE/DISK\_STRUCTURE stages, *Vol 1, D–1*  
 Analyze/Disk\_Structure utility (ANALYZE/DISK\_STRUCTURE)  
     directing output, *Vol 1, 3–6, 3–13*  
     examining shadow sets, *Vol 1, 3–17*  
     file recovery, *Vol 1, 3–5*  
     parameter, *Vol 1, 3–6*  
     preventing updates to a volume, *Vol 1, 3–12*  
     qualifiers, *Vol 1, 3–7 to 3–21*  
         /HOMEBLOCKS, *Vol 1, 3–10*  
         /[NO]CONFIRM, *Vol 1, 3–9*  
         /[NO]LIST, *Vol 1, 3–11*  
         /[NO]LOCK\_VOLUME, *Vol 1, 3–12*  
         /[NO]OUTPUT, *Vol 1, 3–13*  
         /[NO]READ\_CHECK, *Vol 1, 3–14*  
         /[NO]REPAIR, *Vol 1, 3–16*  
         /RECORD\_ATTRIBUTES, *Vol 1, 3–15*  
         /SHADOW, *Vol 1, 3–17*  
         /STATISTICS, *Vol 1, 3–19*  
         /USAGE, *Vol 1, 3–21*  
     repairing disk errors, *Vol 1, 3–4, 3–16*  
     reporting disk errors, *Vol 1, 3–4*  
     requirements and limitations, *Vol 1, 3–7*  
 ANALYZE/ERROR\_LOG/ELV command, *Vol 1, 10–3*  
     See also the Error Log Viewer (ELV)  
 Application port, *Vol 1, 13–11, 13–43, 13–44*  
     displaying, *Vol 1, 13–61*  
     mapping, *Vol 1, 13–45*  
 Application service, *Vol 1, 13–11, 13–15, 13–43, 13–47*  
     defined, *Vol 1, 13–17*  
     setting up, *Vol 1, 13–43*  
 /ARB\_SUPPORT  
     qualifier for CREATE command, *Vol 1, 11–7, 11–23*  
 ARB\_SUPPORT system parameter, *Vol 2, C–7*  
 Arrow keys  
     definition, *Vol 2, 5–71*  
     functions of, *Vol 2, B–3*  
 /ASSIST qualifier  
     in Backup utility, *Vol 1, 7–16*  
 Asterisk (\*)  
     as wildcard character, *Vol 1, 7–8*



- Asynchronous device
  - LAT application ports for, *Vol 1*, 13–11, 13–43
- ATTACH command
  - in LATCP, *Vol 1*, 13–5
  - in SYSMAN, *Vol 2*, 7–17
- Attributes
  - Dynamic, *Vol 1*, 5–20, 5–51, 5–73
  - Holder Hidden, *Vol 1*, 5–20, 5–51, 5–73
  - Name Hidden, *Vol 1*, 5–20, 5–51, 5–73
  - No Access, *Vol 1*, 5–20, 5–51, 5–73
  - Resource, *Vol 1*, 5–20, 5–51, 5–73
    - specifying identifier characteristics, *Vol 1*, 1–17
  - Subsystem, *Vol 1*, 5–20, 5–51, 5–73
- Audit ACEs, *Vol 1*, 1–8
  - specifying access, *Vol 1*, 1–9
  - specifying attributes, *Vol 1*, 1–8
- Audit Analysis utility (ANALYZE/AUDIT), *Vol 1*, 4–3 to 4–38
  - CONTINUE command, *Vol 1*, 4–29
  - default input file, *Vol 1*, 4–4
  - directing output, *Vol 1*, 4–4
  - DISPLAY command, *Vol 1*, 4–30
  - displaying current selection criteria, *Vol 1*, 4–38
  - EXIT command, *Vol 1*, 4–32
  - exiting from, *Vol 1*, 4–4, 4–32
  - HELP command, *Vol 1*, 4–33
  - interactive commands, *Vol 1*, 4–12, 4–28
  - invoking, *Vol 1*, 4–4
  - LIST command, *Vol 1*, 4–34
  - NEXT FILE command, *Vol 1*, 4–35
  - NEXT RECORD command, *Vol 1*, 4–36
  - POSITION command, *Vol 1*, 4–37
  - qualifiers, *Vol 1*, 4–4
    - selecting by access type, *Vol 1*, 4–15
    - selecting by account name, *Vol 1*, 4–15
    - selecting by device name, *Vol 1*, 4–16
    - selecting by facility, *Vol 1*, 4–16
    - selecting by identifier, *Vol 1*, 4–17
    - selecting by identifier holder, *Vol 1*, 4–17
    - selecting by image name, *Vol 1*, 4–18
    - selecting by installation event, *Vol 1*, 4–18
    - selecting by local proxy account, *Vol 1*, 4–18
    - selecting by network attribute, *Vol 1*, 4–22
    - selecting by object, *Vol 1*, 4–20
    - selecting by parent process, *Vol 1*, 4–21
    - selecting by privilege, *Vol 1*, 4–21
    - selecting by process, *Vol 1*, 4–21
    - selecting by success status, *Vol 1*, 4–22
    - selecting by system characteristic, *Vol 1*, 4–23
    - selecting by terminal name, *Vol 1*, 4–23
    - selecting by user name, *Vol 1*, 4–23
    - selecting by volume, *Vol 1*, 4–16, 4–18, 4–19, 4–23
  - SHOW command, *Vol 1*, 4–38
- Audit data packets
  - See Security audit data packets
- Audit event flags
  - See Security audit event messages
- Audit header packets
  - See Security audit header packets
- Auditing security records
  - See Security audit records
- Audit log files
  - See Security audit log files
- Audit messages
  - See Security audit event messages
- Audit reports
  - See Security audit reports
- AUTHORIZE command
  - See Authorize utility
- Authorized privileges
  - assigning to known images, *Vol 1*, 11–24
- Authorize utility (AUTHORIZE)
  - ADD/IDENTIFIER command, *Vol 1*, 5–20
  - ADD/PROXY command, *Vol 1*, 5–22
  - ADD command, *Vol 1*, 5–7
    - commands summary (table), *Vol 1*, 5–6
    - controlling access with identifiers, *Vol 1*, 1–14
  - COPY command, *Vol 1*, 5–24
  - CREATE/PROXY command, *Vol 1*, 5–36
  - CREATE/RIGHTS command, *Vol 1*, 5–37
  - DEFAULT command, *Vol 1*, 5–38
    - default password, *Vol 1*, 5–7
  - DEFAULT record, *Vol 1*, 5–7
  - EXIT command, *Vol 1*, 5–50
    - exiting from, *Vol 1*, 5–5
  - GRANT/IDENTIFIER command, *Vol 1*, 5–51
  - HELP command, *Vol 1*, 5–53
    - invoking, *Vol 1*, 5–4
  - LIST/IDENTIFIER command, *Vol 1*, 5–57
  - LIST/PROXY command, *Vol 1*, 5–59
  - LIST/RIGHTS command, *Vol 1*, 5–60
  - LIST command, *Vol 1*, 5–55
    - login flags, *Vol 1*, 5–10, 5–27, 5–41, 5–64
  - MODIFY/IDENTIFIER command, *Vol 1*, 5–73
  - MODIFY/PROXY command, *Vol 1*, 5–75
  - MODIFY/SYSTEM\_PASSWORD command, *Vol 1*, 5–77
  - MODIFY command, *Vol 1*, 5–61
    - privileges required, *Vol 1*, 5–3
  - REMOVE/IDENTIFIER command, *Vol 1*, 5–79
  - REMOVE/PROXY command, *Vol 1*, 5–80
  - REMOVE command, *Vol 1*, 5–78
  - RENAME/IDENTIFIER command, *Vol 1*, 5–84
  - RENAME command, *Vol 1*, 5–81
    - restrictions, *Vol 1*, 5–3
  - REVOKE/IDENTIFIER command, *Vol 1*, 5–85
  - SHOW/IDENTIFIER command, *Vol 1*, 5–91
  - SHOW/PROXY command, *Vol 1*, 5–93
  - SHOW/RIGHTS command, *Vol 1*, 5–95
  - SHOW command, *Vol 1*, 5–86

Autoconfiguration  
 Alpha and I64 devices, *Vol 2, 7-69*  
 VAXstation 2000 (table), *Vol 2, D-7*  
 VAXstation 2000 and MicroVAX 2000 devices,  
*Vol 2, D-6*

AUTOCONFIGURE command  
 in SYSGEN, *Vol 2, 6-5*

AUTOGEN  
 NEWPARAMS.DAT file, *Vol 1, 6-3*

AUTOGEN.COM command procedure, *Vol 1, 6-3*  
 calculation of parameter values, *Vol 1, 6-9*  
 collection of data, *Vol 1, 6-8, 6-9*  
 displaying page, swap, and dump file size  
 calculations, *Vol 1, 6-9*  
 end phase  
 default, *Vol 1, 6-13*  
 specifying when invoking, *Vol 1, 6-12*  
 examining calculations of, *Vol 1, 6-7*  
 execution mode  
 specifying when invoking, *Vol 1, 6-12*

feedback, *Vol 1, 6-3, 6-7*  
 collection of, *Vol 1, 6-7*  
 data file, *Vol 1, 6-7*  
 parameters affected by, *Vol 1, 6-7*

files used by, *Vol 1, 6-11*  
 generation of known image file list  
 (VMSIMAGES.DAT), *Vol 1, 6-9*

GENFILES phase, *Vol 1, 6-10*  
 GENPARAMS phase, *Vol 1, 6-9*  
 GETDATA phase, *Vol 1, 6-9*  
 HELP phase, *Vol 1, 6-11*  
 input files, *Vol 1, 6-11*  
 invoking, *Vol 1, 6-12*  
 operations performed by, *Vol 1, 6-8*  
 output files, *Vol 1, 6-11*  
 parameters to, *Vol 1, 6-12*  
 phases, *Vol 1, 6-8*  
 specifying when invoking, *Vol 1, 6-12*

REBOOT phase, *Vol 1, 6-10*  
 restriction with changing system files sizes, *Vol 1, 6-10*

SAVPARAMS phase, *Vol 1, 6-8*  
 SETPARAMS phase, *Vol 1, 6-10*  
 SHUTDOWN phase, *Vol 1, 6-10*

start phase  
 default, *Vol 1, 6-13*  
 specifying when invoking, *Vol 1, 6-12*  
 system parameters affected by feedback, *Vol 1, 6-7*

TESTFILES phase, *Vol 1, 6-9*  
 when to run, *Vol 1, 6-7*

AUTOGEN.PAR, *Vol 1, 6-8, 6-10*

Automatic login facility  
 See ALF

AUTO\_DLIGHT\_SAV system parameter, *Vol 2, C-7*  
 AWSMIN system parameter, *Vol 2, C-8*  
 AWSTIME system parameter, *Vol 2, C-8*

## B

---

Backlinks  
 directory  
 definition, *Vol 1, 3-4*

Backup  
 image  
 See Image backup  
 incremental  
 See Incremental backup  
 standalone  
 See Standalone BACKUP

BACKUP command, *Vol 1, 7-5 to 7-95*  
 See also Backup utility  
 element lists, *Vol 1, 7-6*  
 format, *Vol 1, 7-5*  
 input specifier, *Vol 1, 7-5*  
 output specifier, *Vol 1, 7-5*  
 specifying wildcard characters, *Vol 1, 7-8*  
 types of qualifiers, *Vol 1, 7-7*  
 using with save sets, *Vol 1, 7-5*

/BACKUP qualifier  
 in Backup utility, *Vol 1, 7-17*

Backup utility (BACKUP)  
 command line qualifiers, *Vol 1, 7-7*  
 command line syntax, *Vol 1, 7-5*  
 compare operation for, *Vol 1, 7-26*  
 /DELETE qualifier, *Vol 1, 7-33*  
 directing output from, *Vol 1, 7-11*  
 exiting from, *Vol 1, 7-11*  
 improving the speed of, *Vol 1, 7-41*  
 invoking, *Vol 1, 7-11*  
 /JOURNAL qualifier, *Vol 1, 7-59*  
 list operation, *Vol 1, 7-20, 7-43, 7-64*  
 qualifiers  
 command, *Vol 1, 7-7*  
 input file-selection, *Vol 1, 7-7*  
 input save-set, *Vol 1, 7-7*  
 output file, *Vol 1, 7-7*  
 output save-set, *Vol 1, 7-7*  
 position-dependent, *Vol 1, 7-8*  
 selective, *Vol 1, 7-17, 7-18, 7-39, 7-40, 7-68*  
 using to compare files, *Vol 1, 7-26*

BALSETCNT system parameter, *Vol 2, C-8*

Batch execution  
 on remote nodes, *Vol 2, 7-5*

/BEFORE qualifier  
 in Backup utility, *Vol 1, 7-18*

/BEGINNING qualifier, *Vol 2, 5-15*  
 in Monitor utility, *Vol 2, 1-12*

Binary output from Audit Analysis utility, *Vol 1*, 4–6

Blocks
 

- erasing, *Vol 1*, 3–5

/BLOCK\_SIZE qualifier
 

- in Backup utility, *Vol 1*, 7–19

Bootstrap operations, *Vol 2*, 6–3

BORROWLIM system parameter, *Vol 2*, C–8

Break-in attempts
 

- LAT protocol, *Vol 2*, C–24
- Telnet Protocol, *Vol 2*, C–24

BREAKPOINTS system parameter, *Vol 2*, C–9

Brief format listings, *Vol 1*, 4–7

/BRIEF qualifier
 

- in Backup utility, *Vol 1*, 7–20

Buffers
 

- per-connection
  - allocating, *Vol 2*, C–10

/BUFFER\_COUNT qualifier
 

- in Backup utility, *Vol 1*, 7–21

BUGCHECK
 

- dumps System Dump Priority processes early, *Vol 2*, 7–57
- in-memory hardcoded list of priority processes, *Vol 2*, 7–57

BUGCHECKFATAL system parameter, *Vol 2*, C–9

BUGREBOOT system parameter, *Vol 2*, C–9

Bus adapters
 

- generic names, *Vol 2*, 6–31
- listing, *Vol 2*, 6–31
- nexus numbers, *Vol 2*, 6–31
- specifying, *Vol 2*, 6–5, 6–10

Buses
 

- display on Alpha and I64 systems, *Vol 2*, 7–87

/BY\_NODE qualifier
 

- in Monitor utility, *Vol 2*, 1–12

/BY\_OWNER qualifier
 

- as Backup utility input file qualifier, *Vol 1*, 7–22
- as Backup utility output file qualifier, *Vol 1*, 7–23
- as Backup utility output save-set qualifier, *Vol 1*, 7–24

## C

Changing page, swap, and dump file sizes, *Vol 1*, 6–3

Changing system parameters
 

- recommended method, *Vol 1*, 6–3

CHANNELCNT system parameter, *Vol 2*, C–9

Checksumming
 

- enabling on VCs, *Vol 2*, 4–19

CHECK\_CLUSTER system parameter, *Vol 2*, C–9

CHECK\_FEEDBACK mode, *Vol 1*, 6–11

CIRCUITS class
 

- adding, *Vol 2*, 5–23
- fields belonging to, *Vol 2*, 5–3

CIRCUITS class (cont'd)
 

- removing, *Vol 2*, 5–52

Circuit timer, *Vol 1*, 13–33

Classes
 

- SHOW CLUSTER, *Vol 2*, 5–3

Classname qualifier
 

- in Monitor utility
  - /ALL, *Vol 2*, 1–56
  - /CPU, *Vol 2*, 1–42
  - /ITEM, *Vol 2*, 1–26, 1–62
  - /TOPBIO, *Vol 2*, 1–50
  - /TOPCPU, *Vol 2*, 1–50
  - /TOPDIO, *Vol 2*, 1–50
  - /TOPFAULT, *Vol 2*, 1–50
  - /TOPRBS, *Vol 2*, 1–50

Class scheduler database, *Vol 2*, 7–20

CLASS\_PROT system parameter, *Vol 2*, C–9

CLASS\_SCHEDULE ADD command
 

- in SYSMAN, *Vol 2*, 7–19

CLASS\_SCHEDULE DELETE command
 

- in SYSMAN, *Vol 2*, 7–24

CLASS\_SCHEDULE MODIFY command
 

- in SYSMAN, *Vol 2*, 7–25

CLASS\_SCHEDULE RESUME command
 

- in SYSMAN, *Vol 2*, 7–27

CLASS\_SCHEDULE SHOW command
 

- in SYSMAN, *Vol 2*, 7–28

CLASS\_SCHEDULE SUSPEND command
 

- in SYSMAN, *Vol 2*, 7–29

CLEAR DEVICE command
 

- in LANCP utility, *Vol 1*, 12–8

CLEAR DLL command
 

- in LANCP utility, *Vol 1*, 12–9

CLEAR NODE command
 

- in LANCP utility, *Vol 1*, 12–10

CLISYMTBL system parameter, *Vol 2*, C–9

CLOCK\_INTERVAL system parameter, *Vol 2*, C–9

CLOSE LOG command
 

- in LMCP, *Vol 1*, 14–4

CLU\$PARAMS.DAT file, *Vol 1*, 6–4

CLUE
 

- See Crash Log Utility Extractor

CLUSTER class
 

- adding, *Vol 2*, 5–25
- fields belonging to, *Vol 2*, 5–5, 5–33
- of data, *Vol 2*, 5–3
- removing, *Vol 2*, 5–53

CLUSTER class record
 

- in Monitor utility, *Vol 2*, A–13

Cluster performance statistics
 

- in Monitor utility, *Vol 2*, 1–19

CLUSTER\_AUTHORIZE.DAT files, *Vol 2*, 7–30, 7–34

CLUSTER\_CONFIG.COM command procedure, *Vol 2*, 7–30

CLUSTER\_CREDITS system parameter, *Vol 2*, C-10  
 Command mode for Audit Analysis utility  
   manipulating the display interactively, *Vol 1*, 4-12, 4-28  
 Command procedures  
   building, *Vol 2*, 5-64  
   control for SHOW CLUSTER output, *Vol 2*, 5-13  
   correcting system time, *Vol 2*, 7-32  
   executing, *Vol 1*, 12-6; *Vol 2*, 5-22  
     in SYSMAN, *Vol 2*, 7-10  
   executing at system startup, *Vol 2*, 7-136  
   executing DCL commands with DO, *Vol 2*, 7-53  
   parameters for, *Vol 1*, 12-6  
 Commands  
   recalling, *Vol 1*, 13-28  
 Command summary  
   of Authorize utility commands, *Vol 1*, 5-5  
 Command syntax, *Vol 1*, 7-5  
 /COMMENT qualifier  
   in Backup utility, *Vol 1*, 7-25  
   in Monitor utility, *Vol 2*, 1-12  
 Compaction of data  
   See Data compaction  
 Compare operations  
   for BACKUP, *Vol 1*, 7-26  
 /COMPARE qualifier  
   in Backup utility, *Vol 1*, 7-26  
 CONCEAL\_DEVICES system parameter, *Vol 2*, C-10  
 CONFIGURATION SET CLUSTER\_ AUTHORIZATION command  
   in SYSMAN, *Vol 2*, 7-30  
 CONFIGURATION SET TIME command  
   in SYSMAN, *Vol 2*, 7-32  
 CONFIGURATION SHOW CLUSTER\_ AUTHORIZATION command  
   in SYSMAN, *Vol 2*, 7-34  
 CONFIGURATION SHOW TIME command  
   in SYSMAN, *Vol 2*, 7-35  
 Configuration tables  
   rebuilding on Alpha and I64 systems, *Vol 2*, 7-81, 7-84  
 CONFIGURE command  
   in POLYCENTER Software Installation utility, *Vol 2*, 3-6  
   in SYSGEN, *Vol 2*, 6-8  
 Configuring a printer, *Vol 1*, 13-11  
 /CONFIRM qualifier  
   in Analyze/Disk\_Structure utility, *Vol 1*, 3-9  
   in Backup utility, *Vol 1*, 7-28  
 CONNECT/ADAPTER command  
   in SYSGEN, *Vol 2*, 6-10  
 CONNECT command  
   in Point-to-Point Protocol utility, *Vol 2*, 2-5  
   Connecting devices, *Vol 2*, 6-5, 6-10  
     on Alpha and I64 systems, *Vol 2*, 7-71  
 Connections  
   in LAT environment, *Vol 1*, 13-39, 13-66  
   specifying type with LATCP, *Vol 1*, 13-34  
 CONNECTIONS class  
   adding or restoring, *Vol 2*, 5-27  
   fields belonging to, *Vol 2*, 5-5  
   removing, *Vol 2*, 5-54  
 CONNECT NODE command  
   in LANCP utility, *Vol 1*, 12-11  
 CONTINUE command  
   in Audit Analysis utility, *Vol 1*, 4-29  
 /CONTINUOUS qualifier, *Vol 2*, 5-16  
 CONVERT command  
   in Monitor utility, *Vol 2*, 1-6  
 CONVERT DEVICE\_DATABASE command  
   in LANCP utility, *Vol 1*, 12-13  
 CONVERT LOG command  
   in LMCP, *Vol 1*, 14-5  
 CONVERT NODE\_DATABASE command  
   in LANCP utility, *Vol 1*, 12-14  
 /CONVERT qualifier  
   in Backup utility, *Vol 1*, 7-29  
 COPY command  
   in Authorize utility, *Vol 1*, 5-24  
   in POLYCENTER Software Installation utility, *Vol 2*, 3-10  
 Copying a software kit, *Vol 2*, 3-10  
 Counters, *Vol 1*, 13-3  
   displaying  
     for links, *Vol 1*, 13-50  
     for nodes, *Vol 1*, 13-55  
     for ports, *Vol 1*, 13-61  
     for services, *Vol 1*, 13-66  
   resetting, *Vol 1*, 13-70  
 COUNTERS class  
   adding, *Vol 2*, 5-29  
   fields belonging to, *Vol 2*, 5-7  
   removing, *Vol 2*, 5-56  
 CPU power  
   in LAT environment, *Vol 1*, 13-34  
 Crash Log Utility Extractor (CLUE)  
   commands, *Vol 1*, 8-4 to 8-12  
   DELETE command in, *Vol 1*, 8-5  
   DIRECTORY command in, *Vol 1*, 8-6  
   EXIT command in, *Vol 1*, 8-8  
   exiting from, *Vol 1*, 8-3  
   EXTRACT command in, *Vol 1*, 8-9  
   HELP command in, *Vol 1*, 8-10  
   invoking, *Vol 1*, 8-3  
   SHOW command in, *Vol 1*, 8-11  
 /CRC qualifier  
   as Backup utility input save-set qualifier, *Vol 1*, 7-30  
   as Backup utility output save-set qualifier, *Vol 1*, 7-31

CRDENABLE system parameter, *Vol 2*, C-11  
 CRD\_CONTROL system parameter, *Vol 2*, C-10  
 CREATE/PROXY command  
   in Authorize utility, *Vol 1*, 5-36  
 CREATE/RIGHTS command  
   in Authorize utility, *Vol 1*, 5-37  
 CREATE command  
   in Install utility, *Vol 1*, 11-7  
   in SYSGEN, *Vol 2*, 6-13  
 /CREATED qualifier  
   in Backup utility, *Vol 1*, 7-32  
 CREATE LINK command  
   in LATCP, *Vol 1*, 13-7  
     illegal SCSSYSTEMID system parameter,  
       *Vol 1*, 13-7  
 CREATE LOG command  
   in LMCP, *Vol 1*, 14-7  
 CREATE PORT command  
   in LATCP, *Vol 1*, 13-10  
 CREATE SERVICE command  
   in LATCP, *Vol 1*, 13-15  
 Creating  
   disk usage accounting files, *Vol 1*, 3-21  
 Creator ACEs  
   specifying access, *Vol 1*, 1-11  
   specifying attributes, *Vol 1*, 1-10  
 CREDITS class  
   adding, *Vol 2*, 5-31  
   fields belonging to, *Vol 2*, 5-7  
   removing, *Vol 2*, 5-57  
 CSR (Control and Status Register)  
   addresses, *Vol 2*, 6-8, 6-10  
     display on Alpha and I64 systems, *Vol 2*,  
       7-87  
   fixed space, *Vol 2*, D-1  
   floating space, *Vol 2*, D-1  
   vector addresses, *Vol 2*, 6-10  
 CTLIMGLIM system parameter, *Vol 2*, C-11  
 CTLPAGES system parameter, *Vol 2*, C-11  
 CWCREPRC\_ENABLE system parameter, *Vol 2*,  
   C-11

## D

Data collection  
   by AUTOGEN, *Vol 1*, 6-9  
 Data compaction  
   TA90E support, *Vol 1*, 7-67  
 Data field, *Vol 2*, 5-3  
 Data link  
   creating with LATCP, *Vol 1*, 13-7  
   modifying LAT characteristics of, *Vol 1*, 13-31  
   number supported by LAT software, *Vol 1*,  
     13-9  
 Dates  
   See also SYSMAN command CONFIGURATION  
     SHOW TIME  
   displaying in a cluster, *Vol 2*, 7-35

DBGTK\_SCRATCH system parameter, *Vol 2*,  
   C-11  
 DCL commands  
   in SYSMAN, *Vol 2*, 7-54  
 DCL\_CTLFLAGS system parameter, *Vol 2*, C-12  
 DEADLOCK\_WAIT system parameter, *Vol 2*,  
   C-12  
 DECdtm transaction manager, *Vol 2*, A-41  
 DECnet  
   DECNET class record  
     in Monitor utility, *Vol 2*, A-15  
   LAT software  
     effect of SCSSYSTEMID system parameter,  
       *Vol 1*, 13-7  
 DECservers  
   changing device characteristics, *Vol 1*, 13-10,  
     13-43  
 Dedicated port, *Vol 1*, 13-11, 13-43  
   displaying, *Vol 1*, 13-61  
 DEFAULT command  
   in Authorize utility, *Vol 1*, 5-38  
 Default parameter file, *Vol 1*, 6-10  
 Default protection  
   for NET\$PROXY.DAT, *Vol 1*, 5-3  
   for NETPROXY.DAT, *Vol 1*, 5-3  
   for RIGHTSLIST.DAT, *Vol 1*, 5-3  
   for SYSUAF.DAT, *Vol 1*, 5-3  
 Default Protection ACEs, *Vol 1*, 1-10, 1-12  
 Default user authorization records  
   modifying, *Vol 1*, 5-48  
 DEFINE/KEY command, *Vol 2*, B-3  
   in LATCP, *Vol 1*, 13-19  
   in Show Cluster utility, *Vol 2*, 5-40  
   in SYSMAN, *Vol 2*, 7-36  
 DEFINE DEVICE command  
   in LANCP utility, *Vol 1*, 12-15  
 DEFINE NODE command  
   in LANCP utility, *Vol 1*, 12-16  
 Defining keys, *Vol 2*, 7-36  
   to execute SYSMAN commands, *Vol 2*, 7-3  
 DEFMBXBUFQUO system parameter, *Vol 2*,  
   C-12  
 DEFMBXMXMSG system parameter, *Vol 2*, C-13  
 DEFPRI system parameter, *Vol 2*, C-13  
 DEFQUEPRI system parameter, *Vol 2*, C-13  
 DEINSTALL command  
   in SYSGEN, *Vol 2*, 6-14  
 DELETE command  
   in CLUE, *Vol 1*, 8-5  
   in Install utility, *Vol 1*, 11-12  
 DELETE DEVICE command  
   in UCM, *Vol 2*, 8-14  
 DELETE LINK command  
   in LATCP, *Vol 1*, 13-22  
 DELETE PORT command  
   in LATCP, *Vol 1*, 13-23

- /DELETE qualifier
  - in Backup utility, *Vol 1*, 7–33
- DELETE QUEUE\_ENTRY command
  - in LATCP, *Vol 1*, 13–24
- DELETE RECOVERY\_DATA command
  - in POLYCENTER Software Installation utility, *Vol 2*, 3–15
- DELETE SERVICE command
  - in LATCP, *Vol 1*, 13–25
- DELPRC\_EXIT system parameter, *Vol 2*, C–13
- /DENSITY qualifier
  - in Backup utility, *Vol 1*, 7–34
- DESELECT command
  - in Show Cluster utility, *Vol 2*, 5–43
- Device addresses
  - See LAN addresses
- Device drivers
  - addresses, *Vol 2*, 6–36
  - configuring, *Vol 2*, 6–5, 6–35
  - connecting, *Vol 2*, 6–10
  - displaying on Alpha and I64 systems, *Vol 2*, 7–89
  - loading, *Vol 2*, 6–5, 6–10, 6–18
    - on Alpha and I64 systems, *Vol 2*, 7–80
  - replacing with new version of, *Vol 2*, 6–19
  - showing configuration of, *Vol 2*, 6–33
  - showing information, *Vol 2*, 6–35, 6–36
- Device IDs
  - Fibre Channel
    - tape WWIDs, *Vol 2*, 7–75, 7–76, 7–82
- Device names
  - specifying, *Vol 2*, 6–8, 6–10
- Devices
  - configuring
    - VAXstation 2000 and MicroVAX 2000
      - devices, *Vol 2*, D–6
  - connecting on Alpha and I64 systems, *Vol 2*, 7–69, 7–71
  - displaying
    - default, *Vol 2*, 7–126
  - excluding, *Vol 2*, 7–85
  - HP-supplied, *Vol 2*, D–1
  - logical name translation, *Vol 2*, 7–40
  - SCSI and FC
    - checking for changes in paths, *Vol 2*, 7–84
    - setting defaults for file specifications, *Vol 2*, 7–122
    - specifying quotas for, *Vol 2*, 7–38
- Device seed, *Vol 1*, 13–34
- Device table, *Vol 2*, D–1
- Device types
  - codes, *Vol 2*, 6–6
  - descriptions, *Vol 2*, 6–6
- DEVICE\_NAMING system parameter, *Vol 2*, C–13
- DIAL\_OUT command
  - in Point-to-Point Protocol utility, *Vol 2*, 2–6
- Directing ANALYZE/DISK\_STRUCTURE output, *Vol 1*, 3–7
- Directing SYSGEN output, *Vol 2*, 6–4, 6–8, 6–21
- Directories
  - access to files created in, *Vol 1*, 1–10
    - and Default Protection ACEs, *Vol 1*, 1–12
    - assigning a Creator ACE to, *Vol 1*, 1–10
    - creating for a new account, *Vol 1*, 5–19
    - determining access to files created in, *Vol 1*, 1–12
    - ownership
      - changing access to files, *Vol 1*, 1–10
- Directory backlinks
  - definition, *Vol 1*, 3–4
- DIRECTORY command
  - in CLUE, *Vol 1*, 8–6
- Directory default
  - displaying, *Vol 2*, 7–126
  - modifying, *Vol 2*, 7–122
- Directory trees
  - propagating UIC-based protection, *Vol 1*, 1–12
  - restoring, *Vol 1*, 7–9
- DISABLE command
  - in SYSGEN, *Vol 2*, 6–15
- DISABLE\_UPCALLS system parameter, *Vol 2*, C–13
- DISCONNECT command
  - in Point-to-Point Protocol utility, *Vol 2*, 2–8
- DISK class record
  - in Monitor utility, *Vol 2*, A–16
- DISKQUOTA ADD command
  - in SYSMAN, *Vol 2*, 7–38
- DISKQUOTA CREATE command
  - in SYSMAN, *Vol 2*, 7–40
- DISKQUOTA DELETE command
  - See DISKQUOTA REMOVE
  - in SYSMAN, *Vol 2*, 7–42
- DISKQUOTA DISABLE command
  - in SYSMAN, *Vol 2*, 7–43
- DISKQUOTA ENABLE command
  - in SYSMAN, *Vol 2*, 7–45
- DISKQUOTA MODIFY command
  - in SYSMAN, *Vol 2*, 7–47
- DISKQUOTA REBUILD command
  - in SYSMAN, *Vol 2*, 7–49
- DISKQUOTA REMOVE command
  - in SYSMAN, *Vol 2*, 7–50
- Disk quotas
  - creating, *Vol 2*, 7–40
  - disabling, *Vol 2*, 7–43
  - displaying, *Vol 2*, 7–52
  - file
    - adding entry, *Vol 2*, 7–38
    - initializing usage count, *Vol 2*, 7–38
    - modifying an entry, *Vol 2*, 7–47
    - rebuilding, *Vol 2*, 7–49

- Disk quotas (cont'd)
  - removing an entry, *Vol 2, 7-42, 7-50*
- DISKQUOTA SHOW command
  - in SYSMAN, *Vol 2, 7-52*
- Disk statistics
  - in Monitor utility, *Vol 2, 1-26*
- Disk usage accounting file, *Vol 1, E-1*
  - creating
    - example, *Vol 1, 3-21*
- Disk volumes
  - repairing errors, *Vol 1, 3-16*
  - verification, *Vol 1, 3-3*
- DISK\_QUORUM system parameter, *Vol 2, C-14*
- DISPLAY command
  - in Audit Analysis utility, *Vol 1, 4-30*
- /DISPLAY qualifier
  - in Monitor utility, *Vol 2, 1-13*
- Displays
  - refreshing screen with LATCP, *Vol 1, 13-29*
- Displays from SHOW CLUSTER
  - display formatting, *Vol 2, 5-13*
- Distributed lock management statistics
  - in Monitor utility, *Vol 2, 1-29*
- DLCKEXTRASTK system parameter, *Vol 2, C-14*
- DLOCK class record
  - in Monitor utility, *Vol 2, A-18*
- DNVOSII system parameter, *Vol 2, C-14*
- DO command
  - in SYSMAN, *Vol 2, 7-53*
- DORMANTWAIT system parameter, *Vol 2, C-14*
- DOSD (dump off system disk)
  - DUMPSTYLE system parameter, *Vol 2, C-15*
- DR\_UNIT\_BASE system parameter, *Vol 2, C-14*
- Dual-architecture OpenVMS Cluster systems
  - installing images
    - example, *Vol 2, 7-55*
- DUMPBUG system parameter, *Vol 2, C-14*
- DUMP command
  - in LMCP, *Vol 1, 14-9*
- Dump files
  - changing size, *Vol 1, 6-3*
  - displaying the size calculated by AUTOGEN, *Vol 1, 6-9*
  - generating a new file sized by AUTOGEN, *Vol 1, 6-10*
  - setting priority, *Vol 2, 7-57*
- DUMPSTYLE system parameter, *Vol 2, C-15*
- DUMP\_PRIORITY ADD command
  - in SYSMAN, *Vol 2, 7-57*
- DUMP\_PRIORITY LIST command
  - in SYSMAN, *Vol 2, 7-60*
- DUMP\_PRIORITY LOAD command
  - in SYSMAN, *Vol 2, 7-61*
- DUMP\_PRIORITY MODIFY command
  - in SYSMAN, *Vol 2, 7-62*
- DUMP\_PRIORITY REMOVE command
  - in SYSMAN, *Vol 2, 7-64*

- DUMP\_PRIORITY SHOW command
  - in SYSMAN, *Vol 2, 7-65*
- DUMP\_PRIORITY UNLOAD command
  - in SYSMAN, *Vol 2, 7-66*
- Dynamic attributes, *Vol 1, 5-20, 5-51, 5-73*
- Dynamic lock remastering statistics
  - in Monitor utility, *Vol 2, 1-56*
- Dynamic parameters
  - modifying, *Vol 2, 6-45, 7-105*
- Dynamic service rating, *Vol 1, 13-16, 13-48*
  - displaying, *Vol 1, 13-59, 13-67*

## E

---

- EDIT/ACL command, *Vol 1, 1-3*
  - See also ACL editor
- Editing ACLs
  - See ACL editor
- Editing sessions
  - keypad for ACL editing, *Vol 1, A-1*
- EDIT keypad function, *Vol 2, 5-40, B-3*
- EFI Utilities for OpenVMS, *Vol 1, 9-3*
  - VMS\_BCFG utility, *Vol 1, 9-4*
  - VMS\_SET utility, *Vol 1, 9-6*
  - VMS\_SHOW utility, *Vol 1, 9-7*
- ENABLE command
  - in SYSGEN, *Vol 2, 6-16*
- /ENDING qualifier, *Vol 2, 5-17*
  - in Monitor utility, *Vol 2, 1-13*
- END phase, *Vol 2, 7-143*
- End phase of AUTOGEN, *Vol 1, 6-8*
  - default, *Vol 1, 6-13*
- Environment
  - defining
    - in SYSMAN, *Vol 2, 7-120*
- Erasing blocks, *Vol 1, 3-5, 3-10*
- ERLBUFFERPAGES system parameter, *Vol 2, C-16*
- ERLBUFFERPAG\_S2 system parameter, *Vol 2, C-16*
- ERRORLOGBUFFERS system parameter, *Vol 2, C-16*
- ERRORLOGBUFF\_S2 system parameter, *Vol 2, C-16*
- Error Log Viewer (ELV)
  - brief report example, *Vol 1, 10-31*
- CONVERT command, *Vol 1, 10-6*
  - qualifiers, *Vol 1, 10-6 to 10-9*
  - default report example, *Vol 1, 10-31*
- DUMP command, *Vol 1, 10-11*
  - qualifiers, *Vol 1, 10-11 to 10-15*
- event categories, *Vol 1, 10-5*
  - invalid, *Vol 1, 10-5*
  - rejected, *Vol 1, 10-5*
  - selected, *Vol 1, 10-5*
  - valid, *Vol 1, 10-5*
- event classes, *Vol 1, 10-19*
- event header information, *Vol 1, 10-19*

## Error Log Viewer (ELV) (cont'd)

- EXIT command, *Vol 1*, 10–16
- full report example, *Vol 1*, 10–32
- HELP command, *Vol 1*, 10–17
- interactive shell mode, *Vol 1*, 10–4
- invoking, *Vol 1*, 10–4
- one-line-per-event report example, *Vol 1*, 10–30
- report formats
  - standard, *Vol 1*, 10–19
  - summary, *Vol 1*, 10–19
- standard report detail levels, *Vol 1*, 10–19
  - brief, *Vol 1*, 10–19
  - default, *Vol 1*, 10–19
  - full, *Vol 1*, 10–19
  - one-line-per-event, *Vol 1*, 10–19
- supported event types, *Vol 1*, 10–3
- terse format, *Vol 1*, 10–32
- terse report example, *Vol 1*, 10–32
- TRANSLATE command, *Vol 1*, 10–18
  - qualifiers, *Vol 1*, 10–20 to 10–24
- WRITE command, *Vol 1*, 10–25
  - qualifiers, *Vol 1*, 10–25 to 10–28

## Errors

- repair, *Vol 1*, 3–4, 3–9, 3–10, 3–15, 3–16
- reporting, *Vol 1*, 3–4

## ERRORS class

- adding, *Vol 2*, 5–32
- fields belonging to, *Vol 2*, 5–7
- removing, *Vol 2*, 5–58

## Ethernet address

- See LAN addresses

## Ethernet controllers

- See LAN controllers

## Ethernet link

- See Data link

## Event classes

- extracting audit records by, *Vol 1*, 4–8

## Event records

- See Security audit reports

## Event types, *Vol 1*, 4–8

## /EXACT\_ORDER qualifier

- in Backup utility, *Vol 1*, 7–36

## Examples

- annotated, *Vol 1*, D–4 to D–8
- comparing files, *Vol 1*, 7–27
- creating a journal file, *Vol 1*, 7–59
- image save operation, *Vol 1*, 7–52
- incremental save operation, *Vol 1*, 7–54
- initializing an output disk, *Vol 1*, 7–55
- list operation, *Vol 1*, 7–20
- of ANALYZE/DISK\_STRUCTURE/SHADOW command, *Vol 1*, 3–18
- protecting an output save-set, *Vol 1*, 7–74
- repairing errors on a disk volume, *Vol 1*, 3–16
- selective backup, *Vol 1*, 7–17, 7–18
- specifying a save-set label, *Vol 1*, 7–63
- specifying block size, *Vol 1*, 7–19

## Examples (cont'd)

- specifying density for magnetic tape, *Vol 1*, 7–35
- specifying redundancy group blocks, *Vol 1*, 7–45
- using wildcard characters, *Vol 1*, 7–9
- /EXCLUDE qualifier
  - in Backup utility, *Vol 1*, 7–39
- Excluding devices, *Vol 2*, 7–85
- Exclusion list
  - displaying, *Vol 2*, 7–91
- EXECUTE command
  - in Monitor utility, *Vol 2*, 1–7
- Execute procedure (@) command
  - in LANCP utility, *Vol 1*, 12–6
  - in Show Cluster utility, *Vol 2*, 5–22
  - to set up SHOW CLUSTER format, *Vol 2*, 5–22
- Execution modes
  - CHECK\_FEEDBACK, *Vol 1*, 6–11
  - FEEDBACK, *Vol 1*, 6–11
  - INITIAL, *Vol 1*, 6–11
  - NOFEEDBACK, *Vol 1*, 6–11
- EXIT command, *Vol 2*, 7–67
  - in Audit Analysis utility, *Vol 1*, 4–32
  - in Authorize utility, *Vol 1*, 5–50
  - in CLUE, *Vol 1*, 8–8
  - in Install utility, *Vol 1*, 11–13
  - in LANCP utility, *Vol 1*, 12–18
  - in LATCP, *Vol 1*, 13–26
  - in Monitor utility, *Vol 2*, 1–8
  - in Point-to-Point Protocol utility, *Vol 2*, 2–9
  - in Show Cluster utility, *Vol 2*, 5–44
  - in SYSGEN, *Vol 2*, 6–4
  - in SYSMAN, *Vol 2*, 7–67
  - in UCM, *Vol 2*, 8–15
- EXPECTED\_VOTES system parameter, *Vol 2*, C–16
- /EXPIRED qualifier
  - in Backup utility, *Vol 1*, 7–40
- Extensible Firmware Interface
  - See EFI utilities for OpenVMS
- EXTRACPU system parameter, *Vol 2*, C–16
- EXTRACT command
  - in CLUE, *Vol 1*, 8–9
- EXTRACT FILE command
  - in POLYCENTER Software Installation utility, *Vol 2*, 3–17
- EXTRACT PDF command
  - in POLYCENTER Software Installation utility, *Vol 2*, 3–21
- EXTRACT PTF command
  - in POLYCENTER Software Installation utility, *Vol 2*, 3–25
- EXTRACT RELEASE\_NOTES command
  - in POLYCENTER Software Installation utility, *Vol 2*, 3–29



## F

---

### /FAST qualifier

in Backup utility, *Vol 1*, 7-41

FAST\_PATH system parameter, *Vol 2*, C-16

FAST\_PATH\_PORTS system parameter, *Vol 2*, C-17

### FCP class record

in Monitor utility, *Vol 2*, A-19

### Feedback

See AUTOGEN, feedback

FEEDBACK mode, *Vol 1*, 6-3, 6-11

### Field of data

definition, *Vol 2*, 5-33

definition of each, *Vol 2*, 5-34

removing, *Vol 2*, 5-59

### Fields

SHOW CLUSTER, *Vol 2*, 5-3

### File backward link records

in accounting files, *Vol 1*, 2-37

### File forward link records

in accounting files, *Vol 1*, 2-37

### File primitive statistics

in Monitor utility, *Vol 2*, 1-32

### Files

AGEN\$FEEDBACK.DAT, *Vol 1*, 6-7

AGEN\$FEEDBACK.EXE, *Vol 1*, 6-7

AGEN\$PARAMS.REPORT, *Vol 1*, 6-9

ALPHAVMSSYS.OLD, *Vol 1*, 6-10

ALPHAVMSSYS.PAR, *Vol 1*, 6-10

AUTOGEN.PAR, *Vol 1*, 6-8, 6-10; *Vol 2*, 6-43, 6-45

containing AUTOGEN feedback, *Vol 1*, 6-7

containing list of known images, *Vol 1*, 6-9

controlling access to, *Vol 1*, 1-12, 1-14

#### creating

dependency on directory ownership, *Vol 1*, 1-10

in a directory with a Default Protection ACE, *Vol 1*, 1-12

default parameter, *Vol 1*, 6-10

device driver, *Vol 2*, 6-19

format required by ACL editor, *Vol 1*, 1-4

for paging, *Vol 1*, 6-10

for swapping, *Vol 1*, 6-10

for system crash dumps, *Vol 1*, 6-10

I/O driver, *Vol 2*, 6-18

IA64VMSSYS.OLD, *Vol 1*, 6-10

IA64VMSSYS.PAR, *Vol 1*, 6-10

input of device list, *Vol 2*, 6-8

output cluster report, *Vol 2*, 5-19, 5-74

parameter, *Vol 1*, 6-8, 6-10; *Vol 2*, 6-45

PARAMS.DAT, *Vol 1*, 6-9

#### restoring

to original directories, *Vol 1*, 7-10

SETPARAMS.DAT, *Vol 1*, 6-9, 6-10

SHOW\_CLUSTER.LIS, *Vol 2*, 5-19, 5-74

### Files (cont'd)

startup, *Vol 2*, 7-136

disabling, *Vol 2*, 7-138

displaying, *Vol 2*, 7-147

startup command procedure, *Vol 2*, 6-22, 6-38, 6-45

SYSGEN session output, *Vol 2*, 6-4, 6-21

SYSGTTSTR.MSG, *Vol 2*, 6-42

SYSMAN.LIS, *Vol 2*, 7-16, 7-34, 7-35, 7-52, 7-54, 7-55, 7-99, 7-147

system parameter, *Vol 2*, 6-43, 6-45

#### UIC-based protection

propagating in a directory tree, *Vol 1*, 1-12

used by AUTOGEN command procedure, *Vol 1*, 6-11

VAXVMSSYS.OLD, *Vol 1*, 6-10

VAXVMSSYS.PAR, *Vol 1*, 6-10; *Vol 2*, 6-45

VMSIMAGES.DAT, *Vol 1*, 6-9

writing alarm messages to audit logs, *Vol 1*, 1-8

### File system cache (ACP) system parameters

in Monitor utility, *Vol 2*, 1-37

### File system cache statistics

monitoring, *Vol 2*, 1-34

### /FILES\_SELECTED qualifier

in Backup utility, *Vol 1*, 7-42

### File types

specifying for ACL editor, *Vol 1*, 1-4

### FILE\_SYSTEM\_CACHE class record

in Monitor utility, *Vol 2*, A-21

### FIND command

in POLYCENTER Software Installation utility, *Vol 2*, 3-33

### Flags

login, *Vol 1*, 5-10, 5-27, 5-41, 5-64

Floating address, *Vol 2*, D-1

### /FLUSH\_INTERVAL qualifier

in Monitor utility, *Vol 2*, 1-13

### Format for SHOW CLUSTER

specifying processor types, *Vol 2*, 5-38

### Formats

of software kits

changing, *Vol 2*, 3-10

FREEGOAL system parameter, *Vol 2*, C-17

FREELIM system parameter, *Vol 2*, C-18

### /FULL qualifier

in Backup utility, *Vol 1*, 7-43

Function keys, *Vol 1*, 13-19

## G

---

GALAXY system parameter, *Vol 2*, C-18

GBLPAGES system parameter, *Vol 2*, C-18

GBLPAGFIL system parameter, *Vol 2*, C-18

GBLSECTIONS system parameter, *Vol 2*, C-19

### Generic names

showing, *Vol 2*, 6-31, 6-32

GENFILES phase of AUTOGEN, *Vol 1*, 6–10  
 GENPARAMS phase of AUTOGEN, *Vol 1*, 6–9  
 GETDATA phase of AUTOGEN, *Vol 1*, 6–9  
 GH\_EXEC\_CODE system parameter, *Vol 2*, C–19  
 GH\_EXEC\_DATA system parameter, *Vol 2*, C–19  
 GH\_RES\_CODE system parameter, *Vol 2*, C–20  
 GH\_RES\_DATA system parameter, *Vol 2*, C–20  
 GH\_RSRVPGCNT system parameter, *Vol 2*, C–20  
 GLX\_INST\_TMO system parameter, *Vol 2*, C–20  
 GLX\_SHM\_REG system parameter, *Vol 2*, C–20  
 GRANT/IDENTIFIER command  
   in Authorize utility, *Vol 1*, 5–51  
 Group number  
   cluster  
     displaying, *Vol 2*, 7–34  
   local area cluster, *Vol 2*, 7–30  
 Groups  
   access to service nodes, *Vol 1*, 13–40, 13–59  
   access to services in LAT environment, *Vol 1*,  
     13–35  
 /GROUP\_SIZE qualifier  
   in Backup utility, *Vol 1*, 7–45  
 GROWLIM system parameter, *Vol 2*, C–20

## H

---

HANGUP characteristic  
   application LAT devices, *Vol 1*, 13–10  
 /HEADER\_ONLY qualifier  
   in Backup utility, *Vol 1*, 7–46  
 Help  
   online for Analyze Audit, *Vol 1*, 4–33  
 HELP command  
   in Audit Analysis utility, *Vol 1*, 4–33  
   in Authorize utility, *Vol 1*, 5–53  
   in CLUE, *Vol 1*, 8–10  
   in Install utility, *Vol 1*, 11–14  
   in LANCP utility, *Vol 1*, 12–19  
   in LATCP, *Vol 1*, 13–27  
   in LMCP, *Vol 1*, 14–15  
   in Monitor utility, *Vol 2*, 1–9  
   in Point-to-Point Protocol utility, *Vol 2*, 2–10  
   in Show Cluster utility, *Vol 2*, 5–45  
   in SYSMAN, *Vol 2*, 7–68  
   in UCM, *Vol 2*, 8–16  
 HELP phase of AUTOGEN, *Vol 1*, 6–11  
 History of installed software, *Vol 2*, 3–64  
 Holder Hidden attribute, *Vol 1*, 5–20, 5–51, 5–73  
 Home blocks, *Vol 1*, 3–5  
 /HOMEBLOCKS qualifier  
   in Analyze/Disk\_Structure utility, *Vol 1*, 3–10

I/O addresses, *Vol 2*, 6–40  
 I/O class record  
   in Monitor utility, *Vol 2*, A–22  
 I/O databases, *Vol 2*, 6–35  
   displaying on Alpha and I64 systems, *Vol 2*,  
     7–89  
 I/O drivers  
   loading, *Vol 2*, 6–18  
   replacing with new version of, *Vol 2*, 6–19  
 I/O statistics  
   in Monitor utility, *Vol 2*, 1–38  
 ICBM (IOGEN Configuration Building Module),  
   *Vol 2*, 7–85, 7–86, 7–92  
 Identifier ACEs, *Vol 1*, 1–14  
   types of identifiers, *Vol 1*, 1–14  
 Identifiers  
   default, *Vol 1*, 5–7  
   environmental, *Vol 1*, 1–14  
   general, *Vol 1*, 1–14, 1–16  
   granting, *Vol 1*, 1–14, 5–51  
   granting to a process, *Vol 1*, 1–16  
   renaming, *Vol 1*, 5–84  
   resource  
     and directory ownership, *Vol 1*, 1–10  
   revoking, *Vol 1*, 5–85  
   specifying characteristics of, *Vol 1*, 1–17  
   specifying multiple pairs of, *Vol 1*, 1–17  
   types, *Vol 1*, 1–14  
 IEEE\_ADDRESSH system parameter, *Vol 2*, C–21  
 IEEE\_ADDRESS system parameter, *Vol 2*, C–20  
 /IGNORE qualifier  
   in Backup utility, *Vol 1*, 7–48  
 IJOB LIM system parameter, *Vol 2*, C–21  
 Image  
   See System loadable image  
   accounting, *Vol 1*, 11–7, 11–23  
   applying an ACE to, *Vol 1*, 1–16  
   authorized privileges, *Vol 1*, 11–8, 11–24  
   header-resident, *Vol 1*, 11–8, 11–25  
   privileged, *Vol 1*, 11–8, 11–25  
   protecting installed, *Vol 1*, 11–9, 11–25  
 Image backup  
   definition, *Vol 1*, 7–4  
 Image mode, *Vol 1*, 7–51  
 /IMAGE qualifier  
   in Backup utility, *Vol 1*, 7–51  
 IMGIOCNT system parameter, *Vol 2*, C–21  
 IMGREG\_PAGES system parameter, *Vol 2*, C–21  
 Incoming LAT connections, *Vol 1*, 13–34  
 Incremental backup  
   definition, *Vol 1*, 7–4  
 Incremental mode  
   restoring files, *Vol 1*, 7–53

/INCREMENTAL qualifier  
     in Backup utility, *Vol 1*, 7–53  
 Initialization  
     erasing home blocks, *Vol 1*, 3–10  
 Initialization files  
     creating, *Vol 2*, 5–64  
     default specification, *Vol 2*, 7–4  
     establishing SHOW CLUSTER reports, *Vol 2*,  
         5–13  
     use with SYSMAN, *Vol 2*, 7–4  
 Initialization procedures, *Vol 1*, D–1  
 INITIALIZE command  
     in Monitor utility, *Vol 2*, 1–10  
     in Show Cluster utility, *Vol 2*, 5–46  
 /INITIALIZE qualifier  
     in Backup utility, *Vol 1*, 7–55  
 INITIAL mode, *Vol 1*, 6–11  
 Input files  
     security audit log files, *Vol 1*, 4–3  
 /INPUT qualifier  
     in Monitor utility, *Vol 2*, 1–13  
 Input stream  
     switching control to other processes, *Vol 2*,  
         7–17  
 /INPUT\_FILES qualifier  
     in Backup utility, *Vol 1*, 7–57  
 INSTALL command  
     in POLYCENTER Software Installation utility,  
         *Vol 2*, 3–36  
     in SYSGEN, *Vol 2*, 6–17  
 Installing software  
     getting product information, *Vol 2*, 3–64  
     POLYCENTER Software Installation utility,  
         *Vol 2*, 3–36  
     removing software, *Vol 2*, 3–61  
 Install utility (INSTALL)  
     commands  
         ADD, *Vol 1*, 11–6  
         CREATE, *Vol 1*, 11–7  
         DELETE, *Vol 1*, 11–12  
         EXIT, *Vol 1*, 11–13  
         HELP, *Vol 1*, 11–14  
         LIST, *Vol 1*, 11–15  
         PURGE, *Vol 1*, 11–21  
         REMOVE, *Vol 1*, 11–22  
         REPLACE, *Vol 1*, 11–23  
         TRANSLATE, *Vol 1*, 11–3  
     exiting, *Vol 1*, 11–3  
     invoking, *Vol 1*, 11–3  
     /TRANSLATE qualifier, *Vol 1*, 11–3  
 Interactive commands  
     See Command mode for Audit Analysis utility  
 /INTERCHANGE qualifier  
     in Backup utility, *Vol 1*, 7–58  
 /INTERVAL qualifier, *Vol 2*, 5–18  
     in Monitor utility, *Vol 2*, 1–14  
     relationship of qualifier and event, *Vol 2*,  
         1–14  
 INTSTKPAGES system parameter, *Vol 2*, C–21  
 Invoking AUTOGEN, *Vol 1*, 6–12  
 Invoking SYSGEN, *Vol 2*, 6–4  
 Invoking SYSMAN, *Vol 2*, 7–4  
 IO AUTOCONFIGURE command  
     in SYSMAN, *Vol 2*, 7–69  
 IO CONNECT command  
     in SYSMAN, *Vol 2*, 7–71  
 IO CREATE\_WWID command  
     in SYSMAN, *Vol 2*, 7–75  
 IO FIND\_WWID command  
     in SYSMAN, *Vol 2*, 7–76  
 IOGEN Configuration Building Module  
     See ICBM  
 IO LIST\_WWID command  
     in SYSMAN, *Vol 2*, 7–79  
 IO LOAD command  
     in SYSMAN, *Vol 2*, 7–80  
 IO REBUILD command  
     in SYSMAN, *Vol 2*, 7–81  
 IO REPLACE\_WWID command  
     in SYSMAN, *Vol 2*, 7–82  
 IO SCSI\_PATH\_VERIFY command  
     in SYSMAN, *Vol 2*, 7–84  
 IO SET EXCLUDE command  
     in SYSMAN, *Vol 2*, 7–85  
 IO SET PREFIX command  
     in SYSMAN, *Vol 2*, 7–86  
 IO SHOW BUS command  
     in SYSMAN, *Vol 2*, 7–87  
 IO SHOW DEVICE command  
     in SYSMAN, *Vol 2*, 7–89  
 IO SHOW EXCLUDE command  
     in SYSMAN, *Vol 2*, 7–91  
 IO SHOW PREFIX command  
     in SYSMAN, *Vol 2*, 7–92  
 IOTA system parameter, *Vol 2*, C–21  
 IO\_PREFER\_CPUS system parameter, *Vol 2*,  
     C–21  
 IRPCOUNT system parameter, *Vol 2*, C–22  
 IRPCOUNTV system parameter, *Vol 2*, C–22  
  
**J**  


---

 JBOBLIM system parameter, *Vol 2*, C–22  
 JOBCTLD system parameter, *Vol 2*, C–22  
 /JOURNAL qualifier  
     in Backup utility, *Vol 1*, 7–59  
  
**K**  


---

 Keypad definitions, *Vol 2*, 5–13, 5–71  
     SHOW CLUSTER, *Vol 2*, B–1  
 Keypad editing  
     editing commands, *Vol 1*, A–1  
     for ACL editor, *Vol 1*, A–1  
     LK201-specific editing commands, *Vol 1*, A–5

## Keys

- defining, *Vol 2, 7–36*
- defining in LATCP, *Vol 1, 13–19*
- defining in SYSMAN, *Vol 2, 7–3*
- displaying definition, *Vol 2, 7–125*
- KFILSTCNT system parameter, *Vol 2, C–22*
- Known file entries, *Vol 1, 11–3*
- Known file lists, *Vol 1, 11–3*
- Known images, *Vol 1, 11–3*
  - adding, *Vol 1, 11–6*
  - attributes of, *Vol 1, 11–17*
  - creating, *Vol 1, 11–7*
  - deleting, *Vol 1, 11–12*
  - evaluating merits of installing, *Vol 1, 11–15*
  - removing, *Vol 1, 11–22*
  - replacing, *Vol 1, 11–23*
  - substituting, *Vol 1, 11–3*
  - updating, *Vol 1, 11–3, 11–23*
- KSTACKPAGES system parameter, *Vol 2, C–22*

## L

---

### /LABEL qualifier

- in Backup utility, *Vol 1, 7–62*
- LANACP LAN Server process, *Vol 1, 12–3*
  - SET ACP command, *Vol 1, 12–26*
- LAN addresses, *Vol 1, 13–7*
  - displaying, *Vol 1, 13–59*
- LAN controllers
  - address, *Vol 1, 13–59*
  - device name, *Vol 1, 13–8*
  - physical address, *Vol 1, 13–7*
- LAN Control Program (LANCP) utility
  - clearing counters, *Vol 1, 12–9*
  - commands
    - CLEAR DEVICE, *Vol 1, 12–8*
    - CLEAR DLL, *Vol 1, 12–9*
    - CLEAR NODE, *Vol 1, 12–10*
    - CONNECT NODE, *Vol 1, 12–11*
    - CONVERT DEVICE\_DATABASE, *Vol 1, 12–13*
    - CONVERT NODE\_DATABASE, *Vol 1, 12–14*
    - DEFINE DEVICE, *Vol 1, 12–15*
    - DEFINE NODE, *Vol 1, 12–16*
    - @ (Execute Procedure), *Vol 1, 12–6*
    - EXIT, *Vol 1, 12–18*
    - HELP, *Vol 1, 12–19*
    - LIST DEVICE, *Vol 1, 12–20*
    - LIST NODE, *Vol 1, 12–22*
    - PURGE DEVICE, *Vol 1, 12–24*
    - PURGE NODE, *Vol 1, 12–25*
    - SET ACP, *Vol 1, 12–26*
    - SET DEVICE, *Vol 1, 12–27*
    - SET NODE, *Vol 1, 12–37*
    - SHOW ACP, *Vol 1, 12–41*
    - SHOW CONFIGURATION, *Vol 1, 12–42*
    - SHOW DEVICE, *Vol 1, 12–43*

### LAN Control Program (LANCP) utility

- commands (cont'd)
  - SHOW DLL, *Vol 1, 12–47*
  - SHOW LOG, *Vol 1, 12–48*
  - SHOW NODE, *Vol 1, 12–50*
  - SPAWN, *Vol 1, 12–52*
  - TRIGGER NODE, *Vol 1, 12–53*
  - UPDATE DEVICE, *Vol 1, 12–54*
- deleting device information, *Vol 1, 12–8, 12–24*
- deleting node information, *Vol 1, 12–10, 12–25*
- displaying node information, *Vol 1, 12–22, 12–50*
- displaying OPCOM messages, *Vol 1, 12–26*
- displaying status and counters, *Vol 1, 12–22, 12–47, 12–50*
- exiting, *Vol 1, 12–4*
- invoking, *Vol 1, 12–3*
- MOP console carrier, *Vol 1, 12–11*
- MOP trigger boot, *Vol 1, 12–53*
- setting device information, *Vol 1, 12–15*

### LANCP (Local Area Network Control Program)

- See LAN Control Program (LANCP) utility
- LANs (local area networks)
  - clearing counters, *Vol 1, 12–9*
  - deleting device information, *Vol 1, 12–8, 12–24*
  - deleting node information, *Vol 1, 12–10, 12–25*
  - displaying LAN device parameters, *Vol 1, 12–20, 12–43*
  - displaying node information, *Vol 1, 12–22, 12–50*
  - displaying OPCOM messages, *Vol 1, 12–26*
  - displaying status and counters, *Vol 1, 12–22, 12–47, 12–50*
  - LANCP SPAWN command, *Vol 1, 12–52*
  - MOP console carrier, *Vol 1, 12–11*
  - MOP trigger boot, *Vol 1, 12–53*
  - setting device information, *Vol 1, 12–15*
- LAN\_FLAGS system parameter, *Vol 2, C–22*
- Large buffers, *Vol 1, 13–36*
- LAT\$STARTUP.COM command procedure, *Vol 1, 13–41*

### LAT Control Program (LATCP) utility

- announcements, *Vol 1, 13–33*
- exiting, *Vol 1, 13–26*
- help, *Vol 1, 13–27*
- invoking, *Vol 1, 13–3*
- setting SCSSYSTEMID system parameter, *Vol 1, 13–7*

### LATCP

- See LAT Control Program utility

### LAT protocol

- Break-in attempts, *Vol 2, C–24*

### LAT software

- changing DECserver characteristics, *Vol 1, 13–10, 13–43*
- characteristics
  - added or changed, *Vol 1, 13–3*

- LAT software
  - characteristics (cont'd)
    - storing, *Vol 1*, 13–3
  - connections, *Vol 1*, 13–39, 13–66
    - incoming, *Vol 1*, 13–34
    - specifying type of, *Vol 1*, 13–34
  - devices
    - changing HANGUP characteristic, *Vol 1*, 13–10
  - device unit number, *Vol 1*, 13–40
  - outgoing connections, *Vol 1*, 13–34
  - service
    - creating, *Vol 1*, 13–15
    - database, *Vol 1*, 13–37, 13–40
    - deleting, *Vol 1*, 13–25
    - displaying characteristics of, *Vol 1*, 13–67
    - displaying counters, *Vol 1*, 13–66
    - displaying information, *Vol 1*, 13–66
    - identifier, *Vol 1*, 13–15, 13–36, 13–47
    - limiting size of node database, *Vol 1*, 13–37, 13–40
    - modifying characteristics of, *Vol 1*, 13–47
    - naming, *Vol 1*, 13–15, 13–31
    - rating, *Vol 1*, 13–16, 13–48, 13–59, 13–67
  - session limit, *Vol 1*, 13–35, 13–38
  - starting with SET NODE/STATE=ON, *Vol 1*, 13–39
  - stopping with SET NODE/STATE=OFF, *Vol 1*, 13–39
- LCKMGR\_CPUID system parameter, *Vol 2*, C–23
- LGI system parameters
  - LGI\_BRK\_DISUSER, *Vol 2*, C–24
  - LGI\_BRK\_LIM, *Vol 2*, C–24
  - LGI\_BRK\_TERM, *Vol 2*, C–24
  - LGI\_BRK\_TMO, *Vol 2*, C–24
  - LGI\_CALLOUTS, *Vol 2*, C–24
  - LGI\_HID\_TIM, *Vol 2*, C–24
  - LGI\_PWD\_TMO, *Vol 2*, C–25
  - LGI\_RETRY\_LIM, *Vol 2*, C–25
  - LGI\_RETRY\_TMO, *Vol 2*, C–25
- See also System parameters
  - Login security
- LICENSE LOAD command
  - in SYSMAN, *Vol 2*, 7–93
- Licenses
  - deactivating, *Vol 2*, 7–94
- LICENSE UNLOAD command
  - in SYSMAN, *Vol 2*, 7–94
- Limited port, *Vol 1*, 13–11, 13–44
- Links
  - creating with LATCP, *Vol 1*, 13–7
  - deleting with LATCP, *Vol 1*, 13–22
  - displaying characteristics using LATCP, *Vol 1*, 13–53
  - displaying counters using LATCP, *Vol 1*, 13–50
  - displaying information using LATCP, *Vol 1*, 13–50
- Links (cont'd)
  - modifying characteristics using LATCP, *Vol 1*, 13–31
  - name options in LATCP, *Vol 1*, 13–7
  - state of in LAT environment, *Vol 1*, 13–8
- LIST/IDENTIFIER command
  - in Authorize utility, *Vol 1*, 5–57
- LIST/PROXY command
  - in Authorize utility, *Vol 1*, 5–59
- LIST/RIGHTS command
  - in Authorize utility, *Vol 1*, 5–60
- LIST command
  - in Audit Analysis utility, *Vol 1*, 4–34
  - in Authorize utility, *Vol 1*, 5–55
  - in Install utility, *Vol 1*, 11–15
  - in POLYCENTER Software Installation utility, *Vol 2*, 3–42
- LIST DEVICE command
  - in LANCP utility, *Vol 1*, 12–20
- Listing
  - network proxy database, *Vol 1*, 5–59
  - rights database, *Vol 1*, 5–57, 5–60
  - save-set information, *Vol 1*, 7–20, 7–43, 7–64
  - system user authorization file (SYSUAF), *Vol 1*, 5–55
- LIST NODE command
  - in LANCP utility, *Vol 1*, 12–22
- /LIST qualifier, *Vol 1*, 3–6
  - in Analyze/Disk\_Structure utility, *Vol 1*, 3–11
  - in Backup utility, *Vol 1*, 7–64
- LK201-specific editing commands, *Vol 1*, A–5
- LMCP
  - See Log Manager Control Program (LMCP) utility
- LNMPHASHTBL system parameter, *Vol 2*, C–25
- LNMSHASHTBL system parameter, *Vol 2*, C–25
- Load balancing, *Vol 1*, 13–16, 13–48
  - and CPU power, *Vol 1*, 13–34
- LOAD command
  - in SYSGEN, *Vol 2*, 6–18
- Loading device drivers
  - on Alpha and I64 systems, *Vol 2*, 7–69, 7–71, 7–80
- Loading I/O driver, *Vol 2*, 6–18
- LOAD\_PWD\_POLICY system parameter, *Vol 2*, C–25
- LOAD\_SYS\_IMAGES system parameter, *Vol 2*, C–26
- Local area cluster environments
  - setting system date and time, *Vol 2*, 7–32
- LOCAL\_PORTS class
  - adding, *Vol 2*, 5–35
  - fields belonging to, *Vol 2*, 5–8
  - removing, *Vol 2*, 5–60
- LOCK class record
  - in Monitor utility, *Vol 2*, A–24

LOCKDIRWT system parameter, *Vol 2*, C-26  
 LOCKIDTBL system parameter, *Vol 2*, C-26  
 LOCKIDTBL\_MAX system parameter, *Vol 2*, C-26  
 LOCKRETRY system parameter, *Vol 2*, C-26  
 /LOCK\_VOLUME qualifier  
   in Analyze/Disk\_Structure utility, *Vol 1*, 3-12  
 Login command procedures  
   LOGIN.COM, *Vol 1*, 5-49  
 Log Manager Control Program (LMCP) utility, *Vol 1*, 14-1  
   CLOSE LOG command, *Vol 1*, 14-4  
   CONVERT LOG command, *Vol 1*, 14-5  
   CREATE LOG command, *Vol 1*, 14-7  
   DUMP command, *Vol 1*, 14-9  
   EXIT command, *Vol 1*, 14-14  
   HELP command, *Vol 1*, 14-15  
   REPAIR command, *Vol 1*, 14-16  
   SHOW LOG command, *Vol 1*, 14-19  
 /LOG qualifier  
   in Backup utility, *Vol 1*, 7-66  
 LONGWAIT system parameter, *Vol 2*, C-27  
 Lost files  
   recovering, *Vol 1*, 3-5  
 LPBEGIN phase of system startup, *Vol 2*, 7-143  
 LPBETA phase of system startup, *Vol 2*, 7-143  
 LPMAIN phase of system startup, *Vol 2*, 7-143  
 LTA device, *Vol 1*, 13-34  
 LTDRIVER (LAT port driver)  
   turning on and off, *Vol 1*, 13-3

## M

Magnetic tape devices  
   serving within a cluster, *Vol 2*, C-66  
 Management environment  
   defining, *Vol 2*, 7-120  
 MAXBOB\* system parameters, *Vol 2*, C-27  
 MAXBOBMEM system parameter, *Vol 2*, C-27  
 MAXBUF system parameter, *Vol 2*, C-27  
 MAXCLASSPRI system parameter, *Vol 2*, C-27  
 MAXPROCESSCNT system parameter, *Vol 2*, C-27  
 MAXQUEPRI system parameter, *Vol 2*, C-27  
 MAXSYSGROUP system parameter, *Vol 2*, C-28  
 MC\_SERVICES\_P0 system parameter, *Vol 2*, C-28  
 MC\_SERVICES\_P1 system parameter, *Vol 2*, C-28  
 MC\_SERVICES\_P2 system parameter, *Vol 2*, C-28  
 MC\_SERVICES\_P3 system parameter, *Vol 2*, C-28  
 MC\_SERVICES\_P4 system parameter, *Vol 2*, C-28

MC\_SERVICES\_P5 system parameter, *Vol 2*, C-29  
 MC\_SERVICES\_P6 system parameter, *Vol 2*, C-29  
 MC\_SERVICES\_P7 system parameter, *Vol 2*, C-29  
 MC\_SERVICES\_P8 system parameter, *Vol 2*, C-29  
 MC\_SERVICES\_P9 system parameter, *Vol 2*, C-29  
 /MEDIA\_FORMAT qualifier  
   in Backup utility, *Vol 1*, 7-67  
 MEMBERS class  
   adding, *Vol 2*, 5-37  
   fields belonging to, *Vol 2*, 5-9  
   removing, *Vol 2*, 5-61  
 Memory  
   multiport, *Vol 2*, 6-23  
 Memory reservations  
   displays on running system, *Vol 2*, 7-118  
   frees on running system, *Vol 2*, 7-111  
 MINCLASSPRI system parameter, *Vol 2*, C-29  
 MINWSCNT system parameter, *Vol 2*, C-30  
 MMG\_CTLFLAGS system parameter, *Vol 2*, C-30  
 MODES class record  
   in Monitor utility, *Vol 2*, A-25  
 Modes of execution, *Vol 1*, 6-11  
   for SYSMAN startup files, *Vol 2*, 7-136  
 Modes of operation, *Vol 1*, 3-4  
   image, *Vol 1*, 7-51  
   physical, *Vol 1*, 7-73  
 /MODIFIED qualifier  
   in Backup utility, *Vol 1*, 7-68  
 MODIFY/IDENTIFIER command  
   in Authorize utility, *Vol 1*, 5-73  
 MODIFY/PROXY command  
   in Authorize utility, *Vol 1*, 5-75  
 MODIFY/SYSTEM\_PASSWORD command  
   in Authorize utility, *Vol 1*, 5-77  
 MODIFY command  
   in Authorize utility, *Vol 1*, 5-61  
 MODIFY DEVICE command  
   in UCM, *Vol 2*, 8-17  
 Modifying page, swap, and dump file sizes, *Vol 1*, 6-3  
 Modifying system parameters  
   recommended method, *Vol 1*, 6-3  
 MONITOR ALL\_CLASSES command  
   in Monitor utility, *Vol 2*, 1-17  
 MONITOR CLUSTER command  
   in Monitor utility, *Vol 2*, 1-19  
 MONITOR command  
   in Monitor utility, *Vol 2*, 1-11  
 MONITOR DECNET command  
   in Monitor utility, *Vol 2*, 1-24  
 MONITOR DISK command  
   in Monitor utility, *Vol 2*, 1-26

MONITOR DLOCK command  
     in Monitor utility, *Vol 2, 1-29*  
 MONITOR FCP command  
     in Monitor utility, *Vol 2, 1-32*  
 MONITOR FILE\_SYSTEM\_CACHE command  
     in Monitor utility, *Vol 2, 1-34*  
 MONITOR IO command  
     in Monitor utility, *Vol 2, 1-38*  
 MONITOR LOCK command  
     in Monitor utility, *Vol 2, 1-40*  
 MONITOR MODES command  
     in Monitor utility, *Vol 2, 1-42*  
 MONITOR MSCP\_SERVER command  
     in Monitor utility, *Vol 2, 1-46*  
 MONITOR PAGE command  
     in Monitor utility, *Vol 2, 1-48*  
 MONITOR PROCESSES command  
     in Monitor utility, *Vol 2, 1-50*  
 MONITOR RLOCK command  
     in Monitor utility, *Vol 2, 1-56*  
 MONITOR RMS command  
     in Monitor utility, *Vol 2, 1-58*  
 MONITOR SCS command  
     in Monitor utility, *Vol 2, 1-62*  
 MONITOR STATES command  
     in Monitor utility, *Vol 2, 1-66*  
 MONITOR SYSTEM command  
     in Monitor utility, *Vol 2, 1-70*  
 MONITOR TIMER command  
     in Monitor utility, *Vol 2, 1-74*  
 MONITOR TRANSACTION command  
     in Monitor utility, *Vol 2, 1-76*  
 Monitor utility (MONITOR)  
     classname qualifiers  
         /ALL, *Vol 2, 1-56*  
         /CPU, *Vol 2, 1-42*  
         /ITEM, *Vol 2, 1-26, 1-62*  
         /TOPBIO, *Vol 2, 1-50*  
         /TOPCPU, *Vol 2, 1-50*  
         /TOPDIO, *Vol 2, 1-50*  
         /TOPFAULT, *Vol 2, 1-50*  
         /TOPRBS, *Vol 2, 1-50*  
     Cluster performance statistics, *Vol 2, 1-19*  
     collection interval, *Vol 2, 1-14*  
     command qualifiers, *Vol 2, 1-12 to 1-16*  
     commands, *Vol 2, 1-5 to 1-83*  
         CONVERT, *Vol 2, 1-6*  
         EXECUTE, *Vol 2, 1-7*  
         EXIT, *Vol 2, 1-8*  
         HELP, *Vol 2, 1-9*  
         INITIALIZE, *Vol 2, 1-10*  
         MONITOR, *Vol 2, 1-11*  
         MONITOR ALL\_CLASSES, *Vol 2, 1-17*  
         MONITOR CLUSTER, *Vol 2, 1-19*  
         MONITOR DECNET, *Vol 2, 1-24*  
         MONITOR DISK, *Vol 2, 1-26*  
         MONITOR DLOCK, *Vol 2, 1-29*  
         MONITOR FCP, *Vol 2, 1-32*  
         Monitor utility (MONITOR)  
             commands (cont'd)  
                 MONITOR FILE\_SYSTEM\_CACHE, *Vol 2, 1-34*  
                 MONITOR IO, *Vol 2, 1-38*  
                 MONITOR LOCK, *Vol 2, 1-40*  
                 MONITOR MODES, *Vol 2, 1-42*  
                 MONITOR MSCP\_SERVER, *Vol 2, 1-46*  
                 MONITOR PAGE, *Vol 2, 1-48*  
                 MONITOR PROCESSES, *Vol 2, 1-50*  
                 MONITOR RLOCK, *Vol 2, 1-56*  
                 MONITOR RMS, *Vol 2, 1-58*  
                 MONITOR SCS, *Vol 2, 1-62*  
                 MONITOR STATES, *Vol 2, 1-66*  
                 MONITOR SYSTEM, *Vol 2, 1-70*  
                 MONITOR TIMER, *Vol 2, 1-74*  
                 MONITOR TRANSACTION, *Vol 2, 1-76*  
                 MONITOR VBS, *Vol 2, 1-79*  
                 MONITOR VECTOR, *Vol 2, 1-80*  
                 SET DEFAULT, *Vol 2, 1-82*  
                 SHOW DEFAULT, *Vol 2, 1-83*  
             directing output, *Vol 2, 1-4*  
             exiting, *Vol 2, 1-4*  
             invoking, *Vol 2, 1-4*  
             recording file record formats, *Vol 2, A-1 to A-15*  
             relationship of qualifier and event, *Vol 2, 1-14*  
     MONITOR\_SERVER process  
         created on remote cluster node, *Vol 2, 1-20*  
         log file, *Vol 2, 1-21*  
         network startup, *Vol 2, 1-21*  
         purpose, *Vol 2, 1-20*  
         startup time, *Vol 2, 1-21*  
     MOP downline load services  
         clearing counters, *Vol 1, 12-9*  
         console carrier, *Vol 1, 12-11*  
         displaying status and counters, *Vol 1, 12-22, 12-50*  
         trigger boot, *Vol 1, 12-53*  
     MOUNT command  
         avoiding use of /CLUSTER with SYSMAN DO command, *Vol 2, 7-54*  
     Mount error message, *Vol 1, 7-81*  
     Mount verification  
         forced by VERIFY utility, *Vol 1, 3-4*  
     MOVE command  
         in Show Cluster utility, *Vol 2, 5-47*  
     MOVE keypad function, *Vol 2, B-3*  
     MPDEV\_AFB\_INTVL system parameter, *Vol 2, C-30*  
     MPDEV\_D\* system parameters, *Vol 2, C-31*  
     MPDEV\_ENABLE system parameter, *Vol 2, C-31*  
     MPDEV\_LCRETRIES system parameter, *Vol 2, C-31*  
     MPDEV\_POLLER system parameter, *Vol 2, C-31*  
     MPDEV\_REMOTE system parameter, *Vol 2, C-31*

MPW\_HILIMIT system parameter, *Vol 2*, C-31  
 MPW\_IOLIMIT system parameter, *Vol 2*, C-32  
 MPW\_LOLIMIT system parameter, *Vol 2*, C-32  
 MPW\_LOWAITLIMIT system parameter, *Vol 2*, C-32  
 MPW\_PRIO system parameter, *Vol 2*, C-32  
 MPW\_THRESH system parameter, *Vol 2*, C-32  
 MPW\_WAITLIMIT system parameter, *Vol 2*, C-32  
 MPW\_WRTCLUSTER system parameter, *Vol 2*, C-32  
 MSCP\_BUFFER system parameter, *Vol 2*, C-33  
 MSCP\_CMD\_TMO system parameter, *Vol 2*, C-33  
 MSCP\_CREDITS system parameter, *Vol 2*, C-33  
 MSCP\_LOAD system parameter, *Vol 2*, C-33  
 MSCP\_SERVER class record  
   in Monitor utility, *Vol 2*, A-27  
 MSCP\_SERVE\_ALL system parameter, *Vol 2*, C-33  
 Multi-architecture OpenVMS Cluster systems  
   installing images, *Vol 2*, 7-54  
 Multicast address  
   cluster  
     displaying, *Vol 2*, 7-34  
 Multicast messages, *Vol 1*, 13-25, 13-50  
   timer for, *Vol 1*, 13-37  
 Multiport memory  
   connecting, *Vol 2*, 6-23  
   initializing, *Vol 2*, 6-24  
   sharing, *Vol 2*, 6-23  
 MULTIPROCESSING system parameter, *Vol 2*, C-34  
 MULTITHREAD system parameter, *Vol 2*, C-35  
 MVSUPMSG\_INTVL system parameter, *Vol 2*, C-35  
 MVSUPMSG\_NUM system parameter, *Vol 2*, C-35  
 MVTIMEOUT system parameter, *Vol 2*, C-36

## N

Name Hidden attribute, *Vol 1*, 5-20, 5-51, 5-73  
 NET\$PROXY.DAT files  
   description of, *Vol 1*, 5-3  
 NETPROXY.DAT files  
   creating, *Vol 1*, 5-36  
   description of, *Vol 1*, 5-3  
   displaying proxy access, *Vol 1*, 5-93  
   modifying, *Vol 1*, 5-3  
 Network proxy authorization files, *Vol 1*, 5-3  
   creating, *Vol 1*, 5-36  
   modifying, *Vol 1*, 5-3  
 Network statistics  
   in Monitor utility, *Vol 2*, 1-24  
 NET\_CALLOUTS system parameter, *Vol 2*, C-36  
 NEWPARAMS.DAT file, *Vol 1*, 6-3  
   contents, *Vol 1*, 6-5

/NEW\_VERSION qualifier  
   in Backup utility, *Vol 1*, 7-69  
 NEXT FILE command  
   in Audit Analysis utility, *Vol 1*, 4-35  
 NEXT RECORD command  
   in Audit Analysis utility, *Vol 1*, 4-36  
 Nexus numbers  
   showing, *Vol 2*, 6-31, 6-32, 6-39  
 NISCS\_CONV\_BOOT system parameter, *Vol 2*, C-36  
 NISCS\_LOAD\_PEA0 system parameter, *Vol 2*, C-36  
 NISCS\_MAX\_PKTSZ system parameter, *Vol 2*, C-36  
 NISCS\_PORT\_SERV system parameter, *Vol 2*, C-37  
 No Access attribute, *Vol 1*, 5-20, 5-51, 5-73  
 NOAUTOCONFIG system parameter, *Vol 2*, C-37  
 NOCLUSTER system parameter, *Vol 2*, C-37  
 Node names, *Vol 1*, 13-33  
 Node numbers  
   display on Alpha and I64 systems, *Vol 2*, 7-87  
 /NODE qualifier  
   in Monitor utility, *Vol 2*, 1-15  
 Nodes  
   database, *Vol 1*, 13-37, 13-40  
   deleting node information, *Vol 1*, 12-10, 12-25  
   displaying characteristics of, *Vol 1*, 13-58  
   displaying counters, *Vol 1*, 13-55  
   displaying information using LATCP, *Vol 1*, 13-55  
   limit, *Vol 1*, 13-37  
   modifying characteristics of, *Vol 1*, 13-33  
   state, *Vol 1*, 13-39  
   target, *Vol 1*, 13-63  
 NOFEEDBACK mode, *Vol 1*, 6-11  
 NOPGFLSWP system parameter, *Vol 2*, C-38  
 /[NO]CONFIRM qualifier  
   in Analyze/Disk\_Structure utility, *Vol 1*, 3-9  
 /[NO]LIST qualifier  
   in Analyze/Disk\_Structure utility, *Vol 1*, 3-11  
 /[NO]LOCK\_VOLUME qualifier  
   in Analyze/Disk\_Structure utility, *Vol 1*, 3-12  
 /[NO]READ\_CHECK qualifier  
   in Analyze/Disk\_Structure utility, *Vol 1*, 3-14  
 /[NO]REPAIR qualifier  
   in Analyze/Disk\_Structure utility, *Vol 1*, 3-16  
 NPAGECALC system parameter, *Vol 2*, C-38  
 NPAGEDYN system parameter, *Vol 2*, C-38  
 NPAGERAD system parameter, *Vol 2*, C-38  
 NPAGEVIR system parameter, *Vol 2*, C-39  
 NPAG\_AGGRESSIVE system parameter, *Vol 2*, C-39  
 NPAG\_BAP\_MAX system parameter, *Vol 2*, C-39  
 NPAG\_BAP\_MAX\_PA system parameter, *Vol 2*, C-40



NPAG\_BAP\_MIN system parameter, *Vol 2*, C-40  
NPAG\_BAP\_MIN\_PA system parameter, *Vol 2*,  
C-40  
NPAG\_GENTLE system parameter, *Vol 2*, C-40  
NPAG\_INTERVAL system parameter, *Vol 2*, C-40  
NPAG\_RING\_SIZE system parameter, *Vol 2*,  
C-40

## O

---

Object classes  
  security class object, *Vol 1*, 1-3  
Objects  
  controlling access to, *Vol 1*, 1-14  
Object types  
  See Object classes  
OPCOM (Operator Communication Manager)  
  messages  
    generated by LANACP, *Vol 1*, 12-26  
OpenVMS Backup utility  
  See Backup utility (BACKUP)  
OpenVMS Cluster environments  
  command execution, *Vol 2*, 7-54  
  DCL commands interacting with SYSMAN  
    cluster commands, *Vol 2*, 7-5  
  displaying group number, *Vol 2*, 7-34  
  displaying multicast address, *Vol 2*, 7-34  
  displaying network information, *Vol 2*, 7-34  
  displaying system date and time, *Vol 2*, 7-35  
  dual-architecture  
    installing images, *Vol 2*, 7-55  
  group number, *Vol 2*, 7-30  
  modifying security data, *Vol 2*, 7-30  
  multi-architecture  
    installing images, *Vol 2*, 7-54  
  password, *Vol 2*, 7-30  
  setting time, *Vol 2*, 7-32  
  updating security data, *Vol 2*, 7-30  
OpenVMS RMS  
  See RMS  
Operations  
  compare  
    for BACKUP, *Vol 1*, 7-26  
  list, *Vol 1*, 7-20, 7-43, 7-64  
  performed by BACKUP, *Vol 1*, 7-3  
Operator assistance  
  with BACKUP, *Vol 1*, 7-16  
/OUTPUT qualifier, *Vol 2*, 5-19  
  in Monitor utility, *Vol 2*, 1-15  
/OVERLAY qualifier  
  in Backup utility, *Vol 1*, 7-70, 7-71  
/OWNER\_UIC qualifier  
  as Backup utility input file qualifier, *Vol 1*,  
7-72

## P

---

PACKAGE command  
  in POLYCENTER Software Installation utility,  
  *Vol 2*, 3-46  
/PAGE=SAVE qualifier navigation keys, *Vol 1*,  
5-87  
PAGE class record  
  in Monitor utility, *Vol 2*, A-28  
PAGEDYN system parameter, *Vol 2*, C-40  
PAGEFILE.SYS file  
  See Page files  
Page files  
  changing size, *Vol 1*, 6-3  
  displaying the size calculated by AUTOGEN,  
  *Vol 1*, 6-9  
  generating a new file sized by AUTOGEN, *Vol*  
  *1*, 6-10  
Page management statistics  
  in Monitor utility, *Vol 2*, 1-48  
PAGFILCNT system parameter, *Vol 2*, C-40  
PAGTBLPFC system parameter, *Vol 2*, C-40  
PAMAXPORT system parameter, *Vol 2*, C-40  
PAN command  
  in Show Cluster utility, *Vol 2*, 5-49  
PAN keypad function, *Vol 2*, B-3  
PANOPOLL system parameter, *Vol 2*, C-41  
PANUMPOLL system parameter, *Vol 2*, C-41  
PAPOLLINTERVAL system parameter, *Vol 2*,  
C-41  
PAPOOLINTERVAL system parameter, *Vol 2*,  
C-41  
Parameter files  
  ALPHAVMSSYS.OLD, *Vol 1*, 6-10  
  ALPHAVMSSYS.PAR, *Vol 1*, 6-10  
  AUTOGEN.PAR, *Vol 1*, 6-10  
  default, *Vol 1*, 6-10  
  IA64VMSSYS.OLD, *Vol 1*, 6-10  
  IA64VMSSYS.PAR, *Vol 1*, 6-10  
  VAXVMSSYS.OLD, *Vol 1*, 6-10  
  VAXVMSSYS.PAR, *Vol 1*, 6-10  
  writing, *Vol 2*, 6-45, 7-105  
Parameters  
  See also System parameters  
  changing, *Vol 2*, 7-97  
  passing to command procedures, *Vol 1*, 12-6  
PARAMETERS DISABLE CHECKS command  
  in SYSMAN, *Vol 2*, 7-95  
PARAMETERS ENABLE CHECKS command  
  in SYSMAN, *Vol 2*, 7-96  
PARAMETERS SET command  
  in SYSMAN, *Vol 2*, 7-97  
PARAMETERS SHOW command  
  in SYSMAN, *Vol 2*, 7-98  
PARAMETERS USE command  
  in SYSMAN, *Vol 2*, 7-103

PARAMETERS WRITE command  
in SYSMAN, *Vol 2, 7-105*

PARAMS.DAT file, *Vol 1, 6-9, 6-10*

PASANITY system parameter, *Vol 2, C-42*

Password, *Vol 2, 7-120*  
for OpenVMS Cluster, *Vol 2, 7-30*

PASTDGBUF system parameter, *Vol 2, C-42*

Paste buffers, *Vol 1, A-5*

PASTIMOUT system parameter, *Vol 2, C-42*

PCFs (product configuration files)  
CONFIGURE subcommand, *Vol 2, 3-6*

PDB (product database)  
recording a volume label change, *Vol 2, 3-59*

PDFs (product description files)  
of software products, *Vol 2, 3-21*

PEDRIVER  
event-tracing facility, *Vol 2, 4-6*

Per-connection buffers  
allocating, *Vol 2, C-10*

PE system parameters, *Vol 2, C-42*

PFCDEFAULT system parameter, *Vol 2, C-42*

PFN\_COLOR\_COUNT system parameter, *Vol 2, C-43*

PFRATH system parameter, *Vol 2, C-43*

PFRATL system parameter, *Vol 2, C-43*

Phases of AUTOGEN, *Vol 1, 6-8*

Physical mode, *Vol 1, 7-73*

PHYSICALPAGES system parameter, *Vol 2, C-43*

/PHYSICAL qualifier  
in Backup utility, *Vol 1, 7-73*

PHYSICAL\_MEMORY system parameter, *Vol 2, C-43*

PIOPAGES system parameter, *Vol 2, C-44*

PIXSCAN system parameter, *Vol 2, C-44*

Point-to-Point Protocol utility (PPPD)  
commands  
CONNECT, *Vol 2, 2-5*  
DIAL\_OUT, *Vol 2, 2-6*  
DISCONNECT, *Vol 2, 2-8*  
EXIT, *Vol 2, 2-9*  
HELP, *Vol 2, 2-10*  
SET, *Vol 2, 2-11*  
SHOW, *Vol 2, 2-15*  
description, *Vol 2, 2-3*  
exiting, *Vol 2, 2-3*  
overview, *Vol 2, 2-3*

POLYCENTER Software Installation utility, *Vol 2, 3-3*  
See also PRODUCT command

commands  
CONFIGURE, *Vol 2, 3-6*  
COPY, *Vol 2, 3-10*  
DELETE RECOVERY\_DATA, *Vol 2, 3-15*  
EXTRACT FILE, *Vol 2, 3-17*  
EXTRACT PDF, *Vol 2, 3-21*  
EXTRACT PTF, *Vol 2, 3-25*  
EXTRACT RELEASE\_NOTES, *Vol 2, 3-29*  
FIND, *Vol 2, 3-33*

POLYCENTER Software Installation utility  
commands (cont'd)  
INSTALL, *Vol 2, 3-36*  
LIST, *Vol 2, 3-42*  
PACKAGE, *Vol 2, 3-46*  
RECONFIGURE, *Vol 2, 3-50*  
REGISTER PRODUCT, *Vol 2, 3-55*  
REGISTER VOLUME, *Vol 2, 3-59*  
REMOVE, *Vol 2, 3-61*  
SHOW HISTORY, *Vol 2, 3-64*  
SHOW OBJECT, *Vol 2, 3-66*  
SHOW PRODUCT, *Vol 2, 3-69*  
SHOW RECOVERY\_DATA, *Vol 2, 3-72*  
invoking, *Vol 2, 3-3*  
parameters, *Vol 2, 3-3*

POOLCHECK system parameter, *Vol 2, C-44*

POOLPAGING system parameter, *Vol 2, C-44*

Ports  
application, *Vol 1, 13-11, 13-43, 13-44, 13-45, 13-61*  
characteristics, *Vol 1, 13-61*  
counters, *Vol 1, 13-61*  
creating with LATCP, *Vol 1, 13-10*  
dedicated, *Vol 1, 13-11, 13-43, 13-61*  
deleting with LATCP, *Vol 1, 13-23*  
displaying characteristics of, *Vol 1, 13-61*  
displaying information using LATCP, *Vol 1, 13-61*  
limited, *Vol 1, 13-11, 13-44*  
logical name of, *Vol 1, 13-11*  
modifying characteristics of with LATCP, *Vol 1, 13-43*  
name, *Vol 1, 13-43*  
outgoing connection, *Vol 1, 13-61*  
setting type of, *Vol 1, 13-11*  
target, *Vol 1, 13-63*

POSITION command  
in Audit Analysis utility, *Vol 1, 4-37*

POWEROFF system parameter, *Vol 2, C-44*

PPP connections  
establishing, *Vol 2, 2-5*  
setting characteristics, *Vol 2, 2-11*  
terminating, *Vol 2, 2-8*  
viewing characteristics, *Vol 2, 2-15*

PPPD  
See Point-to-Point Protocol utility

PQL system parameters  
PQL\_DASTLM, *Vol 2, C-45*  
PQL\_DBIOLM, *Vol 2, C-45*  
PQL\_DBYTLM, *Vol 2, C-45*  
PQL\_DCPULM, *Vol 2, C-45*  
PQL\_DDIOLM, *Vol 2, C-45*  
PQL\_DENQLM, *Vol 2, C-45*  
PQL\_DFILLM, *Vol 2, C-45*  
PQL\_DJTQUOTA, *Vol 2, C-45*  
PQL\_DPGFLQUOTA, *Vol 2, C-45*  
PQL\_DPRCLM, *Vol 2, C-45*  
PQL\_DTQELM, *Vol 2, C-46*

## PQL system parameters (cont'd)

- PQL\_DWSDEFAULT, *Vol 2*, C-46
- PQL\_DWSEXTENT, *Vol 2*, C-46
- PQL\_DWSQUOTA, *Vol 2*, C-46
- PQL\_MASTLM, *Vol 2*, C-46
- PQL\_MBIOLM, *Vol 2*, C-46
- PQL\_MBYTLM, *Vol 2*, C-46
- PQL\_MCPULM, *Vol 2*, C-46
- PQL\_MDIOLM, *Vol 2*, C-46
- PQL\_MENQLM, *Vol 2*, C-46
- PQL\_MFILLM, *Vol 2*, C-46
- PQL\_MJTQUOTA, *Vol 2*, C-46
- PQL\_MPGFLQUOTA, *Vol 2*, C-47
- PQL\_MPRCLM, *Vol 2*, C-47
- PQL\_MTQELM, *Vol 2*, C-47
- PQL\_MWSDEFAULT, *Vol 2*, C-47
- PQL\_MWSEXTENT, *Vol 2*, C-47
- PQL\_MWSQUOTA, *Vol 2*, C-47
- See also System parameters
  - process limits and quotas
- PRCPOLINTERVAL system parameter, *Vol 2*, C-47
- Prefix list
  - displaying on Alpha and I64 systems, *Vol 2*, 7-92
  - setting on Alpha and I64 systems, *Vol 2*, 7-86
- Printers
  - configuring in a LAT environment, *Vol 1*, 13-11, 13-43
- PRIORITY\_OFFSET system parameter, *Vol 2*, C-47
- Privileges
  - assigning to installed images, *Vol 1*, 11-8, 11-25
  - assigning to known images, *Vol 1*, 11-9, 11-25
  - displaying current, *Vol 2*, 7-126
  - modifying for environment, *Vol 2*, 7-122
  - required by ACL editor, *Vol 1*, 1-4
  - required by SYSMAN, *Vol 2*, 7-4
  - to perform BACKUP operations, *Vol 1*, 7-11
- Process activity statistics
  - in Monitor utility, *Vol 2*, 1-50
- Processes
  - granting identifiers to, *Vol 1*, 1-16
  - hibernating
    - with ATTACH command, *Vol 2*, 7-17
  - switching control of input stream to, *Vol 2*, 7-17
  - transferring control from, *Vol 1*, 13-5
- PROCESSES class record
  - in Monitor utility, *Vol 2*, A-30
- Processor modes statistics
  - in Monitor utility, *Vol 2*, 1-42
- Process states statistics
  - in Monitor utility, *Vol 2*, 1-66

## Process types

- with Accounting utility, *Vol 1*, 2-22
- PROCSCTCNT system parameter, *Vol 2*, C-48
- PRODUCT command, *Vol 2*, 3-3
  - CONFIGURE subcommand, *Vol 2*, 3-6
  - COPY subcommand, *Vol 2*, 3-10
  - DELETE RECOVERY\_DATA subcommand, *Vol 2*, 3-15
  - EXTRACT FILE subcommand, *Vol 2*, 3-17
  - EXTRACT PDF subcommand, *Vol 2*, 3-21
  - EXTRACT PTF subcommand, *Vol 2*, 3-25
  - EXTRACT RELEASE\_NOTES subcommand, *Vol 2*, 3-29
  - FIND subcommand, *Vol 2*, 3-33
  - INSTALL subcommand, *Vol 2*, 3-36
  - LIST subcommand, *Vol 2*, 3-42
  - PACKAGE subcommand, *Vol 2*, 3-46
  - RECONFIGURE subcommand, *Vol 2*, 3-50
  - REGISTER PRODUCT subcommand, *Vol 2*, 3-55
  - REGISTER VOLUME subcommand, *Vol 2*, 3-59
  - REMOVE subcommand, *Vol 2*, 3-61
  - SHOW HISTORY subcommand, *Vol 2*, 3-64
  - SHOW OBJECT subcommand, *Vol 2*, 3-66
  - SHOW PRODUCT subcommand, *Vol 2*, 3-69
  - SHOW RECOVERY\_DATA subcommand, *Vol 2*, 3-72
  - SHOW UTILITY subcommand, *Vol 2*, 3-74
  - UNDO PATCH subcommand, *Vol 2*, 3-75
- Product configuration files
  - See PCFs
- Product description files
  - See PDFs
- Product text files
  - See PTFs
- Profiles, *Vol 2*, 7-120
  - adjusting, *Vol 2*, 7-122
  - displaying, *Vol 2*, 7-126
  - in SYSMAN, *Vol 2*, 7-67
- Protection codes
  - propagating UIC-based protection in a directory tree, *Vol 1*, 1-12
- /PROTECTION qualifier
  - in Backup utility, *Vol 1*, 7-74
- Protocol bit mask
  - error definitions, *Vol 1*, 13-58
- Proxy accounts
  - adding, *Vol 1*, 5-22
  - deleting, *Vol 1*, 5-80
  - modifying, *Vol 1*, 5-75
- Proxy logins, *Vol 1*, 5-22, 5-75
- PSEUDOLOA system parameter, *Vol 2*, C-48
- PTFs (product text files)
  - of software products, *Vol 2*, 3-25

PURGE command  
  in Install utility, *Vol 1*, 11–21  
PURGE DEVICE command  
  in LANCP utility, *Vol 1*, 12–24  
PURGE NODE command  
  in LANCP utility, *Vol 1*, 12–25  
PU\_OPTIONS system parameter, *Vol 2*, C–48

## Q

QDSKINTERVAL system parameter, *Vol 2*, C–48  
QDSKVOTES system parameter, *Vol 2*, C–48  
QUANTUM system parameter, *Vol 2*, C–48  
Quota file  
  creating, *Vol 2*, 7–40  
  disabling, *Vol 2*, 7–43  
  displaying, *Vol 2*, 7–52  
  enabling, *Vol 2*, 7–45  
  modifying, *Vol 2*, 7–47  
  removing an entry, *Vol 2*, 7–42, 7–50  
  updating, *Vol 2*, 7–49

## R

RAD example, *Vol 2*, 7–152  
RAD\_SUPPORT system parameter, *Vol 2*, C–48  
Range checking  
  disabling, *Vol 2*, 7–95  
  enabling, *Vol 2*, 7–96  
/READ\_CHECK qualifier, *Vol 1*, 3–14  
  in Analyze/Disk\_Structure utility, *Vol 1*, 3–14  
REALTIME\_SPTS system parameter, *Vol 2*, C–49  
REBOOT phase of AUTOGEN, *Vol 1*, 6–10  
Rebuilding configuration tables  
  on Alpha and I64 systems, *Vol 2*, 7–81  
RECALL command  
  in LATCP, *Vol 1*, 13–28  
RECNXINTERVAL system parameter, *Vol 2*,  
  C–49  
RECONFIGURE command  
  in POLYCENTER Software Installation utility,  
  *Vol 2*, 3–50  
Recording files  
  class types, *Vol 2*, A–10  
  in Monitor utility  
    class header, *Vol 2*, A–11  
    class prefix for component classes, *Vol 2*,  
    A–12  
    class records, *Vol 2*, A–10  
    data block, *Vol 2*, A–13  
    description, *Vol 2*, A–1  
    file header record, *Vol 2*, A–4  
    HP control records, *Vol 2*, A–3  
    I/O class record, *Vol 2*, A–22  
    node transition record, *Vol 2*, A–9  
    RLOCK class record, *Vol 2*, A–33  
    RMS class record, *Vol 2*, A–34  
    RMS file record, *Vol 2*, A–9

Recording files  
  in Monitor utility (cont'd)  
    SCS class record, *Vol 2*, A–35  
    system information record, *Vol 2*, A–7  
/RECORD qualifier  
  in Backup utility, *Vol 1*, 7–76  
  in Monitor utility, *Vol 2*, 1–15  
Records  
  See also Security audit log files  
  duplicating, *Vol 1*, 5–24  
Record types  
  in Monitor utility, *Vol 2*, A–2  
/RECORD\_ATTRIBUTES qualifier  
  in Analyze/Disk\_Structure utility, *Vol 1*, 3–15  
Recovering lost files, *Vol 1*, 3–5  
REFRESH command  
  in LATCP, *Vol 1*, 13–29  
  in Show Cluster utility, *Vol 2*, 5–51  
REGISTER PRODUCT command  
  in POLYCENTER Software Installation utility,  
  *Vol 2*, 3–55  
REGISTER VOLUME command  
  in POLYCENTER Software Installation utility,  
  *Vol 2*, 3–59  
Release Notes  
  on software products, *Vol 2*, 3–29  
/RELEASE\_TAPE qualifier  
  in Backup utility, *Vol 1*, 7–77  
RELOAD command  
  in SYSGEN, *Vol 2*, 6–19  
  in UCM, *Vol 2*, 8–19  
Remote nodes  
  creating MONITOR\_SERVER process on, *Vol*  
  2, 1–20  
REMOVE (Field) command  
  in Show Cluster utility, *Vol 2*, 5–59  
REMOVE/IDENTIFIER command  
  in Authorize utility, *Vol 1*, 5–79  
REMOVE/PROXY command  
  in Authorize utility, *Vol 1*, 5–80  
REMOVE CIRCUITS command  
  in Show Cluster utility, *Vol 2*, 5–52  
REMOVE CLUSTER command  
  in Show Cluster utility, *Vol 2*, 5–53  
REMOVE command  
  in Authorize utility, *Vol 1*, 5–78  
  in Install utility, *Vol 1*, 11–22  
  in POLYCENTER Software Installation utility,  
  *Vol 2*, 3–61  
REMOVE CONNECTIONS command  
  in Show Cluster utility, *Vol 2*, 5–54  
REMOVE COUNTERS command  
  in Show Cluster utility, *Vol 2*, 5–56  
REMOVE CREDITS command  
  in Show Cluster utility, *Vol 2*, 5–57  
REMOVE ERRORS command  
  in Show Cluster utility, *Vol 2*, 5–58

REMOVE LOCAL\_PORTS command  
     in Show Cluster utility, *Vol 2*, 5–60  
 REMOVE MEMBERS command  
     in Show Cluster utility, *Vol 2*, 5–61  
 REMOVE SYSTEMS command  
     in Show Cluster utility, *Vol 2*, 5–62  
 Removing installed software products, *Vol 2*, 3–61  
 RENAME/IDENTIFIER command  
     in Authorize utility, *Vol 1*, 5–84  
 RENAME command  
     in Authorize utility, *Vol 1*, 5–81  
 REPAIR command  
     in LMCP, *Vol 1*, 14–16  
 Repairing errors, *Vol 1*, 3–4, 3–9, 3–10, 3–15, 3–16  
 /REPAIR qualifier  
     in Analyze/Disk\_Structure utility, *Vol 1*, 3–16  
 REPLACE command  
     in Install utility, *Vol 1*, 11–23  
 /REPLACE qualifier  
     in Backup utility, *Vol 1*, 7–78  
 Report formats from Audit Analysis utility  
     brief, *Vol 1*, 4–7  
     full, *Vol 1*, 4–10  
     summary, *Vol 1*, 4–26  
 Reporting errors, *Vol 1*, 3–4  
 Reports  
     SHOW CLUSTER, *Vol 2*, 5–12  
         command to modify, *Vol 2*, B–3  
         controlling displays, *Vol 2*, 5–12, 5–13  
         default display, *Vol 2*, 5–46  
         formatting, *Vol 2*, 5–70  
         organization of, *Vol 2*, 5–3  
         printing, *Vol 2*, 5–74  
         time period for, *Vol 2*, 5–15, 5–17  
         update interval, *Vol 2*, 5–18, 5–72  
 Requirements and limitations  
     in ANALYZE/DISK\_STRUCTURE, *Vol 1*, 3–7  
 RESALLOC system parameter, *Vol 2*, C–49  
 Reserved Memory Registry  
     adding an entry, *Vol 2*, 7–106  
     adding RADs to a reservation, *Vol 2*, 7–110  
     AUTOGEN processes, *Vol 2*, 7–106  
     modifying an existing entry, *Vol 2*, 7–114  
     previews reservation, *Vol 2*, 7–113  
     removing an entry, *Vol 2*, 7–117  
 RESERVED\_MEMORY commands  
     in SYSMAN, *Vol 2*, 7–106  
 RESERVED\_MEMORY EXTEND command, *Vol 2*, 7–110  
 RESERVED\_MEMORY FREE command, *Vol 2*, 7–111  
 RESERVED\_MEMORY LIST command, *Vol 2*, 7–113  
 RESERVED\_MEMORY MODIFY command, *Vol 2*, 7–114  
 RESERVED\_MEMORY REMOVE command, *Vol 2*, 7–117  
 RESERVED\_MEMORY SHOW command, *Vol 2*, 7–118  
 RESHASH\_TBL system parameter, *Vol 2*, C–50  
 Resource accounting  
     programming information, *Vol 1*, C–1  
 Resource attributes, *Vol 1*, 5–20, 5–51, 5–73  
 Resource use  
     producing reports of, *Vol 1*, 2–9, 2–12, 2–33  
 RESTART command  
     in UCM, *Vol 2*, 8–20  
 Retransmit limit, *Vol 1*, 13–37  
 REVOKE/IDENTIFIER command  
     in Authorize utility, *Vol 1*, 5–85  
 /REWIND qualifier  
     as Backup utility input save-set qualifier, *Vol 1*, 7–79  
     as Backup utility output save-set qualifier, *Vol 1*, 7–80  
 Rights databases  
     See RIGHTS\_LIST.DAT files  
 Rights identifier, *Vol 2*, 7–47  
     for disk quota file, *Vol 2*, 7–38  
 RIGHTS\_LIST.DAT files  
     adding identifiers to, *Vol 1*, 5–20  
     altering identifiers in, *Vol 1*, 5–73  
     creating and maintaining, *Vol 1*, 5–3, 5–37  
     deleting identifiers from, *Vol 1*, 5–79  
     displaying identifiers in, *Vol 1*, 5–91  
     displaying records in, *Vol 1*, 5–95  
     renaming identifiers, *Vol 1*, 5–84  
 RJOB\_LIM system parameter, *Vol 2*, C–50  
 RLOCK class record  
     in Monitor utility, *Vol 2*, A–33  
 RMS class record  
     in Monitor utility, *Vol 2*, A–34  
 RMS statistics  
     in Monitor utility, *Vol 2*, 1–58  
 RMS system parameters  
     RMSD\*, *Vol 2*, C–50  
     RMS\_CONPOLICY, *Vol 2*, C–50  
     RMS\_DFLRL, *Vol 2*, C–50  
     RMS\_DFMBC, *Vol 2*, C–50  
     RMS\_DFMBFIDX, *Vol 2*, C–50  
     RMS\_DFMBFREL, *Vol 2*, C–51  
     RMS\_DFMBFSDK, *Vol 2*, C–51  
     RMS\_DFMBFSMT, *Vol 2*, C–51  
     RMS\_DFMBFSUR, *Vol 2*, C–51  
     RMS\_DFNBC, *Vol 2*, C–51  
     RMS\_EXTEND\_SIZE, *Vol 2*, C–52  
     RMS\_FILEPROT, *Vol 2*, C–52  
     RMS\_HEURISTIC, *Vol 2*, C–52  
     RMS\_PROLOGUE, *Vol 2*, C–52  
     RMS\_SEQFILE\_WBH, *Vol 2*, C–52

RSRVPAGCNT system parameter, *Vol 2*, C-53

## S

S0\_PAGING system parameter, *Vol 2*, C-53

S2\_SIZE system parameter, *Vol 2*, C-53

SAVE command

in Show Cluster utility, *Vol 2*, 5-64

SAVEDUMP system parameter, *Vol 2*, C-53

Save sets

description of, *Vol 1*, 7-3

directing output to, *Vol 1*, 7-11

Save-set specification, *Vol 1*, 7-5

/SAVE\_SET qualifier

as Backup utility input save-set qualifier, *Vol 1*, 7-83

as Backup utility output save-set qualifier, *Vol 1*, 7-84

SAVPARAMS phase of AUTOGEN, *Vol 1*, 6-8

SCA (Systems Communications Architecture)

in SCACP utility, *Vol 2*, 4-3

paths (equivalent to circuits), *Vol 2*, 4-3

SCA Control Program (SCACP)

commands, *Vol 2*, 4-7

EXIT, *Vol 2*, 4-10

HELP, *Vol 2*, 4-11

SET CHANNEL, *Vol 2*, 4-12

SET CIRCUIT, *Vol 2*, 4-14

SET LAN\_DEVICE, *Vol 2*, 4-15

SET PORT, *Vol 2*, 4-16

SET TRACE, *Vol 2*, 4-17

SET VC, *Vol 2*, 4-19

SHOW CHANNEL, *Vol 2*, 4-21

SHOW CIRCUIT, *Vol 2*, 4-23

SHOW LAN\_DEVICE, *Vol 2*, 4-25

SHOW PORT, *Vol 2*, 4-27

SHOW TRACE, *Vol 2*, 4-28

SHOW VC, *Vol 2*, 4-31

SPAWN, *Vol 2*, 4-33

START LAN\_DEVICE, *Vol 2*, 4-34

START TRACE, *Vol 2*, 4-35

STOP LAN\_DEVICE, *Vol 2*, 4-37

STOP TRACE, *Vol 2*, 4-38

event tracing, *Vol 2*, 4-6

invoking, *Vol 2*, 4-8

LAN channels, *Vol 2*, 4-4

LAN circuits, *Vol 2*, 4-4

LAN cluster ports, *Vol 2*, 4-6

overview, *Vol 2*, 4-3

PEDRIVER (LAN cluster driver), *Vol 2*, 4-4

privileges required, *Vol 2*, 4-9

terminology

SCA, *Vol 2*, 4-3

troubleshooting with, *Vol 2*, 4-7

virtual circuits, *Vol 2*, 4-4

SCACP

See SCA Control Program

SCBs (storage control blocks)

updating, *Vol 1*, 3-4

Scheduling classes

creating, *Vol 2*, 7-19

deleting, *Vol 2*, 7-24

displaying, *Vol 2*, 7-28

modifying characteristics, *Vol 2*, 7-25

resuming suspended classes, *Vol 2*, 7-27

suspending temporarily, *Vol 2*, 7-29

SCH\_CTLFLAGS system parameter, *Vol 2*, C-53

Screens

refreshing from LATCP, *Vol 1*, 13-29

SCROLL command

in LATCP, *Vol 1*, 13-30

in Show Cluster utility, *Vol 2*, 5-66

SCROLL keypad function, *Vol 2*, B-3

SCS (Systems Communications Services)

circuits and ports in SCACP utility, *Vol 2*, 4-3

SHOW CLUSTER information, *Vol 2*, 5-3

SCS class record

in Monitor utility, *Vol 2*, A-35

SCSICLUSTER\_P[1-4] system parameter, *Vol 2*, C-55

SCSI\_ERROR\_POLL system parameter, *Vol 2*, C-54

SCSI\_NOAUTO system parameter, *Vol 2*, C-54

SCS statistics

in Monitor utility, *Vol 2*, 1-62

SCSSYSTEMID system parameter

and LATCP CREATE LINK command, *Vol 1*, 13-7

SCS system parameters

SCSBUFFCNT, *Vol 2*, C-53

SCSFLOWCUSH, *Vol 2*, C-54

SCSMAXDG, *Vol 2*, C-55

SCSMAXMSG, *Vol 2*, C-55

SCSNODE, *Vol 2*, C-55

SCSRESPCNT, *Vol 2*, C-55

SCSSYSTEMID, *Vol 2*, C-55

SCSSYSTEMIDH, *Vol 2*, C-56

SECAUDIT.COM command procedure

See Audit Analysis utility

Security Alarm ACEs

See Alarm ACEs

Security alarm messages

specifying access criteria for, *Vol 1*, 1-6, 1-8

Security archive files, analyzing, *Vol 1*, 4-4

Security Audit ACEs

See Audit ACEs

Security audit data packets

fields, *Vol 1*, F-7

format, *Vol 1*, F-6

Security audit event messages

data packets in, *Vol 1*, F-6

event flags, *Vol 1*, F-6

extracting from log files, *Vol 1*, 4-3

format, *Vol 1*, F-1

- Security audit event messages (cont'd)
  - subtypes for events, *Vol 1*, F-2
  - types of events, *Vol 1*, F-2
- Security audit header packets, *Vol 1*, F-2
- Security audit log files
  - advancing to next record, *Vol 1*, 4-37
  - analyzing, *Vol 1*, 4-3, 4-4
  - data packet format, *Vol 1*, F-6
  - displaying records, *Vol 1*, 4-30
  - displaying the full record, *Vol 1*, 4-10
  - excluding information from reports, *Vol 1*, 4-11
  - header packet format, *Vol 1*, F-1
  - opening new input file, *Vol 1*, 4-35
  - resuming processing, *Vol 1*, 4-29
  - returning to previous record, *Vol 1*, 4-37
  - selecting records by event class, *Vol 1*, 4-8
  - selecting records by time, *Vol 1*, 4-5, 4-25
  - specifying duration of record display, *Vol 1*, 4-14
  - writing alarm messages to, *Vol 1*, 1-8
- Security audit records
  - criteria for selecting, *Vol 1*, 4-15
- Security audit reports
  - advancing the display, *Vol 1*, 4-36
  - binary output, *Vol 1*, 4-6
  - brief format, *Vol 1*, 4-7
  - creating, *Vol 1*, 4-3
  - directing output, *Vol 1*, 4-13
  - displaying selection criteria, *Vol 1*, 4-38
  - duration of record display, *Vol 1*, 4-14
  - excluding information, *Vol 1*, 4-11
  - full format, *Vol 1*, 4-10
  - interrupting the display, *Vol 1*, 4-28
  - modifying selection criteria, *Vol 1*, 4-30
  - processing additional log files, *Vol 1*, 4-35
  - resuming the processing, *Vol 1*, 4-28
  - summary of records, *Vol 1*, 4-26
- Security event classes
  - extracting audit records by, *Vol 1*, 4-8
- Security objects
  - classes, *Vol 1*, 1-3
- Security operators
  - sending alarm messages to audit log files, *Vol 1*, 1-8
  - sending alarm messages to terminals, *Vol 1*, 1-6
- SECURITY\_POLICY system parameter, *Vol 2*, C-56
- SELECT command
  - in Show Cluster utility, *Vol 2*, 5-68
- /SELECT qualifier
  - in Backup utility, *Vol 1*, 7-85
- Service node, *Vol 1*, 13-3
  - controlling access to, *Vol 1*, 13-35, 13-40
  - deleting a port, *Vol 1*, 13-23
  - modifying characteristics of, *Vol 1*, 13-33, 13-47
  - naming, *Vol 1*, 13-44
- Service node (cont'd)
  - service announcements, *Vol 1*, 13-36, 13-37
- Services
  - application, *Vol 1*, 13-11, 13-15, 13-17, 13-43, 13-47
  - creating with LATCP, *Vol 1*, 13-15
  - database, *Vol 1*, 13-37, 13-40
  - deleting with LATCP, *Vol 1*, 13-25
  - displaying characteristics of, *Vol 1*, 13-67
  - displaying counters, *Vol 1*, 13-66
  - displaying information using LATCP, *Vol 1*, 13-66
  - general timesharing, *Vol 1*, 13-17
  - identifier, *Vol 1*, 13-15, 13-36, 13-47
  - limiting size of node database, *Vol 1*, 13-37, 13-40
  - modifying characteristics of, *Vol 1*, 13-47
  - naming with LATCP, *Vol 1*, 13-15, 13-31
  - rating, *Vol 1*, 13-16, 13-48, 13-59, 13-67
  - responder, *Vol 1*, 13-38
  - target, *Vol 1*, 13-63
- Session limit, *Vol 1*, 13-38
- Sessions
  - terminating, *Vol 1*, 3-7
- SET/OUTPUT command
  - in SYSGEN, *Vol 2*, 6-21
- SET/STARTUP command
  - in SYSGEN, *Vol 2*, 6-22
- SET ACP command
  - in LANCP utility, *Vol 1*, 12-26
- SET AUDIT command
  - recording alarm messages, *Vol 1*, 1-8
- SET AUTO\_POSITIONING command
  - in Show Cluster utility, *Vol 2*, 5-69
- SET command
  - (Field), *Vol 2*, 5-70
  - in Point-to-Point Protocol utility, *Vol 2*, 2-11
  - in SYSGEN, *Vol 2*, 6-20
- SET DEFAULT command, *Vol 2*, 1-82
  - in Monitor utility, *Vol 2*, 1-82
- SET DEVICE command
  - in LANCP utility, *Vol 1*, 12-27
- SET ENVIRONMENT command
  - in SYSMAN, *Vol 2*, 7-120
- SET FUNCTION command, *Vol 2*, B-3
  - in Show Cluster utility, *Vol 2*, 5-71
- SET INTERVAL command
  - in Show Cluster utility, *Vol 2*, 5-72
- SET LINK command
  - in LATCP, *Vol 1*, 13-31
- SET LOG command
  - in UCM, *Vol 2*, 8-21
- SET NODE command
  - in LANCP utility, *Vol 1*, 12-37
  - in LATCP, *Vol 1*, 13-33
- SETPARAMS.DAT file, *Vol 1*, 6-9, 6-10

SETPARAMS phase of AUTOGEN, *Vol 1*, 6–10  
 SET PORT command  
     in LATCP, *Vol 1*, 13–43  
 SET PROFILE command  
     in SYSMAN, *Vol 2*, 7–122  
 SET QUORUM command  
     avoiding use of /CLUSTER with SYSMAN DO  
     command, *Vol 2*, 7–54  
 SET SCREEN command  
     in Show Cluster utility, *Vol 2*, 5–73  
 SET SECURITY/EDIT command, *Vol 1*, 1–4, 1–18  
 SET SECURITY command  
     /CLASS qualifier, *Vol 1*, 1–4  
     specifying object class, *Vol 1*, 1–4  
 SET SERVICE command  
     in LATCP, *Vol 1*, 13–47  
 SET TIMEOUT command  
     in SYSMAN, *Vol 2*, 7–123  
 SETTIME system parameter, *Vol 2*, C–56  
 Setting page, swap, and dump file sizes, *Vol 1*,  
     6–3  
 SHADOWING system parameter, *Vol 2*, C–57  
 /SHADOW qualifier  
     in Analyze/Disk\_Structure utility, *Vol 1*, 3–17  
 Shadow sets  
     detecting errors, *Vol 1*, 3–18  
     examining, *Vol 1*, 3–17  
     maximum number of, *Vol 2*, C–58  
 SHADOW\_D1-D5 system parameter, *Vol 2*, C–56  
 SHADOW\_ENABLE system parameter, *Vol 2*,  
     C–56  
 SHADOW\_HBMM\_RTC system parameter, *Vol 2*,  
     C–57  
 SHADOW\_MAX\_COPY system parameter, *Vol 2*,  
     C–57  
 SHADOW\_MAX\_UNIT system parameter, *Vol 2*,  
     C–58  
 SHADOW\_MBR\_TMO system parameter, *Vol 2*,  
     C–58  
 SHADOW\_PSM\_DLY system parameter, *Vol 2*,  
     C–57  
 SHADOW\_REC\_DLY system parameter, *Vol 2*,  
     C–58  
 SHADOW\_SYS\_DISK system parameter, *Vol 2*,  
     C–59  
 SHADOW\_SYS\_TMO system parameter, *Vol 2*,  
     C–59  
 SHADOW\_SYS\_UNIT system parameter, *Vol 2*,  
     C–60  
 SHADOW\_SYS\_WAIT system parameter, *Vol 2*,  
     C–60  
 SHARE/INITIALIZE command  
     in SYSGEN, *Vol 2*, 6–24  
 SHARE command  
     in SYSGEN, *Vol 2*, 6–23  
 SHOW/ADAPTER command  
     in SYSGEN, *Vol 2*, 6–31  
 SHOW/BUS command  
     in SYSGEN, *Vol 2*, 6–32  
 SHOW/CONFIGURATION command  
     in SYSGEN, *Vol 2*, 6–33  
 SHOW/DEVICE command  
     in SYSGEN, *Vol 2*, 6–35  
 SHOW/DRIVER command  
     in SYSGEN, *Vol 2*, 6–36  
 SHOW/IDENTIFIER command  
     in Authorize utility, *Vol 1*, 5–91  
 SHOW/PROXY command  
     in Authorize utility, *Vol 1*, 5–93  
 SHOW/RIGHTS command  
     in Authorize utility, *Vol 1*, 5–95  
 SHOW/STARTUP command, *Vol 2*, 6–38  
 SHOW/TURBOCHANNEL command  
     in SYSGEN, *Vol 2*, 6–39  
 SHOW/UNIBUS command  
     in SYSGEN, *Vol 2*, 6–40  
 SHOW ACP command  
     in LANCP utility, *Vol 1*, 12–41  
 SHOW CLUSTER command  
     initialization file, *Vol 2*, 5–13  
     in Show Cluster utility, *Vol 2*, 5–14  
     keypad commands, *Vol 2*, 5–13  
 Show Cluster utility (SHOW CLUSTER)  
     See also Reports, SHOW CLUSTER  
     activation time, *Vol 2*, 5–15, 5–17, 5–18, 5–72  
     classes of data, *Vol 2*, 5–3  
     commands, *Vol 2*, 5–19  
     continuous mode, *Vol 2*, 5–14, 5–16  
     controlling displays, *Vol 2*, 5–12  
     data fields, *Vol 2*, 5–3  
     defining arrow keys, *Vol 2*, B–3  
     defining keypad keys, *Vol 2*, 5–40, B–3  
     deselecting window, *Vol 2*, 5–43  
     directing output, *Vol 2*, 5–14  
     Execute Procedure (@) command, *Vol 2*, 5–22  
     exiting, *Vol 2*, 5–14, 5–44  
     invoking, *Vol 2*, 5–14  
     keypad, *Vol 2*, B–1  
     moving display window, *Vol 2*, 5–47, 5–68  
     /OUTPUT qualifier, *Vol 2*, 5–14  
     panning display window, *Vol 2*, 5–49  
     qualifiers, *Vol 2*, 5–14  
     refreshing the display, *Vol 2*, 5–51  
     removing display information, *Vol 2*, 5–52 to  
     5–62  
     resetting display information, *Vol 2*, 5–46  
     sample display, *Vol 2*, 5–12  
     scrolling display window, *Vol 2*, 5–49, 5–68  
     selecting display fields, *Vol 2*, 5–33  
         CIRCUITS class, *Vol 2*, 5–23  
         CLUSTER class, *Vol 2*, 5–25  
         CONNECTIONS class, *Vol 2*, 5–27  
         COUNTERS class, *Vol 2*, 5–29  
         CREDITS class, *Vol 2*, 5–31  
         ERRORS class, *Vol 2*, 5–32



Show Cluster utility (SHOW CLUSTER)  
 selecting display fields (cont'd)  
   LOCAL\_PORTS class, *Vol 2, 5-35*  
   MEMBERS class, *Vol 2, 5-37*  
   SYSTEMS class, *Vol 2, 5-38*  
 selecting display window, *Vol 2, 5-68*

SHOW command  
 in Audit Analysis utility, *Vol 1, 4-38*  
 in Authorize utility, *Vol 1, 5-86*  
 in CLUE, *Vol 1, 8-11*  
 in Point-to-Point Protocol utility, *Vol 2, 2-15*  
 in SYSGEN, *Vol 2, 6-26*

SHOW CONFIGURATION command  
 in LANCP utility, *Vol 1, 12-42*

SHOW DEFAULT command  
 in Monitor utility, *Vol 2, 1-83*

SHOW DEVICE command  
 in LANCP utility, *Vol 1, 12-43*  
 in UCM, *Vol 2, 8-22*

SHOW DLL command  
 in LANCP utility, *Vol 1, 12-47*

SHOW ENVIRONMENT command, *Vol 2, 7-120, 7-124*  
 in SYSMAN, *Vol 2, 7-124*

SHOW HISTORY command  
 in POLYCENTER Software Installation utility, *Vol 2, 3-64*

SHOW KEY command  
 in SYSMAN, *Vol 2, 7-125*

SHOW LINK command  
 in LATCP, *Vol 1, 13-50*

SHOW LOG command  
 in LANCP utility, *Vol 1, 12-48*  
 in LMCP, *Vol 1, 14-19*

SHOW NODE command  
 in LANCP utility, *Vol 1, 12-50*  
 in LATCP, *Vol 1, 13-55*

SHOW OBJECT command  
 in POLYCENTER Software Installation utility, *Vol 2, 3-66*

SHOW PORT command  
 in LATCP, *Vol 1, 13-61*

SHOW PRODUCT command  
 in POLYCENTER Software Installation utility, *Vol 2, 3-69*

SHOW PROFILE command  
 in SYSMAN, *Vol 2, 7-126*

SHOW QUEUE\_ENTRY command  
 in LATCP, *Vol 1, 13-64*

SHOW RECOVERY\_DATA command  
 in POLYCENTER Software Installation utility, *Vol 2, 3-72*

SHOW SERVICE command  
 in LATCP, *Vol 1, 13-66*

SHOW TIMEOUT command  
 in SYSMAN, *Vol 2, 7-127*

SHOW UTILITY command  
 in POLYCENTER Software Installation utility, *Vol 2, 3-74*

SHOW\_CLUSTER\$INIT:SHOW\_CLUSTER.INI  
 initialization file, *Vol 2, 5-13*

SHOW\_CLUSTER\$INIT logical name, *Vol 2, 5-22*

SHUTDOWN NODE command  
 in SYSMAN, *Vol 2, 7-128*

SHUTDOWN phase of AUTOGEN, *Vol 1, 6-10*

/SINCE qualifier  
 in Backup utility, *Vol 1, 7-86*

SMCI\_FLAGS system parameter, *Vol 2, C-60*

SMCI\_PORTS system parameter, *Vol 2, C-60*

SMP\_CPUSH system parameter, *Vol 2, C-61*

SMP\_CPUS system parameter, *Vol 2, C-61*

SMP\_LNGSPINWAIT system parameter, *Vol 2, C-61*

SMP\_SANITY\_CNT system parameter, *Vol 2, C-61*

SMP\_SPINWAIT system parameter, *Vol 2, C-61*

SMP\_TICK\_CNT system parameter, *Vol 2, C-61*

Software  
 listing  
   POLYCENTER Software Installation utility, *Vol 2, 3-42*

Software products  
 files of  
   deleting recovery data, *Vol 2, 3-15*  
   extracting, *Vol 2, 3-17*  
   removing, *Vol 2, 3-61*

SPAWN command  
 and ATTACH command, *Vol 2, 7-17*  
 in LANCP utility, *Vol 1, 12-52*  
 in LATCP, *Vol 1, 13-69*  
 in SYSMAN, *Vol 2, 7-132*

SPTREQ system parameter, *Vol 2, C-62*

SSINHIBIT system parameter, *Vol 2, C-62*

Stage checks, *Vol 1, D-1*

Standalone BACKUP, *Vol 1, 7-11*

Start phase of AUTOGEN, *Vol 1, 6-8*  
 default, *Vol 1, 6-13*

Startup  
 logging, *Vol 2, 7-145*  
 phase, *Vol 2, 7-136*  
 status, *Vol 2, 7-145*

STARTUP\$STARTUP\_LAYERED logical name, *Vol 2, 7-144*

STARTUP\$STARTUP\_VMS logical name, *Vol 2, 7-144*

STARTUP ADD command  
 in SYSMAN, *Vol 2, 7-136*

Startup command procedure, *Vol 2, 6-22, 6-42, 6-43*  
 setting the name, *Vol 2, 7-97*  
 site-independent file name, *Vol 2, 6-38*

Startup database  
 establishing, *Vol 2, 7-144*  
 files

Startup database files (cont'd)

- adding, *Vol 2, 7-136*
- deleting, *Vol 2, 7-143*
- disabling, *Vol 2, 7-138*
- enabling, *Vol 2, 7-140*
- modifying, *Vol 2, 7-141*
- setting the default, *Vol 2, 7-144*

STARTUP DISABLE command

- in SYSMAN, *Vol 2, 7-138*

STARTUP ENABLE command

- in SYSMAN, *Vol 2, 7-140*

STARTUP MODIFY command

- in SYSMAN, *Vol 2, 7-141*

STARTUP REMOVE command

- in SYSMAN, *Vol 2, 7-143*

STARTUP SET DATABASE command

- in SYSMAN, *Vol 2, 7-144*

STARTUP SET OPTIONS command

- in SYSMAN, *Vol 2, 7-145*

STARTUP SHOW command

- in SYSMAN, *Vol 2, 7-147*

STARTUP\_Pn system parameters, *Vol 2, C-62*

- in SYSMAN, *Vol 2, 7-136*

STATES class record

- in Monitor utility, *Vol 2, A-37*

Static service rating, *Vol 1, 13-16, 13-48*

- displaying, *Vol 1, 13-59, 13-67*

Statistical information

- about volume under verification, *Vol 1, 3-19*

/STATISTICS qualifier

- in Analyze/Disk\_Structure utility, *Vol 1, 3-19*

STATS.DAT file

- created by ANALYZE/DISK\_STRUCTURE/STATISTICS, *Vol 1, 3-19*

Subprocesses

- See also SPAWN command
- creating
  - from LATCP, *Vol 1, 13-5, 13-69*
  - with SPAWN command, *Vol 2, 7-132*
  - with the LANCP SPAWN command, *Vol 1, 12-52*
- switching control of input stream to, *Vol 2, 7-17*

Subsystem ACEs, *Vol 1, 1-16*

- example, *Vol 1, 1-16*
- specifying identifier characteristics, *Vol 1, 1-17*
- specifying identifiers, *Vol 1, 1-16*

Subsystem attribute, *Vol 1, 5-20, 5-51, 5-73*

Subsystems

- controlling access to, *Vol 1, 1-16*

/SUMMARY qualifier

- in Monitor utility, *Vol 2, 1-16*

SWAPFILE.SYS file

- See Swap files

Swap files

- changing size, *Vol 1, 6-3*
- displaying the size calculated by AUTOGEN, *Vol 1, 6-9*
- generating a new file sized by AUTOGEN, *Vol 1, 6-10*

SWPALLOCINC system parameter, *Vol 2, C-63*

SWPFAIL system parameter, *Vol 2, C-63*

SWPFILCNT system parameter, *Vol 2, C-63*

SWPOUTPGCNT system parameter, *Vol 2, C-63*

SWPRATE system parameter, *Vol 2, C-64*

SWP\_PRIO system parameter, *Vol 2, C-63*

Symbol table

- command interpreter, *Vol 2, C-9*

Syntax

- of BACKUP commands, *Vol 1, 7-5*

SYS\$SYSTEM:STARTUP.COM command

- procedure, *Vol 2, D-6*

SYSALF.DAT database in SYSMAN, *Vol 2, 7-14*

SYSBOOT facility

- using during bootstrap operations, *Vol 2, 6-3*

SYSDUMP.DMP file

- See Dump files

SYSGEN

- See System Generation utility (SYSGEN)

SYSLOST.DIR file, *Vol 1, 3-5*

SYSMAN

- See System Management utility (SYSMAN)

SYSMANINI.INI file

- changing file specification, *Vol 2, 7-4*

SYSMWCNT system parameter, *Vol 2, C-64*

SYSPFC system parameter, *Vol 2, C-64*

SYSSER\_LOGGING system parameter, *Vol 2, C-64*

SYSTEM class record

- in Monitor utility, *Vol 2, A-39*

System Dump Priority registry

- adding records, *Vol 2, 7-57*
- description, *Vol 2, 7-57*
- displaying an in-memory copy of the file, *Vol 2, 7-65*
- interaction with BUGCHECK, *Vol 2, 7-57*
- listing records, *Vol 2, 7-60*
- loading contents of file into memory, *Vol 2, 7-61*
- modifying an entry in the file, *Vol 2, 7-62*
- removing an entry in the file, *Vol 2, 7-64*
- unloading an in-memory copy of the file, *Vol 2, 7-66*

System Generation utility (SYSGEN)

- assign parameter values, *Vol 2, 6-20*
- commands
  - AUTOCONFIGURE, *Vol 2, 6-5*
  - CONFIGURE, *Vol 2, 6-8*
  - CONNECT/ADAPTER, *Vol 2, 6-10*
  - CREATE, *Vol 2, 6-13*
  - DEINSTALL, *Vol 2, 6-14*

## System Generation utility (SYSGEN)

### commands (cont'd)

- DISABLE, *Vol 2*, 6–15
- ENABLE, *Vol 2*, 6–16
- INSTALL, *Vol 2*, 6–17
- LOAD, *Vol 2*, 6–18
- RELOAD, *Vol 2*, 6–19
- SET, *Vol 2*, 6–20
- SET/OUTPUT, *Vol 2*, 6–21
- SET/STARTUP, *Vol 2*, 6–22
- SHARE, *Vol 2*, 6–23
- SHARE/INITIALIZE, *Vol 2*, 6–24
- SHOW, *Vol 2*, 6–26
- SHOW/ADAPTER, *Vol 2*, 6–31
- SHOW/BUS, *Vol 2*, 6–32
- SHOW/CONFIGURATION, *Vol 2*, 6–33
- SHOW/DEVICE, *Vol 2*, 6–35
- SHOW/DRIVER, *Vol 2*, 6–36
- SHOW/STARTUP, *Vol 2*, 6–38
- SHOW/TURBOCHANNEL, *Vol 2*, 6–39
- SHOW/UNIBUS, *Vol 2*, 6–40
- TERMINAL, *Vol 2*, 6–42
- USE, *Vol 2*, 6–43
- WRITE, *Vol 2*, 6–45

directing output, *Vol 2*, 6–3, 6–33

### parameters

See also System parameters

showing parameter values, *Vol 2*, 6–26

specifying file output, *Vol 2*, 6–8, 6–21

work area, *Vol 2*, 6–20, 6–43, 6–45

## System loadable image

adding, *Vol 2*, 7–149

removing, *Vol 2*, 7–151

## System management

clearing counters, *Vol 1*, 12–9

creating disk quota files, *Vol 2*, 7–40

defining target system, *Vol 2*, 7–120

deleting device information, *Vol 1*, 12–8, 12–24

deleting node information, *Vol 1*, 12–10, 12–25

displaying LAN device parameters, *Vol 1*,  
12–20, 12–43

displaying node information, *Vol 1*, 12–22,  
12–50

displaying OPCOM messages, *Vol 1*, 12–26

displaying status and counters, *Vol 1*, 12–22,  
12–47, 12–50

displaying target environment, *Vol 2*, 7–124

displaying user profile, *Vol 2*, 7–126

displaying wait time for node response, *Vol 2*,  
7–127

invoking the LANCP utility, *Vol 1*, 12–3

LANCP SPAWN command, *Vol 1*, 12–52

limiting response time, *Vol 2*, 7–123

MOP console carrier, *Vol 1*, 12–11

MOP trigger boot, *Vol 1*, 12–53

setting device information, *Vol 1*, 12–15

setting wait time for node response, *Vol 2*,  
7–123

## System management (cont'd)

### startup procedures

See Startup database

SYSMAN tasks, *Vol 2*, 7–3

## System management environment

See Management environment

## System Management utility (SYSMAN)

### commands

ALF ADD, *Vol 2*, 7–12

ALF REMOVE, *Vol 2*, 7–14

ALF SHOW, *Vol 2*, 7–16

ATTACH, *Vol 2*, 7–17

CLASS\_SCHEDULE, *Vol 2*, 7–19

CONFIGURATION SET CLUSTER\_AUTH,  
*Vol 2*, 7–30

CONFIGURATION SET TIME, *Vol 2*, 7–32

CONFIGURATION SHOW CLUSTER\_  
AUTHORIZATION, *Vol 2*, 7–34

CONFIGURATION SHOW TIME, *Vol 2*,  
7–35

DEFINE/KEY, *Vol 2*, 7–36

DISKQUOTA ADD, *Vol 2*, 7–38

DISKQUOTA CREATE, *Vol 2*, 7–40

DISKQUOTA DELETE, *Vol 2*, 7–42

DISKQUOTA DISABLE, *Vol 2*, 7–43

DISKQUOTA ENABLE, *Vol 2*, 7–45

DISKQUOTA MODIFY, *Vol 2*, 7–47

DISKQUOTA REBUILD, *Vol 2*, 7–49

DISKQUOTA REMOVE, *Vol 2*, 7–50

DISKQUOTA SHOW, *Vol 2*, 7–52

DO, *Vol 2*, 7–53

DUMP\_PRIORITY ADD, *Vol 2*, 7–57

DUMP\_PRIORITY LIST, *Vol 2*, 7–60

DUMP\_PRIORITY LOAD, *Vol 2*, 7–61

DUMP\_PRIORITY MODIFY, *Vol 2*, 7–62

DUMP\_PRIORITY REMOVE, *Vol 2*, 7–64

DUMP\_PRIORITY SHOW, *Vol 2*, 7–65

DUMP\_PRIORITY UNLOAD, *Vol 2*, 7–66

EXIT, *Vol 2*, 7–67

HELP, *Vol 2*, 7–68

IO AUTOCONFIGURE, *Vol 2*, 7–69

IO CONNECT, *Vol 2*, 7–71

IO CREATE\_WWID, *Vol 2*, 7–75

IO FIND\_WWID, *Vol 2*, 7–76

IO LIST\_WWID, *Vol 2*, 7–79

IO LOAD, *Vol 2*, 7–80

IO REBUILD, *Vol 2*, 7–81

IO REPLACE\_WWID, *Vol 2*, 7–82

IO SCSI\_PATH\_VERIFY, *Vol 2*, 7–84

IO SET EXCLUDE, *Vol 2*, 7–85

IO SET PREFIX, *Vol 2*, 7–86

IO SHOW BUS, *Vol 2*, 7–87

IO SHOW DEVICE, *Vol 2*, 7–89

IO SHOW EXCLUDE, *Vol 2*, 7–91

IO SHOW PREFIX, *Vol 2*, 7–92

LICENSE LOAD, *Vol 2*, 7–93

LICENSE UNLOAD, *Vol 2*, 7–94

## System Management utility (SYSMAN)

### commands (cont'd)

- PARAMETERS DISABLE CHECKS, *Vol 2*, 7-95
- PARAMETERS ENABLE CHECKS, *Vol 2*, 7-96
- PARAMETERS SET, *Vol 2*, 7-97
- PARAMETERS SHOW, *Vol 2*, 7-98
- PARAMETERS USE, *Vol 2*, 7-103
- PARAMETERS WRITE, *Vol 2*, 7-105
- RESERVED\_MEMORY ADD, *Vol 2*, 7-106
- RESERVED\_MEMORY EXTEND, *Vol 2*, 7-110
- RESERVED\_MEMORY FREE, *Vol 2*, 7-111
- RESERVED\_MEMORY LIST, *Vol 2*, 7-113
- RESERVED\_MEMORY MODIFY, *Vol 2*, 7-114
- RESERVED\_MEMORY REMOVE, *Vol 2*, 7-117
- RESERVED\_MEMORY SHOW, *Vol 2*, 7-118
- SET ENVIRONMENT, *Vol 2*, 7-120
- SET PROFILE, *Vol 2*, 7-122
- SET TIMEOUT, *Vol 2*, 7-123
- SHOW ENVIRONMENT, *Vol 2*, 7-124
- SHOW KEY, *Vol 2*, 7-125
- SHOW PROFILE, *Vol 2*, 7-126
- SHOW TIMEOUT, *Vol 2*, 7-127
- SHUTDOWN NODE, *Vol 2*, 7-128
- SPAWN, *Vol 2*, 7-132
- STARTUP ADD, *Vol 2*, 7-136
- STARTUP DISABLE, *Vol 2*, 7-138
- STARTUP ENABLE, *Vol 2*, 7-140
- STARTUP MODIFY, *Vol 2*, 7-141
- STARTUP REMOVE, *Vol 2*, 7-143
- STARTUP SET DATABASE, *Vol 2*, 7-144
- STARTUP SET OPTIONS, *Vol 2*, 7-145
- STARTUP SHOW, *Vol 2*, 7-147
- SYS\_LOADABLE ADD, *Vol 2*, 7-149
- SYS\_LOADABLE REMOVE, *Vol 2*, 7-151

common tasks, *Vol 2*, 7-3

executing a command procedure, *Vol 2*, 7-10

executing SYSMAN commands, *Vol 2*, 7-4

exiting from, *Vol 2*, 7-4

initialization file, *Vol 2*, 7-4

invoking, *Vol 2*, 7-4

overview, *Vol 2*, 7-3

privilege requirements, *Vol 2*, 7-4

restrictions, *Vol 2*, 7-4

rights identifiers, *Vol 2*, 7-5

### System overview statistics

in Monitor utility, *Vol 2*, 1-70

### System parameters

See also AUTOGEN.COM command procedure

ACP, *Vol 2*, C-3 to C-7

affected by AUTOGEN feedback, *Vol 1*, 6-7

categories, *Vol 2*, C-1

### System parameters (cont'd)

#### changing

recommended method, *Vol 1*, 6-3

changing during installation, *Vol 1*, 6-3

checking, *Vol 2*, 7-96

checking AUTOGEN's settings, *Vol 1*, 6-7

default parameter file, *Vol 1*, 6-8, 6-10

disabling checking, *Vol 2*, 7-95

displaying, *Vol 2*, 7-98

device configuration, *Vol 2*, 6-33

device driver addresses, *Vol 2*, 6-36

I/O subsystems, *Vol 2*, 6-35

parameter values, *Vol 2*, 6-26

startup procedure, *Vol 2*, 6-38

UNIBUS I/O addresses, *Vol 2*, 6-40

values, *Vol 2*, 6-26

#### file

See Parameter files

modifying, *Vol 2*, 7-105

recommended method, *Vol 1*, 6-3

setting, *Vol 2*, 7-97

parameter addresses, *Vol 2*, 6-20

parameter values, *Vol 2*, 6-3, 6-20, 6-26, 6-43, 6-45

recommended method, *Vol 1*, 6-3

sharing multiport memory, *Vol 2*, 6-23

showing, *Vol 2*, 6-26

specifying startup procedure, *Vol 2*, 6-22, 6-43, 6-45

terminal behavior, *Vol 2*, C-67 to C-71

VIRTUALPAGECNT, *Vol 1*, 3-3, 7-5

VMS, *Vol 2*, C-77

work area, *Vol 2*, 7-103, 7-105

writing contents of, *Vol 2*, 7-105

writing to file, *Vol 2*, 6-45

System passwords, *Vol 1*, 5-77

### SYSTEMS class

adding or restoring, *Vol 2*, 5-38

fields belonging to, *Vol 2*, 5-11

removing, *Vol 2*, 5-62

### Systems Communications Services

See SCS

### System shutdown

to install changed system parameters, *Vol 1*, 6-10

### System time

setting, *Vol 2*, 7-32

SYSTEM\_CHECK system parameter, *Vol 2*, C-64

SYSUAF logical name

defining, *Vol 1*, 5-3

SYSUAFs (system user authorization files)

creating, *Vol 1*, 5-3

default directory entry, *Vol 1*, 5-17

default values, *Vol 1*, 5-17

displaying records in, *Vol 1*, 5-86

modifying, *Vol 1*, 5-3

recreating, *Vol 1*, 5-4

SYSUAFs (system user authorization files) (cont'd)  
renaming records, *Vol 1*, 5–81  
SYS\_LOADABLE ADD command  
in SYSMAN, *Vol 2*, 7–149  
SYS\_LOADABLE REMOVE command  
in SYSMAN, *Vol 2*, 7–151

## T

---

Tape drive  
using /MEDIA\_FORMAT qualifier, *Vol 1*, 7–67  
TAPE\_ALLOCLASS system parameter, *Vol 2*,  
C–65  
/TAPE\_EXPIRATION qualifier  
in Backup utility, *Vol 1*, 7–88  
TAPE\_MVTIMEOUT system parameter, *Vol 2*,  
C–65  
Target node name, *Vol 1*, 13–63  
Target port name, *Vol 1*, 13–63  
Target service name, *Vol 1*, 13–63  
TBSKIPWSL system parameter, *Vol 2*, C–65  
Telnet protocol  
Break-in evasion, *Vol 2*, C–24  
TERMINAL command  
in SYSGEN, *Vol 2*, 6–42  
Terminals  
modifying control-key echo strings, *Vol 2*, 6–42  
sending alarm messages to, *Vol 1*, 1–6  
Terminating a session, *Vol 1*, 3–7  
TESTFILES phase of AUTOGEN, *Vol 1*, 6–9  
Time  
See also SYSMAN command CONFIGURATION  
SHOW TIME  
cluster  
displaying, *Vol 2*, 7–35  
extracting records by, *Vol 1*, 4–5  
setting system, *Vol 2*, 7–32  
updating in a cluster, *Vol 2*, 7–33  
Timeout period  
displaying, *Vol 2*, 7–127  
setting, *Vol 2*, 7–123  
TIMEPROMPTWAIT system parameter, *Vol 2*,  
C–66  
TIMER class record  
in Monitor utility, *Vol 2*, A–40  
Timer queue entries (TQEs)  
monitoring, *Vol 2*, A–40  
Timer Queue Entry (TQE) statistics  
in Monitor utility, *Vol 2*, 1–74  
Timers  
for circuits, *Vol 1*, 13–33  
for multicast message in LAT environment, *Vol*  
*1*, 13–37  
setting in a LAT environment, *Vol 1*, 13–36  
TIME\_CONTROL system parameter, *Vol 2*, C–65  
TIMVCFAIL system parameter, *Vol 2*, C–66

TMSCP\_LOAD system parameter, *Vol 2*, C–66  
TMSCP\_SERVE\_ALL system parameter, *Vol 2*,  
C–67  
TRANSACTION class record  
in Monitor utility, *Vol 2*, A–41  
Transaction logs  
used by DECdtm, *Vol 1*, 14–1  
Transactions  
monitoring, *Vol 2*, 1–76  
TRANSLATE command  
in Install utility, *Vol 1*, 11–3  
Trigger boot  
LANCP MOP downline load service, *Vol 1*,  
12–53  
TRIGGER NODE command  
in LANCP utility, *Vol 1*, 12–53  
TR numbers  
display on Alpha and I64 systems, *Vol 2*, 7–87  
/TRUNCATE qualifier  
in Backup utility, *Vol 1*, 7–89  
TTY system parameters  
See also System parameters  
terminal behavior  
TTY\_ALTALARM, *Vol 2*, C–67  
TTY\_ALTYPAHD, *Vol 2*, C–68  
TTY\_AUTOCHAR, *Vol 2*, C–68  
TTY\_BUF, *Vol 2*, C–68  
TTY\_CLASSNAME, *Vol 2*, C–68  
TTY\_DEFCHAR, *Vol 2*, C–68  
TTY\_DEFCHAR2, *Vol 2*, C–69  
TTY\_DEFCHAR3, *Vol 2*, C–70  
TTY\_DEFPORT, *Vol 2*, C–70  
TTY\_DIALTYPE, *Vol 2*, C–70  
TTY\_DMASIZE, *Vol 2*, C–71  
TTY\_PARITY, *Vol 2*, C–71  
TTY\_RSPEED, *Vol 2*, C–71  
TTY\_SCANDELTA, *Vol 2*, C–71  
TTY\_SILOTIME, *Vol 2*, C–71  
TTY\_SPEED, *Vol 2*, C–71  
TTY\_TIMEOUT, *Vol 2*, C–71  
TTY\_TYPAHDSZ, *Vol 2*, C–71  
TURBOchannel  
device on, *Vol 2*, 6–39

## U

---

UAFALTERNATE system parameter, *Vol 2*, C–71  
UAFs (user authorization files)  
See SYSUAFs  
UCM  
See USB Configuration Manager  
UIC-based protection  
See Protection codes  
UICs (user identification codes)  
for disk quota file, *Vol 2*, 7–38  
identifiers, *Vol 1*, 1–14

- UNDO PATCH command
  - in POLYCENTER Software Installation utility, *Vol 2*, 3–75
- UNIBUS I/O addresses, *Vol 2*, 6–40
- Unloading licenses, *Vol 2*, 7–94
- UPDATE DEVICE command
  - in LANCP utility, *Vol 1*, 12–54
- Usage count
  - updating, *Vol 2*, 7–49
- /USAGE qualifier, *Vol 1*, 3–6, 3–21
  - in Analyze/Disk\_Structure utility, *Vol 1*, 3–21
- USB Configuration Manager (UCM)
  - description, *Vol 2*, 8–3
- USB Configuration Manager (UCM) utility
  - ADD DEVICE command, *Vol 2*, 8–12
  - DELETE DEVICE command, *Vol 2*, 8–14
  - EXIT command, *Vol 2*, 8–15
  - HELP command, *Vol 2*, 8–16
  - MODIFY DEVICE command, *Vol 2*, 8–17
  - RELOAD command, *Vol 2*, 8–19
  - RESTART command, *Vol 2*, 8–20
  - SET LOG command, *Vol 2*, 8–21
  - SHOW DEVICE command, *Vol 2*, 8–22
- USE command
  - in SYSGEN, *Vol 2*, 6–43
- User (security category)
  - controlling types of access, *Vol 1*, 1–14
- User accounts
  - altering, *Vol 1*, 5–61
  - creating, *Vol 1*, 5–7
  - deleting, *Vol 1*, 5–78
- User directories
  - creating for a new account, *Vol 1*, 5–19
- User identification codes
  - See UICs
- User names
  - as identifiers, *Vol 1*, 1–14
  - identifiers added with, *Vol 1*, 5–19
  - specifying with wildcards, *Vol 1*, 5–5
- USER system parameters
  - USER3, *Vol 2*, C–72
  - USER4, *Vol 2*, C–72
  - USERD1, *Vol 2*, C–72
  - USERD2, *Vol 2*, C–72

## V

- VAXCLUSTER system parameter, *Vol 2*, C–72
- VAXVMSSYS.OLD file, *Vol 1*, 6–10
- VAXVMSSYS.PAR file, *Vol 1*, 6–10
- VBN\_CACHE\_S system parameter, *Vol 2*, C–72
- VBS class record
  - in Monitor utility, *Vol 2*, A–42
- VBSS\_ENABLE2 system parameter, *Vol 2*, C–73
- VBSS\_ENABLE system parameter, *Vol 2*, C–73
- VCC\_FLAGS system parameter, *Vol 2*, C–74
- VCC\_MAXSIZE system parameter, *Vol 2*, C–74
- VCC\_MAX\_CACHE system parameter, *Vol 2*, C–74
- VCC\_MAX\_IO\_SIZE system parameter, *Vol 2*, C–75
- VCC\_MAX\_LOCKS system parameter, *Vol 2*, C–75
- VCC\_MINSIZE system parameter, *Vol 2*, C–75
- VCC\_PTES system parameter, *Vol 2*, C–75
- VCC\_READAHEAD system parameter, *Vol 2*, C–76
- VCC\_WRITEBEHIND system parameter, *Vol 2*, C–76
- VCC\_WRITE\_DELAY system parameter, *Vol 2*, C–76
- VECTOR class record
  - in Monitor utility, *Vol 2*, A–44
- Vector CPU time
  - obtaining information regarding processor, *Vol 2*, A–44
  - obtaining information regarding system, *Vol 2*, A–44
- Vector processing
  - monitoring, *Vol 2*, 1–80
- Vectors
  - addresses, *Vol 2*, 6–8, 6–10
  - fixed space, *Vol 2*, D–1
  - floating space, *Vol 2*, D–1
- Verification
  - of disk volumes, *Vol 1*, 3–3
- /VERIFY qualifier
  - in Backup utility, *Vol 1*, 7–90
- VHPT\_SIZE system parameter, *Vol 2*, C–76
- /VIEWING\_TIME qualifier
  - in Monitor utility, *Vol 2*, 1–16
- Virtual memory
  - requirements for bitmaps, *Vol 1*, 3–3
- VIRTUALPAGECNT system parameter, *Vol 1*, 3–3, 7–5; *Vol 2*, C–76
- VMS\$RESERVED\_MEMORY.DATA
  - Reserved Memory Registry file, *Vol 2*, 7–108
- VMS\$SYSTEM\_IMAGES.IDX
  - adding an entry, *Vol 2*, 7–150
  - removing an entry, *Vol 2*, 7–151
- VMS\$VAXcluster
  - communications
    - receiving, *Vol 2*, C–10
- VMSIMAGES.DAT file, *Vol 1*, 6–9
- VMS system parameters
  - See System parameters
- VMS\_BCFG utility, *Vol 1*, 9–4
- VMS\_SET utility, *Vol 1*, 9–6
- VMS\_SHOW utility, *Vol 1*, 9–7
- /VOLUME qualifier
  - in Backup utility, *Vol 1*, 7–91

Volume quota file  
    See Disk quotas  
Volumes  
    repairing errors on a disk volume, *Vol 1*, 3–16  
Volume sets  
    image save operation, *Vol 1*, 7–51  
VOTES system parameter, *Vol 2*, C–77

## W

---

WBM\_MSG\_INT system parameter, *Vol 2*, C–77  
WBM\_MSG\_LOWER system parameter, *Vol 2*,  
    C–77  
WBM\_MSG\_UPPER system parameter, *Vol 2*,  
    C–78  
WBM\_OPCOM\_LVL system parameter, *Vol 2*,  
    C–78  
Wildcard characters, *Vol 1*, 7–8  
    restriction in ACL editor, *Vol 1*, 1–4  
    using to represent directories, *Vol 1*, 7–8  
Windows  
    compressing or expanding, *Vol 2*, 5–73  
    designating, *Vol 2*, 5–68  
    display SHOW CLUSTER information, *Vol 2*,  
        5–12  
    moving, *Vol 2*, 5–69  
    rearranging, *Vol 2*, 5–47  
    scrolling, *Vol 2*, 5–66  
    terminating selection, *Vol 2*, 5–43  
    viewing large display, *Vol 2*, 5–49  
WINDOW\_SYSTEM system parameter, *Vol 2*,  
    C–78  
WLKSYSDSK system parameter, *Vol 2*, C–79  
Work area  
    copying to parameter file, *Vol 2*, 7–105  
    loading parameters, *Vol 2*, 7–103  
WPRE\_SIZE system parameter, *Vol 2*, C–79  
WPTTE\_SIZE system parameter, *Vol 2*, C–79  
WRITABLESYS system parameter, *Vol 2*, C–79  
WRITE command  
    in Show Cluster utility, *Vol 2*, 5–74  
    in SYSGEN, *Vol 2*, 6–45  
WRITESYSPARAMS system parameter, *Vol 2*,  
    C–79  
WSDEC system parameter, *Vol 2*, C–79  
WSINC system parameter, *Vol 2*, C–79  
WSMAX system parameter, *Vol 2*, C–80  
WS\_OPA0 system parameter, *Vol 2*, C–80

## X

---

XA Gateway Control Program (XGCP)  
    commands  
        CREATE\_LOG, *Vol 2*, 9–4  
        EXIT, *Vol 2*, 9–5  
        START\_SERVER, *Vol 2*, 9–6  
        STOP\_SERVER, *Vol 2*, 9–7  
    invoking, *Vol 2*, 9–3

XA Gateway Control Program (XGCP) (cont'd)  
    overview, *Vol 2*, 9–3

### XGCP

    See XA Gateway Control Program (XGCP)  
XQPCTL2 system parameter, *Vol 2*, C–80  
XQPCTLD1 system parameter, *Vol 2*, C–80

## Z

---

ZERO COUNTERS command

    in LATCP, *Vol 1*, 13–70  
ZERO\_LIST\_HI system parameter, *Vol 2*, C–80

