VMS Software

System Management Utilities Reference Manual, Volume I: A–L

Operating System and Version: VSI OpenVMS Alpha Version 8.4-2L1 or higher
VSI OpenVMS IA-64 Version 8.4-1H1 or higher
VSI OpenVMS x86-64 Version 9.2-2 or higher;

VMS Software, Inc. (VSI)
Boston, Massachusetts, USA

System Management Utilities Reference Manual, Volume I: A–L

Legal Notice

Table of Contents

Preface

This document describes reference information for System Management utilities used with the OpenVMS Alpha, IA-64, and x86-64 operating systems.

1. About VSI

VMS Software, Inc. (VSI) is an independent software company licensed by Hewlett Packard Enterprise to develop and support the OpenVMS operating system.

2. Intended Audience

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

3. 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, "Manual Structure"* 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	COPY/RECORDABLE_MEDIA (CDDVD)
9	EFI Utilities for OpenVMS
10	Error Log Viewer (ELV)
11	InfoServer
12	Install (INSTALL)
13	LAN Control Program (LANCP)
14	LAT Control Program (LATCP)
15	Log Manager Control Program (LMCP)

4. Related Documents

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

- *VSI OpenVMS System Management Utilities Reference Manual, Volume II: M–Z* [https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-ii-m-z/]
- *VSI OpenVMS DCL Dictionary: A–M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/] and *VSI OpenVMS DCL Dictionary: N–Z* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-n-z/]
- *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/] and *VSI OpenVMS System Manager's Manual, Volume 2: Tuning, Monitoring, and Complex Systems* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-2-tuning-monitoring-and-complex-systems/]
- *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/]
- *VSI OpenVMS Programming Concepts Manual, Volume I* [https://docs.vmssoftware.com/vsi-openvms-programming-concepts-manual-volume-i/] and *VSI OpenVMS Programming Concepts Manual, Volume II* [https://docs.vmssoftware.com/vsi-openvms-programming-concepts-manual-volume-ii/]
- *VSI OpenVMS Record Management Services Reference Manual* [https://docs.vmssoftware.com/vsi-openvms-record-management-services-reference-manual/]
- *VSI OpenVMS System Services Reference Manual: A–GETUAI* [https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-a-getuai/] and *VSI OpenVMS System Services Reference Manual: GETUTC–Z* [https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-getutc-z/]
- *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/]
- *OpenVMS VAX Device Support Manual* (Archived)

5. VSI Encourages Your Comments

You may send comments or suggestions regarding this manual or any VSI document by sending electronic mail to the following Internet address: <docinfo@vmssoftware.com>. Users who have VSI OpenVMS support contracts through VSI can contact <support@vmssoftware.com> for help with this product.

6. OpenVMS Documentation

The full VSI OpenVMS documentation set can be found on the VMS Software Documentation webpage at https://docs.vmssoftware.com.

7. 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 Alpha and Integrity servers, 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:

Convention	Meaning
Ctrl/*x*	A sequence such as **Ctrl/*x*** indicates that you must hold down the key labeled **Ctrl** while you press another key or a pointing device button.
…	A horizontal ellipsis in examples indicates one of the following possibilities: • Additional optional arguments in a statement have been omitted. • The preceding item or items can be repeated one or more times. • Additional parameters, values, or other information can be entered.
⋮	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
()	In command format descriptions, parentheses indicate that you must enclose choices in parentheses if you specify 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. Do not type the vertical bars on the command line.
Bold type	Bold type represents the name of an argument, an attribute, or a reason. Bold type also represents the introduction of a new term.
Italic type	Italic type indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error *number*), in command lines (**/PRODUCER=*name***), and in command parameters in text (where *dd* represents the predefined code for the device type).
UPPERCASE TYPE	Uppercase type indicates the name of a routine, the name of a file, or the abbreviation for a system privilege.
`Monospace type`	Monospace type indicates code examples, command examples, and interactive screen displays. In text, this type also identifies URLs, UNIX commands and pathnames, PC-based commands and folders, and certain elements of the C programming language.
`Bold monospace type`	Bold monospace type indicates a DCL command or command qualifier.
-	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.
_$	An underscore and dollar sign at the beginning of a command line or code line indicates that the command or statement has been continued from the

Convention	Meaning
	previous line. The underscore and dollar sign will appear automatically when continuing a command from the previous line using a hyphen; do not type them on the command line.
Numbers	All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.

Chapter 1. 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, "ACE Formats"*.) 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.

Syntax

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
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, see the *VSI OpenVMS DCL Dictionary* and the *VSI 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 fora 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, "ACL Editor Keypad Commands"*). 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, "ACL Editor Keypad Editing Commands"*. For information about how to modify the ACL editor by modifying ACL section files, see *Appendix B, "Customizing the ACL Editor"*.

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, see the *VSI 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 *VSI 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 *VSI OpenVMS Programming Concepts Manual*.

Alarm ACE

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.

Syntax

```
(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 *VSI 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

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.

Syntax

```
(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. 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. For a listing of valid access types, see the *VSI OpenVMS Guide to System Security*. 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
```

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.

Description

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 *VSI OpenVMS Guide to System Security* for more information.

Syntax

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

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, see the *VSI OpenVMS Guide to System Security*.

Syntax

```
(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	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 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 *VSI 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.

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, see the *VSI OpenVMS Guide to System Security*.

Syntax

```
(IDENTIFIER=identifier[+identifier...] [,OPTIONS=attributes[+attributes...]] ,ACCE
```

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 *VSI OpenVMS Guide to System Security*. They include batch, network, interactive, local, dialup, and remote.

For more information, see the *VSI OpenVMS Guide to System Security*.

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. Note that an Identifier ACE with the Default attribute has no effect on access.
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 access types that are valid for the object class. Refer to the *VSI OpenVMS Guide to System Security* for a listing of valid access types.

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

Syntax

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

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: • 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.

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, see the *VSI 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.

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 no prompt). 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

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

Syntax

```
/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:

CAPABILITY	A system capability, such as the ability to process vector instructions. Currently, the only defined object name for the CAPABILITY class is VECTOR, 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.
COMMON_EVENT_CLUSTER	A common event flag cluster.
DEVICE	A device, such as a disk or tape drive.
FILE	A file or a directory file. This is the default.
GROUP_GLOBAL_SECTION	A group global section.
LOGICAL_NAME_TABLE	A logical name table.
QUEUE	A batch queue or a device (printer, server, or terminal) queue.
RESOURCE_DOMAIN	A resource domain.
SECURITY_CLASS	A security class.
SYSTEM_GLOBAL_SECTION	A system global section.
VOLUME	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.

Syntax

/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

/MODE — Specifies the use of prompting during the editing session.

Syntax

/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.

Example

$ **EDIT/ACL/MODE=NOPROMPT WEATHERTBL.DAT**

With this command, you initiate an ACL editing session to create an ACL for the file WEATHERTBL.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.

Syntax

/OBJECT_TYPE

/RECOVER

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

Syntax

/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.

Examples

```
$ 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.

Chapter 2. Accounting Utility

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.

Syntax

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
/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 Integrity servers)
/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

/ACCOUNT — Selects or rejects records for the specified account names.

Syntax

/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:

Value	Description
<batch>	Batch job login failure
<det>	Detached process login failure
<login>	Interactive login failure
<net>	Network login failure
<start>	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.

/ADDRESS

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

Syntax

/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

/BEFORE — Selects all records time-stamped before the specified time.

Syntax

```
/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.

/BINARY

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

Syntax

```
/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 Integrity servers-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

/BRIEF — Produces a brief report of the selected records.

Syntax

/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 Integrity servers-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).

Column	Description
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.

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

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

Syntax

/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.

/FULL

/FULL — Produces a full report of the selected records.

Syntax

/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 Integrity servers-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, see the definitions of the symbols that begin PRV$V_ in the $PRVDEF macro in the STARLET library. For example, the $PRVDEF macro defines the PRV$V_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:                     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:                  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.

/IDENT

/IDENT — Selects or rejects records for the specified processes.

Syntax

/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

/IMAGE — Selects or rejects records for the specified images.

Syntax

/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.

/JOB

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

Syntax

```
/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

/LOG — Outputs informational messages.

Syntax

```
/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.

/NODE

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

Syntax

/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 Integrity servers)

/OUTPUT (Alpha and Integrity servers) — Specifies the output file. Requires read and write access to the directory in which the output file is created.

Syntax

/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.

/OWNER

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

Syntax

/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

/PRIORITY — Selects or rejects records for the specified priority.

Syntax

```
/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.

/PROCESS

/PROCESS — Selects or rejects records for the specified types of process.

Syntax

```
/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

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

Syntax

```
/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.

/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.

Syntax

```
/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

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

Syntax

```
/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.

/REPORT

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

Syntax

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

Keyword

resource[,...]

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

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

[2]Present in records of type IMAGE, LOGFAIL, PROCESS, and SYSINIT

[1]Present in records of type PRINT

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.

/SINCE

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

Syntax

/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

/SORT — Sorts the selected records.

Syntax

/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
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

Keyword	Sorts on This Field
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 SYS$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.

/STATUS

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

Syntax

/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

/SUMMARY — Produces a summary report of the selected records.

Syntax

/SUMMARY =(summary_item[,...])

Keyword

summary_item[,...]

Specifies the summary key. The following table lists keywords:

Keyword	Description
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

Keyword	Description
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.

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    Total
                Printed  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        Total
                  Time        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

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

Syntax

/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.

/TITLE

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

Syntax

/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

/TYPE — Selects or rejects the specified types of record.

Syntax

/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:

Keyword	Type of Record	Description of Record
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.

Keyword	Type of Record	Description of Record
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.

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

/UIC — Selects or rejects records for the specified UICs.

Syntax

```
/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 enclosed in parentheses and followed by a comma, all records *except* those with the specified values are accepted.

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].

/USER

/USER — Selects or rejects records for the specified user names.

Syntax

```
/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

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

Syntax

```
/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    PROCESSOR       DIRECT   BUFFERED
           RECORDS     TIME            I/O       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 field sprint correctly.

Chapter 3. 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, see *Appendix D, "ANALYZE/ DISK_STRUCTURE—Stage Checks"*.

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 Integrity servers 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.

For further details, see the *Section 3.1.2, "Detecting Shadow Set Errors"* and to the /SHADOW qualifier documentation.

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
```

Note

VSI 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
```

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 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 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, "ANALYZE/DISK_STRUCTURE—Usage File"*.

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 you enter a WRITE command, it might not be written to all the members when the ANALYZE/DISK/SHADOW processes it.

If a discrepancy is found, a cluster wide 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.

In addition, this utility might report explainable discrepancies between the shadow set members if either of the following occurs:

- The shadow set has not undergone a full merge since the shadow set was created. This occurs if the shadow set was created using the DCL command INITIALIZE/SHADOW without the /ERASE qualifier and if the disk devices had different contents.

  It is important to be aware that this is *not* disk corruption. The blocks that are reported as different have not been written to, but they might contain stale data; the blocks reported as inconsistent might even be allocated to a file because there might be unwritten space between the file's end-of-data location and the end of the allocated space.

- A full merge has not occurred since the shadow set was logically expanded after a new member was added. The following example illustrates this problem:

  - Shadow set DSA1: consists of two members:

    1DGA20:## (18 GB)
    1DGA21:## (36 GB)

  - A second 36-GB member, 1DGA22:, is added to the shadow set with a full copy operation.

  - After the copy completes, 1DGA20: is removed from the shadow set.

  - At this point, if you enter the SET VOLUME/SIZE DSA1: command, the shadow set virtual unit DSA1: increases to 36 GB. Then, ANALYZE/DISK/SHADOW reports discrepancies because only the first 18 GB of the shadow set contents were copied to 1DGA22:.

  The discrepancies reported by ANALYZE/DISK/SHADOW are harmless because the space in question has not yet been written by applications.

You can eliminate inconsistencies by performing a full merge. To initiate a full merge, enter the DCL command SET SHADOW/DEMAND_MERGEDSA `xxx`. If the devices are served by controllers that support controller-based minimerge (for example, HSJ50s), enter this command while the shadow set is mounted on only one node within the cluster. Otherwise, a minimerge occurs, and the discrepancy might not be resolved. When you add members to a single member shadow set, a full copy operation also ensures that the disk is consistent both within and outside the file system.

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.

Syntax

`ANALYZE/DISK_STRUCTURE device-name:[/qualifier]`

Parameter

device-name

Specifies the disk volume or volume set to be verified. If you specify a volume set, all volumes of the volume set must be mounted as Files–11 volumes. For information about the Mount utility, see 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, see the *VSI OpenVMS Guide to System Security*.

Warning

When you use the /REPAIR or /LOCK_VOLUME qualifier, *only* the process running the ANALYZE/DISK_STRUCTURE Utility has access to the file system. This means that active files such as SYSUAF, RIGHTSLIST, log files, and especially AUDIT_SERVER journal and log files that might exist on the target device are stalled while ANALYZE/DISK_STRUCTURE is running.

Stalling includes OPEN, CREATE, CLOSE, file EXTEND, and TRUNCATE operations. Stalling occurs on all nodes within the cluster that have the volume mounted.

If you specify /REPAIR, the utility uses the ACP control lock volume function to prevent creation, deletion, extension, and truncation activity while the volume is being rebuilt. In this way, the volume is prevented from being modified while the operation is in progress.

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 might receive error messages that incorrectly indicate file damage. To avoid this problem, VSI 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 Integrity servers) Prevents updates to a volume while you are analyzing it

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

/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.

Syntax

/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.

/HOMEBLOCKS

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

Syntax

/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

/LIST — Determines whether the Analyze/Disk_Structure utility produces a listing of the index file.

Syntax

/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.

/LOCK_VOLUME (Alpha and Integrity servers)

/LOCK_VOLUME (Alpha and Integrity servers) — Prevents updates to a volume while you are analyzing it.

Syntax

/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, RIGHTSLIST, log files, and AUDIT_SERVER journal and log files. All of these files are stalled while ANALYZE/DISK_STRUCTURE is running.

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

/OUTPUT — Specifies the output file to which the Analyze/Disk_Structure utility is to write the disk structure errors.

Syntax

/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.

/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.

Syntax

/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

/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).

Syntax

/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.

VSI 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
```

/REPAIR

/REPAIR — Determines whether the Analyze/Disk_Structure utility repairs errors that are detected in the file structure of the specified device.

Syntax

/REPAIR

/NOREPAIR

Description

The Analyze/Disk_Structure utility does not perform any repair operation unless you specify the /REPAIR qualifier. The default is /NOREPAIR.

If you specify /REPAIR, the utility uses the ACP control lock volume function to prevent creation, deletion, extension, and truncation activity while the volume is being rebuilt. In this way, the volume is prevented from being modified while the operation is in progress.

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

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

Syntax

/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.

/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 on the entire contents of a shadow set or a specified range of blocks in a shadow set. If a discrepancy is found, a cluster wide WRITE lock is taken on the shadow set, and the questionable blocks are reread. Then one of the following actions occurs:

- If a member of the shadow set experiences connectivity problems for any reason, the ANALYZE/DISK_STRUCTURE command displays the error that it received and then returns the user to the DCL prompt.

  To correct the connectivity problem and run the utility again on the same shadow set, you might need to create a temporary file on the virtual unit before reissuing the ANALYZE/DISK/SHADOW command.

- 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, "Detecting Shadow Set Errors"* 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 _$252$MDA0: _$252$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 252MDAO: and 252DUA716: have identical data in those blocks.

/STATISTICS

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

Syntax

/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.

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]

/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.

Syntax

/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.

Chapter 4. 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, "Security Audit Message Format"*.

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, see the *VSI 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.

Syntax

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.

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

Qualifier	Description
/SUMMARY	Specifies that a summary of the selected records be produced after all records are processed

/BEFORE

/BEFORE — Controls whether records dated earlier than the specified time are selected.

Syntax

```
/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 *VSI 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-2005 -
   _$ SYS$MANAGER:SECURITY.AUDIT$JOURNAL
   ```

   The command in this example selects all records dated earlier than November 25, 2005.

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.

/BINARY

/BINARY — Controls whether output is a binary file.

Syntax

```
/BINARY
```

```
/NOBINARY
```

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=25OCT05.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 25OCT05.AUDIT.

/BRIEF

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

Syntax

```
/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.

/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.

Syntax

```
/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, RIGHTSLIST.DAT, NETPROXY.DAT, or NET$PROXY.DAT)
[NO]BREAKIN	Break-in detection
[NO]CONNECTION	Establishment of a network connection through the System Management utility (SYSMAN), DECwindows, or interprocess communication (IPC) software
[NO]CREATE	Creation of an object
[NO]DEACCESS	Completion of access to an object
[NO]DELETE	Deletion of an object
[NO]INSTALL	Modification of the known file list with the Install utility (INSTALL)
[NO]LOGFAIL	Unsuccessful login attempt
[NO]LOGIN	Successful login
[NO]LOGOUT	Successful logout
[NO]MOUNT	Execution of DCL commands MOUNT or DISMOUNT
[NO]NCP	Modification of the DECnet network configuration databases
[NO]NETPROXY	Modification of the network proxy authorization file (NETPROXY.DAT or NET$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.

/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.

Syntax

/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

/IGNORE — Excludes records from the report that match the specified criteria.

Syntax

/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.

/INTERACTIVE

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

Syntax

/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 non interactive 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

/OUTPUT — Specifies where to direct output from ANALYZE/AUDIT. If you omit the qualifier, the report is sent to SYS$OUTPUT.

Syntax

```
/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.

/PAUSE

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

Syntax

```
/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

/SELECT — Specifies the criteria for selecting records from the audit log file. For a description of how to generate audit records, see the *VSI OpenVMS Guide to System Security*.

Syntax

/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 *VSI OpenVMS Guide to System Security* describes each of these types.

ACCOUNT=(name,...)

Specifies the account name upon which selection is based. You can use wild cards, 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 wild cards to represent all or part of the alarm name.

ASSOCIATION_NAME=(IPC-name,...)

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

AUDIT_NAME=(journal-name,...)

Specifies the audit journal name on which selection is based. You can use wild cards 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, "Description of Audit Event Flags"*.)

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.

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.

When you use the Authorize utility (AUTHORIZE) to copy a user name, NEW_DATA specifies the newly created user name.

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.

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 *VSI 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 *VSI 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.
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.

For valid subtype values, see *Table F.2, "Description of Audit Event Types and Subtypes"*.

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).

UAF_SOURCE=(record-name,...)

Specifies the user name of the source record for an Authorize utility (AUTHORIZE) add, modify, or delete operation.

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.

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.%XFFFFFFFF) .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 ''f$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

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

Syntax

/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 *VSI 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-2005 -**
   _$ **SYS$MANAGER:SECURITY.AUDIT$JOURNAL**

   The command in this example selects records dated later than November 25, 2005.

2. $ **ANALYZE/AUDIT /SINCE=25-NOV-2005:15:00 -**
   _$ **SYS$MANAGER:SECURITY.AUDIT$JOURNAL**

   The command in this example selects records written after 3 P.M. on November 25, 2005.

/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.

Syntax

/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.

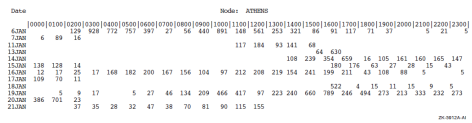
```
Total records read:         9701          Records selected:           9701
Record buffer size:         1031
Successful logins:           542          Object creates:             1278
Successful logouts:          531          Object accesses:            3761
Login failures:               35      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



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

CONTINUE — Resumes processing of event records.

Syntax

CONTINUE

Parameters

None.

Qualifiers

None.

Example

```
COMMAND> DISPLAY/SINCE=25-JAN-2005/SELECT=USERNAME=JOHNSON
COMMAND> CONTINUE
```

The first command in this example selects only event records generated by user JOHNSON after January 25, 2005. The second command in the example displays a report based on the new selection criteria.

DISPLAY

DISPLAY — Changes the criteria used to select event records.

Syntax

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.

EXIT

EXIT — Terminates the session.

Syntax

EXIT

Parameters

None.

Qualifiers

None.

HELP

HELP — Provides online help information for using ANALYZE/AUDIT commands.

Syntax

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.

LIST

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

Syntax

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

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.

Syntax

```
NEXT FILE
```

Parameters

None.

Qualifiers

None.

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.

Syntax

```
NEXT RECORD
```

Parameters

None.

Qualifiers

None.

POSITION

POSITION — Moves the full-format display forward or backward the specified number of event records.

Syntax

```
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.

SHOW

SHOW — Displays information about the selection or exclusion criteria currently being used to select event records.

Syntax

```
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.

Chapter 5. Authorize Utility

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).

Note that you must have read access to the RIGHTSLIST.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 Alpha and Integrity server 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)
```

Updated Quotas for the DEFAULT and SYSTEM Accounts

In OpenVMS Version 8.2 the quotas associated with the DEFAULT and SYSTEM accounts were updated. These updated quotas are seen only on fresh installations of OpenVMS or on the creation of a new SYSUAF data file. Existing SYSUAF data files are not updated.

The updates to the DEFAULT account are as follows:

Quota	Old Value	New Value
ASTLM	250	300
BYTLM	64,000	128,000
ENQLM	2,000	4,000

Quota	Old Value	New Value
FILLM	100	128
PGFLQUOTA	50,000	256,000
TQELM	10	100
WSDEFAULT	2000	4,096
WSQUOTA	4000	8,192

The updates to the SYSTEM account are the same as the DEFAULT account with the exception of the following two quotas:

Quota	Old Value	New Value
BYTLM	64,000	256,000
PGFLQUOTA	50,000	700,000

For upgraded systems with existing SYSUAF files, you might want to update the DEFAULT and SYSTEM account quotas to these new values.

5.1.1. 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.

Syntax

```
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**.

5.1.2. 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
Managing System Resources and User Accounts with SYSUAF	
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.
Managing Network Proxies with NETPROXY.DAT or NET$PROXY.DAT	
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.
Managing Identifiers with RIGHTSLIST.DAT	
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.

Command	Description
Managing System Resources and User Accounts with SYSUAF	
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.
General Commands	
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

ADD — Adds a user record to the SYSUAF and corresponding identifiers to the rights database. ADD/ IDENTIFIER and ADD/PROXY are documented as separate commands.

Syntax

`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 *VSI 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.

/ACCESS	Allows unrestricted access
/NOACCESS=SECONDARY	Allows access on primary days only
/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 *VSI 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 an identifier to the rights database file, RIGHTSLIST.DAT, and also adds a user to the user authorization file, SYSUAF. The /NOADD_IDENTIFIER qualifier does not add an identifier to the RIGHTSLIST.DAT file but does, however, add a user to the SYSUAF user record file. Note that the AUTHORIZE command ADD/IDENTIFIER is quite different: it only adds an entry to the rights database file, RIGHTSLIST.DAT.

/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 250 on Alpha 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 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 128,000 on Alpha and Integrity server 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 *VSI 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 150 on Alpha and Integrity server 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 4000 on Alpha and Integrity server 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 128 on Alpha and Integrity server 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 the *VSI OpenVMS Guide to System Security*. 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_PW_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	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:

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.

/IDENTIFIER

Adds an identifier to the rights database, RIGHTSLIST.DAT. The ADD/IDENTIFIER command does not add a user account to the authorization file, SYSUAF.

The ADD/ADD_IDENTIFIER command, however, adds a user account to the authorization file, SYSUAF, and also adds an identifier to the rights database, RIGHTSLIST.DAT.

/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 Alpha and Integrity server 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 for VSI versions of OpenVMS V8.4, and to 64 characters for OpenVMS V9 and above. 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.

The system also supports case-sensitive and extended-character passwords if the account has the PWDMIX flag set. In this case, the user's new mixed-case primary and secondary passwords must be enclosed in quotation marks (" ").

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 VSI.

/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 256,000 pagelets on Alpha and Integrity server 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 8 on Alpha and Integrity server 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 to63 on Alpha and Integrity server 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 *VSI 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 10.

/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. VSI 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 4096 pagelets on Alpha and Integrity server 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 16384 pagelets on Alpha and Integrity server 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 Integrity server 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.

On Alpha and Integrity servers, the DEFAULT account is as follows:

```
Username: DEFAULT                           Owner:
Account:                                    UIC:    [200,200]
 ([FIELD,USERP])
CLI:      DCL                               Tables: DCLTABLES
Default:  SYS$SYSDEVICE:[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:       100  Bytlm:        64000
Maxacctjobs:      0  Shrfillm:      0  Pbytlm:           0
Maxdetach:        0  BIOlm:       150  JTquota:       4096
Prclm:            8  DIOlm:       150  WSdef:         2000
Prio:             4  ASTlm:       250  WSquo:         4000
Queprio:          0  TQElm:        10  WSextent:     16384
CPU:         (none)  Enqlm:      2000  Pgflquo:      50000
Authorized Privileges:
  NETMBX     TMPMBX
Default Privileges:
  NETMBX     TMPMBX
```

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
  RIGHTSLIST.DAT
%UAF-I-RDBADDMSGU, identifier INV value: [000014,177777] added to
  RIGHTSLIST.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
      RIGHTSLIST.DAT
UAF> MODIFY WELCH/FLAGS=(RESTRICTED,DISNEWMAIL,DISWELCOME, -
_NODISUSER,EXTAUTH)/NODIALUP=SECONDARY/NONETWORK=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 tothe 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.

ADD/IDENTIFIER

ADD/IDENTIFIER — Adds only an identifier to the rights database. It does not add a user account.

Syntax

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 32 alphanumeric characters. The name can contain underscores and dollar signs. It must contain at least one non numeric 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 wild cards is permitted for user names; UICs must be in the form [*,*], [n,*], [*,n], or [n,n]. A wildcard username specification (*) creates identifiers alphabetically by username; 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.

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.

Syntax

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 auser 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, VSI 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, see the *VSI 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.

COPY

COPY — Creates a new SYSUAF record that duplicates an existing UAF record.

Syntax

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 *VSI 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.

/ACCESS	Allows unrestricted access
/NOACCESS=SECONDARY	Allows access on primary days only
/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 *VSI 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 an identifier to the rights database file, RIGHTSLIST.DAT, and also adds a user to the user authorization file, SYSUAF. The /NOADD_IDENTIFIER qualifier does not add an identifier to the RIGHTSLIST.DAT file but does, however, add a user to the SYSUAF user record file. Note that the AUTHORIZE command ADD/IDENTIFIER is quite different: it only adds an entry to the rights database file, RIGHTSLIST.DAT.

/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 300 on Alpha and Integrity server 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 150 on Alpha and Integrity server 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 128,000 on Alpha and Integrity server 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 *VSI 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 150 on Alpha and Integrity server 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 4000 on Alpha and Integrity server 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 128 on Alpha and Integrity server 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 the *VSI OpenVMS Guide to System Security*. 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 system wide 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:

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.

/IDENTIFIER

Adds an identifier to the rights database, RIGHTSLIST.DAT. The ADD/IDENTIFIER command does not add a user account to the authorization file, SYSUAF.

The ADD/ADD_IDENTIFIER command, however, adds a user account to the authorization file, SYSUAF, and also adds an identifier to the rights database, RIGHTSLIST.DAT.

/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 Alpha and Integrity server 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 for VSI versions of OpenVMS V8.4, and to 64 characters for OpenVMS V9 and above. 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.

The system also supports case-sensitive and extended-character passwords if the account has the PWDMIX flag set. In this case, the user's new mixed-case primary and secondary passwords must be enclosed in quotation marks (" ").

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 VSI.

/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 256,000 pagelets on Alpha and Integrity server 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 8 on Alpha and Integrity server 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 63 on Alpha and Integrity server 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 *VSI 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 10.

/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. VSI 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 4096 pagelets on Alpha and Integrity server 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 16384 pagelets on Alpha and Integrity server 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 Integrity server 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 RIGHTSLIST.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.

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. Do not delete NETPROXY.DAT because DECnet Phase IV and many layered products still use it.

Syntax

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 proxyauthorization file.

CREATE/RIGHTS

CREATE/RIGHTS — Creates and initializes the rights database, RIGHTSLIST.DAT.

Syntax

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, see the *VSI OpenVMS Guide to System Security*.

DEFAULT

DEFAULT — Modifies the SYSUAF's DEFAULT record.

Syntax

```
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 *VSI 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.

/ACCESS	Allows unrestricted access
/NOACCESS=SECONDARY	Allows access on primary days only
/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 *VSI 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.

Type	Definition
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 300 on Alpha and Integrity server 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 150 on Alpha and Integrity server 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 128,000 on Alpha and Integrity server 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 *VSI 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 150 on Alpha and Integrity server 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 4000 on Alpha and Integrity server 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 128 on Alpha and Integrity server 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 the *VSI OpenVMS Guide to System Security*. 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).
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:

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.

/IDENTIFIER

Adds an identifier to the rights database, RIGHTSLIST.DAT. The ADD/IDENTIFIER command does not add a user account to the authorization file, SYSUAF.

The ADD/ADD_IDENTIFIER command, however, adds a user account to the authorization file, SYSUAF, and also adds an identifier to the rights database, RIGHTSLIST.DAT.

/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 Alpha and Integrity server 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 for VSI versions of OpenVMS V8.4, and to 64 characters for OpenVMS V9 and above. 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.

The system also supports case-sensitive and extended-character passwords if the account has the PWDMIX flag set. In this case, the user's new mixed-case primary and secondary passwords must be enclosed in quotation marks (" ").

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 VSI.

/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 256,000 pagelets on Alpha and Integrity server 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 8 on Alpha and Integrity server 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 to63 on Alpha and Integrity server 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 *VSI 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 10.

/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. VSI 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 4096 pagelets on Alpha and Integrity server 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 16384 pagelets on Alpha and Integrity server 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 Integrity server 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 VSI-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.
/LGICMD	Specifies the file name of a command procedure to be invoked during the login of the user. 1. 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. 2. 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. 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.
/PRIVILEGES	When users are given different privileges than those supplied by VSI.

Qualifier	Reason for Modification
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.

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**.

Syntax

EXIT

Parameters

None.

Qualifiers

None.

GRANT/IDENTIFIER

GRANT/IDENTIFIER — Assigns the specified identifier to the user and documents the user as a holder of the identifier in the rights database.

Syntax

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.

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

HELP — Displays information concerning the use of AUTHORIZE, including formats and explanations of commands, parameters, and qualifiers.

Syntax

```
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:

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           /PWDLIFETIME
/PWDMINIMUM           /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

LIST — Writes reports for selected UAF records to a listing file, SYSUAF.LIS, which his placed in the current default directory. LIST/IDENTIFIER, LIST/PROXY, and LIST/RIGHTS are documented as separate commands.

Syntax

```
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 (%) wild cards 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. (VSI 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.
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

LIST/IDENTIFIER — Creates a listing file (RIGHTSLIST.LIS) in which identifier names, attributes, values, and holders are written.

Syntax

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.

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

LIST/PROXY — Creates a listing file of the network proxy database entries from the network database file NET$PROXY.DAT.

Syntax

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.

LIST/RIGHTS

LIST/RIGHTS — Lists identifiers held by the specified identifier or, if /USER is specified, all identifiers held by the specified users.

Syntax

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,*], [*,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

MODIFY — Changes values in a SYSUAF user record. Qualifiers not specified in the command remain unchanged. MODIFY/IDENTIFIER, MODIFY/PROXY, and MODIFY/SYSTEM_PASSWORD are documented as separate commands.

Syntax

```
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 *VSI 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.

/ACCESS	Allows unrestricted access
/NOACCESS=SECONDARY	Allows access on primary days only
/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 *VSI 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.

Type	Definition
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 250 on Alpha systems.

/BATCH[=(range[,...])]

Specifies the hours of access permitted for batch jobs. For a description of the range specification, see the /ACCESS qualifier. The default is 300 on Alpha and Integrity server systems.

/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 150 on Alpha and Integrity server 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 128,000 on Alpha and Integrity server 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 150 on Alpha and Integrity server 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 4000 on Alpha and Integrity server 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 128 on Alpha and Integrity server 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 the *VSI OpenVMS Guide to System Security*. 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).
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:

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.

/IDENTIFIER

Adds an identifier to the rights database, RIGHTSLIST.DAT. The ADD/IDENTIFIER command does not add a user account to the authorization file, SYSUAF.

The ADD/ADD_IDENTIFIER command, however, adds a user account to the authorization file, SYSUAF, and also adds an identifier to the rights database, RIGHTSLIST.DAT.

/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 Alpha and Integrity server 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 for VSI versions of OpenVMS V8.4, and to 64 characters for OpenVMS V9 and above. 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.

The system also supports case-sensitive and extended-character passwords if the account has the PWDMIX flag set. In this case, the user's new mixed-case primary and secondary passwords must be enclosed in quotation marks (" ").

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 VSI.

/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 256,000 pagelets on Alpha and Integrity server 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 8 on Alpha and Integrity server 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 63 on Alpha and Integrity server 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 *VSI 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 10.

/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. VSI 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 4096 pagelets on Alpha and Integrity server 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 16384 pagelets on Alpha and Integrity server 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 Integrity server 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.

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

MODIFY/IDENTIFIER — Modifies an identifier name, its associated value, or its attributes in the rights database.

Syntax

```
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.

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, %X80000000is 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.

Example

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

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. You must modify the proxy database from a system running the current OpenVMS system.

Syntax

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.

```
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-NAFADDMSG, 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

MODIFY/SYSTEM_PASSWORD — Changes the systemwide password. 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.

Syntax

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, see the discussion of the SET PASSWORD/SYSTEM command in the *VSI OpenVMS Guide to System Security*.

Example

```
UAF> MODIFY/SYSTEM_PASSWORD=ABRACADABRA
UAF>
```

This command changes the systemwide password to ABRACADABRA.

REMOVE

REMOVE — Deletes a SYSUAF user record and corresponding identifiers in the rights database. The DEFAULT and SYSTEM records cannot be deleted. REMOVE/IDENTIFIER and REMOVE/PROXY are documented as separate commands.

Syntax

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-REMMSG, record removed from SYSUAF.DAT
%UAF-I-RDBREMMSGU, 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

REMOVE/IDENTIFIER — Removes an identifier from the rights database.

Syntax

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-RDBREMMSGU, identifier Q1SALES value %X80010024 removed from
   RIGHTSLIST.DAT
```

The command in this example removes the identifier Q1SALES from the rights database. All of its holder records are removed with it.

REMOVE/PROXY

REMOVE/PROXY — Deletes network proxy access for the specified remote user.

Syntax

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-NAFREMMSG, 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

RENAME — Changes the user name of the SYSUAF record (and, if specified, the corresponding identifier) while retaining the characteristics of the old record. RENAME/IDENTIFIER is documented as a separate command.

Syntax

```
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:

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.

/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.

/PASSWORD=
(password1 [,password2])
/NOPASSWORD

Specifies up to two passwords for login. Passwords can be from 0 to 32 alphanumeric characters in length for VSI versions of OpenVMS V8.4, and to 64 characters for OpenVMS V9 and above. 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.

The system also supports case-sensitive and extended-character passwords if the account has the PWDMIX flag set. In this case, the user's new mixed-case primary and secondary passwords must be enclosed in quotation marks (" ").

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.

RENAME/IDENTIFIER

RENAME/IDENTIFIER — Renames an identifier in the rights database.

Syntax

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

REVOKE/IDENTIFIER — Takes an identifier away from a user.

Syntax

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 RIGHTSLIST.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.

SHOW

SHOW — Displays reports for selected UAF records on the current SYS$OUTPUT device. SHOW/IDENTIFIER, SHOW/PROXY, and SHOW/RIGHTS are documented as separate commands.

Syntax

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, "Screen Control Keys"*. 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, "Screen Control Keys"* to view the entire screen.

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 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  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:         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   000000000011111111112222  Secondary 000000000011111111112222
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 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

SHOW/IDENTIFIER — Displays information about an identifier, such as its name, value, attributes, and holders, on the current SYS$OUTPUT device.

Syntax

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:

/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

SHOW/PROXY — Displays all authorized proxy access for the specified remote user.

Syntax

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.

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

SHOW/RIGHTS — Displays the identifiers held by the specified identifiers or, if /USER is specified, all identifiers held by the specified users.

Syntax

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,*], [*,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 RIGHTSLIST.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
```

Note that the following formats of the command produce the same result:

```
SHOW/RIGHTS/USER=ANDERSON
```

```
SHOW/RIGHTS/USER=[300,015]
```

Chapter 6. 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.

VSI 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, see the *VSI OpenVMS System Management Utilities Reference Manual, Volume II: M–Z.*

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, "Phases"*.

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, see the chapter about managing system parameters in the *VSI 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, "AUTOGEN Execution Modes"* 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, see *Section 6.3, "Feedback"*.

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, "What CLU$PARAMS.DAT Looks Like"*.

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.

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.

Note

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.

VSI 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
```

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, "Product Name"* 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 DEC Windows 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. VSI does not recommend editing these files.

6.2. AUTOGEN Usage Summary

VSI 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, see the chapter on managing page, swap, and dump files in the *VSI OpenVMS System Manager's Manual*.) For installation requirements, see specific product documentation.
- 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 *VSI 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, "Execution Modes"*.

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, "Phases"*.

The system parameters listed in an appendix to this manual indicate whether they are affected by AUTOGEN feedback.

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, "AUTOGEN Phases"* 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
- VSI-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, see the chapter on managing devices in the *VSI 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 SYSPRV, OPER, and CMKRNL 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. VSI 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.

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, see the chapter on managing page, swap and dump files in the *VSI 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 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 Integrity servers, 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, "AUTOGEN Execution Modes"* 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, "Files Used by AUTOGEN" lists the files AUTOGEN uses during each phase.

Table 6.3. Files Used by AUTOGEN

AUTOGEN Phase	Input Files [1]	Output Files [1]
SAVPARAMS	None	AGEN$FEEDBACK.DAT
GETDATA	NEWPARAMS.DAT[2] CLU$PARAMS.DAT	CLU$PARAMS.DAT
	AGEN$FEEDBACK.DAT CLU$PARAMS.DAT MODPARAMS.DAT	PARAMS.DAT[3]
GENPARAMS	PARAMS.DAT	SETPARAMS.DAT VMSIMAGES.DAT AGEN$PARAMS.REPORT
TESTFILES	PARAMS.DAT	SYS$OUTPUT
GENFILES	PARAMS.DAT	PAGEFILE.SYS SWAPFILE.SYS SYSDUMP.DMP

AUTOGEN Phase	Input Files [1]	Output Files [1]
		AGEN$PARAMS.REPORT
SETPARAMS	SETPARAMS.DAT	ALPHAVMSSYS.PAR (Alpha)[d] IA64VMSSYS.PAR (Integrity servers)[e] AUTOGEN.PAR ALPHAVMSSYS.OLD (Alpha)[d] IA64VMSSYS.OLD (Integrity servers)[e]
SHUTDOWN	None	None
REBOOT	None	None

[1]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.
[2]From software installation kit
[3]Also includes collected hardware configuration information
[d]Alpha specific
[e]I64 specific

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.

Syntax

```
@SYS$UPDATE:AUTOGEN [start-phase] [end-phase] [execution-mode]
```

Parameters

start-phase

Specify the phase where AUTOGEN is to begin executing. *Table 6.1, "AUTOGEN Phases"* 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, "AUTOGEN Phases"*. 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, "AUTOGEN Phases"* 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, "AUTOGEN Execution Modes" 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.

Chapter 7. Backup Utility

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, see the *VSI OpenVMS System Manager's Manual*.

Note

Some layered products have their own special backup procedures. For more information, see the layered product documentation.

Also, when a symbolic link is encountered during a backup operation, the symbolic link itself is copied. This is true for all backup types — physical, image, and file. For more information, see the *VSI C Run-Time Library Reference Manual for OpenVMS Systems* [https://docs.vmssoftware.com/vsi-c-run-time-library-reference-manual-for-openvms-systems/].

Using BACKUP 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).

If you have access to the OpenVMS Alpha or Integrity servers CD–ROM, you can use a menu system supplied on the CD–ROM to back up your system disk.

For more information about standalone BACKUP and the menu-driven procedure, see the *VSI OpenVMS System Manager's Manual*.

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.

Note

Beginning in Version 8.2, a restore of a physical backup no longer requires the output disk to have the same geometry (tracks, cylinders). The restore operation works as long as the output has the same or larger capacity.

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 also 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 *VSI OpenVMS User's Manual* describes valid specifications for disk files; the *VSI 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 *VSI OpenVMS User's Manual*.

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, "BACKUP Input and Output by Operation Type" 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

Operation	Format
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[1]	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]

[1]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:

- 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. 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, see the *VSI 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 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:

1. Copies the file MONDAY.DIR to [JAMES]
2. Copies the file TUESDAY.DIR to [JAMES.MONDAY], and
3. Copies the file WEDNESDAY.DIR to [JAMES.MONDAY.TUESDAY].
4. 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...]
```

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 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.
- **Output file qualifiers** change the way output files are restored. Place them immediately after the output 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 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.

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, "Valid Combinations of BACKUP Qualifiers" contains more information about valid combinations of BACKUP qualifiers. *Table 7.2, "Summary of BACKUP Qualifiers by Type"* summarizes BACKUP qualifiers by type.

Table 7.2. Summary of BACKUP Qualifiers by Type

Qualifier	Description
Command Qualifiers	
/[NO]ALIAS	Specifies whether to maintain the previous behavior of multiple processing of alias and primary file entries.
/[NO]ASSIST	Allows operator or user intervention if a request to mount a magnetic tape fails during a BACKUP operation.
/BRIEF	Causes the /LIST qualifier to display the file specification, size (in blocks), and creation date for each file in the save set.
/COMPARE	Causes BACKUP to compare the contents of the first parameter with the contents of the second parameter.
/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.
/ENCRYPT	Creates and restores encrypted save sets.
/FAST	Processes the input specifier using a fast file scan to reduce processing time.
/FULL	Displays the information produced by the /LIST command qualifier in a format similar to that displayed by the DCL command DIRECTORY/FULL.
/IGNORE	Specifies that a BACKUP save or copy operation overrides restrictions placed on files or is not to perform tape label processing checks.
/IMAGE	Directs BACKUP to process an entire volume or volume set.
/INCREMENTAL	Allows you to restore a disk volume from a series of incremental save sets. (Unrelated to /NOINCREMENTAL.)
/[NO]INITIALIZE	Initializes an output disk volume, making its entire previous contents unavailable.
/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.
/IO_LOAD	Increases or decreases the number of simultaneous I/Os issued by the BACKUP utility. The default is 8 I/Os. The minimum is 2 I/Os.

Qualifier	Description
/JOURNAL	Specifies that a BACKUP save operation is to create, or append information to, a BACKUP journal file.
/LIMIT	Specifies the expansion size limit during restore or save operations.
/LIST	Lists information about a BACKUP save set and about the files in a save set.
/[NO]LOG	Displays the file specification of each file processed during the operation on SYS$OUTPUT.
/NOINCREMENTAL	Allows you to control the amount of file data that is saved in a save operation. (Unrelated to /INCREMENTAL.)
/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.
/PROGRESS_REPORT	Displays the progress of a backup operation on the current output device.
/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.
/SIZE	Preserves the logical volume size on the target device or allows you to specify the logical size of the target device.
/[NO]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.
Input File-Selection Qualifiers	
/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.
/BY_OWNER	Causes BACKUP to process files owned by the specified UIC.

Qualifier	Description
/CONFIRM	Displays prompts on your terminal for confirmation before processing each file.
/CONVERT	Converts ODS-5 file names to ODS-2 file names.
/CREATED	Selects files according to the value of the creation date field in each file header record.
/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.
/FILES_SELECTED	Specifies a file that contains a list of the files that are to be selected when a save set is restored.
/HEADER_ONLY	Controls whether BACKUP saves only the file header of shelved and preshelved files.
/MODIFIED	Selects files according to the value of the modified date field (the date the file was last modified) in each file header record.
/SINCE	Selects files dated equal to or later than the specified date and time.
Output File Qualifiers	
/BY_OWNER	Redefines the owner user identification code (UIC) for restored files.
/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.
/OVERLAY	Writes over an existing file when an identically named file is encountered during the restore operation.
/REPLACE	Replaces a file on the output specifier with an identically named file from the input specifier.
Input Save-Set Qualifiers	
/[NO]CRC	Checks the software cyclic redundancy check (CRC) encoded in the save set's data blocks.
/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.
/[NO]REWIND	Rewinds the input tape reel to the beginning-of-tape marker before reading the input volume.
/SAVE_SET	Directs BACKUP to treat the input file as a BACKUP save set.
/SELECT	Selects the specified files for processing.
Output Save-Set Qualifiers	

Qualifier	Description
/BLOCK_SIZE	Specifies the output block size, in bytes, for data records in BACKUP save sets and in disk-to-disk copies.
/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.
/[NO]CRC	Specifies that the CRC is to be computed and stored in the data blocks of the output save set.
/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.
/GROUP_SIZE	Defines the number of blocks BACKUP places in each redundancy group.
/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.
/MEDIA_FORMAT	Controls whether data records are automatically compacted and blocked together.
/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.
/[NO]REWIND	Rewinds the output tape to the beginning-of-tape marker and initializes the output tape.
/SAVE_SET	Directs BACKUP to treat the output file as a BACKUP save set.
/TAPE_EXPIRATION	Writes a file expiration date other than the current date to the file header label of the save set.

7.5. 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 or Integrity servers, 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 or Integrity servers, you must use standalone BACKUP to back up your system disk.

Syntax

```
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. For restrictions on the use of wildcard characters in BACKUP commands, see *Section 7.3.2, "Using Wildcard Characters with BACKUP"*.

Description

To invoke online BACKUP, enter an appropriate BACKUP command at the DCL prompt. For instructions on invoking standalone BACKUP, refer to the *VSI 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.

/ALIAS

/ALIAS — **Command Qualifier**: Specifies that the previous behavior of multiple processing of alias and primary file entries be maintained. 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 VSI support representative.

Syntax

/ALIAS save-set-spec (default)

/NOALIAS

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.

/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.

Syntax

/[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 *VSI 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

/BACKUP — **Input File-Selection Qualifier**: Selects files according to the BACKUP date written in the file header record by the BACKUP/RECORD command.

Syntax

```
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.

/BEFORE

/BEFORE — **Input File-Selection Qualifier**: Selects files dated earlier than the date and time you specify.

Syntax

```
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

/BLOCK_SIZE — **Output Save-Set Qualifier**: Specifies the output block size, in bytes, for data records in BACKUP save sets and in disk-to-disk copies.

Syntax

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).

/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.

Syntax

/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

/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.) VSI recommends that you remove the /BUFFER_COUNT qualifier from command procedures.

Syntax

/BUFFER_COUNT

/BY_OWNER (Select Input File by UIC)

/BY_OWNER (Select Input File by UIC) — **Input or Output File Qualifier, or Output Save-Set Qualifier**: Input File-Selection Qualifier As an input file-selection qualifier, /BY_OWNER causes BACKUP to process files owned by the specified UIC.

Syntax

input-specifier/BY_OWNER[=[uic]] output-specifier

Description

Specify the UIC as octal numbers or in alphanumeric format (in the form [g,m]). Note that the UIC specification must include the brackets. UIC formats are described in the *VSI OpenVMS User's Manual*. If you specify this qualifier without a UIC, the default UIC is the current process UIC. If you do not specify this qualifier, BACKUP processes all files on the volume. Output File Qualifier As an output file qualifier, /BY_OWNER redefines the owner UIC for each file restored during the operation. You can use one of the following options:

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. 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.

Output Save-Set Qualifier As an output save-set qualifier, /BY_OWNER specifies the owner UIC of the save set. If you omit the /BY_OWNER qualifier, the save set receives the UIC of the current process. To use /BY_OWNER as an output save-set qualifier, you must have the SYSPRV user privilege or the UIC must be your own.

Input File-Selection Qualifier

See separate descriptions for /BY_OWNER as an output file qualifier and an output save-set qualifier.

Selects files for processing according to the user identification code (UIC). 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	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 (Redefine Owner UIC for Restored File)

/BY_OWNER (Redefine Owner UIC for Restored File) — **Output File Qualifier**: *See separate descriptions for /BY_OWNER as an input file-selection qualifier and an output save-set qualifier.* Redefines the owner user identification code (UIC) for restored files.

Syntax

```
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.

/BY_OWNER (Specify Owner UIC for Save Set)

/BY_OWNER (Specify Owner UIC for Save Set) — **Output Save-Set Qualifier**: *See separate descriptions for /BY_OWNER as an input file-selection qualifier and an output file qualifier.* Specifies the owner user identification code (UIC) of the save set.

Syntax

```
input-specifier output-save-set-spec/BY_OWNER=uic
```

Description

If the /BY_OWNER qualifier is omitted, the UIC of the current process is used. To use this qualifier on Files–11 save sets, you need the user privilege SYSPRV, or the UIC must be your own.

Specify either a numeric UIC as octal numbers or an alphanumeric UIC in the form [g,m]. Wildcards are permitted. Note that the brackets are required.

[g,m]

g	An octal number in the range 0 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

/COMMENT — Places a comment in an output save set. If the comment string is longer than one word or if it contains non-alphanumeric characters, you must enclose it in quotation marks (" "). A DCL command can contain a maximum of 1024 characters.

Syntax

```
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.

/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.

Syntax

/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.

/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**.

Syntax

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

/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. You can also use the /NOCONVERT qualifier.

Syntax

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.

/CRC

/CRC — **Input or Output Save-Set Qualifier**: As an input save-set qualifier, /CRC causes the software cyclic redundancy check (CRC) encoded in the save set's data blocks to be checked (/CRC) or ignored (/NOCRC). If you ignore the CRC encoding, you reduce processing time at the risk of increasing data error. As an output save-set qualifier, /CRC specifies that software CRC checking code is to be computed and stored in the data blocks of the output save set. To disable CRC checking, use the /NOCRC qualifier. **Input Save-Set Qualifier**: *See a separate description of /CRC as an output save-set qualifier.* Specifies that the software cyclic redundancy check (CRC) is to be performed.

Syntax

```
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

/CRC — **Output Save-Set Qualifier**: *See a separate description of /CRC as an input 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.

Syntax

```
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.

/CREATED

/CREATED — **Input File-Selection Qualifier**: Selects files according to the value of the creation date field in each file header record.

Syntax

```
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).

/DATA_FORMAT

/DATA_FORMAT — **Command Qualifier**: Creates and restores compressed save sets. You can specify the /DATA_FORMAT qualifier anywhere on the BACKUP command line.

Syntax

```
/DATA_FORMAT=COMPRESS=algorithm
```

Description

For compression support specify /DATA_FORMAT=COMPRESS with the algorithm name. The compression algorithm should be mentioned with the qualifier, as of now only one algorithm is supported (that is DEFLATE) and used as default.

Note

The BACKUP compression is supported only for save set file operation on disk and sequential devices.

Examples

1. ```
   $ BACKUP/DATA_FORMAT=COMPRESS SYS$SYSTEM:*.EXE DKA0:[000000]SAVESET.BCK/
   SAVE
   ```

   This command saves the system executable file (SYS$SYSTEM:*.EXE) to a save set named SAVESET.BCK on to the disk DKA0. Since, no compression algorithm is specified the default DEFLATE compression algorithm is used to compress the data.
   ```

2. $ **BACKUP/DATA_FORMAT=COMPRESS=DEFLATE /IMAGE/RECORD DKA0: 2MGA0:SUT746.BCK/SAVE**

   This command creates the image backup of the disk DKA0 in the save set named SUT746.BCK on the magnetic tape labeled SUT746. Since the qualifier /DATA_FORMAT=COMPRESS is mentioned the created save set is compressed with this algorithm.

3. $ **BACKUP/DATA_FORMAT=COMPRESS SAVESET.BCK/SAVE DKA100:[000000...]**

   In this command, the save set SAVESET.BCK is restored to the directory tree DKA0:[000000]. /DATA_FORMAT=COMPRESS qualifier is specified here, in case if the save set is an uncompressed save set the qualifier is ignored and restore operation is continued.

   On the other hand, if save set is compressed and /DATA_FORMAT= COMPRESS is not specified during restore, BACKUP identifies the save set as a compressed save set and restores it successfully.

/DELETE

/DELETE — **Command Qualifier**: Specifies that a BACKUP save or copy operation is to delete the selected input files from the input volume after all files have been successfully processed.

Syntax

/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

/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.

Syntax

```
input-specifier output-save-set-spec/DENSITY=keyword
```

Description

The following table shows the densities that are supported for tapes.

Keyword	Meaning
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 Integrity server systems.	
TK85	DLT Tx85: 10625 BPI – Cmpt III - Alpha and Integrity servers only
TK86	DLT Tx86: 10626 BPI – Cmpt III - Alpha and Integrity servers only
TK87	DLT Tx87: 62500 BPI – Cmpt III - Alpha and Integrity servers only
TK88	DLT Tx88: (Quantum 4000) – Cmpt IV - Alpha and Integrity servers only
TK89	DLT Tx89: (Quantum 7000) – Cmpt IV - Alpha and Integrity servers only
QIC	All QIC drives are drive-settable only - Alpha and Integrity servers only
8200	Exa-Byte 8200 - Alpha and Integrity servers only
8500	Exa-Byte 8500 - Alpha and Integrity servers only
DDS1	Digital Data Storage 1 – 2G - Alpha and Integrity servers only
DDS2	Digital Data Storage 2 – 4G - Alpha and Integrity servers only
DDS3	Digital Data Storage 3 – 8-10G - Alpha and Integrity servers only
DDS4	Digital Data Storage 4 - Alpha and Integrity servers only
AIT1	Sony Advanced Intelligent Tape 1 - Alpha and Integrity servers only
AIT2	Sony Advanced Intelligent Tape 2 - Alpha and Integrity servers only

Keyword	Meaning
AIT3	Sony Advanced Intelligent Tape 3 - Alpha and Integrity servers only
AIT4	Sony Advanced Intelligent Tape 4 - Alpha and Integrity servers only
DLT8000	DLT 8000 - Alpha and Integrity servers only
8900	Exabyte 8900 - Alpha and Integrity servers only
SDLT	SuperDLT1 - Alpha and Integrity servers only
SDLT320	SuperDLT320 - Alpha and Integrity servers only

Note that tape density keywords cannot be abbreviated.

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.

/ENCRYPT

/ENCRYPT — **Command Qualifier**: Creates and restores encrypted save sets. Specify the /ENCRYPT qualifier anywhere on the BACKUP command line. Standalone BACKUP, which is a version of the BACKUP utility that runs without the support of the OpenVMS operating system, does not support the /ENCRYPT qualifier.

Syntax

```
/ENCRYPT=([key] [,ALGORITHM=algorithm])
```

Description

Optionally, you can specify either a key name or a key value, but not both. If you have already defined a key value using the DCL command ENCRYPT/CREATE_KEY, you can specify /ENCRYPT=NAME= *key-name* to identify the key name that was created and stored in the key storage table.

To define a key value interactively, specify /ENCRYPT=VALUE= *key-value*, where *key-value* is one of the following:

- 1 to 243 alphanumeric characters enclosed in quotation marks (“”). Dollar signs ($) and underscores (_) are valid characters. The key is not case sensitive.

- A hexadecimal constant using the digits 0 to 9 and A to F.

Note

For additional security, specify the /ENCRYPT qualifier with no parameters and press Return. The command prompts you for a key value. When you enter a value, the software does not echo what you type and, for verification, prompts you to retype the value.

ALGORITHM Keyword

Optionally, you can use ALGORITHM= *algorithm* to specify DES or AES algorithms:

- Data encryption standard (DES)

  Use DES to encrypt the initialization vector and the key you supply. Possible values for *algorithm* are as follows:

  - DESCBC (default) — Cipher block chaining
  - DESECB — Electronic code book
  - DESCFB — Cipher feedback

- Advanced encryption standard (AES)

  Use an AES algorithm to encrypt both the data and the user-provided key using the AES algorithm. Possible values for *algorithm* are as follows:

  - AESCBC — Cipher block chaining
  - AESECB — Electronic code book
  - AESCFB — Cipher feedback
  - AESOFB — Output feedback

  You can also specify one of the following three lengths for AES:

  128
  192
  256

  When you use an AES value, BACKUP places the result of the encryption operation in the save set as a BACKUP attribute subrecord of the BACKUP summary record. At the time of a save set restore or listing operation, BACKUP uses the key you supplied to get to the encrypted key to decrypt the data key and the initialization vector value.

Using /ENCRYPT and /SAVE_SET Qualifiers

The BACKUP command qualifier /SAVE_SET is both an input save set qualifier and an output save set qualifier, as follows:

- When you specify the /SAVE_SET and /ENCYRPT qualifiers with an output save set specification, BACKUP writes file data (including file names and attributes) in an encrypted form into the save set.

- When you specify /SAVE_SET with an input save set specification, BACKUP uses the decryption key specified to access the file name, attributes, and data from the save set records. The ENCRYPT option decrypts the data files after BACKUP reads the data files from the save set media and processes them according to the remaining qualifiers of the BACKUP command.

Restoring Files

When you encrypt a save set, BACKUP does not store the encryption key in the save set header. Consequently, to decrypt an encrypted save set, specify /ENCRYPT in the restore operation so that BACKUP searches for the data encryption control record.

If you restore an unencrypted save set and mistakenly specify /ENCRYPT, BACKUP ignores the incorrect qualifier. If you try to restore an encrypted save set without the /ENCRYPT qualifier or with a key name, the system displays the following error message:

```
 %BACKUP-F-ENCSAVSET, save set is encrypted, /ENCRYPT must be specified.
```

BACKUP tries to decrypt an encrypted save set by:

1. Decrypting the encryption data saved in an attribute subrecord.
2. Comparing a 32-bit checksum of the decrypted data key with the stored value.
3. If there is a match, BACKUP assumes the data key is valid and restores the save set.
4. If BACKUP finds a mismatch, which is likely if the data key or algorithm you specified in the BACKUP command is incorrect, the utility displays the following error message:

```
%BACKUP-F-ENCKEYMAT, the supplied decryption key does not yield a readable
 save set
```

Examples

1. ```
   $ ENCRYPT/CREATE_KEY my_key "This is my private encryption key"/AES/LOG
   %ENCRYPT-S-KEYDEF, key defined for key name = MY_KEY
   $ BACKUP *.COM COMS.BCK/SAVE/ENCRYPT=(name=my_key,alg=AES_/LOG)
   ```

   This example creates an encrypted save set.

2. ```
   $ BACKUP *.COM COMS.BCK/SAVE/ENCRY=ALG=AES
   Enter key value:
   Verification:
   $
   ```

   In this example, the BACKUP command line does not contain a key name or key value; therefore, BACKUP prompts for an encryption key.

3. ```
   $ BACKUP DKA100: DKA100.BCK/SAV/IMA/ENCRY=(VALUE="THIS IS MY  ENCRYPTION
    KEY")
   ```

   In this example, the image BACKUP of DKA100 is encrypted in a save set with a key value that uses the default DESCBC algorithm.

4. ```
   $ BACKUP DKA100: DKA100.BCK/SAV/IMA/ENCRY=(VALUE="THIS IS MY  ENCRYPTION
    KEY",ALGO=AESCFB192)
   ```

In this example, the image BACKUP of DKA100 is encrypted in a saveset with a key value that uses the AESCFB algorithm with a 192-bit encryption key.

/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.

Syntax

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.

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>
```

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

/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.

Syntax

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.

/EXPIRED

/EXPIRED — **Input File-Selection Qualifier**: Selects files according to the value of the expiration date field in each file header record.

Syntax

```
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

/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.

Syntax

```
/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.

/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.

Syntax

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

/FULL — **Command Qualifier**: Lists the file information produced by the command qualifier /LIST in the format provided by the DCL command DIRECTORY/FULL.

Syntax

/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
  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

/GROUP_SIZE — **Output Save-Set Qualifier**: Defines the number of blocks BACKUP places in each redundancy group.

Syntax

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.

/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.

Syntax

```
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 *preshelve* 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, see the 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.

/IGNORE

/IGNORE — **Command Qualifier**: Specifies that a BACKUP save or copy operation will override restrictions placed on files or will not perform tape label processing checks. 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.

Syntax

/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.
LABEL_PROCESSING	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.
LIMIT	Prevents the target device from inheriting the volume expansion limit.
NOBACKUP	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, see the *VSI 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. 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

/IMAGE — **Command Qualifier**: Directs BACKUP to process an entire volume or volume set. Beginning in Version 8.2, this qualifier has been supported for Integrity servers disk. The image of an Integrity servers disk can be saved and restored on either Alpha or Integrity servers.

Syntax

/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 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.

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

/INCREMENTAL — **Command Qualifier**: Allows you to restore an incremental save set. /INCREMENTAL is valid only in restore operations. It is not related to the /NOINCREMENTAL qualifier, which is valid only in BACKUP save operations.

Syntax

/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. (VSI 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.

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 *VSI 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

/INITIALIZE — **Command Qualifier**: Initializes an output disk or tape volume, making its entire previous contents unavailable. (/REWIND performs the same function for output tapes.)

Syntax

/[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.

Note

The BACKUP/NOINITIALIZE command does not preserve the dynamic volume expansion characteristics of the output device. The reason is that the target device is mounted foreign, preventing OpenVMS from obtaining the expansion size and the logical size. To overcome this restriction, use the /LIMIT and /SIZE qualifiers.

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.

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_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.

Syntax

```
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
```

/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. (You can also specify /NOCONVERT with the /INTERCHANGE qualifier.)

Syntax

```
/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.

/IO_LOAD

/IO_LOAD — **Command Qualifier**: Beginning in OpenVMS Version 8.3, BACKUP is optimized to work more efficiently with new storage controllers. You can use the /IO_LOAD qualifier to increase or decrease the number of simultaneous I/Os issued by the BACKUP utility. The default is 8 I/Os. The minimum is 2 I/Os. If the /IO_LOAD qualifier is omitted from the command line, the default number of outstanding I/Os is still 8.

Syntax

/IO_LOAD=*n*

The value for *n* is an integer between 1 and the process AST limit. The default value is 8.

Example

```
$ BACKUP DKA100: DKA400: /IMAGE /IO_LOAD=8
```

In this example, the /IO_LOAD=8 qualifier maintains 8 threads of I/O reading data from the source disk. (BACKUP does not exceed 8 outstanding I/Os.)

/JOURNAL

/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.

Syntax

/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.

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: Return
%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.

/LABEL

/LABEL — **Output Save-Set Qualifier**: Specifies the volume labels for the magnetic tapes to which the save set is written.

Syntax

```
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
MARK	MARK, MARKnn

Label Specified in the Command Line	Matching Volume Labels
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.

/LIMIT

/LIMIT — **Command Qualifier**: The /LIMIT qualifier allows you to specify the expansion size limit during restore or save operations. Therefore, you can override the value stored in the saveset header. (This matches the way the /LIMIT qualifier of the INITIALIZE utility works.)

Syntax

/LIMIT=*n*

The value for *n* is the expansion size of the device. There are no limits on this value.

Specifying /LIMIT without a value instructs BACKUP that the target device is to inherit the expansion size. This is the opposite of specifying /IGNORE=LIMIT, which prevents the target device from inheriting the expansion limit on a restore operation.

/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. You can use this qualifier with any operation (save, restore, copy, compare, or journal). If you specify /LIST by itself (not in conjunction with another operation), the input specifier must be a save set; you cannot specify an output specifier. You can use /LIST with either /BRIEF or /FULL command qualifiers. The default is /BRIEF. Do not use /LOG together with /LIST when the output for /LIST is directed to the terminal; you will receive confusing output.

Syntax

/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
[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.

/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.

Syntax

/[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

/MEDIA_FORMAT — **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, see 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.

Syntax

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 Integrity server system, 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.

/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.

Syntax

```
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

/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.

Syntax

```
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. VSI 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.

/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. /NOINCREMENTAL is valid only in BACKUP save operations. It is not related to the /INCREMENTAL qualifier, which is valid only in restore operations.

Syntax

```
/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

/OVERLAY — **Output File Qualifier**: Writes the input file over a file with an identical specification at the output location.

Syntax

input-specifier output-specifier/OVERLAY

Description

If BACKUP attempts to copy or restore a file when a file with an identical directory name, file name, type, and version number already exists, the new version of the file is written over the existing version. The file identification of the new version is the same as the file identification of the file that is overwritten.

The physical location of the file on disk does not change. If /OVERLAY is specified, and the new file is larger than the one already present, BACKUP allocates more blocks on the disk and extends the file.

When you do not use /OVERLAY, /REPLACE, or /NEW_VERSION, and the version number of the file being restored is identical to the version number of the existing file, BACKUP reports an error in copying or restoring the file.

Example

```
$ BACKUP DRA1:MAR30SAV.BCK/SAVE_SET [RECORDS...]/OVERLAY
```

The sequential-disk save set MAR30SAV.BCK is restored to the directory tree [RECORDS...]. If a file from the save set has a specification that is identical to a file that already exists in [RECORDS...], the /OVERLAY qualifier directs BACKUP to write over the existing version.

/OWNER_UIC

/OWNER_UIC — The /OWNER_UIC qualifier has been superseded by /BY_OWNER. VSI recommends that you substitute /BY_OWNER for /OWNER_UIC in command procedures and operator instructions. See the description of /BY_OWNER for more information.

Syntax

/OWNER_UIC

/PHYSICAL

/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.

Syntax

/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; VSI 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.

/PROGRESS_REPORT

/PROGRESS_REPORT — **Command Qualifier**: Use the /PROGRESS_REPORT qualifier to display the progress of a BACKUP operation on the current output device in every 'n' seconds. This qualifier expects an integer value from the user and does not have any default value. (When you use BACKUP to back up or restore data interactively, press **Ctrl/T** to display the progress of the operation.)

Syntax

/PROGRESS_REPORT=*n*

The value for *n* is the frequency of the message display, in seconds.

Example

```
$ BACKUP SAVESET.BCK/SAVESET DKA100:/IMAGE/PROGRESS_REPORT=60

%BACKUP-I-PROGRESS, progress report generated at 18-JAN-2006 18:07:55.08
Restoring file: DKA100:[KITS.CDSA]VSI-I64VMS-CDSA-T0202-134-1.PCSI
$COMPRESSED;1
Saveset volume:1, saveset block:1705 (32256 byte blocks)
  52.44MB restored out of 1.23GB, 4% completed
Restore rate: 895KB/sec, estimated completion time: 18:30:59.10

%BACKUP-I-PROGRESS, progress report generated at 18-JAN-2006 18:08:55.08
Restoring file: DKA100:[KITS.DWMOTIF]VSI-I64VMS-DWMOTIF-L0106-1.PCSI
$COMPRESSED;1
Saveset volume:1, saveset block:3547 (32256 byte blocks)
109.11MB restored out of 1.23GB, 8% completed
Restore rate: 967KB/sec, estimated completion time: 18:29:16.05
```

In this example, the command restores a save set called SAVESET.BCK into DKA100. When you specify /PROGRESS=60, a progress report is displayed on the screen every 60 seconds, indicating the progress of the operation.

/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.)

Syntax

```
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, see the *VSI 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 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: Return
```

   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.

/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.

Syntax

/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

1. $ **BACKUP/RECORD DBA1:[*...]/SINCE=BACKUP MTA0:13MAY.BCK**

2. $ **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

/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).

Syntax

/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.

/REPAIR

/REPAIR — Helps BACKUP to reset the RMS file attribute of a saveset file which gets corrupted while saveset is transferred through World Wide Web (WWW) or copied using FTP or compressed and uncompressed using ZIP. You can use the /REPAIR qualifier repair the saveset attributes.

Syntax

```
/REPAIR
```

/REPLACE

/REPLACE — **Output File Qualifier**: Replaces a file on the output specifier with an identically named file from the input specifier.

Syntax

```
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

/REWIND — **Input Save-Set Qualifier**: *See a separate description of /REWIND as an output save-set qualifier.* Rewinds the input tape reel to the beginning-of-tape marker before reading the input volume. **Input or Output Save-Set Qualifier**: As an input save-set qualifier, causes the input tape reel to be rewound (/REWIND) or not rewound (/NOREWIND) to beginning-of-tape (BOT) before BACKUP searches for the save-set name specified in the input specifier. As an output save-set qualifier, specifies that the output magnetic tape is to be rewound and initialized before the save operation begins (/REWIND) or that the tape is neither to be rewound nor initialized before the save operation begins (/NOREWIND). Initializing the tape removes access to any existing data on the tape. If you want to start processing at BOT, and the magnetic tape is already positioned beyond BOT, specify /REWIND.

Otherwise, the magnetic tape begins (or resumes) processing from the logical end-of-tape (EOT) marker. Use the /[NO]REWIND qualifier for magnetic tape save sets only.

Syntax

```
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

/REWIND — **Output Save-Set Qualifier**: *See a separate description of /REWIND as an input 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.

Syntax

```
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:

Label Specified in the Command Line	**Matching Volume Labels**
MAR	MAR, MAR_, MAR_nn
MAR_	MAR_, MAR_nn
MARK	MARK, MARKnn
MARKER	MARKER, MARKnn

You can specify more than one label with the /LABEL qualifier. If any label specified in the BACKUP command line matches the volume label of the tape and the tape is expired, BACKUP overwrites the volume label of the tape with the same volume label.

By overwriting the tape's volume label, BACKUP initializes the tape, removing access to any data that previously resided on the tape and preparing the tape to receive new data. During the initialization process, BACKUP writes the values specified with the output save-set qualifiers /TAPE_EXPIRATION, /PROTECTION, and /BY_OWNER to the volume header record. (If these qualifiers are not specified, the default tape expiration date is today, the default protection is none, and the owner UIC of the tape is the UIC of the current process.) After initializing the tape, BACKUP writes the save set to the tape.

If the label in the BACKUP command line did not match the volume label of the tape, BACKUP displays the following message and prompt on your terminal if you specified the command qualifier /NOASSIST, or on the operator terminal if you did not specify /NOASSIST:

```
%BACKUP-W-MOUNTERR, volume 'number' on 'device' was not mounted because
its label does not match the one requested
Specify option (QUIT, NEW tape or OVERWRITE tape)

BACKUP>
```

If you enter QUIT at the BACKUP> prompt, BACKUP aborts, unloads the magnetic tape, and issues the following message:

```
%BACKUP-F-ABORT, operator requested abort on fatal error
```

If you enter NEW at the BACKUP> prompt, BACKUP unloads the magnetic tape and issues the following prompt for a new tape:

```
%BACKUP-I-READYWRITE, mount volume 'volume-number' on
_'device-name': for writing Enter "YES" when ready:
```

If you enter OVERWRITE at the BACKUP> prompt, BACKUP overwrites the old volume label with the new volume label. (OVERWRITE instructs BACKUP to ignore the fact that either the tape has not expired or that the labels do not match.) By overwriting the tape's volume label, BACKUP initializes the tape, removing access to any data that previously resided on the tape and preparing the tape to receive new data.

During the initialization process, BACKUP writes the values specified with the output save-set qualifiers /TAPE_EXPIRATION, /PROTECTION, and /BY_OWNER to the volume header record. After initializing the tape, BACKUP writes the save set to the tape.

If the tape is not expired, BACKUP displays the following message and prompt on your terminal if you specified the command qualifier /NOASSIST, or on the operator terminal if you did not specify /NOASSIST:

```
%BACKUP-W-MOUNTERR, volume 'number' on 'device' was not mounted because
its expiration date is in the future
Specify option (QUIT, NEW tape or OVERWRITE tape)

BACKUP>
```

Always specify /REWIND when the output tape has a non-ANSI or non-ISO label or when the output tape has never been initialized.

The /NOREWIND qualifier directs BACKUP to compare the volume label of the tape with the label you specified in the BACKUP command before performing the save operation. You can specify a label explicitly with the /LABEL qualifier; otherwise, BACKUP uses the first six characters of the save-set name as the volume label. If the volume label does not match the label you specified, BACKUP displays the following message and prompt on your terminal if you specified the command qualifier /NOASSIST, or on the operator terminal if you did not specify /NOASSIST:

```
%BACKUP-W-MOUNTERR, volume 'number' on 'device' was not mounted because
its label does not match the one requested
Specify option (QUIT, NEW tape or OVERWRITE tape)

BACKUP>
```

If you choose the OVERWRITE option, BACKUP ignores the fact that the volume labels do not match. If the labels match, or if you choose the OVERWRITE option, BACKUP winds the tape forward to the logical end-of-tape (the end of the last save set stored on the tape) and writes the save set to the tape. If the logical end-of-tape is also the physical end of the tape, BACKUP requests a new tape. Because BACKUP searches for the end of data on the tape, you cannot write a new save set to a tape if it ends with a save set that is continued onto another tape.

Although the /NOREWIND qualifier does not initialize the first tape in a multivolume save set, BACKUP initializes subsequent tapes in a multivolume save set. BACKUP ensures that the tape is expired and that the tape labels match before initializing subsequent volumes in a multivolume save set.

The default is /NOREWIND. You must specify /REWIND to rewind and initialize a magnetic tape volume.

Example

```
$ BACKUP
_From: *.RNO
_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

/SAVE_SET — **Input Save-Set Qualifier**: *See a separate description of /SAVE_SET as an output 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.

Syntax

```
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.

/SAVE_SET

/SAVE_SET — **Output Save-Set Qualifier**: *See a separate description of /SAVE_SET as an input 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.

Syntax

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

/SELECT — **Input Save-Set Qualifier**: Selects the specified files for processing.

Syntax

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.

/SINCE

/SINCE — **Input File-Selection Qualifier**: Selects files dated equal to or later than the specified date and time.

Syntax

```
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.

/SIZE

/SIZE — **Command Qualifier**: The logical volume size is recorded in the save-set header during a backup operation. By default, during a restore operation, the logical volume size is not preserved because restoring a save set of 2 GB to a 4-GB disk, for example, results in only 2 GB of available disk space. The /SIZE qualifier, however, allows you to preserve the logical volume size on the target device. Alternatively, you can specify the logical volume size of the target device by using the optional keyword value *n*.

Syntax

/SIZE (=[*n*])

Description

When you specify /SIZE, the geometry of the target device is determined by the logical size of the source original rather than calling $GETDVI to find out the physical limits of the device.

The /SIZE qualifier can use an optional value *n* as the new logical size of the target device. This new value overrides any existing value in the save set. (This matches the way the /SIZE qualifier in the INITIALIZE utility works.)

Example

```
$ BACKUP LEEHE.BCK/SAVE DKA100:/IMAGE/SIZE=8000000
```

In this example, a save set is restored to device DKA100. The operation initializes the logical volume size of DKA100 to be 8,000,000 blocks.

/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.

Syntax

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.

VSI 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

/TRUNCATE — **Command Qualifier**: Controls whether a copy or restore operation truncates a sequential output file at the end-of-file (EOF) when creating it.

Syntax

```
/[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.

/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.

Syntax

/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

/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.

Syntax

```
/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.

Examples

1. ```
   $ BACKUP
   _From: DBA1:[000000...]
   _To: MTA0:13NOVBAK.BCK,MTA1:/LABEL=(DLY101,DLY102)
   ```

   This BACKUP command saves the entire contents of the disk DBA1 onto a multivolume tape save set. This command does not initialize the first tape in the save set, but does initialize subsequent tapes. The first tape in the save set must be labeled DLY101 and the second tape in the save set must be labeled DLY102.

2. ```
   $ BACKUP [PLI.WORK]/SINCE=YESTERDAY/MODIFIED [PLI.SAV]
   ```

   This BACKUP command copies selected files in the directory [PLI.WORK] to the directory [PLI.SAV]. BACKUP processes only those files that have been modified since 24 hours preceding midnight last night. Note that the /MODIFIED qualifier is not required, as its action is the default when the /SINCE qualifier is specified.

3. `$ BACKUP DBA2:[PLI.WORK]*.*;  DBA3:[SAVE]23MAR82.BCK/SAVE_SET`

   This BACKUP command saves the highest version of each file in directory [PLI.WORK] to a disk save set on DBA3. The /SAVE_SET qualifier is required because the output save set is on a Files-11 disk.

4. `$ BACKUP/PHYSICAL DYA0:  DYA1:`

   This BACKUP command copies the entire RX02 floppy disk mounted on device DYA0 to the RX02 floppy disk mounted on device DYA1. Both devices must have been mounted with the DCL command MOUNT/FOREIGN.

5. ```
   $ BACKUP
   _From: DB1:[SCHED]*.*
   _To: DENVER::DBA2:[SAVE]SCH.BCK/SAVE_SET
   ```

   This BACKUP command saves all files in the directory SCHED on disk DB1 at the local node to the network save set SCH.BCK at remote node DENVER. The /SAVE_SET qualifier is required to identify the output specifier as a save set on a Files-11 disk.

6. `$ BACKUP/LIST DENVER::DBA2:[SAVE]SCH.BCK/SAVE_SET`

   This BACKUP command lists the BACKUP summary information, the original BACKUP command used, and the file name, size, and creation date for each file in the save set created in the previous example. The /SAVE_SET qualifier is required to identify the input specifier as a save set on a Files-11 device.

7. `$ BACKUP MTA0:NOV2SAVE.BCK/REWIND/SELECT=[USER...] [USER...]`

   This command restores files from the magnetic tape save set named NOV2SAVE.BCK to the directory tree from which they were saved (the [USER] directory tree). The /REWIND qualifier directs BACKUP to rewind the tape to the beginning-of-tape before beginning the restore operation.

8. ```
   $ BACKUP WORK$:[TESTFILES...]*.*;*  MUA0:TEST.SAV  -
   _$ /MEDIA_FORMAT=COMPACTION /REWIND
   ```

   This command saves all files in the directory [TESTFILES] and its subdirectories to 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.

9. ```
   $ 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 tasks:

   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.

10.
```
$ 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.

/WINDOW

/WINDOW — Specifies the number of mapping pointers to be allocated for file windows.

Syntax

/WINDOW=FULLLIMITED

Description

By default, file window has 7 mapping pointers and it increases as file is more fragmented. Window is taken from nonpaged pool and is charged against process BYTLM quota. This may result in performance degrade when BACKUP tries to map the 8th extent of the file. Contiguous files are always completely mapped, but noncontiguous files may or may not be completely mapped.

You can use the following keywords with the /WINDOW qualifier:

FULL	Default full mapping.
LIMITED	Optional limited size window.

7.6. BACKUP Examples

Table 7.3, "Save Operation Quick Reference" 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`

Command Action	Command Format and Example

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`

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, "Restore Operation Quick Reference" 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`

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, "Copy Operation Quick Reference" 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

Command Action	Command Format and Example
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`

Command Action	Command Format and Example
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, "Compare Operation Quick Reference" 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

Command Action	Command Format and Example
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, "List Operation Quick Reference" 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

Command Action	Command Format and Example
Lists the files in a save set at the terminal	BACKUP/LIST save-set-specifier `$ BACKUP/LIST MTA0:1618FEB2.BCK`
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`

Chapter 8. COPY/RECORDABLE_MEDIA (CDDVD) Utility

8.1. CDDVD Description

The COPY/RECORDABLE_MEDIA (CDDVD) Utility allows users to create Compact Disk (CD) and Digital Versatile Disk (DVD) media directly on OpenVMS, using an optional optical disk recorder.

CDDVD generates ISO/IEC 10149 Mode 1 (2048-byte blocks, data) single-session optical media recordings.

CDDVD supports the recording of various optical media formats, including CD Recordable (CD-R), CD Rewritable (CD-RW), DVD Recordable (DVD+R) and DVD Rewritable (DVD+RW) formats. For a successful recording operation, one or more of these formats must be available within the target optical disk recording device. Compatible recording media must also be loaded into the recording device.

8.1.1. Media Limitations

As part of the recording operation, you must create a master for the recording operation, or otherwise provide all the data required for the entire target disk. This master must fit onto the target media, and the input must also fit evenly into the target disk sector. The input disk image or the input master device must both fit onto the target recording media, and the recording operation cannot partially fill the last sector of the target media.

Unlike a traditional read-write random-access storage device, you must completely populate the recording media in a single continuous recording operation. You cannot incrementally update the optical media through a series of individual copy operations, nor can you interrupt a recording operation once it has begun. You cannot modify nor partially erase the contents of the recorded optical media, and you cannot access the media for write access using DCL commands such as INITIALIZE/ERASE nor through operations that include opening a file on the media for read-write access.

With rewritable media, you can reformat and repopulate the media.

8.1.2. Mastering

A recording master is usually created by using the OpenVMS Logical Disk (LD) utility and an associated LD pseudo-disk device. The structures and files to be included on the optical media eventually are staged within the LD device using DCL tools and commands such as INITIALIZE/ERASE, CREATE/DIRECTORY, and COPY.

Opens the specified input disk image file or input master device and records the entire contents to the specified CD-R, CD-RW, DVD+R, and DVD+RW media formats.

Syntax

COPY/RECORDABLE_MEDIA source-path-name target-path-name

Parameter

source-path-name

This is the data source for the recording operation.

Specify the name of a disk file containing a disk image to be copied onto the target recording media, or the device name of the input device containing the disk volume master for the recording.

On OpenVMS systems, this is usually a Logical Disk (LD) Utility LDAu: device.

target-device-name

The device name of the target recordable media device.

This is usually the name of an ATAPI (DQcu:) SCSI (DKcu:), or USB (DNcu:): CD-R/RW or DVD +R/RW recording device, or both.

Description

The COPY/RECORDABLE_MEDIA command records the entire contents of the specified input disk image file or input device onto the media loaded into the specified output CD or DVD recording device.

The output media format is sensed automatically, and the utility automatically configures the recording appropriately for the particular target device and for the output media that are loaded.

You cannot record more than the capacity of the target media permits. Therefore, you need to select the size of the input disk image or the master appropriate for the capacity of the target media. The input data source must also be an even multiple of the sector size on optical media; the size of the input must be a multiple of four blocks.

The recording operation is independent of the input volume structure or input file data used for the master, and is based solely on the block-level contents of the specified input master.

Qualifiers

/BELL

Sounds an audible signal when the requested recording operation completes successfully.

/FORMAT[=keyword]
/NOFORMAT (default)

Requests that rewritable (RW) media be formatted or reformatted prior to use. This qualifier is required for writing to blank rewritable media or rewriting rewritable media.

If the target media cannot be formatted, this command qualifier is ignored.

If not specified, the appropriate keyword is automatically selected for the fastest formatting speed available for the target recording media.

Table 8–1 lists available keywords.

Table 8.1. Keywords for the /FORMAT Qualifier

Keyword	Function, Comments
WAIT	Applies to DVD+RW. The default for the /FORMAT qualifier is not to wait for the formatting to complete because waiting is usually unnecessary and far slower. Selecting WAIT causes the entire format to run synchronously to completion before beginning the recording operation. The default is to: ● Operate asynchronously ● Perform background formatting ● Run both the media format operation and the recording operations concurrently
ERASE	Applies to CD-RW. The default for the /FORMAT qualifier is to perform a quick erasure because a full erasure is usually both unnecessary and far slower. Selecting ERASE causes the CD-RW rewritable disk to be entirely erased as part of the format operation. This erasure is performed and is completed before the recording operation begins. The default is to perform a quick erasure.

/LOG (default)
/NOLOG

Shows basic device information and the progress of the recording operation. Use /NOLOG to disable the normal output from the utility.

/SPEED

If you must use the lower-speed or poor-quality CD recording media, the /SPEED qualifier is often required for successful completion of the recording process. You might need to select a recording speed below the rated speed of the CD drive itself. Specifically, you might need to select a recording speed that is compatible with both the CD drive and the CD recording media loaded in the drive.

The /SPEED qualifier accepts a single keyword for a requested device speed:

1X
2X
4X
8X
16X
32X
MAXIMUM

The CDDVD utility attempts to match the requested speed to a speed that the device supports. (Not all devices support all speeds, including the lowest speed, 1X, or the highest speed available.) The

default speed is the maximum speed that the target device supports. DVD+R/RW drives select the maximum recording speed based on information encoded on the media.

You need to specify this qualifier only under one of the following circumstances:

- When incompatibilities or recording errors are reported during a previous failed recording operation.
- If the CD media in use has a rated recording speed below the drive default recording settings.
- If CDDVD application, processor, or system I/O performance constraints exist.

CD drives can select speeds faster than those supported by the particular media loaded in the drive. VSI recommends that you select only media that match the recording capabilities of the drive. In other words, do not attempt to exceed the recording speed limits of the particular CD media. Selecting faster media will not make a slow drive record any faster, and selecting faster speeds with slow media can trigger recording errors and corrupt media.

If the recording process fails during the recording operation, discard the write-once media and try a slower recording speed. (Note that you can attempt to reformat and rerecord on rewritable media.)

/VERIFY

Specifies that the contents of the output media be compared to the contents of the input source after the recording operation. Any data comparison errors detected are displayed.

/WRITE (default)
/NOWRITE

Allows you to test the system and device I/O throughput and the command syntax without recording on the target media.

If you specify /NOWRITE and if the target drive supports the underlying test-write hardware capability, all I/O operates as usual although /NOWRITE disables writing to the media.

/WRITE is the default, and causes the target optical media to be written.

Examples

1.
```
$
$ COPY/RECORDABLE_MEDIA -
$_ [/BELL] -
$_ [/DATA_CHECK=WRITE] -
$_ [/DIAGNOSTICS=(DETAILS,COMMANDS,ALL)] -
$_ [/EXTENSIONS[=(keywords)]] -
$_ [/[NO]LOG] -
$_ [/SPEED={1X|2X|4X|8X|16X|32X|MAXIMUM}] -
$_ source-path-name target-device-name
$
```

This example shows the generic format of the COPY/RECORDABLE_MEDIA command.

2.
```
$ COPY/RECORDABLE_MEDIA/FORMAT LDA1 DQA1

HP OpenVMS CD-R/RW and DVD+R/RW Utility V1.0-0
Copyright 1976, 2006 Hewlett-Packard Development Company, L.P.
```

```
Output device vendor: HP
Output device product name: DVD Writer 740b
Commencing media format operation
Formatting may require up to an hour
Output medium format: DVD+RW
Input data being read from: LDA1:
Output data being written to: DQA1:
Input data size: 1200000 blocks

Starting operation at: 15:28:16

16 sectors written

30000 sectors written; estimated completion in 00:06:52; at 15:35:55
37000 sectors written; estimated completion in 00:06:54; at 15:36:07
46000 sectors written; estimated completion in 00:06:36; at 15:36:03
57000 sectors written; estimated completion in 00:06:08; at 15:35:51
71000 sectors written; estimated completion in 00:06:00; at 15:36:04
88000 sectors written; estimated completion in 00:05:26; at 15:35:56
110000 sectors written; estimated completion in 00:04:55; at 15:35:58
137000 sectors written; estimated completion in 00:04:12; at 15:35:56
171000 sectors written; estimated completion in 00:03:14; at 15:35:48
213000 sectors written; estimated completion in 00:02:10; at 15:35:48
266000 sectors written; estimated completion in 00:00:54; at 15:35:50
300000 sectors written; operation completed

Operation completed at: 15:35:47
Elapsed time for operation: 00:07:30
Synchronizing with output device cache
Processing completed
```

This example demonstrates recording the contents of LDA1: device onto the DVD+RW media loaded into device DQA1:.

Chapter 9. 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 Integrity servers:

- 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

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.

Syntax

```
VMS_BCFG driver | boot [dump][add # device-name -fl x,y "desc"][rm #] [mv # #
```

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 $1$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

VMS_SET — Sets the dump device and the debug device to the specified OpenVMS device name.

Syntax

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

VMS_SHOW — Displays the equivalent OpenVMS device name for devices mapped by the EFI console.

Syntax

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. ```
   fs1:\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. ```
   fs1:\efi\vms> vms_show dump
   VMS: DGA3730      COMPAQ HSV110 (C)COMPAQ3014
   EFI: fs1: Acpi(000222F0,200)/Pci(1|1)/Fibre(50001FE10011B15D),Lun(D)
   ```

   This example shows the settings for the OpenVMS DUMP_DEV device.

Chapter 10. 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.

10.2. ELV Usage Summary

You can execute ELV commands directly from the DCL command level or from ELV's interactive shell mode.

Syntax

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, see 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

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.

Syntax

```
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 isSYS$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 *VSI 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.

Keyword	Description
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: • System power failure restarts • Time stamps • System startups • $SNDERR messages (system service to send messages to error log)

Keyword	Description
	• Operator messages • Network messages • ERRLOG.SYS created
CPU_ENTRIES	Reject CPU-related entries. CPU entries include the following entry types: • SBI alerts/faults • Undefined interrupts • MBA/UBA adapter errors • Asynchronous write errors • UBA errors
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 VSI 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.

/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:

Keyword	Description
	• System power failure restarts • Time stamps • System startups • $SNDERR messages (system service to send messages to error log) • Operator messages • Network messages • ERRLOG.SYS created
CPU_ENTRIES	Select CPU-related entries. CPU entries include the following entry types: • SBI alerts/faults • Undefined interrupts • MBA/UBA adapter errors • Asynchronous write errors • UBA errors
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 VSI 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, see *Section 10.2, "ELV Usage Summary"*.

/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, see *Section 10.3, "Understanding Categories of Events"*.

/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 *VSI OpenVMS User's Manual* for details on specifying times.

If you omit a date and time, TODAY is used.

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

DUMP — Writes events from one or more binary error log files to a single new ASCII output file in an OpenVMS dump-style format.

Syntax

```
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 *VSI 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.

/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.

Keyword	**Description**
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: • System power failure restarts • Time stamps • System startups • $SNDERR messages (system service to send messages to error log) • Operator messages • Network messages • ERRLOG.SYS created
CPU_ENTRIES	Reject CPU-related entries. CPU entries include the following entry types: • SBI alerts/faults • Undefined interrupts • MBA/UBA adapter errors • Asynchronous write errors • UBA errors
DEVICE_ERRORS	Reject device error entries.
ENVIRONMENTAL_ENTRIES	Reject environmental entries.
MACHINE_CHECKS	Reject machine check entries.

Keyword	Description
MEMORY	Reject memory errors.
SNAPSHOT_ENTRIES	Reject snapshot entries.
SYNDROME	Reject firmware-generated entries that describe a symptom set used by VSI 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.

/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: ● System power failure restarts ● Time stamps ● System startups ● $SNDERR messages (system service to send messages to error log) ● Operator messages ● Network messages ● ERRLOG.SYS created
CPU_ENTRIES	Select CPU-related entries. CPU entries include the following entry types: ● SBI alerts/faults ● Undefined interrupts ● MBA/UBA adapter errors

Keyword	Description
	• Asynchronous write errors • UBA errors
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 VSI 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, see *Section 10.2, "ELV Usage Summary"*.

/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, see *Section 10.3, "Understanding Categories of Events"*.

/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, see *Section 10.3, "Understanding Categories of Events"*.

/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 *VSI 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.

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.

Syntax

EXIT

Example

```
ELV> EXIT
$
```

The command in this example terminates the ELV session and returns control to the DCL command level.

HELP

HELP — Provides online help information for using ELV.

Syntax

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.

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, "ELV Sample Reports"* contains examples of reports.

Syntax

```
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, "Report Formats"*.

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
Standard	A report that contains detailed bit-to-text translation information for each event. To specify the detail level of a standard report, use the/BRIEF, /FULL, or /ONE_LINE qualifier with the TRANSLATE command. In addition to these qualifiers, you can use the /TERSE qualifier to obtain a report that contains less interpretation of the data, regardless of detail level. At a minimum, the following header information is included for each event in a standard report: ● Event Number ● Event Type ● Timestamp ● Node

Format	Description
	• Event Class With the exception of Event Type, the header information is useful when using the following event filtering and interval qualifiers:/BEFORE, /ENTRY, /EXCLUDE, /INCLUDE, /[NO]NODE, /SINCE. For Event Class, see the keywords listed for the /EXCLUDE or /INCLUDE qualifier. **Standard Report Detail Levels:** • If you specify /ONE_LINE, the header information is the only event information that is included in a standard report. • If you specify /BRIEF, only the most essential event information is included with the header information. • If you do not specify a standard report detail level qualifier, only the most commonly useful event information is included with the header information. • If you specify /FULL, all event information is included with the header information.
Summary	Summary information is broken down by node where applicable, followed by summary information for the entire file. See the example following this table.

```
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 *VSI 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.

Keyword	Description
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: ● System power failure restarts ● Time stamps ● System startups ● $SNDERR messages (system service to send messages to error log) ● Operator messages ● Network messages

Keyword	Description
	• ERRLOG.SYS created
CPU_ENTRIES	Reject CPU-related entries. CPU entries include the following entry types: • SBI alerts/faults • Undefined interrupts • MBA/UBA adapter errors • Asynchronous write errors • UBA errors
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 VSI 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.

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:

Keyword	Description
	• System power failure restarts • Time stamps • System startups • $SNDERR messages (system service to send messages to error log) • Operator messages • Network messages • ERRLOG.SYS created
CPU_ENTRIES	Select CPU-related entries. CPU entries include the following entry types: • SBI alerts/faults • Undefined interrupts • MBA/UBA adapter errors • Asynchronous write errors • UBA errors
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 VSI 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, see *Section 10.2, "ELV Usage Summary"*.

/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, see *Section 10.3, "Understanding Categories of Events"*.

/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 *VSI 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.
- 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, "/TERSE Example"*.

Examples

1. `$ ANALYZE/ERROR_LOG/ELV TRANSLATE /ONE_LINE /PAGE`

   The command in this example translates selected events from the default SYS $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 SYS $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 SYS $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 SYS $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 SYS $ERRORLOG:ERRLOG.SYS file and writes the resulting summary report to the terminal.

   The selected events include only the VOLUME_CHANGES event class.

WRITE

WRITE — Performs an image copy of events from one or more binary error log files to a single new binary error log file.

Syntax

```
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 *VSI 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.

Keyword	Description
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:

Keyword	Description
	• System power failure restarts • Time stamps • System startups • $SNDERR messages (system service to send messages to error log) • Operator messages • Network messages • ERRLOG.SYS created
CPU_ENTRIES	Reject CPU-related entries. CPU entries include the following entry types: • SBI alerts/faults • Undefined interrupts • MBA/UBA adapter errors • Asynchronous write errors • UBA errors
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 VSI 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.

/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: • System power failure restarts • Time stamps • System startups • $SNDERR messages (system service to send messages to error log) • Operator messages • Network messages • ERRLOG.SYS created
CPU_ENTRIES	Select CPU-related entries. CPU entries include the following entry types: • SBI alerts/faults • Undefined interrupts • MBA/UBA adapter errors • Asynchronous write errors • UBA errors
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 VSI 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, see *Section 10.2, "ELV Usage Summary"*.

/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, see *Section 10.3, "Understanding Categories of Events"*.

/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 *VSI 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.

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, "Report Formats"* 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
```

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$COMMON:[SYSEXE.ERRLOGS]EXAMPLE.DAT;1
EVENT  EVENT_TYPE_____________________________  TIMESTAMP_______________  NODE__
 EVENT_CLASS__________
1      Volume Mount                            14-AUG-2003 13:31:39.12  FRANZ   VOLUME_CHANGES
DESCRIPTION_____________________________   RANGE___   VALUE_____________
 TRANSLATED_VALUE_______
   Logging CPU                                        3
   Number of CPU's in Active Set                      4
   System Marketing Model                             1968                COMPAQ
 AlphaServer GS160
   Error Mask                                <31:00>:   0x00000003
   Operating System Version                           X9WY-SSB
   Device Unit Number                                 200
   Device Generic Name                                FRANZ$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_____________________________   RANGE___   VALUE_____________
 TRANSLATED_VALUE_________
   Hardware Architecture                              4                   Alpha
   Hardware System Type                               35                  Wildfire
   Logging CPU                                        3
   Number of CPU's in Active Set                      4
   System Marketing Model                             1968                COMPAQ
 AlphaServer GS160
   Error Mask                                <31:00>:   0x00000003
```

```
   Seconds Since Boot                                17
   Error Sequence Number                             46
   DSR String                                        AlphaServer GS160 6/731
   Operating System Version                          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
```

10.5.4. /FULL Example

The report produced when you use the /FULL qualifier includes all event information along with the header information for each event.

```
Output for SYS$COMMON:[SYSEXE.ERRLOGS]EXAMPLE.DAT;1
EVENT  EVENT_TYPE_____________________  TIMESTAMP_____________  NODE__
 EVENT_CLASS_____________
1      Volume Mount                     14-AUG-2003 13:31:39.12  FRANZ   VOLUME_CHANGES
DESCRIPTION_____________________________   RANGE___   VALUE_____________
 TRANSLATED_VALUE________
   Operating System Type                             2                   OpenVMS
   Hardware Architecture                             4                   Alpha
   Vendor ID                                         0x00000DEC          hp
   Hardware System Type                              35                  Wildfire
   Logging CPU                                       3
   Number of CPU's in Active Set                     4
   System Marketing Model                            1968                COMPAQ AlphaServer
 GS160
   Error Mask                              <31:00>:  0x00000003
   Seconds Since Boot                                17
   Chip Type                                         11                  EV67 (21264A)
   Error Sequence Number                             46
   DSR String                                        AlphaServer GS160 6/731
   DDR String
 00000000000000000000000000000000000000000000000000000000000000000000
                                                     0x0000
   System Serial Number                              0x00000000000000000000000000000000000000000000
   Time - ISO 8601 Format                            20030814133139, 12-0400
   Operating System Version                          X9WY-SSB
   Computer Name                                     FRANZ
   Owner UIC of the Volume                           65537
   Unit Operation Count                              378
   Device Unit Number                                200
   Device Generic Name                               FRANZ$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
```

10.5.5. /TERSE Example

The following modifications are made to the standard report when you use the /TERSE qualifier:

- Translations of field values are omitted, resulting in a more compressed report layout.
- Field values are displayed in hex rather than their default format.
- Fields that are usually suppressed when they contain a value of zero are output regardless of their value.
- A concatenated field is output as separate fields rather than as a single field.

The example that follows shows the results of using the /TERSE qualifier with the /FULL qualifier. You can also use /TERSE with standard reports of any detail level to output event information in terse format.

```
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
                                                        372F362030363153472O7
                                                        00000000000000003133
                                                        0x0
   DDR String                                           00000000000000000000
                                                        00000000000000000000
                                                        00000000000000000000
                                                        0x0
   System Serial Number                                 00000000000000000000
                                                        0x000000000000000000
   Time - ISO 8601 Format                               33331343138303303032
                                                        3034302D32312C3933313
                                                        0x30
   Operating System Version                             0x4253532D59573958
   Computer Name                                        000002020205A4E415246
                                                        0x00000000000
   Owner UIC of the Volume                              0x00010001
   Unit Error Count                                     0x00000000
   Unit Operation Count                                 0x0000017A
   Device Unit Number                                   0x00C8
   Device Generic Name                                  000424B44245A4E415246
                                                        00000000000000000000
                                                        0000000000000000000
                                                        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
```

Chapter 11. InfoServer Utility (Alpha and Integrity servers Only)

11.1. InfoServer Description

On Alpha and Integrity server systems, the InfoServer application allows you to create virtual disk services on the local area network.

Virtual disk services can be created for the following devices:

- DVD drives
- Disk drives: SCSI, Fibre Channel, or IDE
- CD drives
- Partitions (the equivalent of container files)

Comparison of InfoServer Hardware and the InfoServer Application

The new InfoServer application on OpenVMS differs from previous InfoServer hardware in a number of important ways. Some of the most notable are the following:

- The use of DCL-style command syntax
- The requirement that a device must be mounted systemwide before you can create a service for it
- Support for creating services for DVD drives
- No support for tape devices
- No support for CD-R (CD-recordable) drives.
- No automount support

11.2. InfoServer Usage Summary

ESS$INFOSERVER is the user interface for the LASTport/Disk server implemented as an application on OpenVMS. Its behavior is similar to that of the hardware InfoServer product.

You can use the InfoServer utility commands to do the following:

- Create and delete services for virtual disk devices on a local area network.
- Save a list of active InfoServer services.
- Modify the attributes of existing services.
- Display information about the server and the nodes connected to services on the server.
- Display service-specific information about one or more services.

- Start the LASTport/disk server and set various server and cache characteristics.

To run the Infoserver, enter the following command:

```
$ RUN SYS$SYSTEM:ESS$INFOSERVER
```

The system displays the following prompt:

```
InfoServer>
```

If your system does not display the InfoServer prompt but displays an error message instead, ask your system manager to start the InfoServer server.

Following the InfoServer prompt, you can enter any InfoServer command; for example:

```
InfoServer> SHOW SERVER
```

You can also enter HELP to obtain help on InfoServer commands within the utility:

```
InfoServer> HELP SHOW SERVER
```

Command

Specifies an InfoServer command. If you do not specify a command, the utility displays its prompt and waits for command input until you exit the utility.

11.3. InfoServer Commands

The following sections describe and provide examples of InfoServer commands.

CREATE SERVICE

CREATE SERVICE — Creates a service for a specified device or partition.

Syntax

CREATE SERVICE serviceName device-or-partitionName

Description

Usage rules:

- All devices must be mounted systemwide to prevent them from being dismounted when a process logs out.
- A device that has read/write service must be mounted /FOREIGN so that it is not visible to OpenVMS.
- A device that has read-only service must be mounted with either the /NOWRITE qualifier or the /FOREIGN qualifier so that no one can change it locally.
- A partition can be served off a disk that is mounted for either read-only or read/write access to OpenVMS.
- Support for partitions is limited in this release.

Parameters

serviceName

The name by which the service is known to the local area network. The service name can consist of alphanumeric characters and dollar signs ($). It can be 255 characters or fewer in length.

device-or-partitionName

The name of the OpenVMS disk device or partition name is the name being served to the local area network. The name of the device or partition must have been created previously.

Explanations of device and partition names follow.

- Device names

  Devices served to the local area network are OpenVMS disk devices; use OpenVMS device names when you specify an InfoServer device name.

  A disk specification must end with a colon.

- Partition names

  Partitions are container files that are served to the network. As such, they have OpenVMS file names with a default file type of .ESS$PARTITION. Partition names, including the device, directory, and file name, can be no more than 242 characters in length.

  Support for partitions is limited in this version. VSI strongly suggests that you use LD devices to support partitioned hard drives. See the DCL command LD HELP for more information.

Qualifiers

/CLASS=className

Specifies a subset of the complete LASTport Disk (LAD) name space.

The purpose of class names is to subdivide name spaces so that clients see only those names that are meaningful to them. The use of class names also allows two services to have the same name and not conflict with one another.

For example, you can use different class names for different on-disk structures that several client systems use. You might use SERVICEA/CLASS=ODS-2 for some client systems and SERVICEA/CLASS=ISO_9660 for other client systems. The service has the same name (SERVICEA), but the class names are different.

The class name you use depends on the client systems that connect to the service being created. The default class name is ODS_2. For example, OpenVMS systems use the ODS_2 name space when attempting to mount an InfoServer device. Note that OpenVMS clients can solicit only those services that are in the ODS_2 service class.

Valid class names are the following:

```
V2.0           Names understood by PCSA MS-DOS Clients
Unformatted    Virtual disk has no format
MSDOS          MSDOS virtual disks
ODS_2          VMS virtual disks
UNIX           UNIX virtual disks
```

```
ISO_9660        ISO 9660 CD format
HIGH_SIERRA     MS-DOS CD format
APPLE           Macintosh HFS format
SUN             Sun format
```

/ENCODED_PASSWORD=hexstring

The SAVE command creates this qualifier. Because passwords are not stored in plain text, the hashed password value is written out as part of the SAVE operation so that the service can be recreated without revealing the password.

Note that if you edit the command procedure that the SAVE command creates and change the service name, the encoded password value is no longer valid. You need to set another password on the service using the /PASSWORD qualifier.

/PASSWORD=passwordstring
/NOPASSWORD (default)

Specifies an optional access control password for the service. The client system must specify the password to access the service.

The password-string can be 39 alphanumeric ASCII characters in length. If no password is specified, the client system is not required to provide a password to access the service.

The text password is hashed and stored in encrypted form in memory with the other service information.

/RATING=DYNAMIC
/RATING=STATIC=value

Clients use the service rating to select a service in the case of multiple matching services. The service with the highest service rating is selected.

The system adjusts the dynamic service rating based on load. You can also set a static rating between 0 and 65535. The system does not adjust static ratings.

One use of static ratings is to migrate clients from one copy of a service to another. If you set a static rating of 0 on services you want to migrate clients away from, no new clients will connect to a 0-rated service; instead, they will connect to higher-rated services. When all current clients have disconnected from a service, you can safely delete it.

/READAHEAD (default)
/NOREADAHEAD

When a disk read is required to fill a cache block, the /READAHEAD qualifier specifies that the read is to be from the first block requested to the end of the bucket boundary. Read-ahead can speed up sequential operations by preloading disk blocks that are needed into the cache.

If you specify both the /READAHEAD and the /READBEHIND qualifiers, any block requested within a cache bucket causes the entire bucket range of blocks to be read into the cache.

/READBEHIND
/NOREADBEHIND (default)

When a disk read is required to fill a cache block, the /READBEHIND qualifier specifies that the read is to include all blocks from the beginning of the cache bucket boundary up to and including the requested blocks.

If you specify both the /READAHEAD and /READBEHIND qualifiers, any block requested within a cache bucket causes the entire bucket range of blocks to be read into the cache.

/READERS=number (default is READERS=1000)
/NOREADERS

Specifies the maximum number of simultaneous client connections allowed for read access. The default is 1000 readers. A value of 0 indicates write-only access.

If a client requests read-only or read/write access to a service, the system counts this as one reader.

/WRITERS
/NOWRITERS (default)

Specifies that the service is to allow access to a single writer.

Examples

1.
```
$ SHOW DEVICE MOVMAN$DQA0:/full



Disk MOVMAN$DQA0:, device type Compaq  CRD-8322B, is online, file-
oriented
    device, shareable, served to cluster via MSCP Server, error logging
 is
    enabled.
Error count                 0    Operations completed
Owner process              ""    Owner UIC                  [SYSTEM]
Owner process ID     00000000    Dev Prot        S:RWPL,O:RWPL,G:R,W
Reference count             0    Default buffer size             512
Total blocks         16515072    Sectors per track                63
Total cylinders         16384    Tracks per cylinder              16

$ MOUNT/SYSTEM dqa0 OVMSIPS11

Volume is write locked
OVMSIPS11 mounted on _MOVMAN$DQA0:

$ InfoServer
InfoServer> CREATE SERVICE VMS_SIPS_V11 _MOVMAN$DQA0:

%INFOSRVR-I-CRESERV, service VMS_SIPS_V11 [ODS-2] created for
_MOVMAN$DQA0:.
```

   This example shows how to create a service for a CD device:

   - The SHOW DEVICE …/FULL command displays a complete list of information about the _MOVMAN$DQA0 CD.
   - The MOUNT/SYSTEM command mounts the OVMSIPS11 volume on the _MOVMAN $DQA0: CD.
   - The InfoServer CREATE SERVICE command creates the VMS_SIPS_V11 service on the _MOVMAN$DQA0 CD.

2.
```
$ LD CREATE KIT1/SIZE-100000
$ DIRECTORY KIT1
```

```
Directory DKB0:[DISKS]
KIT1.DSK;1       100000/100008   29-APR-2005 14:14:43.49
Total of 1 file, 100000/100008 blocks.

$ LD CONNECT KIT1

%LD-I-UNIT, Allocated device is MOVMAN$LDA1:

$ INITIALIZE/SYSTEM MOVMAN$LDA1: kit1
$ MOUNT/SYSTEM/NOWRITE MOVMAN$LDA1: kit1

%MOUNT-I-MOUNTED, KIT1 mounted on _MOVMAN$LDA1:

$ CREATE SERVICE TEST_KIT_1 MOVMAN$LDA1:

%INFOSRVR-I-CRESERV, service TEST_KIT_1 [ODS-2] created for
_MOVMAN$LDA1:
```

This example shows how to create a service for a logical disk (LD) device:

- The LD CREATE KIT1 command creates a contiguous file, KIT1, that can be used as a logical disk.
- The DIRECTORY KIT1 command provides information about KIT1.
- The LD CONNECT KIT1 connects the logical disk file, KIT1, to the logical disk device MOVMAN$LDA1:.
- The INITIALIZE command formats the MOVMAN$LDA1: LD device.
- The MOUNT command makes the LD device available for processing.
- The CREATE SERVICE command creates the TEST_KIT_1 service on the _MOVMAN$LDA1 LD device.

DELETE SERVICE

DELETE SERVICE — Deletes one or more services.

Syntax

DELETE SERVICE serviceName device-or-partitionName

Parameters

serviceName

The name by which the service is known to the local area network. The service name can consists of alphanumeric characters and dollar signs ($). It can be up to 255 characters long. Wildcards are permitted in this command.

device-or-partitionName

The name of the OpenVMS disk device or partition is the name as it is to be known to the local area network. The name of the device or partition that you enter must have been created previously.

- Device names

  Devices served to the local area network are OpenVMS disk devices; use OpenVMS device names when you specify an InfoServer device name. Note that the device name must either match exactly the name that the SHOW SERVICES command displays or must contain wildcards. (Wildcards are permitted in this command.)

  A disk specification must end with a colon.

- Partition names

  Partitions are container files that are served to the network. As such, they have OpenVMS file names with a default file type of .ESS$PARTITION. Partition names, including the device, directory, and file name, can be no more than 242 characters in length.

  The partition name can be used to further identify the specific service selected. Support for partitions is limited in this version, however. VSI strongly recommends that you use LD devices to support partitioned hard drives. See the DCL command LD HELP for more information.

Qualifiers

/CLASS=className

Specifies a subset of the complete LASTport Disk (LAD) name space.

The purpose of class names is to subdivide name spaces so that clients see only those names that are meaningful to them. The use of class names also allows two services to have the same name and not conflict with one another.

For example, you can use different class names for different on-disk structures that several client systems use. You might use SERVICEA/CLASS=ODS-2 for some client systems and SERVICEA/CLASS=ISO_9660 for other client systems. The service has the same name, SERVICEA, but the class names are different.

The class name you use depends on the client systems that connect to the service being created. The default class name is ODS_2. For example, OpenVMS systems use the ODS_2 name space when attempting to mount an InfoServer device. Note that OpenVMS clients can solicit only those services that are in the ODS_2 service class.

Valid class names are the following:

```
V2.0            Names understood by PCSA MS-DOS Clients
Unformatted     Virtual disk has no format
MSDOS           MSDOS virtual disks
ODS_2           VMS virtual disks
UNIX            UNIX virtual disks
ISO_9660        ISO 9660 CD format
HIGH_SIERRA     MS-DOS CD format
APPLE           Macintosh HFS format
SUN             Sun format
```

/CONFIRM (default)
/NOCONFIRM

Confirms the deletion of a service. If there are any connections, even though /NOCONFIRM has been entered, the system forces a confirmation.

Controls whether a request is issued before each delete operation to confirm that the operation should be performed on that service. The following responses are valid:

```
YES       NO        QUIT
TRUE      FALSE     Ctrl/Z
1         0         ALL
    Return (key)
```

Usage notes:

- You can use any combination of uppercase and lowercase letters for word responses. Word responses can be abbreviated to one or more letters (for example, T, TR, or TRU for TRUE); however, these abbreviations must be unique.
- Affirmative answers are YES, TRUE, and 1. Negative answers include NO, FALSE, 0, and pressing **Return**.
- To stop processing the command at a particular point, enter QUIT or press **Ctrl/Z**.
- When you respond by entering ALL, the command continues to process, but no further prompts are displayed.

/DISCONNECT
/NODISCONNECT (default)

Overrides the default prompting for confirmation if you attempt to delete a service that has sessions connected to it. If a service has connected sessions and the /DISCONNECT qualifier is not supplied, you are prompted to confirm service deletion.

To delete services without being prompted at all, specify both the /NOCONFIRM and /DISCONNECT qualifiers.

Example

```
$ InfoServer SHOW SERVICES

Service Name          [Service Class] Device or File
--------------------- --------------- --------------
CURRENT_KIT           [ODS-2]         _MOVMAN$LDA2:
CURRENT_KIT           [ODS-2]         _MOVMAN$LDA42:
%INFOSRVR-I-FOUND, 2 services found.

$ DELETE SERVICE CURRENT_KIT _MOVMAN$LDA42:
Delete service CURRENT_KIT [ODS-2] for _MOVMAN$LDA42:? [N]:Y
%INFOSRVR-I-DELSERV, service CURRENT_KIT [ODS-2] deleted for
_MOVMAN$LDA42:.
```

The first command displays two services.

The second command deletes the CURRENT_KIT service on the _MOVMAN$LDA42:device. The system prompts you to confirm your deletion command. After you do so, the system displays a messages indicating that the service has been deleted.

EXIT

EXIT — Terminates the program. Alternatively, you can press **Ctrl/Z** to exit the program.

Syntax

EXIT

HELP

HELP — Starts InfoServer online help. ESS$INFOSERVER is the user interface for the LASTport/Disk server implemented as an application on OpenVMS. Its behavior is similar to that of the hardware InfoServer product.

Syntax

HELP topic

Parameters

topic

The topic for which help is requested.

Example

```
$ INFOSERVER HELP SHOW SESSIONS

SHOW
  SESSIONS
      Displays information about client nodes that are connected
      to services.
      Format:
        SHOW SESSIONS [serviceName] [dev-or-partitionName]
      Additional information available:
      Parameters Qualifiers
      /ALL
```

This command displays help about the InfoServer command SHOW SESSIONS.

SAVE

SAVE — Saves the current set of active services as a set of commands in a command procedure. You can then run the command procedure to reproduce the current services when you reboot the system.

Syntax

SAVE procedureName

Parameters

procedureName

Creates a command procedure that restores the current server state. The procedure name is the OpenVMS file name of the command procedure to be created. If you do not specify a file type, the type defaults to .COM.

The default procedure name is ESS$LAD_SERVICES.COM.

Examples

```
$ SHOW SERVICES


Service Name          [Service Class] Device or File
--------------------- --------------- --------------
BASELEVEL_A           [ODS-2]         _INFOS$LDA1:
BASELEVEL_B           [ODS-2]         _INFOS$LDA2:
BASELEVEL_C           [ODS-2]         _INFOS$LDA3:
BASELEVEL_D           [ODS-2]         _INFOS$LDA4:
FIELD_TEST_BASELEVEL [ODS-2]          _INFOS$LDA2:
CURRENT_BASELEVEL     [ODS-2]         _INFOS$LDA3:
EXPERIMENTAL_BASELEVEL
                      [ODS-2]         _INFOS$LDA4:
%INFOSRVR-I-FOUND, 7 services found.

$ SAVE BASELEVELS



$! Created by the OpenVMS InfoServer SAVE command on 22-APR-2005
14:34:02.48
$ Set NoOn$ Infoserver := $ESS$INFOSERVER
$!
$! The comment for each service includes the current device name.
$!
$!************************************************************
$!  BASELEVEL_A [ODS_2] - _BILBO$LDA1: ❶
$!************************************************************
$ LD Connect/Symbol _BILBO$DKB0:[DISKS]BASELEVEL_A.DSK;1 ❷
$ LD_UNIT_1 := LDA'LD_UNIT': ❸
$ If $STATUS Then Mount/System/NoWrite 'LD_UNIT_1' BASELEVELA ❹
 $ INFOSERVER Create Service BASELEVEL_A 'LD_UNIT_1' - ❺
        /Class=ODS_2/Readers=1000/NoWriters -
        /Readahead/NoReadbehind -
        /Rating=Dynamic
$!************************************************************
$!  BASELEVEL_B [ODS_2] - _BILBO$LDA2:
$!************************************************************
$ LD Connect/Symbol _BILBO$DKB0:[DISKS]BASELEVEL_B.DSK;1
$ LD_UNIT_2 := LDA'LD_UNIT':
$ If $STATUS Then Mount/System/NoWrite 'LD_UNIT_2' BASELEVELB
$ INFOSERVER Create Service BASELEVEL_B 'LD_UNIT_2' -
        /Class=ODS_2/Readers=1000/NoWriters -
        /Readahead/NoReadbehind -
        /Rating=Dynamic
$!************************************************************
$!  BASELEVEL_C [ODS_2] - _BILBO$LDA3:
$!************************************************************
$ LD Connect/Symbol _BILBO$DKB0:[DISKS]BASELEVEL_C.DSK;1
$ LD_UNIT_3 := LDA'LD_UNIT':
$ If $STATUS Then Mount/System/NoWrite 'LD_UNIT_3' BASELEVELC
$ INFOSERVER Create Service BASELEVEL_C 'LD_UNIT_3' -
        /Class=ODS_2/Readers=1000/NoWriters -
        /Readahead/NoReadbehind -
```

```
        /Rating=Dynamic
$!*************************************************************
$!  BASELEVEL_D [ODS_2] - _BILBO$LDA4:
$!*************************************************************
$ LD Connect/Symbol _BILBO$DKB0:[DISKS]BASELEVEL_D.DSK;1
$ LD_UNIT_4 := LDA'LD_UNIT':
$ If $STATUS Then Mount/System/NoWrite 'LD_UNIT_4' BASELEVELD
$ INFOSERVER Create Service BASELEVEL_D 'LD_UNIT_4' -
        /Class=ODS_2/Readers=1000/NoWriters -
        /Readahead/NoReadbehind -
        /Rating=Dynamic -
        /Encoded_Password=481C6B9081E742C2
           ! Invalid if service name changes ❻
$!*************************************************************
$!  FIELD_TEST_BASELEVEL [ODS_2] - _BILBO$LDA2:
$!*************************************************************
$ INFOSERVER Create Service FIELD_TEST_BASELEVEL 'LD_UNIT_2' - ❼
        /Class=ODS_2/Readers=1000/NoWriters -
        /Readahead/NoReadbehind -
        /Rating=Dynamic
$!*************************************************************
$ INFOSERVER Create Service CURRENT_BASELEVEL 'LD_UNIT_3' -
        /Class=ODS_2/Readers=1000/NoWriters -
        /Readahead/NoReadbehind -
        /Rating=Dynamic
$!*************************************************************
$!  EXPERIMENTAL_BASELEVEL [ODS_2] - _BILBO$LDA4:
$!*************************************************************
$ INFOSERVER Create Service EXPERIMENTAL_BASELEVEL 'LD_UNIT_4' -
        /Class=ODS_2/Readers=1000/NoWriters -
        /Readahead/NoReadbehind -
        /Rating=Dynamic -
        /Encoded_Password=01F1D7374C0B81EC
          ! Invalid if service name changes ❽
$ Exit
```

The SHOW SERVICES command displays the services that are currently offered by the server. There is a set of software base levels, each on its own logical disk and served to the LAN. The base levels are labeled a through d, but, in addition, names help users so that they do not need to remember the corresponding letters.

Note that devices LDA2, LDA3, and LDA4 have two services assigned to each one.

The numbers in the third part of the example correspond to the numbers of the following explanations.

❶ The comment for each device contains the name of the device at the time the SAVE command was executed. LD devices are pseudo disk devices and might change unit numbers every time they are connected.

❷ This command connects an LD device to the container file and assigns the unit number to the DCL symbol LD_UNIT.

❸ A unique symbol is created for each device assigned to a container file.

❹ This command mounts the device specifying the label of the volume that the device had at the time of the SAVE command.

❺ The InfoServer service is re-created for the device.

❻ The experimental base level services are password protected. For security, the password is stored in the command procedure in prehashed format. Note that both services have the same password, but the hash is different.

❼ Because FIELD_TEST_BASELEVEL and BASELEVEL_B point to the same LD device, no attempt is made to create another device, and the correct unit (symbol LD_UNIT_2) is used to refer to the previously created unit.

❽ See callout 6.

SET SERVICE

SET SERVICE — Modifies the attributes of an existing service.

Syntax

```
SET SERVICE serviceName device-or-partitionName
```

Parameters

serviceName

The name by which the service is known to the local area network. The service name can consist of alphanumeric characters or dollar signs ($). It can be up to 255 characters in length.

device-or-partitionName

The name of the OpenVMS disk device or partition is the name as it is to be known to the local area network. The name of the device or partition that you enter must have been created previously.

- Device names

  Devices served to the local area network are OpenVMS disk devices; use OpenVMS device names when you specify an InfoServer device name. Note that the device name must either match exactly the name that the SHOW SERVICES command displays or must contain wildcards.

  In the InfoServer utility, wildcards, where supported, are those used in OpenVMS. The percent (%) character matches exactly one character. The asterisk (*) character matches zero or more characters.

  A disk specification must end with a colon.

- Partition names

  Partitions are container files that are served to the network. As such, they have OpenVMS file names with a default file type of ".ESS$PARTITION". Partition names, including the device, directory, and file name, can be no more than 242 characters in length.

  The partition name can be used to further identify the specific service selected. Support for partitions is limited in this version, however. VSI strongly recommends that you use LD devices to support partitioned hard drives. See the DCL command LD HELP for more information.

Qualifiers

/CLASS=className

Specifies a subset of the complete LASTport Disk (LAD) name space.

The purpose of class names is to subdivide name spaces so that clients see only those names that are meaningful to them. The use of class names also allows two services to have the same name and not conflict with one another.

For example, you can use different class names for different on-disk structures that several client systems use. You might use SERVICEA/CLASS=ODS-2 for some client systems and SERVICEA/CLASS=ISO_9660 for other client systems. The service has the same name (SERVICEA), but the class names are different.

The class name you use depends on the client systems that connect to the service being created. The default class name is ODS_2. For example, OpenVMS systems use the ODS_2 name space when attempting to mount an InfoServer device. Note that OpenVMS clients can solicit only those services that are in the ODS_2 service class.

Valid class names are the following:

```
V2.0            Names understood by PCSA MS-DOS Clients
Unformatted     Virtual disk has no format
MSDOS           MSDOS virtual disks
ODS_2           VMS virtual disks
UNIX            UNIX virtual disks
ISO_9660        ISO 9660 CD format
HIGH_SIERRA     MS-DOS CD format
APPLE           Macintosh HFS format
SUN             Sun format
```

/PASSWORD=passwordString
/NOPASSWORD

Specifies an optional service access control password. The client system must specify the password to access the service.

The password-string can be up to 39 alphanumeric ASCII characters in length. If no password is specified, the client is not required to provide a password to access the service.

The text password is hashed and stored in encrypted form in memory with the other service information.

/RATING=DYNAMIC
/RATING=STATIC=value

Clients use service rating to select a service in the case of multiple matching services. The service with the higher service rating is selected.

The system adjusts the dynamic service rating based on load.

A static rating between 0 and 65535 can also be set. Static ratings are not adjusted by the system.

/READAHEAD
/NOREADAHEAD

When a disk read is required to fill a cache lock, specifies that the read should be from the first block requested to the end of the bucket boundary. Read-ahead can speed up sequential operations by preloading disk blocks that are needed into the cache.

If both the /READAHEAD and /READBEHIND qualifiers are specified, any block requested within a cache bucket causes the entire bucket range of blocks to be read into the cache.

/READBEHIND
/NOREADBEHIND

When a disk read is required to fill a cache block, specifies that the read should include all blocks from the beginning of the cache bucket boundary up to and including the requested block.

If both the /READAHEAD and /READBEHIND qualifiers are specified, any block requested within a cache bucket causes the entire bucket range of blocks to be read into the cache.

/READERS=number

Specifies the maximum number of client connections allowed for read access.

Example

```
$ INFOSERVER SET SERVICE FUNDY/NOPASSWORD

Service FUNDY [ODS-2] modified.

$ INFOSERVER SHOW SERVICES FUNDY/FULL

FUNDY [ODS-2]                          Access: Read-only
  File or device: _MOVERS$LDA1: [750000 blocks]
  Flags: 00000000D2 {No Writers,Static Rating,Readbehind,
                                      Readahead}
  Rating:      Static, 42      Password:          Disabled
  Max Readers:         1000      Max Writers:            0
  Curr Readers:           0      Curr Writers:           0
  Reads:                  0      Writes:                 0
  Blocks Read:            0      Blocks Written:         0
```

The first command in this example modifies the FUNDY service so that the client does not need to enter a password to access the service. The second command displays the FUNDY service, which indicates that the use of a password has been disabled.

SHOW SERVER

SHOW SERVER — Displays information about the server (the system that provides services).

Syntax

SHOW SERVER

Example

```
$ INFOSERVER SHOW SERVER

Node MOVERS [COMPAQ Professional Workstation XP1000] running
OpenVMS XALD-BL2

LASTport/Disk Server Version 1.2

Max Services:       64       Write Quota:          0
Cache Buckets:    4096       Cache Bucket Size:   32 blocks
Cache Size:   67108864 bytes
Hits:              478       Hit Percentage:      59%
```

```
Misses:             328
Current Sessions:    0          Peak Sessions:           1
                   Read              Write
Requests:            40                  0
Blocks:             319                  0
Errors:               0                  0
Aborted:              0                  0
Conflicts:            0                  0
```

This command displays information about the server that provides services to the client. The information displayed includes the following:

- The maximum number of services this server can offer simultaneously
- The current size of the cache
- Cache effectiveness statistics
- Current and maximum historical number of clients connected simultaneously
- I/O statistics

SHOW SERVICES

SHOW SERVICES — The SHOW SERVICES command displays service-specific information for one or all services offered by the server. This information includes the device associated with the service, the service class, and the number of connected sessions. The SHOW SERVICES command supports wildcard expressions. In the InfoServer utility, wildcards, where supported, are those used in OpenVMS. The percent (%) character matches exactly one character. The asterisk (*) character matches zero or more characters.

Syntax

SHOW SERVICES serviceName options...

Parameters

serviceName

The name by which the service is known to the local area network. The service name consists of alphanumeric characters or dollar signs ($). It can be up to 255 characters in length. If omitted, the service name defaults to all services.

In the InfoServer utility, wildcards, where supported, are those used in OpenVMS. The percent (%) character matches exactly one character. The asterisk (*) character matches zero or more characters.

Qualifiers

/BRIEF (default)

The BRIEF option provides an abbreviated, one-line summary of information for each service selected. BRIEF is the default.

/FULL

The FULL option provides all the service-specific information for the services selected.

Example

```
INFOSERVER> SHOW SERVICES

Service Name          [Service Class] Device or File
-------------------- --------------- --------------
HUDSON                 [ODS-2]         _MOVERS$LDA1:
BAFFIN                 [ODS-2]         _MOVERS$LDA1:
FUNDY                  [ODS-2]         _MOVERS$LDA1:

3 services found.

INFOSERVER> SHOW SERVICES/FULL

HUDSON [ODS-2]                           Access: Read-only
  File or device: _MOVERS$LDA1: [750000 blocks]
  Flags: 0000000082 {No Writers,Readahead}
  Rating:       Dynamic, 65535    Password:         Disabled
  Max Readers:            1000    Max Writers:             0
  Curr Readers:              0    Curr Writers:            0
  Reads:                     0    Writes:                  0
  Blocks Read:               0    Blocks Written:          0

BAFFIN [ODS-2]                           Access: Read-only
  File or device: _MOVERS$LDA1: [750000 blocks]
  Flags: 0000000082 {No Writers,Readahead}
  Rating:       Dynamic, 65535    Password:         Disabled
  Max Readers:            1000    Max Writers:             0
  Curr Readers:              0    Curr Writers:            0
  Reads:                     0    Writes:                  0
  Blocks Read:               0    Blocks Written:          0

FUNDY [ODS-2]                            Access: Read-only
  File or device: _MOVERS$LDA1: [750000 blocks]
  Flags: 00000000D2 {No Writers,Static Rating,Readbehind,
                     Readahead}
  Rating:        Static,    42    Password:         Enabled
  Max Readers:            1000    Max Writers:             0
  Curr Readers:              0    Curr Writers:            0
  Reads:                     0    Writes:                  0
  Blocks Read:               0    Blocks Written:          0

3 services found.
```

The first command displays the one-line default BRIEF summary of all the services that are connected.

The second command displays all of the service-specific information for all the services that are connected. Note that passwords are disabled on the HUDSON and BAFFIN services and enabled on the FUNDY service.

SHOW SESSIONS

SHOW SESSIONS — Displays information about client nodes that are connected to services.

Syntax

SHOW SESSIONS serviceName device-or-partitionName

Parameters

serviceName

The name by which the service is known to the local area network. The service name can consist of alphanumeric characters, dollar signs ($), and wildcards. It can be up to 255 characters in length. If omitted, the service name defaults to all services.

In the InfoServer utility, wildcards, where supported, are those used in OpenVMS. The percent (%) character matches exactly one character. The asterisk (*) character matches zero or more characters.

device-or-partitionName

The name of the OpenVMS disk device or partition name is the name as it is to be known to the local area network. The name of the device or partition that you enter must have been created previously.

- Device names

  Devices served to the local area network are OpenVMS disk devices; use OpenVMS device names when you specify an InfoServer device name. Note that the device name must either match exactly the name that the SHOW SERVICES command displays or must contain wildcards.

  In the InfoServer utility, wildcards, where supported, are the same as those used in OpenVMS. The percent (%) character matches exactly one character. The asterisk (*) character matches zero or more characters.

  A disk specification must end with a colon.

- Partition names

  Partitions are container files that are served to the network. As such, they have OpenVMS file names with a default file type of .ESS$PARTITION. Partition names, including the device, directory, and file name, can be no more than 242 characters in length.

  Support for partitions is limited in this version. VSI strongly suggests that you use LD devices to support partitioned hard drives. See the DCL command LD HELP for more information.

Qualifiers

/ALL

Display all services that match the selection criteria even if no clients have connections. If this qualifier is omitted, only those services with clients connected are displayed.

Example

```
$ INFOSERVER SHOW SESSIONS

HUDSON        [ODS-2]     _MOVERS$LDA1: [ 1 Connection]
1 service found.

$ INFOSERVER SHOW SESSIONS/ALL

HUDSON        [ODS-2]     _MOVERS$LDA1: [ 1 Connection]
BAFFIN        [ODS-2]     _MOVERS$LDA1:
FUNDY         [ODS-2]     _MOVERS$LDA1:
```

```
3 services found.
```

The first command displays only the session that has a client connection, HUDSON. The second command displays all sessions, even those with no client connections.

SPAWN

SPAWN — Spawns a process to execute a DCL command. If you do not enter a command, the command terminal is attached to the spawned process. If you do enter a command, that command is executed and, upon completion of the command, control returns to the parent process.

Syntax

SPAWN DCL Command

Example

```
InfoServer> SPAWN DIRECTORY

.
.
.
(output)
.
.
.

InfoServer>
```

This command spawns a process to execute a DCL command DIRECTORY. Following execution of the command, control returns to the InfoServer process.

START SERVER

START SERVER — This command starts the LASTport/Disk server and sets various server and cache characteristics. Usually, this command is executed by SYS$STARTUP:ESS$LAD_STARTUP.COM using data from SYS$STARTUP:ESS$LAD_STARTUP.DAT. VSI strongly recommends that you make all modifications in the SYS$STARTUP:ESS$LAD_STARTUP.DAT file. You can use the START SERVER command interactively to use its qualifiers to change server settings as long as no services are currently defined. If you enter the START SERVER command directly, you must specify all four qualifiers. Otherwise, the system reverts to using default values.

Syntax

START SERVER

Qualifiers

/BUFFER_SIZE=n

The InfoServer block cache is structured as an array of fixed-size buffers (also called **buckets**.) The /BUFFER_SIZE qualifier determines the size of each bucket. (The /CACHE qualifier determines the number of buckets.)

The numeric value of this parameter is an integer between 3 and 8, inclusive, representing the bucket size in 512-byte blocks. For example:

```
3 -   8 blocks (default)
4 -  16 blocks
5 -  32 blocks
6 -  64 blocks
7 - 128 blocks
8 - 256 blocks
```

The default value is 3.

Bucket sizes that are larger than 32 blocks are not appropriate for most users. The OpenVMS client segments I/O requests that are larger than 31 blocks into 31-block chunks, and the default bucket read ahead behavior might result in unnecessary I/O activity to the disk.

/CACHE = number-of-buckets (default = 512)

The InfoServer block cache is structured as an array of fixed-size buffers (also called **buckets**. The /CACHE qualifier determines the number of buckets in the cache. The /BUFFER_SIZE qualifier determines the size of each bucket.

Numbers larger than 16384 can adversely affect performance. Consider increasing the /BUFFER_SIZE qualifier to reach the desired cache size.

/MAXIMUM_SERVICES = maxservice (default = 256)

Sets the maximum service count for the server. This is the maximum number of services that can be defined at one time. Each service descriptor consumes nonpaged pool; however, unused service slots consume only 4 bytes each.

The maximum value is 1024.

/WRITE_QUOTA = n (default = 0)

Number of simultaneous synchronous writes permitted within the server. The default of zero means that all write operations are performed synchronously.

Example

```
$ InfoServer SHOW SERVER

Node BILBO [HP rx2600  (900MHz/1.5MB)] running OpenVMS XAR8-D2Y
LASTport/Disk Server Version 1.2
Max Services:        64        Write Quota:          0
Cache Buckets:     2048        Cache Bucket Size:   32 blocks
Cache Size:    33554432 bytes
Hits:                0         Hit Percentage:       0%
Misses:              0
Current Sessions:    0         Peak Sessions:        0
                   Read              Write
Requests:             0                 0
Blocks:               0                 0
Errors:               0                 0
Aborted:              0                 0
Conflicts:            0                 0
```

```
$ InfoServer START SERVER/MAXIMUM_SERVICES=128/CACHE=2048/BUFF=5/WRITE=0

%INFOSRVR-I-STARTED, LASTport/Disk server started.

$ InfoServer SHOW SERVER

Node BILBO [HP rx2600  (900MHz/1.5MB)] running OpenVMS XAR8-D2Y
LASTport/Disk Server Version 1.2
Max Services:    128        Write Quota:          0
Cache Buckets:  2048        Cache Bucket Size:   32 blocks
Cache Size: 33554432 bytes
Hits:               0       Hit Percentage:       0%
Misses:             0
Current Sessions:   0       Peak Sessions:        0
                 Read            Write
Requests:           0               0
Blocks:             0               0
Errors:             0               0
Aborted:            0               0
Conflicts:          0               0
```

The first command in this example displays the current information about the server. The second command starts the server and increases the maximum number of services for the server. The third command displays the new information about the server, showing the increased maximum number of services.

Chapter 12. Install Utility

12.1. 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.

12.2. 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
```

12.3. INSTALL Qualifiers

This section describes the /TRANSLATE qualifier and provides examples of its use.

/TRANSLATE

/TRANSLATE — The /TRANSLATE qualifier forces INSTALL to attempt a logical name translation of a file that you specify. Before you use the /TRANSLATE qualifier, you must invoke INSTALL as a foreign command: `$ INSTALL = "$INSTALL"`

Syntax

/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.

12.4. INSTALL Commands

This section describes and provides examples of the INSTALL commands. The following table summarizes the INSTALL command functions:

Command	Function
ADD	Synonym for the CREATE command
CREATE	Installs the specified image as a known image
DELETE	Synonym for the REMOVE command
EXIT	Exits from INSTALL

Command	Function
HELP	Describes how to use INSTALL
LIST	Displays a description of each specified known image, global sections, and the addresses of known image data structures
PURGE	Deletes all known images installed without the /NOPURGE qualifier
REMOVE	Deletes a known image
REPLACE	Associates a known image with the latest version of the image file or modifies the attributes of an installed image

ADD

ADD — Installs the specified image file as a known image. Requires the CMKRNL privilege. Also requires the SYSGBL privilege to create system global sections and the PRMGBL privilege to create permanent global sections.

Syntax

```
ADD
```

CREATE

CREATE — Installs the specified image file as a known image. Requires the CMKRNL privilege. Also requires the SYSGBL privilege to create system global sections and the PRMGBL privilege to create permanent global sections.

Syntax

```
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 Integrity server 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:

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, see 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 *VSI 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 *VSI 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.

/RESIDENT[=([NO]CODE,[NO]DATA)]

On Alpha and Integrity server 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.

You must have PFNMAP privilege to install the image with /RESIDENT qualifier.

/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.

/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.

DELETE

DELETE — Deletes a known image. The DELETE command is a synonym for the REMOVE command.

Syntax

DELETE

EXIT

EXIT — Terminates INSTALL and returns control to the DCL command level. You can also exit from INSTALL by pressing **Ctrl/Z**.

Syntax

EXIT

HELP

HELP — Displays information about how to use INSTALL.

Syntax

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

LIST — Displays a description of each specified known image or, if no file is specified, all known images.

Syntax

```
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, see the *VSI 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.

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 ❶                Prv ❷
   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
.
.
.
```

   ❶ File specification of the known image

   ❷ Attribute of known image, as follows:

Attribute	Meaning
ACNT	Image accounting is enabled for the image (/ACCOUNTING).
HDR	Image header is permanently resident (/HEADER_RESIDENT).
LNKBL	Image is not executable; it is a shareable (linkable) image set by the OpenVMS Linker.
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).

Attribute	Meaning
PROT	Image contains protected code (/PROTECTED).
PRV	Image has privileges (/PRIVILEGED).
[ddag]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, see the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials*.
SHAR	Image is shared (/SHARED).
WRT	Image is writable (/WRITABLE).
XONLY	Only execute access to image is allowed (/EXECUTE_ONLY).

[ddag]Alpha and Integrity servers 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

 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.

```
SYS$DISK:<SYS0.SYSCOMMON.SYSEXE>.EXE

    LOGINOUT;3       Open Hdr    Shar Priv
        Entry access count         = 44                    ❶
        Current / Maximum shared   = 3 / 5                  ❷
        Global section count       = 2                     ❸
        Privileges = CMKRNL SYSNAM TMPMBX EXQUOTA SYSPRV   ❹
```

   ❶ Number of times known file entry has been accessed by this node since it was installed.

   ❷ 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.

   ❸ Number of global sections created for the known file; appears only if the image is installed with the /SHARED qualifier.

   ❹ 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>SLS$USSSHR.EXE ❷
 INS$86D9BFB0_003 ❸ (14000001) ❹       PRM SYS ❺   Pagcnt/Refcnt=1/1 ❻
 INS$86D9BFB0_002    (14000001)          PRM SYS      Pagcnt/Refcnt=3/3
 INS$86D9BFB0_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 Integrity servers) in the section and number of page table entries currently mapped to this global section. For a more detailed discussion of mapping global sections, see the *VSI 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.

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

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.

Syntax

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.

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.

Syntax

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

REPLACE — Replaces a known image entry with another version of the image, or with modified attributes of the image. Requires the CMKRNL privilege. Also requires the SYSGBL privilege to create system global sections and the PRMGBL privilege to create permanent global sections.

Syntax

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 Integrity server 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:

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, see 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 *VSI 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 *VSI 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 Integrity server 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.

You must have PFNMAP privilege to install the image with /RESIDENT qualifier.

/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.

/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.

Chapter 13. LAN Control Program (LANCP) Utility

Important

OpenVMS x86-64 is currently the primary platform for Local Area Network Control Program (LANCP) and LAN Auxiliary Control Program (LANACP) feature enhancements and new functionality. As a result, some LANCP and LANACP features described in this manual may be available on OpenVMS x86-64 systems before they are available on OpenVMS Alpha or OpenVMS IA-64 systems.

Refer to the LANCP online help for the most current platform-specific feature information.

13.1. 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 OpenVMS Alpha, IA-64, and x86-64 systems, LANCP and LANACP functionality is broadly equivalent.

Refer to the *VSI LAN Bandwidth Monitoring Guide for OpenVMS x86-64* [https://docs.vmssoftware.com/vsi-lan-bandwidth-monitoring-guide-for-openvms-x86-64/] and *VSI OpenVMS LAN Driver Tracing Guide* [https://docs.vmssoftware.com/vsi-openvms-lan-driver-tracing-guide/] for more information on the use of LANCP in the context of these topics.

13.2. 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.
- Provide MOP downline load support for booting satellites in an OpenVMS cluster environment (Alpha and IA-64 only).

To invoke LANCP, enter the following command at the DCL command prompt:

```
$ RUN SYS$SYSTEM:LANCP.EXE
```

The LANCP utility displays the LANCP> prompt, at which you can enter any LANCP command described in this chapter.

To define LANCP as a foreign command, enter the following command (either at the DCL prompt or in a startup or login command file):

```
$ LANCP :== $SYS$SYSTEM:LANCP.EXE [command]
```

Then you can enter the **LANCP** command at the DCL prompt to invoke the utility and enter LANCP commands.

Alternatively, invoke LANCP using the **MCR** command. Enter the following command at the DCL prompt:

```
$ MCR LANCP [command]
```

When you enter the **LANCP** or **MCR LANCP** command:

- If you do not specify a command, the LANCP utility displays the LANCP> prompt, at which you can enter commands.
- If you do specify a command, the LANCP utility executes the command and returns to the DCL command prompt.

Note

Some LANCP commands require SYSPRV or PHY_IO privilege.

To exit LANCP and return to the DCL command level, 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.

13.3. LANCP Commands

This section describes and provides examples of the LANCP commands. The following table summarizes the LANCP commands:

Table 13.1. LANCP Commands

Command	Function
@ (Execute Procedure)	Reads and executes commands from the specified file.
BANNER	Displays the current date and time, followed by any text that was specified on the command line.
CLEAR DEVICE	Deletes a device from the LAN volatile device database.
CLEAR DLL **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 remote node, such as a terminal server, that implements a console interface using the MOP console carrier protocol.

Command	Function
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 LANCP, returning control to DCL.
HELP	Provides help for the LANCP utility.
LIST BANDWIDTH	Displays bandwidth monitoring settings in the LAN permanent device database.
LIST DEVICE	Displays information in the LAN permanent device database.
LIST DLL **LIST MOPDLL**	Displays MOP downline load settings in the LAN permanent device database.
LIST FAILOVER **LIST LLAN**	Displays LAN Failover settings in the LAN permanent device database.
LIST NODE	Displays information from the LAN permanent node database.
LIST TEST	Displays LAN testing settings in the LAN permanent device database.
LIST TRACE	Displays LAN tracing settings in the LAN permanent device database.
LIST VLAN	Displays VLAN pseudo devices in the LAN permanent device database.
PURGE DEVICE	Deletes a device from the LAN permanent device database.
PURGE NODE	Deletes node data from the LAN permanent node database.
SET ACP	Modifies the operation of the LANACP LAN Server process.
SET DEVICE	Sets LAN device parameters, creating or updating the volatile device database entry as needed.
SET NODE	Enters a node into the LAN volatile node database or modifies an existing entry.
SET VERIFY	Controls whether command lines and data lines in command procedures are displayed in the terminal or are printed in a batch job log.
SHOW ACP	Displays the settings of various options for LANCP and LANACP.
SHOW BANDWIDTH	Displays bandwidth monitoring settings for all devices.

Command	Function
SHOW CONFIGURATION	Displays a list of LAN devices on the system.
SHOW DEVICE	Displays LAN device information from the LAN volatile device database and from the requested LAN devices.
SHOW DLL **SHOW MOPDLL**	Displays the status of MOP downline load services, states, and counters.
SHOW FAILOVER **SHOW LLAN**	Displays LAN Failover settings in the LAN volatile device database.
SHOW LOG	Displays the SYS$MANAGER:LAN$ACP.LOG file.
SHOW NODE	Displays information from the LAN volatile node database.
SHOW TEST	Displays LAN device information from the LAN volatile device database for the current test status.
SHOW TRACE	Shows trace settings for devices.
SHOW VLAN	Shows VLAN settings for devices.
SPAWN	Creates a subprocess of the current process.
START DEVICE	Starts a test on a LAN device.
TRIGGER NODE	Issues a reboot request to a remote node.
UPDATE DEVICE	Updates a device, specifically to issue a reset request to the device driver.
WAIT	Waits the specified number of seconds.

@ (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 60
@COUNT
[Ctrl/Z]
$ RUN SYS$SYSTEM:LANCP.EXE
LANCP> @COUNT.COM

Device Counters EIA0 (1-MAR-2026 08:57:32.06):
                  Value  Counter
                 129503 Seconds since last zeroed
           166154128848 Bytes received
            67782124143 Bytes sent
              123088201 Packets received
               55670215 Packets sent
              496902925 Multicast bytes received
               17028824 Multicast bytes sent
                5238209 Multicast packets received
                 123592 Multicast packets sent
                3774643 Unrecognized multicast destination packets
                      1 Link up transitions (27-FEB-2026 20:51:17.50)
27-FEB-2026 20:51:17.50 Time of last link up
```

This example creates and runs a command procedure, `COUNT.COM`, that displays EIA device counters once per minute.

BANNER

BANNER — Displays the current date and time, followed by any text that was specified on the command line. This is useful in command files to record the time and additional text.

Format

BANNER [*text*]

Parameter

text

The rest of the input line, enclosed in quotes if needed.

Qualifiers

None.

Example

```
LANCP> BANNER "Test"
26-MAR-2026 09:22:23.60 "Test"
```

This command displays the current date and time, followed by the specified message `"Test"`.

CLEAR DEVICE

CLEAR DEVICE — Deletes device data from the LAN volatile device database. If no qualifier is present to select the type of data to delete, all settings in the device entry are deleted. Requires SYSPRV privilege.

Format

CLEAR DEVICE *device-name*

Parameter

device-name

Specifies the LAN device name. The device name has the form *ddcu*:, where *dd* is the device identifier, *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 EI Ethernet device is specified as EIA0, the second as EIB0, and so on. It is possible to specify the first EI device, for example, as EIA, EIA0, or EIA0:.

A partial device name, accompanied by the **/ALL** qualifier, selects all matching devices. For example, specify E to select all Ethernet devices, V for VLAN devices, EI for Ethernet EI devices, or no *device-name* to select all LAN devices.

Qualifiers

/ALL

Deletes requested data for all LAN devices in the LAN volatile device database. If a device name is specified, data is deleted for all matching devices. For example, specifying **EI/ALL** deletes data for all EI LAN devices.

/BANDWIDTH

Deletes device settings for bandwidth monitoring and disables the monitoring.

/CHARACTERISTICS

Deletes device characteristics settings for the LAN device from the LAN volatile device database. The settings include speed, duplex mode, LAN Failover device settings, and other device parameters. Clearing these characteristics does not directly affect the LAN device, but is done for future use.

/DLL

Deletes MOP downline load settings for the LAN device and disables MOP downline load (if enabled).

/MOPDLL

Equivalent to the **/DLL** qualifier.

/PARAMETERS

Equivalent to the **/CHARACTERISTICS** qualifier.

/TEST

Deletes test context for the LAN device. Clearing this context does not directly affect the functionality for the LAN device, but is done for future use.

/TRACE

Deletes driver tracing settings for the LAN device and stops tracing for the device.

Example

```
LANCP> CLEAR DEVICE EIA0/BANDWIDTH
```

This command deletes bandwidth monitoring settings for device `EIA0` from the LAN volatile device database. The **/BANDWIDTH** qualifier specifies that bandwidth settings should be deleted and turns off bandwidth monitoring for the device.

CLEAR DLL

CLEAR DLL — Clears MOP downline load counters from the LAN volatile device and node databases for all nodes and devices. Equivalent to the **CLEAR MOPDLL** 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.

CLEAR MOPDLL

CLEAR MOPDLL — Clears MOP downline load counters from the LAN volatile device and node databases for all nodes and devices. Equivalent to the **CLEAR DLL** command. Requires SYSPRV privilege.

Format

CLEAR MOPDLL

Parameters

None.

Qualifiers

None.

CLEAR NODE

CLEAR NODE — Deletes a node from the LAN volatile node database. Requires SYSPRV privilege.

Format

CLEAR NODE *node-name*

Parameter

node-name

Specifies the name of a node in the LAN volatile node database.

Qualifier

/ALL

Deletes all LAN nodes in the LAN volatile node database. If a node name is specified, all matching nodes are deleted. For example, **CLEAR NODE A/ALL** deletes all nodes whose name begins with the letter A.

Example

```
LANCP> CLEAR NODE VMSSYS
```

This command deletes the node `VMSSYS` from the LAN volatile node database.

CONNECT NODE

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

Specifies 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 does not have to be defined in the LAN volatile node database. The address consists of six hexadecimal byte characters separated by hyphens or colons.

Qualifiers

/DEVICE=*device-name*

Specifies the LAN device name to be used for the connection. The device name has the form *ddcu*:, where *dd* is the device identifier, *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 EI Ethernet device is specified as EIA0, the second as EIB0, and so on. It is possible to specify the first EI device, for example, as EIA, EIA0, or EIA0:.

/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.

/INPUTFILE=*file-spec*

Supplies command input from the specified input file. Input is taken up to end-of-file or a disconnect character. If no disconnect character is encountered, the command input continues from the local terminal. If a local terminal does not exist (if the command is executing in batch mode), end-of-file disconnects the console carrier connection.

/PASSWORD=*16hexdigits*

Specifies the password to be used when the connection is initiated, in hexadecimal (for example, **/PASSWORD=0123456789ABCDEF**). The default password is 0000000000000000 or (with leading zeros suppressed) simply 0.

/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.

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.

3. `LANCP> CONNECT NODE TERM_SRV/DEVICE=EWA0/INPUT=LOGOUT_PORT_3.COM`

   This command attempts a console-carrier connection to node `TERM_SRV` to send the contents of the command file `LOGOUT_PORT_3.COM`.

CONVERT DEVICE_DATABASE

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 does not convert an entry in it unless

the entry is changed. The conversion is necessary when the contents of the device entry change between OpenVMS releases. Usually, LANCP and LANACP automatically update entries as required. Requires SYSPRV privilege.

Format

CONVERT DEVICE_DATABASE

Parameters

None.

Qualifiers

None.

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 does not convert an entry in it unless the entry is changed. The conversion is necessary when the contents of the node entry change between OpenVMS releases. Usually, LANCP and LANACP automatically update entries as required. Requires SYSPRV privilege.

Format

CONVERT NODE_DATABASE

Parameters

None.

Qualifiers

None.

DEFINE DEVICE

DEFINE DEVICE — Enters a device into the LAN permanent device database or modifies an existing device entry. Requires SYSPRV privilege.

Format

DEFINE DEVICE *device-name*

Parameter

device-name

Specifies the LAN device name. The device name has the form *ddcu*:, where *dd* is the device identifier, *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 EI Ethernet device is specified as EIA0, the second as EIB0, and so on. It is possible to specify the first EI device, for example, as EIA, EIA0, or EIA0:.

A partial device name, accompanied by the **/ALL** qualifier, selects all matching devices. For example, specify E to select all Ethernet devices, V for VLAN devices, EI for Ethernet EI devices, or no *device-name* to select all LAN devices.

Qualifiers (General)

/ALL

Sets data for all LAN devices. If a device name is specified, all matching LAN devices are selected. For example, specify E to select all Ethernet devices, V for VLAN devices, or EW to select all Ethernet EW devices.

/BANDWIDTH=(*hours-option*, *threshold-option*, *default-option*, *restart-option*)
/NOBANDWIDTH

Specifies bandwidth monitoring settings for devices. Refer to the *VSI LAN Bandwidth Monitoring Guide for OpenVMS x86-64* [https://docs.vmssoftware.com/vsi-lan-bandwidth-monitoring-guide-for-openvms-x86-64/] for more information.

Bandwidth is recorded in a buffer allocated from S2 space, a list of 32-byte buckets. Each bucket contains data for one second. A bucket may be extended across multiple seconds according to the *threshold-option*, to extend coverage of the bandwidth buffer during idle periods (see the description of *threshold-option*).

Note that one bucket is reserved for bandwidth context.

If the recording buffer cannot be allocated, bandwidth recording is not done.

The default bandwidth buffer size is determined by the LAN_FLAGS system parameter value extracted from bits <11:9>. This occurs either when the LAN device is configured (usually at boot time), or when a LANCP **SET DEVICE/BANDWIDTH** command is issued. The buffer size is set according to the following table:

LAN_FLAGS <11:9>	Buckets	Buffer Size (Bytes)	Minimum Hours[1]
0x00000000	2048	65536	0.6
0x00000200	32768	1048576	9.1
0x00000400	65536	2097152	18.2
0x00000600	98304	3145728	27.3
0x00000800	131072	4194304	36.4
0x00000A00	163840	5242880	45.5
0x00000C00	196608	6291456	54.6
0x00000E00	229376	7340032	63.7

[1]The time spanned by the bandwidth buffer when the threshold is set to 0 bytes, indicating one second per bucket.

You can specify the following keywords with this qualifier:

- *hours-option*

HOURS=*value* (Default is 1 hour)

Specifies the size of the bandwidth recording buffer as hours of data collected, 32 bytes per second, 115200 bytes per hour.

If the calculated buffer is different from the current size, then the existing buffer is deallocated, a new buffer is allocated with the requested size, and bandwidth recording is restarted.

The maximum hours that can be specified is 8760 (one year of data collection with a threshold value of zero). The buffer size is set according to the following table:

Hours Value	Buffer Size (Bytes)	Number of Buckets
1	115200	3600
2	230400	7200
24 (day)	2764800	86400
744 (month)	85708800	2678400
8760 (year)	1009152000	31536000

Specifying the *hours-option* overrides the LAN_FLAGS setting.

Specifying HOURS=0 is equivalent to **SET DEVICE/NOBANDWIDTH**.

- *threshold-option*

  THRESHOLD=*bytes* (Default is 10000 bytes)

  Specifies the threshold at which a new bandwidth entry is started. Activity below this threshold is considered noise and is not recorded in detail. When the amount of transmit and receive data over the last second is less than the threshold, the total is accumulated in the current bandwidth bucket and we do not advance to the next bucket.

  Note that 0 bytes in one second does get recorded, as that is an unusual amount of activity. For example, a 20-second period with 0 bytes of activity would be recorded as a 20-second-long bucket with no activity.

  For a nonzero threshold value, each bucket is one of the following:

  - Data for consecutive seconds where bytes were 0.
  - Data for one second where bytes were greater than or equal to the threshold value.
  - Data for consecutive seconds where bytes were nonzero and less than the threshold value.

  For a threshold value of 0, all bandwidth buckets contain one second of activity.

- *default-option*

  DEFAULT

  Restarts bandwidth monitoring with default settings, discarding any data recorded already.

- *restart-option*

RESTART

Restarts bandwidth monitoring, discarding any data recorded already.

/DESCRIPTION="*quoted-string*"

The quoted string provides additional information that is displayed by the LANCP **SHOW CONFIGURATION** command. An example of usage is to identify the switch connection for a device.

/DEVICE_SPECIFIC=(FUNCTION="*xxxx*",VALUE=*n*)
/NODEVICE_SPECIFIC

Allows device-specific parameters to be adjusted. These are used for debugging purposes or to override default device characteristics.

Specify **/NODEVICE_SPECIFIC** to clear all device-specific parameter data.

Refer to the *VSI LAN Devices, Counters, and Functions Reference Manual for OpenVMS Alpha and IA-64* [https://docs.vmssoftware.com/vsi-lan-devices-counters-and-functions-reference-openvms-alpha-ia64/] or *VSI LAN Devices, Counters, and Functions Reference Manual for OpenVMS x86-64* [https://docs.vmssoftware.com/vsi-lan-devices-counters-and-functions-reference-openvms-x86/] for a description of these functions.

/DLL=(*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)

  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 and DECnet-Plus implementations that include MOP support).

Note

When you select NOEXCLUSIVE, LANACP responds only to MOP downline load requests directed to nodes defined in the LAN node database.

- *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, minus the 802E header and CRC (26 bytes) and MOP the protocol overhead (10 bytes).

  You can override the size on a per-node basis. See the **DEFINE NODE** and **SET NODE** commands for details.

- *knownclientsonly-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.

  Note

  This option is not available when NOEXCLUSIVE mode is selected. In NOEXCLUSIVE mode, LANACP services MOP downline load requests only for clients defined in the LAN 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.

/PORT=*value*

Sets the port number to use when defining a UDP pseudo device.

/TAG=*value* (VLAN Devices Only)

Specifies the IEEE 802.1Q tag for the VLAN device.

A VLAN tag consists of a 16-bit TPID (Tag Protocol Identifier), a 3-bit priority, a 1-bit CFI (Canonical Format Indicator), and a 12-bit VLAN ID.

The VLAN TPID, priority, and CFI tag fields must be zero. As a result, the VLAN tag field is only the VLAN ID and is restricted to values 1–4095.

Once set, the tag cannot be changed using this qualifier. In order to change it, either deactivate the VLAN device to reuse the existing device name or create a new VLAN device.

/TEST
/LOOP
/RECEIVE
/TOGGLE
/TRANSMIT

These qualifiers implement test functionality built into LANCP for purposes of debugging and validation. For a complete description of the test functionality, refer to the LANCP online help.

/TRACE=(*size-option*, *mask-option*, *stop-option*, *keywords*)
/NOTRACE

Provides the LAN driver trace settings for the device. By default, the LAN drivers perform tracing of error conditions and state changes. Tracing is controlled by an event mask that selects events to trace, a stop mask that specifies when to stop tracing, and the size of the trace buffer. Keywords provide a user-friendly way of specifying the event mask that consists of two hexadecimal values.

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; it is more substantial when all events are selected, and yet more so when full packet tracing is enabled.

The command **`SHOW DEVICE/TRACE`** displays trace results. The trace settings are displayed by **`SHOW DEVICE/TRACE/HEADER`**.

Stop tracing with the command **`SET DEVICE/NOTRACE`**.

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 (32 bytes each). The default is 2048 entries. The initial and minimum size is 512 entries and the maximum is 1000000.

- *mask-option*

  MASK=(*value1*,*value2*)

  Specifies the trace mask to select which 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 this device.

  The mask can be specified by a numerical value and/or a list of keywords that identify the bits to be set.

- *stop-option*

  STOP=(*value1*,*value2*)

Specifies the trace mask to select which type of entries should stop tracing. When a trace entry that matches one of the bits in the stop mask is made, the trace mask is cleared so that you can look at the trace data accumulated so far.

The mask can be specified by a numerical value or a list of keywords that identify the bits to be set.

- *keywords*
  - DEFAULT – Requests that the trace mask be set to the LAN driver default trace mask.
  - PAUSE – Stops tracing but does not deallocate the trace buffer, allowing it to be read without being disturbed by more tracing.
  - RTINT – Selects receive done, transmit issue, interrupt events.
  - RFTINT – Selects the same events as RTINT, plus fork start and fork done events.
  - PK – Selects full size transmit and receive packets.
  - SMALLPK – Selects transmit and receive packets, saving to 32 bytes.
  - MEDIUMPK – Selects transmit and receive packets, saving up to 64 bytes.
  - LARGEPK – Selects transmit and receive packets, saving up to 96 bytes.
  - XFULL, RFULL – Selects transmit, receive packets full size.
  - XSMALL, RSMALL – Selects transmit, receive packets up to 32 bytes.
  - XMEDIUM, RMEDIUM – Selects transmit, receive packets up to 64 bytes.
  - XLARGE, RLARGE – Selects transmit, receive packets up to 96 bytes.
  - MALL – Selects all bits for the trace mask.
  - M00–M63 – Selects individual bits for the trace mask.
  - MNONE – Selects no bits for the trace mask.
  - SALL – Selects all bits for the trace stop mask.
  - S00–S63 – Selects individual bits for the trace stop mask.
  - SNONE – Selects no bits for the trace stop mask.

  The remaining events are listed in the trace header displayed by the command **`SHOW DEVICE/TRACE/HEADER`**. The keyword is listed in uppercase and precedes the description of the trace entry. All of the keywords except the ones above can be negated.

/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_DATABASE** qualifier. You can combine the **/UPDATE** and **/VOLATILE_DATABASE** qualifiers in a single **DEFINE DEVICE** command.

/VLAN_DEVICE=*device-name*
/NOVLAN_DEVICE

Specifies the physical LAN device that will host the VLAN device. The **/NOVLAN_DEVICE** qualifier requests deactivation of the VLAN device.

When deactivating the VLAN device, the *device-name* is not required because the VLAN driver knows which device is hosting the VLAN device.

When creating a new VLAN device or reusing a deactivated device, the **/TAG** qualifier must be included to supply the VLAN ID.

Notes

A VLAN device cannot be deactivated if any applications continue to use the device.

The LAN Failover device (LL*x*) is considered a physical LAN device. It acts as a single failover device and represents a set of physical LAN devices.

Be aware that an OpenVMS virtual machine considers hypervisor NICs to be physical LAN devices.

/VOLATILE_DATABASE (DEFINE DEVICE Only)

Updates the device entries in the LAN permanent device database with any data currently set in the volatile database. This allows a system manager 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.

Qualifiers (Ethernet Devices)

Note

For virtual devices, auto-negotiation, duplex mode, and flow control settings are determined by the hypervisor and associated network device driver. The OpenVMS LAN driver overrides any requested setting to set auto-negotation to enabled, full-duplex mode, and flow control to enabled.

/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.

After you issue a command using this qualifier, the Ethernet driver may redo 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. Gigabit and faster Ethernet devices support jumbo frames.

The jumbo frame size is 9018 bytes (MAC header + data + CRC, excluding VLAN tags).

/MEDIA=*value*

Selects the cable connection. Normally, the selection is made during device initialization using a limited autosensing algorithm that selects twisted pair, but it 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 qualifier allows you to change the selection without rebooting.

Acceptable values are AUI (10Base2, 10Base5), TWISTEDPAIR (10BaseT), BNC, UTP, STP, and AUTOSENSE. The default value is AUTOSENSE.

/SPEED=*value*

Sets the speed of the LAN, in Mbit/second.

Valid values are 10, 100, 1000, 2500, 5000, 10000, 25000, 40000, 100000, or AUTONEGOTIATE.

AUTONEGOTIATE indicates that the driver should determine the speed either by auto-negotiation or by device characteristics.

Emulated virtual devices are nominally 1000 Mbit/second, but a system manager can override the nominal speed.

After you issue a command using this qualifier, the Ethernet driver may redo link setup and auto-negotiation for the device.

Qualifiers (LAN Failover Devices)

/DISABLE

Disables the LAN Failover set, deactivating the active LAN Failover device and disallowing I/O to it.

/ENABLE

Enables a LAN Failover set, activating the LAN Failover 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 LAN Failover device.

/FAILOVER_SET=(*device-name*[,...])
/NOFAILOVER_SET=(*device-name*[,...])

Specifies the members of a LAN Failover set to add or remove.

The active LAN device in the LAN Failover set cannot be removed without disabling the failover set or making another device active (see the **/SWITCH** qualifier).

/PRIORITY=*value*

Sets the failover priority of a LAN device. Specify a 32-bit signed integer for the *value* parameter. When choosing among available link up devices, the device with the highest priority is selected.

When a LAN device with a higher priority becomes available, a failover transition to the newly available device is performed. This allows a system manager to set a preferred device by setting one LAN device to a higher priority than others.

This qualifier is not necessary for LAN Failover devices, as they inherit the priority of the attached LAN device when the LAN Failover set is enabled.

/SIZE=*value*

Sets the packet size of the LAN Failover device.

Specify one of the following values:

- STANDARD (Default) – The Ethernet maximum packet size of 1518 bytes (MAC header + data + CRC, excluding VLAN tags).
- JUMBO – The oversize packet size, supported by Gigabit and faster Ethernet devices, of 9018 bytes (MAC header + data + CRC, excluding VLAN tags).

  JUMBO is allowed if all of the LAN devices in the LAN Failover set support jumbo frames.

Note

The **/SIZE** qualifier controls the jumbo setting of the LAN Failover set. The jumbo setting of the individual members of the LAN Failover set is irrelevant.

Examples

1. `LANCP> DEFINE DEVICE EIA0/DLL=(ENABLE,EXCLUSIVE)`

   This command defines LAN device `EIA0` 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/DLL=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.

DEFINE NODE

DEFINE NODE — Enters a node into the LAN permanent node database or modifies an existing entry. Similar to the **SET NODE** command. 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. Requires SYSPRV privilege.

Format

DEFINE NODE *node-name*

Parameter

node-name

Specifies the name of a node in the LAN permanent node database. When entering a new node, you can use the system node name recorded in the system parameter SCSNODE or another name limited to 63 characters in length.

Qualifiers

/ADDRESS=*node-address*
/NOADDRESS (Default)

Associates a LAN address with the node name. The address is six 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 specified as a node name with an extension. For example, VMSSYS.EIA for the EIA device on node VMSSYS, or VMSSYS_1 for the first LAN device on node VMSSYS.

If the **/ADDRESS** qualifier is omitted, the setting remains unchanged. The **/NOADDRESS** qualifier clears the field.

/ALL

Sets data for all nodes in the LAN volatile node database. If a node name is specified, all matching nodes are selected. For example, **DEFINE NODE A/ALL** selects all nodes beginning with the letter A.

/BOOT_TYPE=*boot-option*
/NOBOOT_TYPE

Indicates the type of processing required for downline load requests. Specify one of the following keywords with this qualifier:

- ALPHA_SATELLITE – The OpenVMS cluster satellite is an Alpha system.
- I64_SATELLITE – The OpenVMS cluster satellite is an IA-64 system.
- 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 to be appended to the load image given by the **/FILE** qualifier. The default value is OTHER.

If the **/BOOT_TYPE** qualifier is omitted, the setting remains unchanged. The **/NOBOOT_TYPE** qualifier clears the field.

Note

IA-64 satellites boot using TFTP rather than MOP services; this boot type is included to maintain information in the node database.

/DECNET_ADDRESS=*value*
/NODECNET_ADDRESS

Associates a DECnet address with the node name. The address is specified in DECnet notation (*xx.xxxx*).

If the **/DECNET_ADDRESS=*value*** qualifier is omitted, the setting remains unchanged. The **/NODECNET_ADDRESS** qualifier clears the field.

/FILE=*file-spec*
/NOFILE

Specifies 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.

If the **/FILE** qualifier is omitted, the setting remains unchanged. The **/NOFILE** qualifier clears the field.

/IP_ADDRESS=*value*
/NOIP_ADDRESS

Associates an IP address with the node name. The address is specified in the standard IPv4 notation (*xxx.xxx.xxx.xxx*).

If the **/IP_ADDRESS=*value*** qualifier is omitted, the setting remains unchanged. The **/NOIP_ADDRESS** qualifier clears the filed.

/ROOT=*directory-spec*
/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 the **/ROOT** qualifier is omitted, 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. A larger size ensures better load performance and less server overhead.

If the **/SIZE** qualifier is omitted, 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 that have a problem with MOP Version 4 booting to load. This qualifier causes the requesting node to fail over from MOP Version 4 to MOP Version 3 when no response has been received to a MOP Version 4 load request.

If the **/V3** qualifier is omitted, 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 a system manager 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> DEFINE NODE ZAPNOT/ADDRESS=08-00-2B-11-22-33/FILE=APB.EXE -
_LANCP> /ROOT=$1$DGA14:<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 1DGA14:<SYS10.SYSCOMMON.SYSEXE>. Note that <SYSCOMMON.SYSEXE> is supplied by the LANACP LAN Server process and is not included in the root definition.

2.
```
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`.

   Since the file does not include a directory specification, the logical name LAN$DLL defines the location. You can include the directory specification in the file name or use the **/ROOT** qualifier.

   Note that specifying the file name explicitly in the boot command overrides the file name specified in the node database entry.

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

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

       Enters a device into the LAN permanent device database or
       modifies an existing device entry.

       Format

         DEFINE DEVICE device-name

                                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.
```

```
    Additional information available:

    Parameter  Examples

DEFINE DEVICE Subtopic?
```

This command provides online help for the LANCP command **DEFINE DEVICE**.

LIST BANDWIDTH

LIST BANDWIDTH — Displays bandwidth monitoring settings in the LAN permanent device database.

Format

LIST BANDWIDTH

Parameters

None.

Qualifier

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

Example

```
LANCP> LIST BANDWIDTH

Device Listing, permanent database (26-MAR-2026 15:04:57.97):
              --- Bandwidth Settings ---
  Device   Hours(minimum)   Threshold(bytes)
   EIA0             2              20000
```

In this example, the permanent device database settings override the defaults for `EIA0`.

LIST DEVICE

LIST DEVICE — Displays information in the LAN permanent device database.

Format

LIST DEVICE *device-name*

Parameter

device-name

Specifies the LAN device name. The device name has the form *ddcu*:, where *dd* is the device identifier, *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 EI Ethernet device is specified as EIA0, the second as EIB0, and so on. It is possible to specify the first EI device, for example, as EIA, EIA0, or EIA0:.

A partial device name selects all matching devices. For example, specify E to select all Ethernet devices, V for VLAN devices, EI for Ethernet EI devices, or no *device-name* to select all devices.

Qualifiers

/ALL

Lists all matching LAN devices in the LAN permanent device database. If no device name is provided, all devices are listed.

LIST and **SHOW** commands implicitly include this qualifier.

/BANDWIDTH

Lists bandwidth monitoring settings for devices.

/CHARACTERISTICS

Lists device characteristics.

/DLL

Displays MOP downline load settings.

/FAILOVER

Displays LAN Failover settings.

/LLAN

Equivalent to the **/FAILOVER** qualifier.

/MOPDLL

Equivalent to the **/DLL** qualifier.

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

/PARAMETERS

Equivalent to the **/CHARACTERISTICS** qualifier.

/TEST

Lists test characteristics set for each device.

/TRACE

Lists driver trace parameters set for each device.

/VLAN

Lists VLAN devices and characteristics.

Examples

1. `LANCP> LIST DEVICE/DLL`

```
Device Listing, permanent  database:
              --- MOP Downline Load Service Characteristics ---
   Device     State   Access Mode      Clients           Data Size
   ------     -----   -----------      -------           ---------
   EIA0      Enabled  Exclusive    KnownClientsOnly      1400 bytes
   EWA0      Disabled NoExclusive  NoKnownClientsOnly     246 bytes
```

This command displays MOP downline load information in the LAN permanent device database for all known devices.

2. `LANCP> LIST DEVICE/DLL EIA0`

```
Device Listing, permanent database:
              --- MOP Downline Load Service Characteristics ---
   Device     State   Access Mode      Clients           Data Size
   ------     -----   -----------      -------           ---------
   EIA0      Enabled  Exclusive    KnownClientsOnly      1400 bytes
```

This command displays MOP downline load information in the LAN permanent device database for device `EIA0`.

LIST DLL

LIST DLL — Displays MOP downline load settings in the LAN permanent device database. Equivalent to the **LIST MOPDLL** command.

Format

LIST DLL

Parameters

None.

Qualifiers

None.

Example

```
LANCP> LIST DLL

Device Listing, permanent database (5-APR-2026 19:41:43.46):
                --- Downline Load Service Characteristics ---
  Device     State   Access Mode      Clients           Data Size
   EIA0     Disabled NoExclusive   NoKnownClientsOnly       0 bytes
```

This command displays MOP downline load settings in the LAN permanent device database.

LIST FAILOVER

LIST FAILOVER — Displays LAN Failover devices and status for the LAN permanent device database. Equivalent to the **LIST LLAN** command.

Format

LIST FAILOVER

Parameters

None.

Qualifier

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

Example

```
LANCP> LIST FAILOVER

Device Listing, permanent database (5-APR-2026 19:42:31.09):
                 --- Failover Characteristics ---
  Device     ------- Failset Members --------      Enabled      Size
   LLA0    EIA EIB                                  Yes     Jumbo(9018)
   LLB0    EIF EIG EIH                              Yes     Standard(1518)
```

This command displays LAN Failover devices and status for the LAN permanent device database.

LIST LLAN

LIST LLAN — Displays LAN Failover devices and status for the LAN permanent device database. Equivalent to the **LIST FAILOVER** command.

Format

LIST LLAN

Parameters

None.

Qualifier

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

LIST MOPDLL

LIST MOPDLL — Displays MOP downline load settings in the LAN permanent device database. Equivalent to the **LIST DLL** command.

Format

LIST MOPDLL

Parameters

None.

Qualifiers

None.

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 name of a node in the LAN permanent node database.

A partial node name selects all matching nodes. For example, specify **LIST NODE A** to select all nodes starting with the letter A, or no *node-name* to select all nodes.

Qualifiers

/ALL

Lists all matching nodes in the LAN permanent node database. If no node name is provided, all nodes are listed.

LIST and **SHOW** commands implicitly include this qualifier.

/COMFILE=*file-spec*

Writes command output to the specified file in the form of a list of **DEFINE NODE** commands. The resulting command file can be used to populate the LAN node database.

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

Example

```
LANCP> LIST NODE
Node Listing:

GALAXY (08-00-2B-2C-51-28):
  MOP DLL: Load file:   APB.EXE
           Load root:   $1$DGA24:<SYS11.>
           Boot type:   Alpha 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-2025 09:11:17
               on EIA0 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-2025 09:11:17.29
```

This example shows output from a **LIST NODE** command issued on a system with two nodes defined (GALAXY 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.

LIST TEST

LIST TEST — Lists test settings in the LAN permanent device database.

Format

LIST TEST

Parameters

None.

Qualifier

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

LIST TRACE

LIST TRACE — Lists trace settings in the LAN permanent device database.

Format

LIST TRACE

Parameters

None.

Qualifier

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

Example

```
LANCP> DEFINE DEVICE/TRACE=(PK,SIZE=10000) EIC

LANCP> LIST DEVICE/TRACE EIC

Device Listing, permanent database (29-MAR-2026 09:57:34.69):
                     - Trace Settings -
  Device       Trace Mask         Stop Mask     Buffer Size
   EIC0    02020000 00000000   00000000 00000000     10000
```

In this example, tracing is set for `EIC0` to record transmit and receive packets.

LIST VLAN

LIST VLAN — Lists VLAN settings in the LAN permanent device database.

Format

LIST VLAN

Parameters

None.

Qualifier

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

Example

```
LANCP> LIST VLAN

Device Listing, permanent database (29-MAR-2026 09:55:36.49):
    --- VLAN Characteristics ---
  Device   Parent         VLAN tag
   VLA0      EIH      0x00000022 34
```

This command lists VLAN settings in the LAN permanent device database.

PURGE DEVICE

PURGE DEVICE — Deletes device data from the LAN permanent device database. If you do not select particular data to delete via qualifier, the entire device entry is deleted. Requires SYSPRV privilege.

Format

PURGE DEVICE *device-name*

Parameter

device-name

Specifies the LAN device name. The device name has the form *ddcu*:, where *dd* is the device identifier, *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 EI Ethernet device is specified as EIA0, the second as EIB0, and so on. It is possible to specify the first EI device, for example, as EIA, EIA0, or EIA0:.

A partial device name, accompanied by the **/ALL** qualifier, selects all matching devices. For example, specify E to select all Ethernet devices, V for VLAN devices, EI for Ethernet EI devices, or no *device-name* to select all LAN devices.

Qualifiers

/ALL

Deletes requested data for all LAN devices in the LAN permanent device database. If a device name is specified, data is deleted for all matching LAN devices. For example, **EI/ALL** deletes data for all EI LAN devices.

/CHARACTERISTICS

Deletes device characteristics settings for each LAN device, such as speed, duplex mode, and other device parameters.

/DLL

Deletes MOP downline load settings for each LAN device.

/MOPDLL

Equivalent to the **/DLL** qualifier.

/PARAMETERS

Equivalent to the **/CHARACTERISTICS** qualifier.

/TEST

Deletes test settings for each LAN device.

/TRACE

Deletes driver tracing settings for each LAN device.

Example

```
LANCP> PURGE DEVICE/ALL
```

This command deletes all devices from the LAN permanent device database.

PURGE NODE

PURGE NODE — Deletes a node from the LAN permanent node database. Requires SYSPRV privilege.

Format

PURGE NODE *node-name*

Parameter

node-name

Specifies the name of a node in the LAN permanent node database.

A partial node name selects all matching nodes. For example, specify **PURGE NODE A** to select all nodes starting with the letter A, or no *node-name* to select all nodes.

Qualifier

/ALL

Deletes all LAN nodes in the LAN permanent node database. If a node name is specified, all matching nodes are deleted. For example, **PURGE NODE A/ALL** deletes all nodes whose name begins with the letter A.

Example

```
LANCP> PURGE NODE/ALL
```

This command deletes all nodes from the LAN permanent node database.

SET ACP

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 with Cluster Parameters message 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:

```
LANCP> 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 console and included in the log file SYS$MANAGER:LAN$ACP.LOG written by LANACP.

/START

Starts the LANACP process by executing the SYS$STARTUP:LAN$STARTUP.COM command procedure.

/STOP

Stops the LANACP process. It can be restarted by executing the following command:

```
LANCP> SET ACP/START
```

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

SET DEVICE — Sets LAN device parameters, creating or updating the volatile device database entry as needed. Requires PHY_IO privilege.

Format

SET DEVICE *device-name*

Parameter

device-name

Specifies the LAN device name. The device name has the form *ddcu*:, where *dd* is the device identifier, *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 EI Ethernet device is specified as EIA0, the second as EIB0, and so on. It is possible to specify the first EI device, for example, as EIA, EIA0, or EIA0:.

A partial device name, accompanied by the **/ALL** qualifier, selects all matching devices. For example, specify E to select all Ethernet devices, V for VLAN devices, EI for Ethernet EI devices, or no *device-name* to select all LAN devices.

Qualifiers (General)

/ALL

Sets data for all LAN devices. If a device name is specified, all matching LAN devices are selected. For example, specify E to select all Ethernet devices, V for VLAN devices, or EW to select all Ethernet EW devices.

/BANDWIDTH=(*hours-option*, *threshold-option*, *default-option*, *restart-option*)
/NOBANDWIDTH

Specifies bandwidth monitoring settings for devices. Refer to the *VSI LAN Bandwidth Monitoring Guide for OpenVMS x86-64* [https://docs.vmssoftware.com/vsi-lan-bandwidth-monitoring-guide-for-openvms-x86-64/] for more information.

Bandwidth is recorded in a buffer allocated from S2 space, a list of 32-byte buckets. Each bucket contains data for one second. A bucket may be extended across multiple seconds according to the *threshold-option*, to extend coverage of the bandwidth buffer during idle periods (see the description of *threshold-option*).

Note that one bucket is reserved for bandwidth context.

If the recording buffer cannot be allocated, bandwidth recording is not done.

The default bandwidth buffer size is determined by the LAN_FLAGS system parameter value extracted from bits <11:9>. This occurs either when the LAN device is configured (usually at boot time), or when a LANCP **SET DEVICE/BANDWIDTH** command is issued. The buffer size is set according to the following table:

LAN_FLAGS <11:9>	Buckets	Buffer Size (Bytes)	Minimum Hours[1]
0x00000000	2048	65536	0.6
0x00000200	32768	1048576	9.1
0x00000400	65536	2097152	18.2
0x00000600	98304	3145728	27.3
0x00000800	131072	4194304	36.4
0x00000A00	163840	5242880	45.5
0x00000C00	196608	6291456	54.6

LAN_FLAGS <11:9>	Buckets	Buffer Size (Bytes)	Minimum Hours[1]
0x00000E00	229376	7340032	63.7

[1]The time spanned by the bandwidth buffer when the threshold is set to 0 bytes, indicating one second per bucket.

You can specify the following keywords with this qualifier:

- *hours-option*

  HOURS=*value* (Default is 1 hour)

  Specifies the size of the bandwidth recording buffer as hours of data collected, 32 bytes per second, 115200 bytes per hour.

  If the calculated buffer is different from the current size, then the existing buffer is deallocated, a new buffer is allocated with the requested size, and bandwidth recording is restarted.

  The maximum hours that can be specified is 8760 (one year of data collection with a threshold value of zero). The buffer size is set according to the following table:

Hours Value	Buffer Size (Bytes)	Number of Buckets
1	115200	3600
2	230400	7200
24 (day)	2764800	86400
744 (month)	85708800	2678400
8760 (year)	1009152000	31536000

  Specifying the *hours-option* overrides the LAN_FLAGS setting.

  Specifying HOURS=0 is equivalent to **`SET DEVICE/NOBANDWIDTH`**.

- *threshold-option*

  THRESHOLD=*bytes* (Default is 10000 bytes)

  Specifies the threshold at which a new bandwidth entry is started. Activity below this threshold is considered noise and is not recorded in detail. When the amount of transmit and receive data over the last second is less than the threshold, the total is accumulated in the current bandwidth bucket and we do not advance to the next bucket.

  Note that 0 bytes in one second does get recorded, as that is an unusual amount of activity. For example, a 20-second period with 0 bytes of activity would be recorded as a 20-second-long bucket with no activity.

  For a nonzero threshold value, each bucket is one of the following:

  - Data for consecutive seconds where bytes were zero.
  - Data for one second where bytes were greater than or equal to the threshold value.
  - Data for consecutive seconds where bytes were nonzero and less than the threshold value.

  For a threshold value of zero, all bandwidth buckets contain one second of activity.

- *default-option*

  DEFAULT

  Restarts bandwidth monitoring with default settings, discarding any data recorded already.

- *restart-option*

  RESTART

  Restarts bandwidth monitoring, discarding any data recorded already.

/DESCRIPTION="*quoted-string*"

The quoted string provides additional information that is displayed by the LANCP **SHOW CONFIGURATION** command. An example of usage is to identify the switch connection for a device.

/DEVICE_SPECIFIC=(FUNCTION="*xxxx*",VALUE=*n*)
/NODEVICE_SPECIFIC

Allows device-specific parameters to be adjusted. These are used for debugging purposes or to override default device characteristics.

Specify **/NODEVICE_SPECIFIC** to clear all device-specific parameter data.

Refer to the *VSI LAN Devices, Counters, and Functions Reference Manual for OpenVMS Alpha and IA-64* [https://docs.vmssoftware.com/vsi-lan-devices-counters-and-functions-reference-openvms-alpha-ia64/] or *VSI LAN Devices, Counters, and Functions Reference Manual for OpenVMS x86-64* [https://docs.vmssoftware.com/vsi-lan-devices-counters-and-functions-reference-openvms-x86/] for a description of these functions.

/DLL=(*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)

  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 and DECnet-Plus implementations that include MOP support).

Note

When you select NOEXCLUSIVE, LANACP responds only to MOP downline load requests directed to nodes defined in the LAN node database.

- *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, minus the 802E header and CRC (26 bytes) and MOP the protocol overhead (10 bytes).

  You can override the size on a per-node basis. See the **DEFINE NODE** and **SET NODE** commands for details.

- *knownclientsonly-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.

Note

This option is not available when NOEXCLUSIVE mode is selected. In NOEXCLUSIVE mode, LANACP services MOP downline load requests only for clients defined in the LAN 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_DATABASE (SET DEVICE Only)

Updates the device entries in the LAN volatile device database with any data currently set in the permanent database. This allows a system manager 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.

/PORT=*value*

Sets the port number to use when defining a UDP pseudo device.

/TAG=*value* (VLAN Devices Only)

Specifies the IEEE 802.1Q tag for the VLAN device.

A VLAN tag consists of a 16-bit TPID (Tag Protocol Identifier), a 3-bit priority, a 1-bit CFI (Canonical Format Indicator), and a 12-bit VLAN ID.

The VLAN TPID, priority, and CFI tag fields must be zero. As a result, the VLAN tag field is only the VLAN ID and is restricted to values 1–4095.

Once set, the tag cannot be changed using this qualifier. In order to change it, either deactivate the VLAN device to reuse the existing device name or create a new VLAN device.

/TEST
/LOOP
/RECEIVE
/TOGGLE
/TRANSMIT

These qualifiers implement test functionality built into LANCP for purposes of debugging and validation. For a complete description of the test functionality, refer to the LANCP online help.

/TRACE=(*size-option*, *mask-option*, *stop-option*, *keywords*)
/NOTRACE

Provides the LAN driver trace settings for the device. By default, the LAN drivers perform tracing of error conditions and state changes. Tracing is controlled by an event mask that selects events to trace, a stop mask that specifies when to stop tracing, and the size of the trace buffer. Keywords provide a user-friendly way of specifying the event mask that consists of two hexadecimal values.

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; it is more substantial when all events are selected, and yet more so when full packet tracing is enabled.

The command **`SHOW DEVICE/TRACE`** displays trace results. The trace settings are displayed by **`SHOW DEVICE/TRACE/HEADER`**.

Stop tracing with the command **`SET DEVICE/NOTRACE`**.

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 (32 bytes each). The default is 2048 entries. The initial and minimum size is 512 entries and the maximum is 1000000.

- *mask-option*

MASK=(*value1*,*value2*)

Specifies the trace mask to select which 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 this device.

The mask can be specified by a numerical value and/or a list of keywords that identify the bits to be set.

- *stop-option*

  STOP=(*value1*,*value2*)

  Specifies the trace mask to select which type of entries should stop tracing. When a trace entry that matches one of the bits in the stop mask is made, the trace mask is cleared so that you can look at the trace data accumulated so far.

  The mask can be specified by a numerical value or a list of keywords that identify the bits to be set.

- *keywords*
  - DEFAULT – Requests that the trace mask be set to the LAN driver default trace mask.
  - PAUSE – Stops tracing but does not deallocate the trace buffer, allowing it to be read without being disturbed by more tracing.
  - RTINT – Selects receive done, transmit issue, interrupt events.
  - RFTINT – Selects the same events as RTINT, plus fork start and fork done events.
  - PK – Selects full size transmit and receive packets.
  - SMALLPK – Selects transmit and receive packets, saving to 32 bytes.
  - MEDIUMPK – Selects transmit and receive packets, saving up to 64 bytes.
  - LARGEPK – Selects transmit and receive packets, saving up to 96 bytes.
  - XFULL, RFULL – Selects transmit, receive packets full size.
  - XSMALL, RSMALL – Selects transmit, receive packets up to 32 bytes.
  - XMEDIUM, RMEDIUM – Selects transmit, receive packets up to 64 bytes.
  - XLARGE, RLARGE – Selects transmit, receive packets up to 96 bytes.
  - MALL – Selects all bits for the trace mask.
  - M00–M63 – Selects individual bits for the trace mask.
  - MNONE – Selects no bits for the trace mask.
  - SALL – Selects all bits for the trace stop mask.
  - S00–S63 – Selects individual bits for the trace stop mask.

- SNONE – Selects no bits for the trace stop mask.

The remaining events are listed in the trace header displayed by the command **SHOW DEVICE/TRACE/HEADER**. The keyword is listed in uppercase and precedes the description of the trace entry. All of the keywords except the ones above can be negated.

/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_DATABASE** qualifier. You can combine the **/UPDATE** and **/PERMANENT_DATABASE** qualifiers in a single **SET DEVICE** command.

/VLAN_DEVICE=*device-name*
/NOVLAN_DEVICE

Specifies the physical LAN device that will host the VLAN device. The **/NOVLAN_DEVICE** qualifier requests deactivation of the VLAN device.

When deactivating the VLAN device, the *device-name* is not required because the VLAN driver knows which device is hosting the VLAN device.

When creating a new VLAN device or reusing a deactivated device, the **/TAG** qualifier must be included to supply the VLAN ID.

Notes

A VLAN device cannot be deactivated if any applications continue to use the device.

The LAN Failover device (LL*x*) is considered a physical LAN device. It acts as a single failover device and represents a set of physical LAN devices.

Be aware that an OpenVMS virtual machine considers hypervisor NICs to be physical LAN devices.

Qualifiers (Ethernet Devices)

Note

For virtual devices, auto-negotiation, duplex mode, and flow control settings are determined by the hypervisor and associated network device driver. The OpenVMS LAN driver overrides any requested setting to set auto-negotation to enabled, full-duplex mode, and flow control to enabled.

/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.

After you issue a command using this qualifier, the Ethernet driver may redo 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. Gigabit and faster Ethernet devices support jumbo frames.

The jumbo frame size is 9018 bytes (MAC header + data + CRC, excluding VLAN tags).

/MEDIA=*value*

Selects the cable connection. Normally, the selection is made during device initialization using a limited autosensing algorithm that selects twisted pair, but it 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 qualifier allows you to change the selection without rebooting.

Acceptable values are AUI (10Base2, 10Base5), TWISTEDPAIR (10BaseT), BNC, UTP, STP, and AUTOSENSE. The default value is AUTOSENSE.

/SPEED=*value*

Sets the speed of the LAN, in Mbit/second.

Valid values are 10, 100, 1000, 2500, 5000, 10000, 25000, 40000, 100000, or AUTONEGOTIATE.

AUTONEGOTIATE indicates that the driver should determine the speed either by auto-negotiation or by device characteristics.

Emulated virtual devices are nominally 1000 Mbit/second, but a system manager can override the nominal speed.

After you issue a command using this qualifier, the Ethernet driver may redo link setup and auto-negotiation for the device.

Qualifiers (LAN Failover Devices)

/DISABLE

Disables the LAN Failover set, deactivating the active LAN Failover device and disallowing I/O to it.

/ENABLE

Enables a LAN Failover set, activating the LAN Failover 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 LAN Failover device.

/FAILOVER_SET=(*device-name*[,...])
/NOFAILOVER_SET=(*device-name*[,...])

Specifies the members of a LAN Failover set to add or remove.

The active LAN device in the LAN Failover set cannot be removed without disabling the failover set or making another device active (see the **/SWITCH** qualifier).

/PRIORITY=*value*

Sets the failover priority of a LAN device. Specify a 32-bit signed integer for the *value* parameter. When choosing among available link up devices, the device with the highest priority is selected.

When a LAN device with a higher priority becomes available, a failover transition to the newly available device is performed. This allows a system manager to set a preferred device by setting one LAN device to a higher priority than others.

This qualifier is not necessary for LAN Failover devices, as they inherit the priority of the attached LAN device when the LAN Failover set is enabled.

/SIZE=*value*

Sets the packet size of the LAN Failover device.

Specify one of the following values:

- STANDARD (Default) – The Ethernet maximum packet size of 1518 bytes (MAC header + data + CRC, excluding VLAN tags).
- JUMBO – The oversize packet size, supported by Gigabit and faster Ethernet devices, of 9018 bytes (MAC header + data + CRC, excluding VLAN tags).

  JUMBO is allowed if all of the LAN devices in the LAN Failover set support jumbo frames.

Note

The **/SIZE** qualifier controls the jumbo setting of the LAN Failover set. The jumbo setting of the individual members of the LAN Failover set is irrelevant.

/SWITCH (SET DEVICE Only)

Forces a LAN Failover transition to another member of the LAN Failover set.

You can test LAN Failover operation by using this qualifier to switch to another device in the failover set. The switch is accomplished by resetting the active device. The LAN Failover driver then selects the next available device, according to link state and device priority.

To force a LAN Failover transition to any member of the set, adjust the priority of the devices before using the **/SWITCH** qualifier.

Examples

1. `LANCP> SET DEVICE/MEDIA=TWISTEDPAIR EWB0`

   The command in this example sets the media type to twisted pair for `EWB0`.

2. `LANCP> SET DEVICE EIA0/DLL=ENABLE`

   The command in this example enables MOP downline load service for `EIA0`, leaving the remaining `MOPDLL` parameters unchanged.

3. `LANCP> SET DEVICE EIA0/DLL=(ENABLE,EXCLUSIVE,SIZE=1482)`

   The command in this example enables MOP downline load service for `EIA0` in exclusive mode with the data transfer size of `1482` bytes, leaving the remaining `MOPDLL` parameters unchanged.

4. `LANCP> SET DEVICE EIA0/DLL=(ENABLE,NOEXCLUSIVE)`

   The command in this example enables LANACP MOP downline load service for `EIA0` in nonexclusive mode.

SET NODE

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

Specifies 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 the one specified by the system parameter SCSNODE, but this is not a requirement. The node name is limited to 63 characters in length.

Qualifiers

/ADDRESS=*node-address*
/NOADDRESS (Default)

Associates a LAN address with the node name. The address is six 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 specified as a node name with an extension. For example, VMSSYS.EIA for the EIA device on node VMSSYS, or VMSSYS_1 for the first LAN device on node VMSSYS.

If the **/ADDRESS** qualifier is omitted, the setting remains unchanged. The **/NOADDRESS** qualifier clears the field.

/ALL

Sets data for all nodes in the LAN volatile node database. If a node name is specified, all matching nodes are selected. For example, **SET NODE A/ALL** selects all nodes beginning with the letter A.

/BOOT_TYPE=*boot-option*
/NOBOOT_TYPE

Indicates the type of processing required for downline load requests. Specify one of the following keywords with this qualifier:

- ALPHA_SATELLITE – The OpenVMS cluster satellite is an Alpha system.
- I64_SATELLITE – The OpenVMS cluster satellite is an IA-64 system.
- 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 to be appended to the load image given by the **/FILE** qualifier. The default value is OTHER.

If the **/BOOT_TYPE** qualifier is omitted, the setting remains unchanged. The **/NOBOOT_TYPE** qualifier clears the field.

Note

IA-64 satellites boot using TFTP rather than MOP services; this boot type is included to maintain information in the node database.

/DECNET_ADDRESS=*value*
/NODECNET_ADDRESS

Associates a DECnet address with the node name. The address is specified in DECnet notation (*xx.xxxx*).

If the **/DECNET_ADDRESS=*value*** qualifier is omitted, the setting remains unchanged. The **/NODECNET_ADDRESS** qualifier clears the field.

/FILE=*file-spec*
/NOFILE

Specifies 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.

If the **/FILE** qualifier is omitted, the setting remains unchanged. The **/NOFILE** qualifier clears the field.

/IP_ADDRESS=*value*
/NOIP_ADDRESS

Associates an IP address with the node name. The address is specified in the standard IPv4 notation (*xxx.xxx.xxx.xxx*).

If the **/IP_ADDRESS=*value*** qualifier is omitted, the setting remains unchanged. The **/NOIP_ADDRESS** qualifier clears the filed.

/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 a system manager 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-spec*
/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 the **/ROOT** qualifier is omitted, 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. A larger size ensures better load performance and less server overhead.

If the **/SIZE** qualifier is omitted, 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 that have a problem with MOP Version 4 booting to load. This qualifier causes the requesting node to fail over from MOP Version 4 to MOP Version 3 when no response has been received to a MOP Version 4 load request.

If the **/V3** qualifier is omitted, the setting remains unchanged. The **/NOV3** qualifier clears the setting.

Examples

1.
```
LANCP> SET NODE VMSSYS/ADDRESS=08-00-2B-11-22-33/FILE=APB.EXE -
_LANCP> /ROOT=$1$DGA14:<SYS10.>/BOOT_TYPE=ALPHA_SATELLITE
```

   This command sets up node VMSSYS for booting as an Alpha satellite into an OpenVMS cluster.

   The APB.EXE file is actually located on 1DGA14:<SYS10.SYSCOMMON.SYSEXE>. Note that <SYSCOMMON.SYSEXE> is supplied by the LANACP LAN Server process and is not included in the root definition.

2.
```
LANCP> SET NODE VMSSYS/ADDRESS=08-00-2B-11-22-33/NOROOT
```

This command changes the LAN address associated with node VMSSYS and clears the current root specification.

3. LANCP> 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 a file name is received from node CALPAL.

   Since 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 use the **/ROOT** qualifier.

   Note that specifying the file name explicitly in the boot command overrides the file name specified in the node database entry.

SET VERIFY

SET VERIFY — Controls whether command lines and data lines in command procedures are displayed in the terminal or are printed in a batch job log.

Format

SET [NO]VERIFY

Parameters

None.

Qualifiers

None.

SHOW ACP

SHOW ACP — Displays the settings of various options for LANCP and LANACP.

Format

SHOW ACP

Parameters

None.

Qualifier

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

Example

```
LANCP> SHOW ACP
```

```
LANCP/LANACP options (26-FEB-2026 09:32:36.61):
  Verify is OFF
  OPCOM messages are ENABLED
  DLL packet tracing is DISABLED
```

The command in this example displays how LANCP and LANACP are currently configured.

SHOW BANDWIDTH

SHOW BANDWIDTH — Displays bandwidth monitoring settings for all devices.

Format

SHOW BANDWIDTH

Parameters

None.

Qualifier

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

Example

```
LANCP> SHOW BANDWIDTH

Bandwidth Settings (26-MAR-2026 09:18:07.59):
                                    - Bandwidth Settings -
  Device     Buckets  Hours(minimum)  Threshold(bytes)   Oldest Monitor Data     Wrapped  Current Bucket
   EIA0       2047        0.6             10000        25-MAR-2026 21:14:22        1         1842
   EIB0       2047        0.6             10000        25-MAR-2026 21:14:34        1         1867
   EIC0       2047        0.6             10000        25-MAR-2026 21:14:32        1         1898
   EID0       2047        0.6             10000        25-MAR-2026 21:14:31        1         1889
   EIE0       2047        0.6             10000        25-MAR-2026 21:14:40        1         1910
   EIF0       2047        0.6             10000        25-MAR-2026 21:14:23        1         1891
```

This command displays bandwidth monitoring settings for all devices. Refer to the *VSI LAN Bandwidth Monitoring Guide for OpenVMS x86-64* [https://docs.vmssoftware.com/vsi-lan-bandwidth-monitoring-guide-for-openvms-x86-64/] for further information on **SHOW BANDWIDTH**.

SHOW CONFIGURATION

SHOW CONFIGURATION — Displays a list of LAN devices and characteristics on the system.

Format

SHOW CONFIGURATION

Parameters

None.

Qualifiers

/ALL

Equivalent to **SHOW CONFIGURATION** with the **/BANDWIDTH** and **/STATUS** qualifiers.

/BANDWIDTH

Shows configuration and bandwidth information for each LAN device, including uptime, packets, bytes, highest bandwidth, and interval data.

/CONFIGURATION

Shows configuration and status information for each LAN device, including link state, buffer size, MAC address, name, and so on. This is the default for **SHOW CONFIGURATION** and **SHOW CONFIGURATION/USER**.

/OUTPUTFILE=*file-spec*

Creates the specified file and directs output to it.

/STATUS

Shows configuration and status information for each LAN device, including uptime, packets, bytes, and errors.

/USERS

Shows configuration information and the protocols running on each LAN device.

Example

```
LANCP> SHOW CONFIGURATION

LAN Configuration (26-FEB-2026 09:30:48.16):
         Parent or
  Device  PrefCPU  Medium/User Version  ----LinkState----  BufSize   MAC Address       Name
   EIA0    17      Ethernet    X-53    Up    1gb Fdx Auto  1500  00-50-56-AC-06-61  i82545 VMware
   EIB0    14      Ethernet    X-19    Up   10gb Fdx Auto  1500  00-50-56-AC-94-A0  VMXNET3
   EIC0    11      Ethernet    X-19    Up   10gb Fdx Auto  1500  00-50-56-AC-15-6B  VMXNET3
   EID0    10      Ethernet    X-19    Up   10gb Fdx Auto  1500  00-50-56-AC-10-F7  VMXNET3
```

This example shows the output from a **SHOW CONFIGURATION** command that was entered on a node that has four LAN devices. The version is typically the device driver version. For example, `X-19` is version 19 of the `VMXNET3` driver.

SHOW DEVICE

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

Specifies the LAN device name. The device name has the form *ddcu*:, where *dd* is the device identifier, *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 EI Ethernet device is specified as EIA0, the second as EIB0, and so on. It is possible to specify the first EI device, for example, as EIA, EIA0, or EIA0:.

A partial device name selects all matching devices. For example, specify E to select all Ethernet devices, V for VLAN devices, EI for Ethernet EI devices, or no *device-name* to select all LAN devices.

Qualifiers

/ALL

Shows all devices. If a device name is specified, all matching LAN devices are displayed. For example, specify E to select all Ethernet devices, V for VLAN devices, or EI to select all Ethernet EI devices.

LIST and **SHOW** commands implicitly include this qualifier.

/BANDWIDTH

Displays bandwidth data collected. Bandwidth data is collected once per second for each LAN device. The output includes a column characterizing the bandwidth as zero, <1Mb, 1Mb, and so on.

You can specify the following additional qualifiers with **/BANDWIDTH** to affect what data is displayed:

- **/EXPANDED**

  Expands compressed data and converts the raw data into one-second intervals.

- **/SUMMARY**

  Provides summary data only. Summary data is collected on one-second intervals (**/EXPANDED** is implied).

- **/SINCE=*time* /DURATION=*time* /LAST=*time***

  Displays results for a period of time. **/SINCE** defaults to TODAY and can be explicitly specified as YESTERDAY or TODAY. **/DURATION** and LAST are equivalent. **/DURATION** or **/LAST** defaults to 60 seconds if no value is provided.

- **/[NO]ZERO /1MB /10MB /100MB /1GB /2GB /5GB /10GB**

  Selects a specific bandwidth. For example, **/100MB** selects intervals with a bandwidth of 100 MBits/sec or greater.

- **/DUMPFILE**

  Writes the binary bandwidth data to the specified output file, including the system name, LAN device name, and binary bandwidth data as read from the LAN device.

Note

If more than one device is selected, only the first device found is processed.

- **/INPUTFILE=*file-spec***

  Reads the binary bandwidth data from the specified file (created using **/DUMPFILE** and **/OUTPUTFILE**) rather than from the LAN device.

The default bandwidth buffer size is 2048 buckets, a total 64 KB per LAN device. The LAN_FLAGS system parameter sets the default buffer size according to the following table:

LAN_FLAGS <11:9>	Buckets	Buffer Size (Bytes)	Minimum Hours[1]
0x00000000	2048	65536	0.6
0x00000200	32768	1048576	9.1
0x00000400	65536	2097152	18.2
0x00000600	98304	3145728	27.3
0x00000800	131072	4194304	36.4
0x00000A00	163840	5242880	45.5
0x00000C00	196608	6291456	54.6
0x00000E00	229376	7340032	63.7

[1]The time spanned by the bandwidth buffer when the threshold is set to 0 bytes, indicating one second per bucket.

/CHARACTERISTICS

Displays status and related information about the device.

/CLEAR_COUNTERS

Clears device counters and internal driver counters after a **SHOW DEVICE/COUNTERS** or **SHOW DEVICE/INTERNAL_COUNTERS** command. Subsequent **SHOW** commands show the counter values accumulated since the previous clear.

This affects counters displayed by LANCP only. Used with the **/COUNTERS** and **/INTERNAL_COUNTERS** qualifiers.

See the **/RESTORE_COUNTERS** qualifier to restore counters to the original values.

/COUNTERS

Displays device counters. By default, it does not display zero counters. To see all counters, including zero, use the additional qualifier **/ZERO**.

/DEBUG

Shows additional counters that may be used for debug purposes.

Applies to the **/INTERNAL_COUNTERS** qualifier.

/DLL

Displays LAN volatile device database information related to MOP downline load for the device.

/FAILOVER

Shows LAN Failover settings.

/GRAND_TOTAL

Displays the grand total of device counters for specified devices. By default, it does not display zero counters. To see all counters, including zero, use the additional qualifier **/ZERO**.

Applies to the **/COUNTERS** qualifier.

/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**.

/MOPDLL

Equivalent to the **/DLL** qualifier.

/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.

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

/PARAMETERS

Equivalent to the **/CHARACTERISTICS** qualifier.

/RESTORE_COUNTERS

Restores device counters and internal driver counters prior to a **SHOW DEVICE/COUNTERS** or **SHOW DEVICE/INTERNAL_COUNTERS** command.

This affects counters displayed by LANCP only. Used with the **/COUNTERS** and **/INTERNAL_COUNTERS** qualifiers.

See the **/CLEAR_COUNTERS** qualifier to clear the counters.

/TEST

Displays test status and configuration for the test functionality built into LANCP for purposes of debug and validation. For a complete description of the test functionality, refer to the LANCP online help.

/TRACE

Displays LAN driver trace data.

You can specify the following additional qualifiers with **/TRACE** to affect the trace data output:

- **/PCAPFILE=*file-spec***

  Without affecting any other output, LANCP writes the transmit and receive packet trace entries to the specified file in binary PCAPNG format, version 1.0. This file can be read by other utilities that process PCAPNG files, such as Wireshark.

- **/DUMPFILE**

  Writes the trace data to the file specified by the **/OUTPUTFILE** qualifier. The resultant file can be read using the **/INPUTFILE** qualifier (so the data can be viewed later) and with the **/SELECT** qualifier (to select particular entries).

- **/INPUTFILE=*file-spec***

  Reads trace data from the specified file that was created using the **/DUMPFILE** qualifier.

- **/REVERSE**

  Displays trace data in reverse order.

- **/SELECT=(*mask-option*,*keywords*)**

  Specifies which trace events to display.

  You can specify the following keywords with this qualifier:

  - *mask-option*

    MASK=(*value1*,*value2*)

    Use MASK=(*value1*,*value2*) to specify the trace mask to select which entries should be displayed from the trace data. 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 device.

    The mask can be specified by a numerical value and/or a list of keywords that identify the bits to be set.

  - *keywords*

    Any event can be specified. The relevant events are listed in the trace header.

- **/[NO]HEADER**

  By default, the trace masks are displayed first, then the trace data. To suppress this trace header data, specify **/NOHEADER**.

  To display only the trace header data, specify **/HEADER**.

/VERSION

Displays the current LAN driver and device version information if available or applicable. Not all LAN drivers maintain version information.

/VLAN

Displays a list of IEEE 802.1Q tags that are configured on the switch port connected to the specified physical LAN device. LANCP listens for the GARP (Generic Attribute Registration Protocol) VLAN Registration Protocol (GVRP) packets that contain the configuration information and displays the configured tags.

The switch periodically sends GVRP packets to provide VLAN configuration information in compliance with the IEEE 802.1Q specification. If GVRP is not configured on the switch, or if the

LAN device is not connected to a switch, this command displays only the list of tags that the VLAN driver has configured.

/ZERO

Shows the counter, even if zero. By default, zero counter values are not displayed.

Applies to the **/COUNTERS** and **/INTERNAL_COUNTERS** qualifiers.

Examples

1.
```
LANCP> SHOW DEVICE/COUNTERS EIC0

Device Counters EIC0 (26-FEB-2026 09:36:23.68):
                  Value  Counter
                1294951 Seconds since last zeroed
           6643476148827 Bytes received
           2119225471944 Bytes sent
              6082943995 Packets received
              1901616462 Packets sent
              2179961029 Multicast bytes received
               171713274 Multicast bytes sent
                14358202 Multicast packets received
                 1244671 Multicast packets sent
                       1 Link up transitions (11-FEB-2026 08:26:52.06)
11-FEB-2026 08:26:52.06 Time of last link up
```

   This command displays counters for Ethernet device `EIC0`.

2.
```
LANCP> SHOW DEVICE/DLL

Device Listing, volatile database:
              --- MOP Downline Load Service Characteristics ---
   Device     State   Access Mode      Clients           Data Size
   ------     -----   -----------      -------           ---------
   EIA0      Enabled  Exclusive    KnownClientsOnly      1400 bytes
   EWA0      Disabled NoExclusive  NoKnownClientsOnly     246 bytes
```

   This command displays MOP downline load information in the LAN volatile device database for all known devices.

3.
```
LANCP> SHOW DEVICE/DLL EIA0

Device Listing, volatile database:
              --- MOP Downline Load Service Characteristics ---
   Device     State   Access Mode      Clients           Data Size
   ------     -----   -----------      -------           ---------
   EIA0      Enabled  Exclusive    KnownClientsOnly      1400 bytes
```

   This command displays MOP downline load information in the LAN volatile device database for device `EIA0`.

4.
```
LANCP> SHOW DEVICE/VERSION EIC0

Device Versions EIC0 (26-FEB-2026 09:34:52.44):
                  Value  Component
                00000000  Device hardware version
      09230049 10000001  Port driver version
```

```
        09230240 10000002  LAN common routines version
```

This command displays version information for Ethernet device `EIC0`.

5. `LANCP> SHOW DEVICE/COUNTERS/INTERNAL_COUNTERS/FAILOVER`

   This command shows that you can select multiple displays on the same command line.

SHOW DLL

SHOW DLL — Displays the current state of MOP downline load services for the system, including devices for which MOP loading is enabled and counters information. Equivalent to the **SHOW MOPDLL** command.

Format

SHOW DLL

Parameters

None.

Qualifier

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

Example

```
LANCP> SHOW DLL

LAN MOP DLL Status:
   EIA enabled in exclusive mode for known nodes only, data size 1482 bytes
   EIB disabled

       #Loads   Packets        Bytes     Last load time       Last loaded
       ------   -------        -----  --------------------  ---------------
   EIA      5      1675      4400620  10-JAN-2026 10:27.51  GALAXY
   EIB      0         0            0
```

On this node, there are two LAN devices, `EIA` and `EIB`. MOP downline load service is enabled on `EIA` 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, no downline loads are recorded for EIB, which is currently disabled for downline load service.

SHOW FAILOVER

SHOW FAILOVER — Displays LAN Failover devices and status for the LAN volatile device database. Equivalent to the **SHOW LLAN** command.

Format

SHOW FAILOVER

Parameters

None.

Qualifier

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

Example

```
LANCP> SHOW FAILOVER

LAN Failover Settings (21-JAN-2026 18:48:23.57):
                            --- Failover Status ---
  Device     ------- Failset Members --------        Failset State              Size
   LLA0    EIA* EIB                                 Enabled/Active          Standard(1518)
   LLB0    EID* EIC                                 Enabled/Active          Standard(1518)
   LLC0                                             Disabled/NoFailset      Jumbo(9018)
```

This command displays LAN Failover devices and status for the LAN volatile device database.

Once created, LL*x* devices are present as long as the system remains online, even if there are no NICs present (see `LLC0`). They may be enabled or disabled by the system manager (indicated in the `State` column). The asterisk (`*`) indicates the currently active NIC in the LAN Failover device. The `Size` refers to the Ethernet packet size, which is the MTU plus the Ethernet header.

SHOW LLAN

SHOW LLAN — Displays LAN Failover devices and status for the LAN volatile device database. Equivalent to the **SHOW FAILOVER** command.

Format

SHOW LLAN

Parameters

None.

Qualifier

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

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

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

Example

```
LANCP> SHOW LOG

SYS$MANAGER:LAN$ACP.LOG latest contents (26-FEB-2026 09:31:25.89):
11-FEB-2026 08:34:37.13 Defined LAN$DLL to be SYS$SYSROOT:[MOM$SYSTEM]
11-FEB-2026 08:34:37.15 Found LAN device EIA0, hardware address 00-50-56-AC-06-35
11-FEB-2026 08:34:37.15 Found LAN device EIB0, hardware address 00-50-56-AC-00-41
11-FEB-2026 08:34:37.16 Found LAN device EIC0, hardware address 00-50-56-AC-19-22
11-FEB-2026 08:34:37.16 %EIA0, Link up: 1000 mbit, 00-50-56-AC-06-35
11-FEB-2026 08:34:37.16 %EIB0, Link up: 1000 mbit, 00-50-56-AC-00-41
11-FEB-2026 08:34:37.16 %EIC0, Link up: 1000 mbit, 00-50-56-AC-19-22
11-FEB-2026 08:34:37.16 Node database file, LAN$NODE_DATABASE, not found
11-FEB-2026 08:34:37.19 LANACP initialization complete
```

This command displays the last 2048 bytes of log data written to the log file SYS$MANAGER:LAN$ACP.LOG.

SHOW MOPDLL

SHOW MOPDLL — Displays the current state of MOP downline load services for the system, including devices for which MOP loading is enabled and counters information. Equivalent to the **SHOW DLL** command.

Format

SHOW MOPDLL

Parameters

None.

Qualifier

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

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 a node name is specified, all matching nodes are selected. For example, **SHOW NODE A/ALL** selects all nodes beginning with the letter A.

LIST and **SHOW** commands implicitly include this qualifier.

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

/TOTAL

Display only counter totals for the selected nodes.

Examples

1.
```
LANCP> SHOW NODE
Node Listing:

GALAXY (08-00-2B-2C-51-28):
 MOP DLL: Load file: APB.EXE
          Load root: $1$DGA14:<SYS11.>
          Boot type: Alpha 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-2025 09:11:17
              on EIA0 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 12-JUN-2025 09:11:17.29
```

This example shows output from a command issued on a local node on which there are two nodes defined (GALAXY 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 VMSSYS

   The command in this example displays node characteristics and counters information from the LAN volatile node database for node VMSSYS.

3. LANCP> SHOW NODE/ALL VMS

   The command in this example displays node characteristics and counters information from the LAN volatile node database for all nodes whose name begins with VMS.

4. LANCP> SHOW NODE/ALL

   The command in this example displays node characteristics and counters information from the LAN volatile node database for all nodes.

5. LANCP> SHOW NODE/ALL/OUTPUTFILE=TMP.INI

   The command in this example writes a list of all nodes to the file TMP.INI.

SHOW TEST

SHOW TEST — Shows test settings for devices.

Format

SHOW TEST

Parameters

None.

Qualifier

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

Example

```
LANCP> SHOW TEST
```

```
Test Settings (26-MAR-2026 09:15:10.14):
                                                    - Test Characteristics -
              Buffer  --Mode--  XmtChain    IO    Test   IO/Sec ...  Packet   Stats
  Device API Alignment AMC Prom  Segments  Count Duration  Rate  ... SizeRange (Sec) User
   EIA0  QIO       0     N    N        1      0        0   100  ...  46-46       10 60-05
   EIB0  QIO       0     N    N        1      0        0   100  ...  46-46       10 60-05
   EIC0  QIO       0     N    N        1      0        0   100  ...  46-46       10 60-05
   EID0  QIO       0     N    N        1      0        0   100  ...  46-46       10 60-05
   EIE0  QIO       0     N    N        1      0        0   100  ...  46-46       10 60-05
   EIF0  QIO       0     N    N        1      0        0   100  ...  46-46       10 60-05
```

This command shows test settings for devices.

SHOW TRACE

SHOW TRACE — Shows trace settings for devices.

Format

SHOW TRACE

Parameters

None.

Qualifier

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

Example

```
LANCP> SHOW TRACE

Trace Settings (26-MAR-2026 09:12:10.41):
                    - Trace Settings -
  Device       Trace Mask    Default Trace Mask      Stop Mask     Buffer Size
   EIA0    00000000 00000000  9C2027F0 000063EF  00000000 00000000       2048
   EIB0    00000000 00000000  9C2027F0 000063EF  00000000 00000000       2048
   EIC0    00000000 00000000  9C2027F0 000063EF  00000000 00000000       2048
   EID0    00000000 00000000  9C2027F0 000063EF  00000000 00000000       2048
   EIE0    BC7C7FFF 00000FFF  9C2027F0 00000FFF  00000000 00000000       2048
   EIF0    00000000 00000000  9C2027F0 000063EF  00000000 00000000       2048
```

This command shows trace settings for devices. Refer to the *VSI OpenVMS LAN Driver Tracing Guide* [https://docs.vmssoftware.com/vsi-openvms-lan-driver-tracing-guide/] for further information on **SHOW TRACE**.

SHOW VLAN

SHOW VLAN — Shows VLAN settings for devices.

Format

SHOW VLAN

Parameters

None.

Qualifier

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

Example

```
LANCP> SHOW VLAN

VLAN Settings (26-MAR-2026 15:00:43.49):
    --- VLAN Characteristics ---
  Device   Parent         VLAN tag
   VLA0      EIA      0x00000021   33
```

This command shows the VLAN devices and settings for each.

SPAWN

SPAWN — Creates a subprocess of the current process. Portions of the current process context are copied to the subprocess.

Format

SPAWN [*command-string*]

Parameter

command-string

Specifies a command string of less than 132 characters that is to be executed in the context of the created subprocess. When the command completes execution, the subprocess terminates and control returns to LANCP.

Qualifiers

None.

Example

```
LANCP> SPAWN

$ MCR 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, the user enters the node information for a node, but is unsure of the root. They use the **SPAWN** command to spawn a subprocess, allowing them to search MODPARAMS.DAT for the node name and correct the root.

START DEVICE

START DEVICE — Starts a test on a LAN device.

Format

START DEVICE *device-name*

Parameter

device-name

Specifies the LAN device name. The device name has the form *ddcu*:, where *dd* is the device identifier, *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 EI Ethernet device is specified as EIA0, the second as EIB0, and so on. It is possible to specify the first EI device, for example, as EIA, EIA0, or EIA0:.

A partial device name accompanied by the **/ALL** qualifier selects all matching devices. For example, specify E to select all Ethernet devices, V for VLAN devices, EI for Ethernet EI devices, or no *device-name* to select all LAN devices.

Qualifiers

/ALL

Starts test activity on all LAN devices. If a device name is specified, all matching LAN devices are selected. For example, specify E to select all Ethernet devices, V for VLAN devices, or EW to select all Ethernet EW devices.

/OUTPUTFILE=*file-spec*

Writes command output to the specified file.

/TEST
/LOOP
/RECEIVE
/TOGGLE
/TRANSMIT

These qualifiers implement test functionality built into LANCP for purposes of debugging and validation. For a complete description of the test functionality, refer to the LANCP online help.

Example

```
LANCP> SET DEVICE EIA/TEST=(ECHO=HEADER,TARGET=52-54-00-AD-7D-8D,PROMISCUOUS)

LANCP> START DEVICE EIA/RECEIVE

Running on EIA0 (52-54-00-3D-C8-C5) node L4K2, Red Hat KVM, 2 CPUs, XH4A, Cluster ...
```

```
Starting QIO receive test, Promiscuous Echo=header PTY=60-05 Size=46 Pipe=1 Rate= ...
 Target 52-54-00-AD-7D-8D
Rcv   36 bytes, DA 01-80-C2-00-00-00, SA E8-2C-6D-F9-61-F1,  DSAP 42, SSAP 42, CTL 03
Rcv  152 bytes, DA 01-00-5E-72-0A-BF, SA 52-54-00-AD-7D-8D,  PTY 08-00
Rcv  230 bytes, DA 01-00-5E-00-00-07, SA 9C-8C-6E-E8-F1-2D,  PTY 08-00
Rcv   63 bytes, DA 48-21-0B-5D-7D-BC, SA 9C-8C-6E-E8-F1-2D,  PTY 08-00
Rcv   63 bytes, DA 98-E7-F4-61-14-73, SA 9C-8C-6E-E8-F1-2D,  PTY 08-00
Rcv  120 bytes, DA AB-00-04-01-BF-0B, SA 52-54-00-AD-7D-8D,  PTY 60-07
Rcv   36 bytes, DA 01-80-C2-00-00-00, SA E8-2C-6D-F9-61-F1,  DSAP 42, SSAP 42, CTL 03
.
.
.
```

In this example, the **/TEST** qualifier is used with the **SET DEVICE** command to set test parameters. The **START DEVICE** command is then used to start receiving in promiscuous mode and displaying packet header data.

TRIGGER NODE

TRIGGER NODE — Issues a request to reboot to a remote node. Rather than specifying 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

Specifies 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 does not have to be defined in the LAN volatile node database. The address consists of six hexadecimal byte characters separated by hyphens or colons.

Qualifiers

/DEVICE=*device-name*

Specifies the LAN device name to be used for sending the trigger boot messages. The device name has the form *ddcu*:, where *dd* is the device identifier, *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 EI Ethernet device is specified as EIA0, the second as EIB0, and so on. It is possible to specify the first EI device, for example, as EIA, EIA0, or EIA0:.

/PASSWORD=*16hexdigits*

Specifies the password to be used when the connection is initiated, in hexadecimal (for example, **/PASSWORD=0123456789ABCDEF**). The default password is 0000000000000000 or (with leading zeros suppressed) simply 0.

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 the specified password.

UPDATE DEVICE

UPDATE DEVICE — Does an update operation on a LAN device. Currently, the only operation supported is reset.

Format

UPDATE DEVICE *device-name*

Parameter

device-name

Specifies the LAN device name. The device name has the form *ddcu*:, where *dd* is the device identifier, *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 EI Ethernet device is specified as EIA0, the second as EIB0, and so on. It is possible to specify the first EI device, for example, as EIA, EIA0, or EIA0:.

Qualifier

/RESET

Specifies that the device is to be reset.

Example

```
LANCP> UPDATE DEVICE EWA0/RESET
```

This command resets Ethernet device EWA0.

WAIT

WAIT — Waits the specified number of seconds. The **WAIT** command is used in command files.

Format

WAIT [*value*]

Parameter

value

Specifies the time to wait in seconds. For example, **WAIT 3** waits for 3 seconds.

If no value is specified, **WAIT** waits for 1 second.

Qualifiers

None.

Example

```
LANCP> WAIT 5
LANCP>
```

This command waits for 5 seconds.

Chapter 14. LAT Control Program (LATCP) Utility

14.1. 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

14.2. 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. (For more information, see the *VSI OpenVMS System Manager's Manual*.)

Syntax

```
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.

14.3. 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 you 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.

Command	Function
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

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.

Syntax

```
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.

Example

```
$ SET PROCESS/NAME="TOP_LEVEL"
$ SPAWN RUN SYS$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

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.

Syntax

```
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 SHOWLINK 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 VSI).
- 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.

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 fora 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. VSI 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.

CREATE PORT

CREATE PORT — Creates a logical port on your local node that connects with are mote 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.

Syntax

```
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.

Note

When creating a port, note the following points:

- VSI 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. For a description of programming an application service, see the *VSI OpenVMS I/O User's Reference Manual*.

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 job wide logical name table.
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. For more details, see the *VSI OpenVMS System Manager's Manual*.

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 time sharing 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.

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

CREATE SERVICE — Creates a service on a service node. You must have OPER privilege to use this command.

Syntax

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 DEC server 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.

/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 nodeless 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 time sharing 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 DEC server 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. (For more information about creating an application service, see the *VSI OpenVMS System Manager's Manual*. The *VSI 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.

2. LATCP> **CREATE SERVICE/APPLICATION GRAPHICS**

   This command creates the service GRAPHICS on the local node. Use the CREATEPORT/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

DEFINE/KEY — Assigns a command string to a function key. For example, you can assign the LATCP command SHOW NODE to a function key.

Syntax

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 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.

DELETE LINK

DELETE LINK — Deletes a logical link from a node. You must have OPER privilege to use this command.

Syntax

```
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

DELETE PORT — Deletes a logical port from a node. You must have OPER privilege to use this command.

Syntax

```
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.

DELETE QUEUE_ENTRY

DELETE QUEUE_ENTRY — Deletes an incoming queued request, or entry, from the local node.

Syntax

```
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

DELETE SERVICE — Deletes a service that your service node currently offers. You must have OPER privilege to use this command.

Syntax

```
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.

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.

Syntax

```
EXIT
```

Parameters

None.

Example

```
LATCP> EXIT
```

This command exits the LATCP program and returns control to the DCL command level.

HELP

HELP — Provides online help information for using the LATCP commands.

Syntax

```
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 displays tops 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

RECALL — Displays previously entered LATCP commands on the screen so that you can execute them again.

Syntax

```
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

REFRESH — Refreshes the display screen so that any output from some other source (such as a broadcast message) is erased from the screen.

Syntax

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

SCROLL — Retrieves information that has scrolled off the screen, either up or down.

Syntax

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

SET LINK — Changes the characteristics of LAT data links. You must have OPER privilege to use this command.

Syntax

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 fora 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

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

SET NODE — Specifies the LAT characteristics of your local node. You must have OPER privilege to use this command.

Syntax

```
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$NODE. A LAT node name should be the same as the DECnet node name. If the node is not running DECnet but will be in the future, VSI recommends that you define SYS$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, VSI 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.

The default value of 80 msecs 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.
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. VSI 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), VSI 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 /NODECNET).
 $!
 $ LCP CREATE LINK FDDI_1 /DEVICE=FCA0 /NODECNET
 $ LCP CREATE LINK FDDI_2 /DEVICE=FCB0 /NODECNET
 $!
 $! 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).

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.) VSI 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:

INCOMING= *value*	Sets the session limit for incoming connections only. The default is no limit (a value of 0).
OUTGOING= *value*	Sets the session limit for outgoing connections only. The default is no limit (a value of 0).
INCOMING= *value*, OUTGOING= *value*	Sets the limit for both outgoing and incoming connections. Enclose both options in parentheses; for example, /SESSION_LIMIT=(INCOMING=20, OUTGOING=25).

- 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 specify 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. VSI 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. (For more information, see the *VSI OpenVMS System Manager's Manual*.) 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 (LATSYM) 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).

/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:

ENABLE= *group-code*[,...]	Gives your node access to the listed user groups.
DISABLE= *group-code*[,...]	Prevents your node from accessing the listed groups. The listed groups were enabled previously.
ENABLE= *group-code*[,...], DISABLE=group-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 *VSI 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 "NODEDUKE, 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.

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

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.

Syntax

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. For a description of configuring remote printers on a terminal server, see the *VSI OpenVMS System Manager's Manual*.

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.

/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 are mote 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 servicenode. 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.

Exampled

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.

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. For information about setting up print queues, see the description of print operations in the *VSI OpenVMS System Manager's Manual*.

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

SET SERVICE — Dynamically changes the characteristics of a locally offered service. You must have OPER privilege to use this command.

Syntax

SET SERVICE service-name

Parameter

service-name

Specifies the service whose characteristics are to be modified. If a servicename 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).

/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 nodeless 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 then 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 FORCETIMESHARING 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.

SHOW LINK

SHOW LINK — Displays the status and LAT characteristics of links on the local node.

Syntax

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.

Counter	Description
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.
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.

Flag	Description
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.

Flag	Description
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.

Flag	Description
Remote failure to defer	A remote station failed to defer frames transmission. Could indicate a misconfigured network.
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:	**Single collision—**The total number of times a frame was successfully transmitted on the second attempt after a normal collision on the first attempt. **Multiple collision—**The total number of times a frame was successfully transmitted on the third or later attempt after normal collisions on previous attempts. **Initially deferred—**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.

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.
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.

Counter	Description
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 data link address, DECnet address, or counters.

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     Messages Sent:              3550507
Multicast Msgs Received:       15096805     Multicast Msgs Sent:         413178
Bytes Received:              1994694325     Bytes Sent:               290838585
Multicast Bytes Received:    1528077909     Multicast Bytes Sent:      32637472
System Buffer Unavailable:         8724     User Buffer Unavailable:       6269
Unrecognized Destination:             0     Data Overrun:                     0

Receive Errors -                          Transmit Errors -
   Block Check Error:                No        Excessive Collisions:        No
   Framing Error:                    No        Carrier Check Failure:       No
   Frame Too Long:                   No        Short Circuit:               No
   Frame Status Error:               No        Open Circuit:                No
   Frame Length Error:               Yes       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

SHOW NODE — Displays the status and LAT characteristics of a node.

Syntax

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.
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.

Counter	Description
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.
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.

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:

```
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

SHOW PORT — Displays the status and LAT characteristics of ports on the local node.

Syntax

```
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).

Description

If a port is an application port, the display lists the remote nodename, remote port name, and remote service name that you specified in the SETPORT 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 nodename 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:

-------------------------------------------------------------------------------
```

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)
```

SHOW QUEUE_ENTRY

SHOW QUEUE_ENTRY — Displays information about requests, or entries, queued on the local node.

Syntax

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
```

SHOW SERVICE

SHOW SERVICE — Displays the status and LAT characteristics of LAT services known to the local node.

Syntax

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:

Counter	Description
Remote Counters	
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.
Local Counters	
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.

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

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.

Syntax

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.

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.

Syntax

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.

Chapter 15. Log Manager Control Program (LMCP) Utility

15.1. 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.

To understand the reasons for using LMCP and how to use it safely, see the *VSI OpenVMS System Manager's Manual*.

15.2. 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**.

15.3. 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

Command	Description
EXIT	Exits LMCP
HELP	Gives help on LMCP commands
REPAIR	Changes the state of transactions
SHOW LOG	Displays information about transaction logs

CLOSE LOG

CLOSE LOG — Closes the transaction log and stops the TP_SERVER process. Requires the SYSNAM privilege.

Syntax

```
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

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. If a node already has a transaction log, using the CONVERT LOG command to create a new one can corrupt data. For information about how to use the CONVERT LOG command safely, see the *VSI OpenVMS System Manager's Manual*. The CONVERT LOG command requires the CMKRNL privilege, read access to the existing transaction log and the directory it is in, and read and write access to the directory in which the new transaction log is to be created.

Syntax

```
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.

Qualifiers

/OWNER=uic

Specifies the owner of the new transaction log.

Specify the owner using the standard UIC format, as described in the *VSI 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

CREATE LOG — Creates a new transaction log. 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.

Syntax

```
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.

Specify the owner using the standard UIC format, as described in the *VSI 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

DUMP — Displays the contents of a transaction log. Requires read access to the transaction log and the directory it is in.

Syntax

```
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.

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 15.1, "Sample Transaction Log "* is a sample of a transaction log, with the important fields identified.

Example 15.1. 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)

Record number 1 (00000001),❸ 114 (0072) bytes ❹
Transaction state (1):  PREPARED ❺
Transaction ID: 1D017140-2676-11C9-9F34-08002B174360 ❻
```

```
(8-JUL-2002 14:08:29.14)
DECdtm Services Log Format V1.1 ❼
Type ( 2): CHILD ❽          Log ID: F1469720-4A0C-11CC-8001-AA000400B7A5 ❾
Name (13): "SYSTEM$WESTRN" (4E 52545345 57244D45 54535953) ❿
Type ( 8): CHILD NODE ❽     Log ID: F1469720-4A0C-11CC-8001-AA000400B7A5 ❾
Name (6): "WESTRN" (4E52 54534557) ❿
Type ( 3): LOCAL RM ❽       Log ID: 037100C0-0019-0003-0100-000000000000 ❾
Name (6): "ORANGE" (4547 4E41524F) ❿
```

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.

❽ 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.

❾ 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.

❿ 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 Journalling for OpenVMS is participating.

```
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
```

EXIT

EXIT — Exits LMCP.

Syntax

EXIT

HELP

HELP — Provides help on LMCP commands.

Syntax

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

REPAIR — Changes the state of transactions. 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 and read and write access to the transaction log and the directory it is in.

Syntax

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:

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

SHOW LOG — Displays information about transaction logs. Requires read access to the transaction logs and the directories they are in.

Syntax

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.

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:        000000F1O5FC 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.
```

Appendix A. 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, "Customizing the ACL Editor"* for more information). To get help on the ACL editor keypad commands, press PF2.

A.1. ACL Editor Keypad Commands

Figure A.1, "Keypad for an LK201-Series Keyboard" 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, "Keypad for an LK201-Series Keyboard"*; VT100 terminals, however, do not have the supplemental editing keypad (keys E1 through E6).

Figure A.1. Keypad for an LK201-Series Keyboard

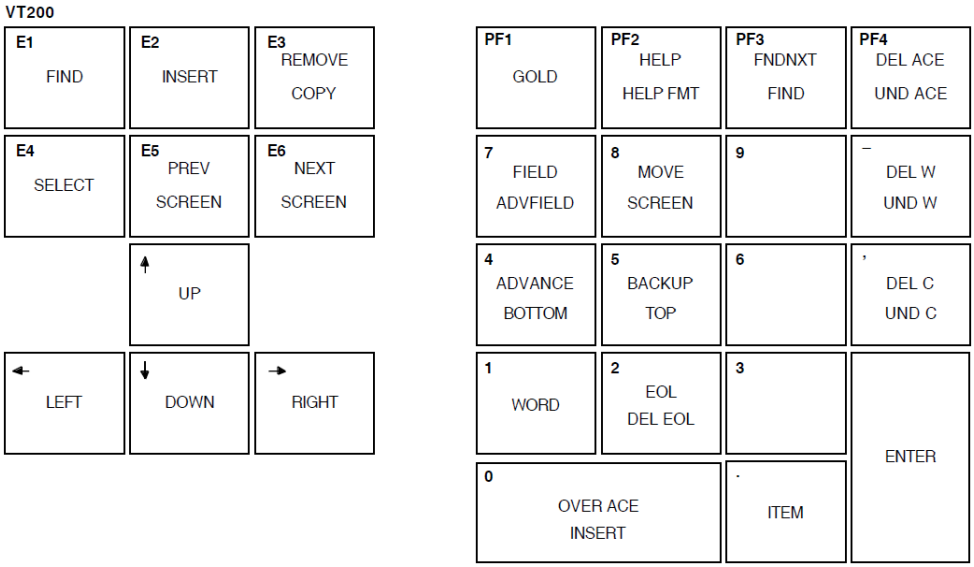


Table A.1, "ACL Editor Keypad Commands" describes each of the keypad commands you can use with the ACL editor. In this table, KP *n* refers to a keypad key labeled with the number *n*. For example, KP4 refers to the keypad key labeled with the number 4.

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

Command	Key or Key Sequence	Description
		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, MOVESCREEN, 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.
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

Command	Key or Key Sequence	Description
		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).

Command	Key or Key Sequence	Description
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).

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, "Additional ACL Editing Keys and Key Sequences"* 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 inline 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 inline 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.

Key or Sequence	Action Taken When Key or Sequence Is Pressed
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.
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, "Customizing the ACL Editor"*).

Table A.3, "ACL Editing Keys on the Supplemental Keypad" 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.
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.

Appendix B. 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, "ACL Section File Variables" lists the ACL section file variables and their defaults.

Table B.1. ACL Section File Variables

Variable	Meaning	
ACLEDIT $X_CHECK_DUPLICATES	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.
ACLEDIT $X_CHECK_MODIFY	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.
ACLEDIT $X_DIRECTORY_FILE	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.
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.

Variable	Meaning	
	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, "ACL Section File Variables"*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. For more information about creating your own CALL_USER routine, see the description of the CALL_USER routine in the *DEC Text Processing Utility Reference Manual*.

Table B.2, "CALL_USER Function Codes" 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.”
18153478	ACLEDIT$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 ACLEDIT $C_CHECK_DUP.
18153479	ACLEDIT$C_CHECK_DUP	Parses the input string (ACE) and returns the string “PARSE_ERROR” if

Function Code	Mnemonic	Description
		an error was encountered. Otherwise, the parsed (binary) ACE is compared with the candidate ACE set by the CALL_USER function ACLEDIT $C_SET_CANDIDATE. Returns the string “DUPLICATE_ACE” if the ACE is a duplicate, or “UNIQUE_ACE” if it is not a duplicate.
18153482	ACLEDIT$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.

Appendix C. Accounting Information for Programmers

Table C.1, "Summary of Accounting System Services" 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, "Format of an Accounting Record" illustrates the general format of an accounting record. *Table C.2, "Fields in an Accounting Record Header"* describes the fields in the record header. The type field in the record header is subdivided into five fields, described in *Table C.3, "ACR$W_TYPE Fields in an Accounting Record Header"*.

Figure C.1. Format of an Accounting Record

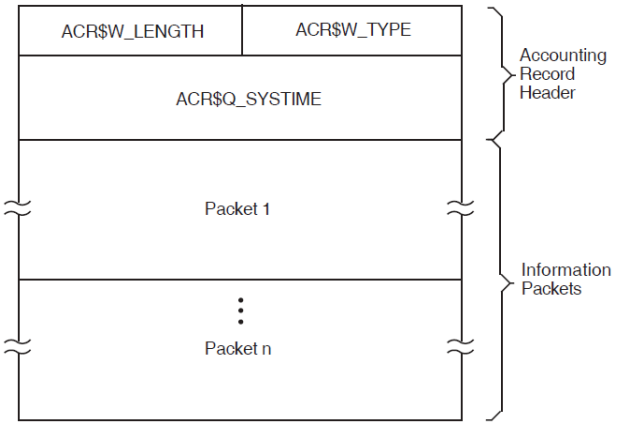


Table C.2. Fields in an Accounting Record Header

Symbolic Offset	Description
ACR$W_TYPE	Identifies the type of the record. This field is subdivided into five fields, described in *Table C.3, "ACR$W_TYPE Fields in an Accounting Record Header"*. (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, "Types of Accounting Record"*. (7 bits)	
ACR$V_SUBTYPE	Identifies the type of process with which the record is associated. The subtypes (4 bits) are:	
	Symbol	**Meaning**
	ACR$K_BATCH	Batch process
	ACR$K_DETACHED	Detached process
	ACR$K_INTERACTIVE	Interactive process
	ACR$K_NETWORK	Network process
	ACR$K_SUBPROCESS	Subprocess
	Note that this field is only meaningful for records of type ACR $K_IMGDEL and ACR$K_PRCDEL.	
ACR$V_VERSION	Identifies the version of the accounting file record structure. The versions (3 bits) are:	
	Symbol	Meaning
	ACR$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$K_VERSION4	OpenVMS V8.4 Alpha and Integrity server Version 4.0 of accounting file record structure.
ACR$V_CUSTOMER	Identifies whether the record was written by VSI software or by customer software. If this bit is 0, the record was written by VSI software. If this bit is 1, the record was written by customer software. (1 bit)	

Note

ACR$K_CURVER = Current version. Set equal to ACR$K_VERSION4 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, "Types of Accounting Record"*. 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
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, "General Format" describes the general format of an information packet. *Section C.1.2.2, "File Name Packet (ACR$K_FILENAME)"* to *Section C.1.2.7, "User Data Packet (ACR $K_USER_DATA)"* 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, "Format of an Information Packet"* 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, "File Name Packet (ACR$K_FILENAME)"* to *Section C.1.2.7, "User Data Packet (ACR$K_USER_DATA)"* 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, "Fields in an Information Packet Header"* and *Table C.6, "ACR$W_TYPE Fields in an Information Packet Header"*).

Figure C.2. Format of an Information Packet

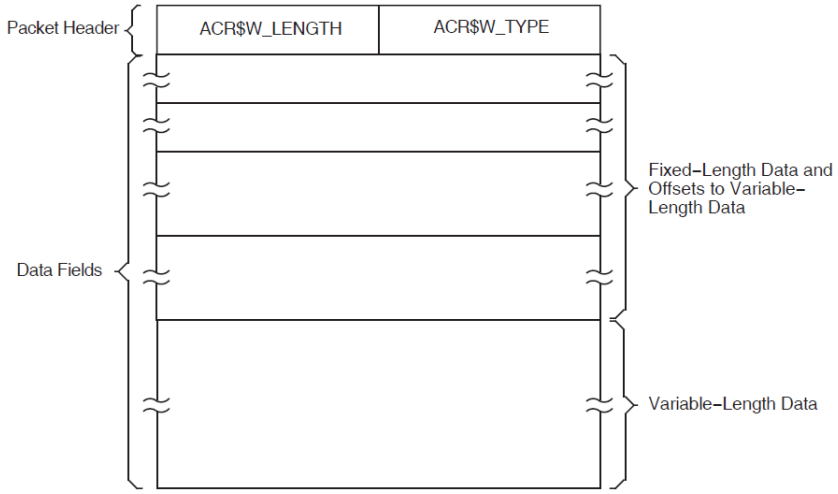


Table C.5. Fields in an Information Packet Header

Symbolic Offset	Description
ACR$W_TYPE	Identifies the type of the packet. This field is subdivided into five fields, described in *Table C.6, "ACR$W_TYPE Fields in an Information Packet Header"*. (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:

Symbolic Offset	Description	
	Symbol	**Description**
	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
ACR$V_SUBTYPE	Identifies the packet subtype; reserved for future use. (4 bits)	
ACR$V_VERSION	See *Table C.3, "ACR$W_TYPE Fields in an Accounting Record Header"*.	
ACR$V_CUSTOMER	See *Table C.3, "ACR$W_TYPE Fields in an Accounting Record Header"*.	

C.1.2.2. File Name Packet (ACR$K_FILENAME)

The file name packet contains the name of the accounting file. *Figure C.3, "Format of a File Name Packet"* shows the format of the file name packet. *Table C.7, "Data Fields in a File Name Packet"* describes the field contained in the packet. See *Section C.1.2.1, "General Format"* for information on the packet header.

Figure C.3. Format of a File Name Packet

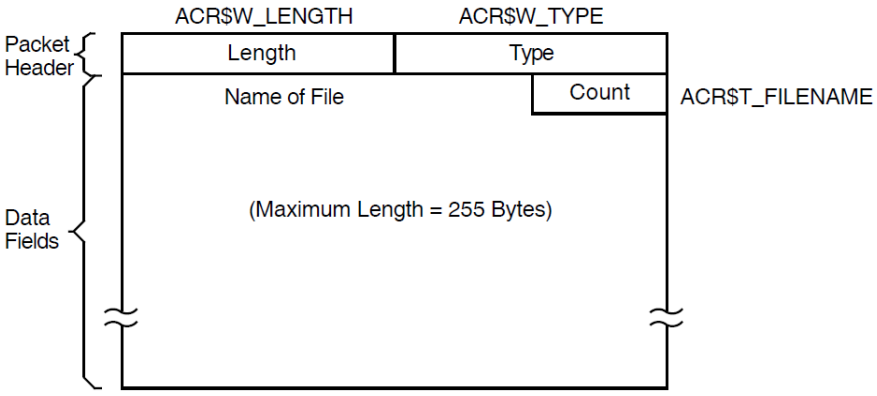


Table C.7. Data Fields in a File Name Packet

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, "Format of an Identification Packet" shows the format of the identification packet. *Table C.8, "Data Fields in an Identification Packet"* describes the fields contained in the packet. See *Section C.1.2.1, "General Format"* for information on the packet header.

Figure C.4. Format of an Identification Packet

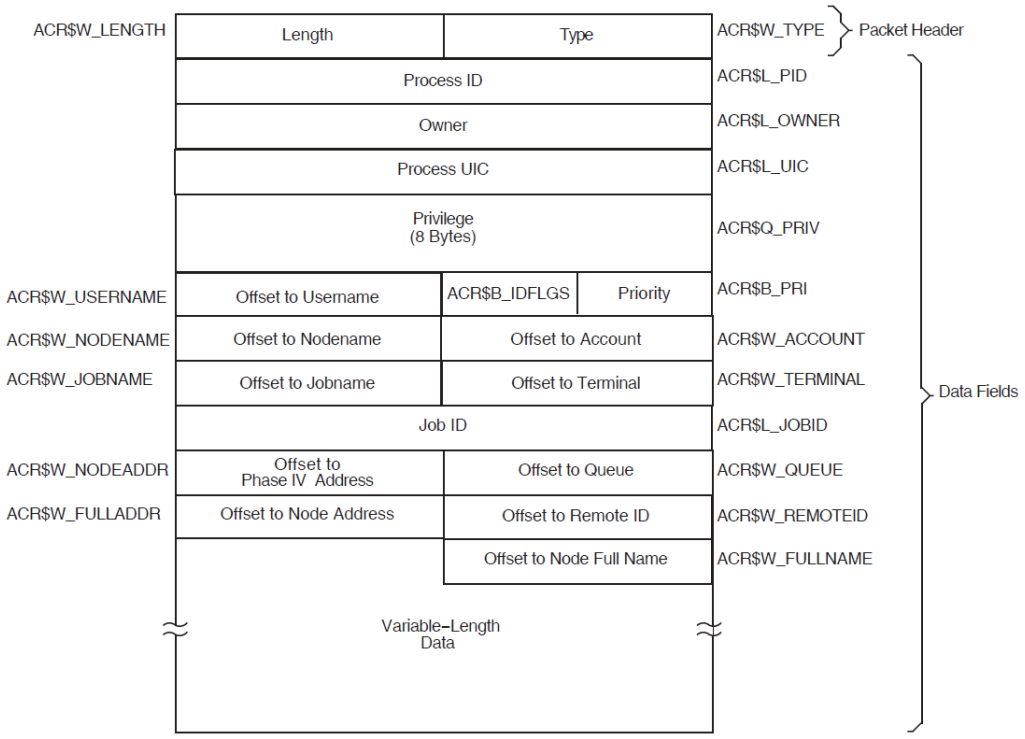


Table C.8. Data Fields in an Identification Packet

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)
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).Starting from OpenVMS Version 8.4 this symbolic offset is changed to a format where the first four bytes that store the terminal name is an ASCII string and the remaining four bytes of (unit number) data is converted to an integer and stored in integer format. This change is to support large device names (more than 8

Symbolic Offset	Description
	characters) in accounting records. For example, in the following new format the TNA55 terminal name is stored as TNA37.
	Integer Part | **ASCII part**
	00 | 00 | 00 | 37 | (41)A | (4E)N | (54)T | Count
	Note: The ASCII value (3A) of colon (:) is not stored in the new format. This should be taken care while processing the records.
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, "Format of an Image Name Packet" shows the format of the image name packet. *Table C.9, "Data Field in an Image Name Packet"* describes the field contained in the packet. See *Section C.1.2.1, "General Format"* for information on the packet header.

Figure C.5. Format of an Image Name Packet

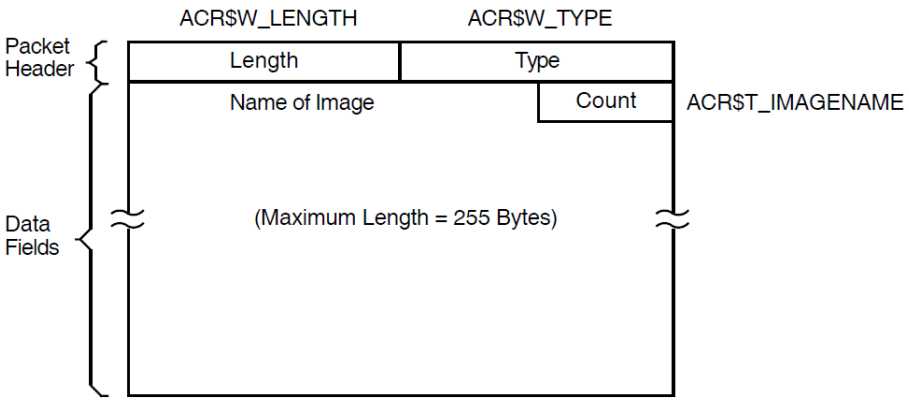


Table C.9. Data Field in an Image Name Packet

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, "Format of a Print Resource Packet" shows the format of the print resource packet. *Table C.10, "Data Fields in a Print Resource Packet"* describes the fields contained in the packet. See *Section C.1.2.1, "General Format"* for information on the packet header.

Figure C.6. Format of a Print Resource Packet

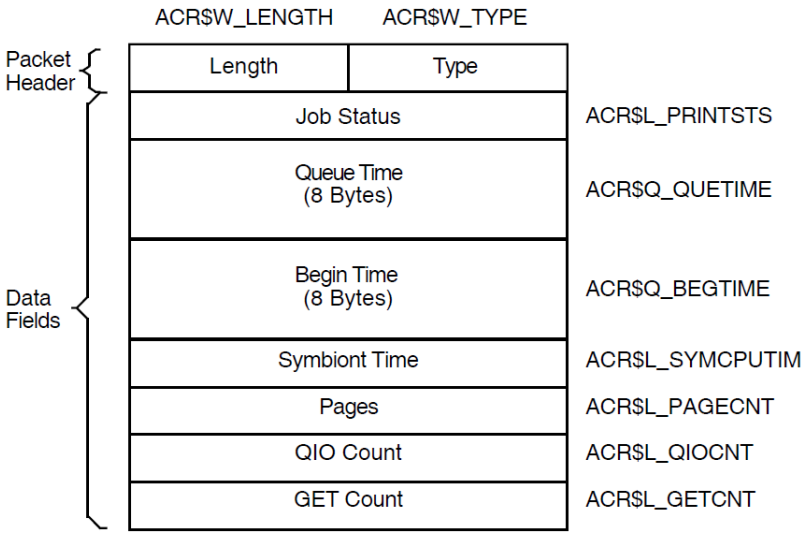


Table C.10. Data Fields in a Print Resource Packet

Symbolic Offset	Description
ACR$L_PRINTSTS	Status of the print job. (longword)
ACR$Q_QUETIME	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, "Format of a Resource Packet" shows the format of a resource packet. *Table C.11, "Data Fields in a Resource Packet"* describes the fields contained in the packet. See *Section C.1.2.1, "General Format"* for information on the packet header.

Figure C.7. Format of a Resource Packet

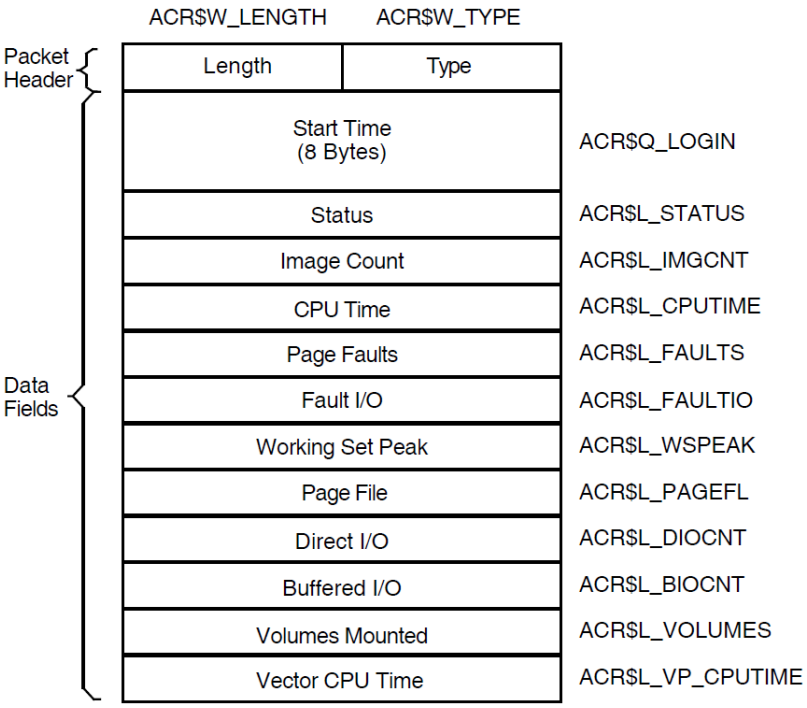


Table C.11. Data Fields in a Resource Packet

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, "Format of a User Data Packet" shows the format of the user data packet. *Table C.12, "Data Field in a User Data Packet"* describes the fields contained in the packet. See *Section C.1.2.1, "General Format"* for information on the packet header.

Figure C.8. Format of a User Data Packet

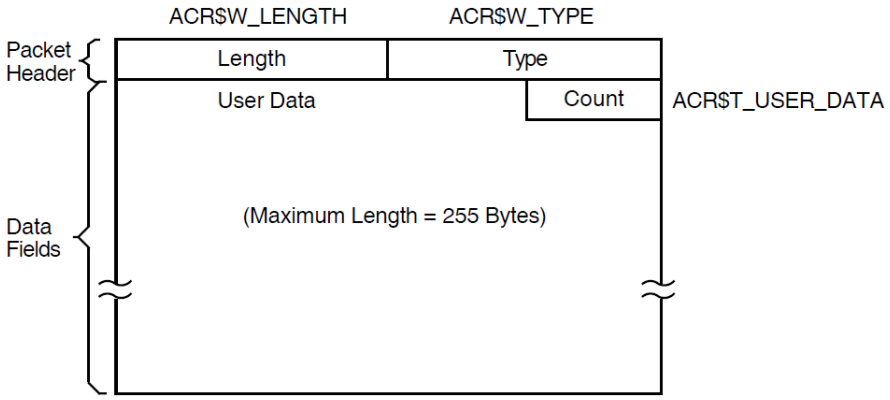


Table C.12. Data Field in a User Data Packet

Symbolic Offset	Description
ACR$T_USER_DATA	Up to 255 bytes of data (counted string).

Appendix D. 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.

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, "Stage 3 Maps"* briefly describes each map built in Stage 3.

Table D.1. Stage 3 Maps

Bitmap	Function
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

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
        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 ❷
        LBN 442398 to 445538, RVN 1
%VERIFY-I-BADHEADER, file (487,0,1) MAIL$0004008EEAEE0572.MAI;1  ❸
        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
        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
%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 ❹
%VERIFY-I-BADDIRENT, invalid file identification in directory entry
 [BLAIN.BOOTS]LOADER.OBJ;1
%VERIFY-I-BADDIRENT, invalid file identification in directory entry
 [BLAIN.BOOTS]SYSGEN.OBJ;1
```

```
%VERIFY-I-BADDIRENT, invalid file identification in directory entry
 [BLAIN]MAIL_20600841.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.TMPSRC]BACKDEF.SDL;1
%VERIFY-I-BACKLINK, incorrect directory back link [CLABIN.BACKUP.TMPSRC]COMMON.REQ;1
%VERIFY-I-BACKLINK, incorrect directory back link [CLABIN.BACKUP.TMPSRC]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 [SYSLOST.BOOTS] has invalid format
%VERIFY-I-BADDIRENT, invalid file identification in directory entry
 [THOEN]NETSERVER.LOG;374
%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 ❺
%VERIFY-I-INCQUOTA, QUOTA.SYS indicates 69663 blocks used, actual use is 69740 blocks for
 [11,402] ❻
%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]
```

❶ 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.

❷ 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.

❸ 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.

❹ 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.

❺ 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].

❻ 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.

Appendix E. 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, "Identification Record Format (Length USG$K_IDENT_LEN)"* 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, "File Record Format (Length USG$K_FILE_LEN)"* contains a description of all the fields in these records.

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.

Field	Meaning
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.

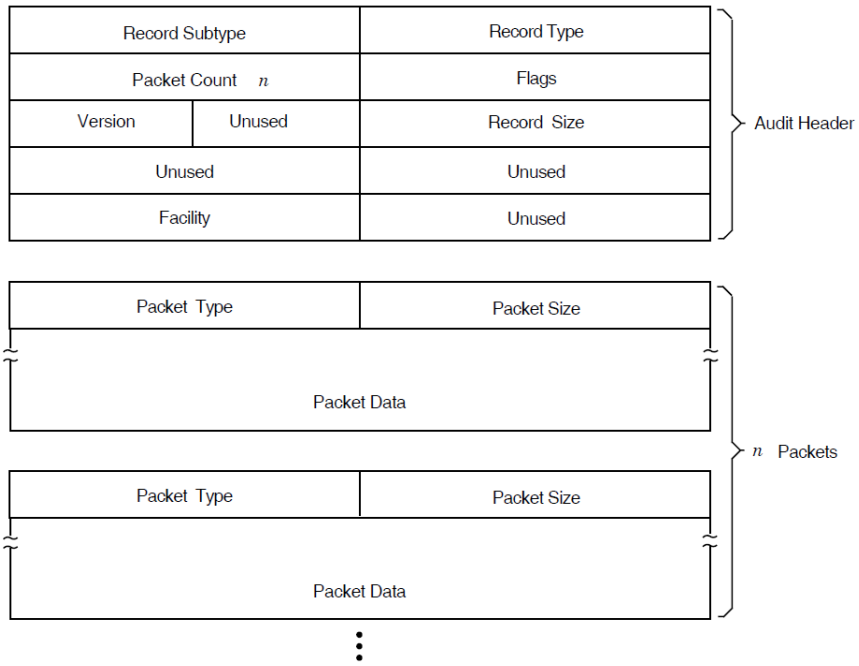
The symbolic names referenced in both the identification and the file summary records are defined in the system definition macro $USGDEF. The length of the identification record is USG $K_IDENT_LEN. The length of a file summary record is USG$K_FILE_LEN.

Appendix F. 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, "Format of a Security Audit Message"*. 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



F.1. Audit Header Packet

*Table F.1, "Description of the Audit Header Fields"*describes the fields contained in *Figure F.2, "Audit Header Packet Format"*.

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

Table F.1. Description of the Audit Header Fields

Field	Symbolic Offset	Contents
Type	NSA$W_RECORD_TYPE	Indicates the type of event that has occurred. See *Table F.2, "Description of Audit Event Types and Subtypes"* for details.
Subtype	NSA$W_RECORD_SUBTYPE	Further defines the type of event that has occurred. See *Table F.2, "Description of Audit Event Types and Subtypes"* for details.
Flags	NSA$W_FLAGS	Identifies any flags associated with the audited event. See *Table F.3, "Description of Audit Event Flags"* for details. Reserved to VSI. (Word)
Packet count	NSA$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, "Description of Audit Event Types and Subtypes"*.

Symbols representing the types or subtypes of security events are listed in *Table F.2, "Description of Audit Event Types and Subtypes"*. 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

Symbols for Event Types and Subtypes		Meaning
	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[1]	System snapshot attempt has aborted
	SNAPSHOT_ACCESS [1]	Snapshot file access/deaccess
	SNAPSHOT_SAVE [1]	System snapshot save in progress
	SNAPSHOT_STARTUP [1]	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

Symbols for Event Types and Subtypes		Meaning
	CNX_IPC_CLOSE	Interprocess communication association closed
	CNX_IPC_OPEN	Interprocess communication association opened
	CNX_REJECT	Connection rejected
	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

Symbols for Event Types and Subtypes		Meaning
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
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

Symbols for Event Types and Subtypes		Meaning
	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
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

[1]Obsolete as of OpenVMS Version 7.1

Table F.3, "Description of Audit Event Flags" 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.

Symbol	Meaning
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 VSI.

F.2. Audit Data Packets

Figure F.3, "Audit Data Packet Format" 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, see the appendix of alarm messages in the *VSI OpenVMS Guide to System Security*.

Figure F.3. Audit Data Packet Format

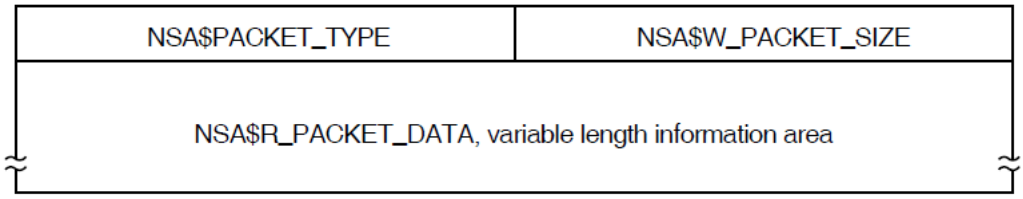


Table F.4, "Description of the Audit Data Packet" describes the fields contained in these packets.

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, "Types of Data in Audit Packets"*.
Packet data	NSA$R_PACKET_DATA	Variable length field containing the packet data.

Table F.5, "Types of Data in Audit Packets" 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)

Symbol	Packet Contents
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 VSI. (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$_DEFAULT_USERNAME	Default local user name for incoming network proxy requests (String of 1-32 characters)
NSA$_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)

Symbol	Packet Contents
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)
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)

Symbol	Packet Contents
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$_OLD_PRIVILEGES	Former privileges (Quadword)
NSA$_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)

Symbol	Packet Contents
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

Symbol	Packet Contents
	time that the system was saved (String of 1-255 characters)
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$_SYSTEM_SERVICE_NAME	Name of the system service associated with the event (String of 1-256 characters)
NSA$_SYSTIM_NEW	New system time (Quadword)
NSA$_SYSTIM_OLD	Old system time (Quadword)
NSA$_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)

Symbol	Packet Contents
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 VSI. (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)
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)

Appendix G. 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, "Command Qualifiers Used in Save Operations"* shows command qualifiers used in save operations.
- *Figure G.2, "Input File-Selection Qualifiers Used in Save Operations"* shows input file-selection qualifiers used in save operations.
- *Figure G.3, "Output Save-Set Qualifiers Used in Save Operations"* shows output save-set qualifiers used in save operations.
- *Figure G.4, "Command Qualifiers Used in Restore Operations"* shows command qualifiers used in restore operations.
- *Figure G.5, "Input Save-Set Qualifiers Used in Restore Operations"* shows input save-set qualifiers used in restore operations.
- *Figure G.6, "Output File Qualifiers Used in Restore Operations"* shows output file qualifiers used in restore operations.
- *Figure G.7, "Command Qualifiers Used in Copy Operations"* shows command qualifiers used in copy operations.
- *Figure G.8, "Input File-Selection Qualifiers Used in Copy Operations"* shows input file-selection qualifiers used in copy operations.
- *Figure G.9, "Output File Qualifiers Used in Copy Operations"* shows output file qualifiers used in copy operations.
- *Figure G.10, "Command Qualifiers Used in Compare Operations"* shows command qualifiers used in compare operations.
- *Figure G.11, "Input File-Selection Qualifiers Used in Compare Operations"* shows input file-selection qualifiers used in compare operations.
- *Figure G.12, "Input Save-Set Qualifiers Used in Compare Operations"* shows input save-set qualifiers used in compare operations.

Figure G.1. Command Qualifiers Used in Save Operations

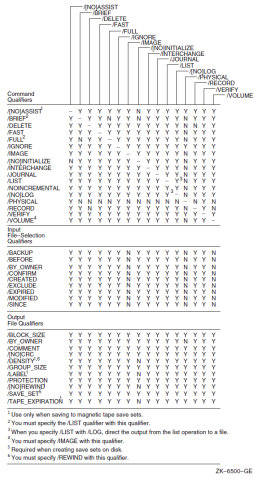


Figure G.2. Input File-Selection Qualifiers Used in Save Operations

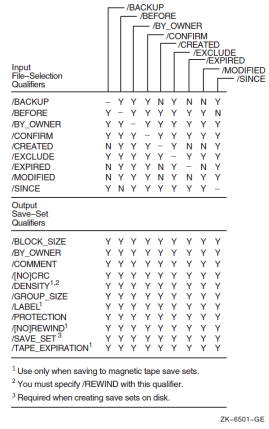


Figure G.3. Output Save-Set Qualifiers Used in Save Operations

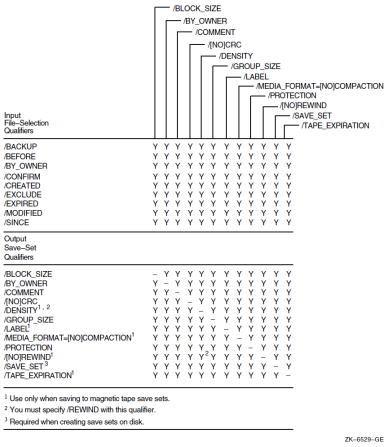
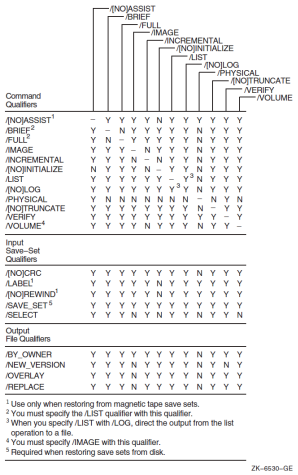


Figure G.4. Command Qualifiers Used in Restore Operations



Command Qualifiers	/[NO]ASSIST	/BRIEF	/FULL	/IMAGE	/INCREMENTAL	/[NO]INITIALIZE	/LIST	/[NO]LOG	/PHYSICAL	/[NO]TRUNCATE	/VERIFY	/VOLUME
/[NO]ASSIST[1]	–	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
/BRIEF[2]	Y	–	N	Y	Y	Y	Y	Y	N	Y	Y	Y
/FULL[2]	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[3]	N	Y	Y	Y
/[NO]LOG	Y	Y	Y	Y	Y	Y	Y[3]	–	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[4]	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	–
Input Save–Set Qualifiers												
/[NO]CRC	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
/LABEL[1]	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
/[NO]REWIND[1]	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
/SAVE_SET[5]	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												
/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

[1] Use only when restoring from magnetic tape save sets.
[2] You must specify the /LIST qualifier with this qualifier.
[3] When you specify /LIST with /LOG, direct the output from the list operation to a file.
[4] You must specify /IMAGE with this qualifier.
[5] Required when restoring save sets from disk.

ZK–6530–GE

Figure G.5. Input Save-Set Qualifiers Used in Restore Operations

Input Save Set Qualifiers	/[NO]CRC	/[NO]REWIND	/SAVE_SET	/SELECT
/[NO]CRC	–	Y	Y	Y
/[NO]REWIND[1]	Y	–	Y	Y
/SAVE_SET[2]	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

[1] Use only when restoring from magnetic tape save sets.
[2] Required when restoring save sets from disk.

VM-0949A-AI

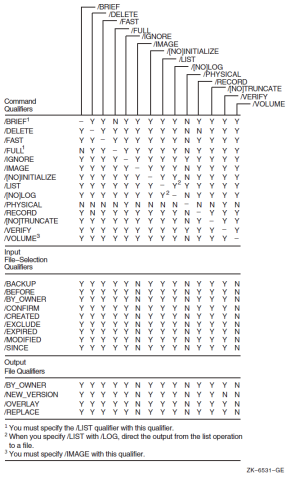
Figure G.6. Output File Qualifiers Used in Restore Operations

Input Save–Set Qualifiers	/BY_OWNER	/NEW_VERSION	/OVERLAY	/REPLACE
/[NO]CRC	Y	Y	Y	Y
/[NO]REWIND[1]	Y	Y	Y	Y
/SAVE_SET[2]	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	–

[1] Use only when restoring from magnetic tape save sets.
[2] Required when restoring save sets from disk.

ZK–0837A–GE

Figure G.7. Command Qualifiers Used in Copy Operations

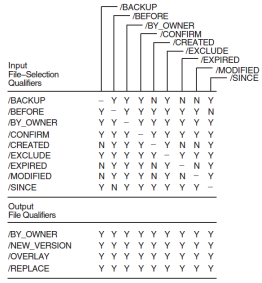


Command Qualifiers	/BRIEF	/DELETE	/FAST	/FULL	/IGNORE	/IMAGE	/[NO]INITIALIZE	/LIST	/[NO]LOG	/PHYSICAL	/RECORD	/[NO]TRUNCATE	/VERIFY	/VOLUME
/BRIEF[1]	–	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	Y	Y
/DELETE	Y	–	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y	Y
/FAST	Y	Y	–	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y
/FULL[1]	N	Y	Y	–	Y	Y	Y	Y	Y	N	Y	Y	Y	Y
/IGNORE	Y	Y	Y	Y	–	Y	Y	Y	Y	Y	Y	Y	Y	Y
/IMAGE	Y	Y	Y	Y	Y	–	Y	Y	Y	N	Y	Y	Y	Y
/[NO]INITIALIZE	Y	Y	Y	Y	Y	Y	–	Y	Y	N	Y	Y	Y	Y
/LIST	Y	Y	Y	Y	Y	Y	Y	–	Y[2]	Y	Y	Y	Y	Y
/[NO]LOG	Y	Y	Y	Y	Y	Y	Y	Y[2]	–	N	Y	Y	Y	Y
/PHYSICAL	N	N	N	N	Y	N	N	N	N	–	N	N	Y	N
/RECORD	Y	N	Y	Y	Y	Y	Y	Y	Y	N	–	Y	Y	Y
/[NO]TRUNCATE	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	–	Y	Y
/VERIFY	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	–	Y
/VOLUME[3]	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	–
Input File–Selection Qualifiers														
/BACKUP	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	N
/BEFORE	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	N
/BY_OWNER	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	N
/CONFIRM	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	N
/CREATED	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	N
/EXCLUDE	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	N
/EXPIRED	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	N
/MODIFIED	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	N
/SINCE	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	N
Output File Qualifiers														
/BY_OWNER	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	N
/NEW_VERSION	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	N	N
/OVERLAY	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	N
/REPLACE	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	N

[1] You must specify the /LIST qualifier with this qualifier.
[2] When you specify /LIST with /LOG, direct the output from the list operation to a file.
[3] You must specify /IMAGE with this qualifier.

ZK–6531–GE

Figure G.8. Input File-Selection Qualifiers Used in Copy Operations

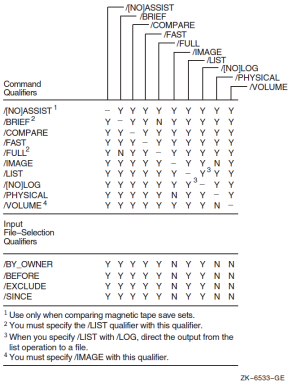


Input File–Selection Qualifiers	/BACKUP	/BEFORE	/BY_OWNER	/CONFIRM	/CREATED	/EXCLUDE	/EXPIRED	/MODIFIED	/SINCE
/BACKUP	–	Y	Y	N	Y	N	N	Y	
/BEFORE	Y	–	Y	Y	Y	Y	Y	N	
/BY_OWNER	Y	Y	–	Y	Y	Y	Y	Y	
/CONFIRM	Y	Y	Y	–	Y	Y	Y	Y	
/CREATED	N	Y	Y	Y	–	Y	N	N	
/EXCLUDE	Y	Y	Y	Y	Y	–	Y	Y	
/EXPIRED	N	Y	Y	Y	N	Y	–	N	
/MODIFIED	N	Y	Y	Y	N	Y	N	–	
/SINCE	Y	N	Y	Y	Y	Y	Y	Y	–
Output File Qualifiers									
/BY_OWNER	Y	Y	Y	Y	Y	Y	Y	Y	
/NEW_VERSION	Y	Y	Y	Y	Y	Y	Y	Y	
/OVERLAY	Y	Y	Y	Y	Y	Y	Y	Y	
/REPLACE	Y	Y	Y	Y	Y	Y	Y	Y	

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
Output File Qualifiers				
/BY_OWNER	–	Y	Y	Y
/NEW_VERSION	Y	–	N	N
/OVERLAY	Y	N	–	N
/REPLACE	Y	N	N	–

Figure G.10. Command Qualifiers Used in Compare Operations



Command Qualifiers	/[NO]ASSIST [1]	/BRIEF [2]	/COMPARE	/FAST	/FULL [2]	/IMAGE	/LIST	/[NO]LOG	/PHYSICAL	/VOLUME [4]
/[NO]ASSIST [1]	–	Y	Y	Y	Y	Y	Y	Y	Y	Y
/BRIEF [2]	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 [2]	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 [3]	Y	Y
/[NO]LOG	Y	Y	Y	Y	Y	Y	Y	–	Y	Y
/PHYSICAL	Y	Y	Y	Y	Y	N	Y	Y	–	Y
/VOLUME [4]	Y	Y	Y	Y	Y	Y	Y	Y	N	–
Input File–Selection Qualifiers										
/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

[1] Use only when comparing magnetic tape save sets.
[2] You must specify the /LIST qualifier with this qualifier.
[3] When you specify /LIST with /LOG, direct the output from the list operation to a file.
[4] You must specify /IMAGE with this qualifier.

ZK–6533–GE

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				
/[NO]REWIND [1]	Y	Y	Y	Y
/SAVE_SET [2]	Y	Y	Y	Y
/SELECT	Y	Y	Y	Y

[1] Use only when comparing magnetic tape save sets.
[2] 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 [1]	–	Y	Y
/SAVE_SET [2]	Y	–	Y
/SELECT	Y	Y	–

[1] Use only when comparing magnetic tape save sets.
[2] Required when comparing save sets on disk.

ZK–0947A–GE