VMS Software

DCL Dictionary: N–Z

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

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

DCL Dictionary: N–Z

Copyright © 2025 VMS Software, Inc. (VSI), Boston, Massachusetts, USA

Legal Notice

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

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

All other trademarks and registered trademarks mentioned in this document are the property of their respective holders.

Table of Contents

Preface vii
1. About VSI vii
2. Intended Audience vii
3. Documents Structure vii
4. Related Documents vii
5. VSI Encourages Your Comments viii
6. OpenVMS Documentation viii
7. Conventions viii
DCL Commands 1
NCS 1
ON 1
OPEN 4
PASSWORD 7
PATCH 8
PHONE 12
PIPE 13
PPPD 21
PRINT 22
PRODUCT 35
PURGE 36
READ 40
RECALL 43
RENAME 47
REPLY 51
REQUEST 60
RETURN 62
RUN (Image) 63
RUN (Process) 65
RUNOFF 75
SEARCH 76
SET 85
SET ACCOUNTING 88
SET AUDIT 91
SET BOOTBLOCK (IA-64 Only) 103
SET BROADCAST 104
SET CACHE/RESET 105
SET CARD_READER 106
SET CLUSTER/EXPECTED_VOTES 107
SET COMMAND 109
SET CONTROL 112
SET CPU 114
SET DAY 120
SET DEFAULT 121
SET DEVICE 124
SET DEVICE/SERVED 131
SET DIRECTORY 132
SET DISPLAY 136
SET DUMP_OPTIONS (x86-64 Only) 148
SET ENTRY 149
SET FILE 159

SET HOST 170
SET HOST/DTE 173
SET HOST/DUP 180
SET HOST/HSC 181
SET HOST/LAT 182
SET HOST/RLOGIN 186
SET HOST/TELNET 188
SET HOST/TN3270 189
SET IMAGE 190
SET KEY 194
SET LOGINS 194
SET MAGTAPE 195
SET MESSAGE 199
SET NETWORK 201
SET ON 204
SET OUTPUT_RATE 205
SET PASSWORD 205
SET PREFERRED_PATH 209
SET PREFIX 211
SET PRINTER 212
SET PROCESS 215
SET PROMPT 228
SET PROTECTION/DEFAULT 228
SET QUEUE 229
SET RESTART_VALUE 237
SET RIGHTS_LIST 239
SET RMS_DEFAULT 241
SET ROOT 247
SET SECURITY 247
SET SERVER ACME_SERVER 256
SET SERVER REGISTRY_SERVER (Alpha/IA-64 only) 262
SET SERVER SECURITY_SERVER 265
SET SHADOW 266
SET SYMBOL 277
SET TERMINAL 280
SET TIME 293
SET VERIFY 295
SET VOLUME 297
SET WORKING_SET 303
SHOW 305
SHOW ACCOUNTING 307
SHOW AUDIT 308
SHOW BROADCAST 312
SHOW CLUSTER 314
SHOW CPU 314
SHOW DEFAULT 323
SHOW DEVICES 324
SHOW DEVICES/SERVED 335
SHOW DISPLAY 340
SHOW DUMP_OPTIONS (x86-64 Only) 343
SHOW ENTROPY (x86-64 Only) 344
SHOW ENTRY 345

SHOW ERROR 350
SHOW FASTPATH (Alpha/IA-64 only) 353
SHOW IMAGE 354
SHOW INTRUSION 357
SHOW KEY 359
SHOW LICENSE 361
SHOW LOGICAL 372
SHOW MEMORY 377
SHOW NETWORK 394
SHOW PRINTER 398
SHOW PROCESS 401
SHOW PROTECTION 410
SHOW QUEUE 411
SHOW QUEUE/CHARACTERISTICS 417
SHOW QUEUE/FORM 418
SHOW QUEUE/MANAGERS 420
SHOW QUOTA 422
SHOW RMS_DEFAULT 423
SHOW ROOT 424
SHOW SECURITY 424
SHOW SERVER ACME_SERVER 428
SHOW SERVER REGISTRY_SERVER (Alpha/IA-64 only) 431
SHOW SHADOW 432
SHOW STATUS 440
SHOW SYMBOL 441
SHOW SYSTEM 443
SHOW TERMINAL 451
SHOW TIME 454
SHOW TRANSLATION 454
SHOW USERS 456
SHOW WORKING_SET 460
SORT 461
SPAWN 461
START/CPU 467
START/NETWORK 469
START/QUEUE 469
START/QUEUE/MANAGER 481
STOP 484
STOP/CPU 487
STOP/NETWORK 489
STOP/QUEUE 490
STOP/QUEUE/ABORT 491
STOP/QUEUE/ENTRY 492
STOP/QUEUE/MANAGER/CLUSTER 493
STOP/QUEUE/NEXT 494
STOP/QUEUE/REQUEUE 495
STOP/QUEUE/RESET 497
STOP/QUEUES/ON_NODE 498
SUBMIT 500
SUBROUTINE 512
SYNCHRONIZE 513
TYPE 514

VIEW 522
WAIT 523
WRITE 524
XAUTH (Alpha/IA-64 only) 526

Appendix A. Obsolete Commands 529

Appendix B. Handling Numeric Values Outside the Range of DCL Integer Representation 531

B.1. Preface 531
B.2. Introduction 531
B.2.1. Problem Statement 531
B.2.2. Prerequisites 532
B.3. DCL Examples 532
B.3.1. DCL code checking a value in range 0 to 2147483647 532
B.3.2. DCL code checking a value in range 0 to 4294967295 532
B.3.3. Checking the percentage of free blocks on a disk 533
B.3.4. F$CUNITS() lexical function description 534
B.4. Identifying Suspect DCL Code 537
B.4.1. DCL Code Search Example 537

Preface

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 all users of the VSI OpenVMS operating system. It includes descriptions of all DIGITAL Command Language (DCL) commands and lexical functions. If a command has any restrictions or requires special privileges, they are noted in reference information for that command.

Readers of this manual should be familiar with the material covered in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/].

3. Documents Structure

This manual contains detailed descriptions of each command and lexical function. The commands are listed in alphabetical order. The lexical functions are grouped under the heading Lexical Functions and are listed alphabetically within that grouping.

The hardcopy version of the *VSI OpenVMS DCL Dictionary* is a two-part manual. The first volume contains commands beginning with the letters A to M (including the lexical functions); the second volume contains commands beginning with the letters N to Z.

Appendix A of this manual (in the second volume of the hardcopy manual) lists the obsolete DCL commands and the current services that replace them.

The commands that invoke language compilers and other OpenVMS optional software products are not included in this manual; they are included in the documentation provided with those products.

4. Related Documents

For an introduction to the OpenVMS operating system and for information on using DCL, refer to the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/]. This manual is especially recommended for novice users or users lacking experience with interactive computer systems.

The *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/] provides an overview of DCL command language concepts and defines and illustrates good practices in constructing command procedures with DCL commands and lexical functions.

Refer to the various utilities reference manuals for detailed information about utilities. These manuals describe the DCL commands that invoke the various utilities, describe any commands that you can enter while running a utility, and provide reference information. The *VSI OpenVMS DCL Dictionary* provides only a brief description and format information for each utility.

For message descriptions, use the online Help Message utility.

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

The following conventions are 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.
PF1 *x*	A sequence such as **PF1 *x*** indicates that you must first press and release the key labeled **PF1** and then press and release 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

Convention	Meaning
	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 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.

DCL Commands

NCS

NCS — Invokes the OpenVMS National Character Set (NCS) utility, which provides a convenient method of implementing alternative (non-ASCII) string collating sequences, typically using subsets of the DEC Multinational character set. NCS also facilitates the implementation of string conversion functions. For more information about the NCS utility, see the *OpenVMS National Character Set Utility Manual* or online help.

Format

NCS [*filespec*[,...]]

ON

ON — Performs a specified action when a command or program executed within a command procedure encounters an error condition or is interrupted by **Ctrl/Y**. The specified actions are performed only if the command interpreter is enabled for error checking or **Ctrl/Y** interrupts (the default conditions). Use the **ON** command only in a command procedure.

Format

ON *condition* THEN [$] *command*

Parameters

condition

Specifies either the severity level of an error or a **Ctrl/Y** interrupt. Specify one of the following keywords, which may be abbreviated to one or more characters:

WARNING	Return status of warning occurs ($SEVERITY equals 0).
ERROR	Return status of error occurs ($SEVERITY equals 2).
SEVERE_ERROR	Return status of error occurs ($SEVERITY equals 4).
CONTROL_Y	**Ctrl/Y** character occurs on SYS$INPUT.

The default error condition is **ON ERROR THEN EXIT**.

command

Specifies the DCL command line to be executed. Optionally, you can precede the command line with a dollar sign ($).

If you specified an error condition as the condition parameter, the action is taken when errors equal to or greater than the specified level of error occur.

Description

During the execution of a command procedure, the command interpreter checks the condition code returned from each command or program that executes. With the **ON** command, you can establish a course of action for the command interpreter to take based on the result of the check.

The system places condition codes in the global symbol $STATUS. The severity of the condition code is represented in the first 3 low-order bits of $STATUS. This severity level is also represented by the global symbol $SEVERITY. See the description of the **EXIT** command in the *VSI OpenVMS DCL Dictionary: A–M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_110] for information on severity level values.

If an **ON** command action specifies the severity level of an error, the command interpreter executes the ON command action for errors at the specified severity level or greater. For example, the following command causes a procedure to exit on warnings, errors, or severe errors:

```
$ ON WARNING THEN EXIT
```

The default action is as follows:

```
$ ON ERROR THEN EXIT
```

That is, the command interpreter continues when a warning occurs, and executes an **EXIT** command when an error or severe error occurs. An **ON** command action that specifies a severity level is executed only once; after the **ON** command action is taken, the default **ON** action is reset. There is an exception to the default **ON** action. If you use the **CALL**, **GOSUB**, or **GOTO** commands and specify a label that does not exist in the current command procedure, the procedure issues a warning message and exits.

The action specified by an **ON** command applies only within the command procedure in which the command is executed; therefore, if you execute an **ON** command in a procedure that calls another procedure, the **ON** command action does not apply to the nested procedure. An **ON** command executed at any command procedure level does not affect the error condition handling of procedures at any other level.

To disable error checking with the **ON** command, use the **SET NOON** command. You can enable error checking with the **SET ON** command, or by entering another **ON** command.

The **ON** command also provides a way to define an action routine for a **Ctrl/Y** interrupt that occurs during execution of a command procedure. The default (**Ctrl/Y**) action is to prompt for command input at the **Ctrl/Y** command level. The **Ctrl/Y** command level is a special command level where you can enter DCL commands. If you enter a command that is executed within the command interpreter, you can resume execution of the procedure with the **CONTINUE** command. (For a list of commands that are executed within the command interpreter, see the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/].)

If you enter any other DCL command, the command interpreter returns to command level 0 and executes the image invoked by the command. If you interrupt the command procedure while it is executing an image that contains an exit handler, the exit handler is allowed to execute before the new command (image) is executed. (However, if you enter the **STOP** command after you interrupt a command procedure by pressing **Ctrl/Y**, exit handlers declared by the interrupted image are not executed.)

You can use the **ON** command to change the default action for a **Ctrl/Y** interrupt. If you change the default **Ctrl/Y** action, the execution of a **Ctrl/Y** interrupt does not automatically reset the default **Ctrl/Y** action. A **Ctrl/Y** action remains in effect until one of the following occurs:

- The procedure terminates (as a result of an **EXIT** or **STOP** command, or as a result of a default error condition handling action).
- Another **ON CONTROL_Y** command is executed.
- The procedure executes the **SET NOCONTROL=Y** command.

A **Ctrl/Y** action can be specified for each active command level; the **Ctrl/Y** action specified for the currently executing command level overrides actions specified for previous levels.

Note

The **ON CONTROL_Y** and **SET NOCONTROL=Y** commands are intended for special applications. VSI does not recommend, in general, that you disable **Ctrl/Y** interrupts. For example, if a procedure that disables **Ctrl/Y** interrupts begins to loop uncontrollably, you cannot gain control to stop the procedure from your terminal.

Examples

1.
```
$ ON SEVERE_ERROR THEN CONTINUE
```

   A command procedure that contains this statement continues to execute normally when a warning or error occurs during execution. When a severe error occurs, the **ON** statement signals the procedure to execute the next statement anyway. Once the statement has been executed as a result of the severe error condition, the default action (**ON ERROR THEN EXIT**) is reinstated.

2.
```
$ ON ERROR THEN GOTO BYPASS
$ RUN A
$ RUN B
   .
   .
   .
$ EXIT
$ BYPASS:
$      RUN C
```

   If either program `A` or program `B` returns a status code with a severity level of error or severe error, control is transferred to the statement labeled `BYPASS` and program `C` is run.

3.
```
$ ON WARNING THEN EXIT
   .
   .
   .
$ SET NOON
$ RUN [SSTEST]LIBRA
$ SET ON
   .
   .
   .
```

   The **ON** command requests that the procedure exit when any warning, error, or severe error occurs. Later, the **SET NOON** command disables error checking before executing the **RUN** command. Regardless of any status code returned by the program `LIBRA.EXE`, the procedure continues. The next command, **SET ON**, re-enables error checking and reestablishes the most recent **ON** condition.

4.
```
$ ON CONTROL_Y THEN GOTO CTRL_EXIT
```

```
   .
   .
   .
$ CTRL_EXIT:
$ CLOSE INFILE
$ CLOSE OUTFILE
$ EXIT
```

The **ON** command specifies action to be taken when **Ctrl/Y** is pressed during the execution of this procedure: the **GOTO** command transfers control to the line labeled CTRL_EXIT. At CTRL_EXIT, the procedure performs cleanup operations (in this example, closing files and exiting).

OPEN

OPEN — Opens a file for reading, writing, or both; assigns a logical name to a file; and places the name in the process logical name table. **See the qualifier descriptions for restrictions.**

Format

OPEN *logical-name*[:] *filespec*

Parameters

***logical-name*[:]**

Specifies the logical name and a character string to be assigned to the file.

filespec

Specifies the name of the file or device being opened for input or output. The file type defaults to DAT. The asterisk (*) and the percent sign (%) wildcard characters are not allowed.

To create a new, sequential file, specify the **/WRITE** qualifier. See the description of the **/WRITE** qualifier for more information.

Description

A file can be opened for either reading or writing, or for both reading and writing. After the file is opened, it is available for input or output at the command level with the **READ** and **WRITE** commands.

The **OPEN** command opens files as process permanent. Therefore, these files remain open until you close them with the **CLOSE** command, or until you log out. If a command procedure that opens a file terminates without closing an open file, the file remains open; the command interpreter does not automatically close it. The **OPEN** command uses OpenVMS RMS to open files, and is subject to RMS restrictions on using process-permanent files. The **OPEN** command opens sequential, relative, or indexed sequential files.

The logical devices SYS$INPUT, SYS$OUTPUT, SYS$COMMAND, and SYS$ERROR do not have to be opened explicitly before they can be read or written at the command level. All other files must be opened explicitly.

Do not use the same logical name when you open different files. If you specify a logical name with the **OPEN** command and the logical name is currently assigned to another file, no warning message is issued;

however, the file is not opened, and the next **READ** request will access the file to which the logical name was originally assigned.

You can enter more than one **OPEN** command for the same file and assign it different logical names if you use the **/SHARE** qualifier the first time the file is opened. Also, if you open the file by using the **/SHARE=READ** or the **/SHARE=WRITE** qualifier, other users can access the file with the **TYPE** or the **SEARCH** command.

When you use the **OPEN** command to create a new file, variable fixed control (VFC) record format is used. Concatenating a file of this record format with a file of another record format might be impossible due to record format incompatibilities. To avoid the VFC format, use the **CREATE** command to create the file.

When the **OPEN** command is specified on an existing file, the record type of that file is used.

Qualifiers

/APPEND

Opens an existing file for writing and positions the record pointer at the end-of-file (EOF). New records are added to the end of the file.

Only sequential files allow more than one user to append records concurrently.

Use the **/APPEND** qualifier only to add records to an existing file. The **/APPEND** and the **/WRITE** qualifiers are mutually exclusive.

/ERROR=*label*

Transfers control to the location specified by the label keyword (in a command procedure) if the open operation results in an error. The error routine specified for this qualifier overrides any **ON** condition action specified. If the **/ERROR** qualifier is not specified, the current **ON** condition action is taken.

If an error occurs and the target label is successfully given control, the global symbol $STATUS retains the code for the error that caused the error path to be taken.

/READ (default)

Opens the file for reading. If you open a file with **/READ**, other users are also allowed read access to the file, but no user is allowed write access. If you open a file with **/READ/WRITE**, no other users are allowed access while the file is open. If you specify the **/READ** qualifier without the **/WRITE** qualifier, you must specify an existing file.

/SHARE[=*option*]
/NOSHARE (Alpha/IA-64 only)

Opens the specified file as a shareable file to allow other users read or write access. If you specify the **/SHARE=READ** qualifier, other users are allowed read (R) access to the file, but not write (W) access. If you specify the **/SHARE=WRITE** or the **/SHARE** qualifier with no option, users are allowed read and write access to the specified file.

To open a file with no shared access, specify **/NOSHARE** or use **OPEN/READ/WRITE**.

/WRITE

Opens the file for writing. The following restrictions apply to the **/WRITE** qualifier:

- Use the **/WRITE** qualifier to open and create a new, sequential file. If the file specification on an **OPEN/WRITE** command does not include a file version number, and if a file with the specified file name and file type already exists, a new file with a version number one greater than the existing file is created.
- Use the **/READ** qualifier with the **/WRITE** qualifier to open an existing file. While the file is open, no other user will have access to it. When the file is first opened, the pointer is positioned to the beginning of the file. (This differs from **OPEN/APPEND**, which positions the pointer at the end of the file.) You cannot use **OPEN/READ/WRITE** to create a new file.
- The **/WRITE** and the **/APPEND** qualifiers are mutually exclusive.

Examples

1.
```
$ OPEN INPUT_FILE AVERAGE.DAT
$ READ_LOOP:
$ READ/END_OF_FILE=ENDIT  INPUT_FILE  NUM
    .
    .
    .
$ GOTO READ_LOOP
$ ENDIT:
$ CLOSE INPUT_FILE
```

   The **OPEN** command opens the file named AVERAGE.DAT as an input file and assigns it the logical name INPUT_FILE. The file is opened with read access because the **/READ** qualifier is present by default. The **READ** command reads a record from the logical file INPUT_FILE into the symbol named NUM. The procedure executes the lines between the labels READ_LOOP and ENDIT until the end of the file is reached. At the end of the file, the **CLOSE** command closes the file.

2.
```
$ OPEN/WRITE/ERROR=OPEN_ERROR  OUTPUT_FILE  TEMP.OUT
$ COUNT = 0
$ WRITE_LOOP:
$ COUNT = COUNT + 1
$ IF COUNT .EQ. 11 THEN GOTO ENDIT
$ WRITE OUTPUT_FILE "Count is ''COUNT'."
    .
    .
    .
$ GOTO WRITE_LOOP
$ ENDIT:
$ CLOSE OUTPUT_FILE
$ EXIT
$
$ OPEN_ERROR:
$ WRITE SYS$OUTPUT "Cannot open file TEMP.OUT"
$ EXIT
```

   The **OPEN** command with the **/WRITE** qualifier creates the file TEMP.OUT and assigns it the logical name OUTPUT_FILE. TEMP.OUT is a sequential file.

   The **/ERROR** qualifier specifies that if any error occurs while opening the file, the command interpreter should transfer control to the line at the label OPEN_ERROR. The command procedure writes records to the file TEMP.OUT until the symbol COUNT equals 11.

3.
```
$ OPEN/READ INPUT_FILE TRNTO::DKA0:[COST]INVENTORY.DAT
```

```
$ READ_LOOP:
$ READ/END_OF_FILE=ENDIT  INPUT_FILE  NUM
$ FIRST_CHAR = F$EXTRACT(0,1,NUM)
$ WRITE SYS$OUTPUT FIRST_CHAR
$ GOTO READ_LOOP
$ ENDIT:
$ CLOSE INPUT_FILE
```

This command procedure opens the file INVENTORY.DAT located at remote node TRNTO as an input file, and assigns it the logical name INPUT_FILE. The **READ** command reads a record from the logical file INPUT_FILE into the symbol named NUM. The next two commands extract the first character from the record and write the character to the SYS$OUTPUT device. These two steps occur for all records in the file until the procedure reaches the end-of-file (EOF). At this point, the **CLOSE** command closes the file and deassigns the logical name INPUT_FILE.

PASSWORD

PASSWORD — Provides the password associated with the user name that you specify with the **JOB** card when you submit a batch job through a card reader. Although the **PASSWORD** card is required, the password on the card is optional if the account has a null password. **The PASSWORD command is valid only in a batch job submitted through a card reader and requires that a dollar sign ($) precede the PASSWORD command on the card.**

Format

PASSWORD [*password*]

Parameter

password

Specifies the password associated with the user name specified with the **JOB** command. The password can be 1 to 31 characters long.

If you are submitting the job from an account with a null password, omit the password specifier on the **PASSWORD** card.

Description

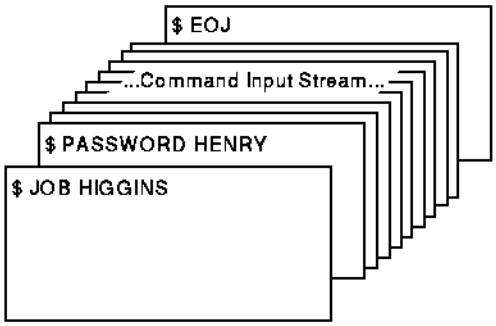
The **PASSWORD** command is used in conjunction with the **JOB** command. The **JOB** card identifies the user submitting the batch job through a card reader and is followed by a **PASSWORD** card giving the password. The password is checked by the system to verify that it matches the password associated with the user name on the **JOB** card. If the passwords do not match, the job is rejected.

Note that you might want to suppress printing when you originally key punch the **PASSWORD** card to prohibit other users from seeing the password when the **PASSWORD** card is in use.

Note

To change your password, use the **SET PASSWORD** command. For information on this command, see the description of **SET PASSWORD**.

Example



ZK-0786-GE

The **JOB** and **PASSWORD** commands precede a batch job submitted from the card reader. An **EOJ** command marks the end of the job.

PATCH

PATCH — On Alpha and IA-64, invokes **PATCH/ABSOLUTE** by default, which patches a file at absolute virtual addresses. For more information, see the online help inside the Patch Utility.

Format

PATCH ***filespec***

Parameter

filespec

Specifies the image file to be patched or a command procedure that contains both the name of the image file to be patched and PATCH commands.

If the file specification denotes an image file, the file specification must contain the file name. If you omit the remaining fields (device, directory, file type, and version number), PATCH uses your default device and directory, assumes a file type of EXE, and uses the highest version of the image file.

If the file specification denotes a command procedure, the *filespec* parameter must be preceded by an at sign (@). Only the file name is required. If you omit the remaining fields (device, directory, file type, and version number), PATCH uses your default device and directory, assumes a file type of COM, and locates the highest version of the command procedure.

The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification.

Qualifiers

/ABSOLUTE

Patches a file at absolute virtual addresses. This is the default for Alpha and IA-64 systems. The **/ABSOLUTE** function allows a user to patch any file (not just image files) at absolute virtual addresses relative to the beginning of the file. This feature allows replacement of existing data with

new data of the same length. If the data is smaller than that of the original data, PATCH uses the appropriate fill character for the mode in use. For example, if the current mode is instruction mode, a NOP is used for fill; if it is data (numeric or ASCII) mode, a NULL is used for fill. Any PATCH operation that results in a data replacement longer than the length of the original data generates an error message and terminates the command in progress; either the PATCH or DCL prompt is then displayed, whichever is appropriate.

Also, note that there is no default patch area, and none will be created, because of the tendency to corrupt a file. Patch area is meaningless in any file other than an image file.

If you patch a file in absolute mode, remember that there are no symbols available to assist you in locating data locations. You must exercise great care to ascertain that the correct locations are modified.

Most **PATCH** commands will work in their normal fashion. However, only **REPLACE** and **DEPOSIT** should be used for write operations; other commands are acceptable for read operations. Commands that attempt to expand the file, such as **ALIGN** and **INSERT**, should be avoided because they will probably corrupt the file. (These commands will be trapped by PATCH and an error message will be issued indicating that the replacement data must not exceed the length of the original data.)

File attributes are propagated from the original input file to the output file. These include ALQ, TYPE, MRS, RAT, RFM, and RAC.

/JOURNAL[=*filespec*]

Indicates an alternate journal file specification. By default, PATCH creates a journal file with a file specification that consists of the current defaults. Use the **/JOURNAL** qualifier when you want to specify an alternate file specification. If you omit fields in the file specification, PATCH supplies the following default values:

Field	Default Value
device and directory	Defaults of current process
file name	Name of input image file
file type	JNL
version	1

Subsequent PATCH sessions append information to the journal file, rather than create a new version of this file.

The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification.

/NEW_VERSION (default)
/NONEW_VERSION

Controls whether a new version of the patched file is created or the contents of the existing file are modified in place. The **/NEW_VERSION** qualifier is used in conjunction with the **/ABSOLUTE** qualifier to control whether a new version of the patched file is created or the contents of the existing file are modified in place. **/NEW_VERSION** is the default. If **/NONEW_VERSION** is selected, the PATCH command **UPDATE** will act as a checkpoint operation, that is, all modifications made to the file are written back to the file instead of waiting until image exit. If **/ABSOLUTE** is not specified with **/NONEW_VERSION**, **/NONEW_VERSION** is ignored, that is, a new version of

the file will be created. **/NONEW_VERSION** will need to be used when patching large data files when there is not enough disk space to create a new version of the patched file.

Note

If **/NEW_VERSION** is specified, the file will be overwritten. No attempt on the part of the user, including pressing **Ctrl/Y**, will prevent this result. Therefore, you should have a back up copy of the file before making any attempt to patch it.

PATCH will always issue an informational message at image exit, indicating that the file is being overwritten.

/OUTPUT[=*filespec*]

Indicates the output image file specification. By default, PATCH creates an output file with a file specification that consists of the current defaults. Use the **/OUTPUT** qualifier when you want to specify an alternate file specification.

If you omit fields in the file specification, PATCH supplies the following default values:

Field	Default Value
device and directory	Defaults of current process
file name	Name of input image file
file type	.EXE
version	One greater than the most recent copy of the input image file

The output image file is created only when you issue the PATCH command **UPDATE** at the end of the PATCH session. You can issue multiple **UPDATE** commands in a single session. The first **UPDATE** command creates the output image file; subsequent **UPDATE** commands overwrite this file.

The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification.

/VOLUME[=*n*]

Requests that the output file be placed on a specified relative volume number of a multi-volume set. If you specify **/VOLUME** without a number, the number defaults to the relative volume number of the input image file.

If the **/VOLUME** qualifier is not specified, the file is placed in an arbitrary position within the multi-volume set.

Examples

1.
```
$ PATCH/ABSOLUTE IMAGE.EXE
PATCH>EX/INS 604
00000604: BBSS #07,R1,00000608
PATCH>REPLACE/INS 604='BBSS #07,R1,0608'
NEW> 'BBSS #07,R1,0608'
NEW> 'CLRL R0'
NEW> EXIT
old: 00000604: BBSS #07,R1,00000608
```

```
%PATCH-E-DATTOOLNG, length of new data may not exceed length of
old data
PATCH>EX/INS 684
00000684: MOVB #01,(R5)+
PATCH>REPLACE/INS 684='MOVB #01,(R5)+'
NEW> 'MOVB #02,(R5)+'
NEW> EXIT
old: 00000684: MOVB #01,(R5)+
new: 00000684: MOVB #02,(R5)+
PATCH>EX/INS 687
00000687: MOVB #00,(R5)+
PATCH>DEPOSIT/INS 687
NEW> 'CLRB (R5)+'
NEW> EXIT
old: 00000687: MOVB #00,(R5)+
new: 00000687: CLRB (R5)+
PATCH>INSERT/INS 68D
OLD> 'MOVB #10,(R5)+'
NEW> 'MOVB #20,(R5)+'
NEW> EXIT
old: 0000068D: MOVB #10,(R5)+
%PATCH-E-DATTOOLNG, length of new data may not exceed length of
old data
PATCH>UPDATE
%PATCH-I-WRTFIL, updating image file
DISK$STARWORK01:[NASR.PATCH]IMAGE.EXE;2
PATCH>EXIT
```

Example of the **/ABSOLUTE** qualifier with **/NEW_VERSION** as the default. Note the error messages returned when the command tries to expand the file.

2.
```
$ PATCH/ABSOLUTE/NONEW_VERSION LOGIN.COM
PATCH>EX/ASCII 57
00000057: 'MANA'
PATCH>REPLACE/ASCII 57='MANA'
NEW> 'mana'
NEW> 'test'
NEW> EXIT
old: 00000057: 'MANA'
%PATCH-E-REPLACEERR, replacement value too large for location
PATCH>replace/ascii 57='MANA'
NEW> 'mana'
NEW> EXIT
old: 00000057: 'MANA'
new: 00000057: 'mana'
PATCH>EX/ASCII 24
00000024: 'F$MO'
PATCH>INSERT/ASCII 24='F$MO'
NEW> 'test'
NEW> EXIT
%PATCH-E-INVCMD, invalid command
PATCH>UPDATE
%PATCH-I-OVERLAY, DISK$STARWORK01:[NASR.PATCH]LOGIN.COM;1 being
overwritten
PATCH>EX 68:75
00000068: 4349544F
0000006C: 58542E45
00000070: 00010054
```

```
00000074: 00100024
PATCH>REPLACE 68
OLD> 4349544F
OLD> 58542E45
OLD> 00010054
OLD> EXIT
NEW> 6369746F
NEW> 68642E65
NEW> 00010074
NEW> EXIT
old: 00000068: 4349544F
old: 0000006C: 58542E45
old: 00000070: 00010054
new: 00000068: 6369746F
new: 0000006C: 68642E65
new: 00000070: 00010074
PATCH>EX/ASCII 68
00000068: 'otic'
PATCH>UPDATE
%PATCH-I-OVERLAY, DISK$STARWORK01:[NASR.PATCH]LOGIN.COM;1 being
overwritten
PATCH>EXIT
%PATCH-I-OVERLAY, DISK$STARWORK01:[NASR.PATCH]LOGIN.COM;1 being
overwritten
$
```

Example of a **PATCH/ABSOLUTE/NONEW_VERSION** command. Note the error messages that are returned when the command tries to expand the file, and when the commands **UPDATE** and **EXIT** are performed; that is, `(file) being overwritten`.

3. `$ PATCH AVERAGE /JOURNAL=TEST /OUTPUT=TEST`

   This command invokes PATCH for an interactive PATCH session with the image file `AVERAGE.EXE`. The journal file and output image file created by this session are both named `TEST` and reside in the default device and directory. The journal file has a file type of JNL, and the output image file has a file type of .EXE.

4. ```
   $ PATCH /UPDATE=(100,102) @ORION
   $
   ```

   This **PATCH** command executes the command procedure `ORION.COM`. The **/UPDATE** qualifier requests that only the patches identified by the ECO levels `100` and `102`, contained in `ORION.COM`, be processed. The first record in `ORION.COM` must specify the input image file. The second DCL prompt (`$`) indicates that the patches were successfully applied.

PHONE

PHONE — Invokes the Phone utility, which lets you communicate with other users on your system or any other system connected to your system by DECnet-Plus or DECnet for OpenVMS. For more information about the Phone utility, see online help.

Format

PHONE [*phone-command*]
   ```

PIPE

PIPE — Executes one or more DCL command strings from the same command line. The **PIPE** command enables you to perform UNIX-style command processing, such as command pipelining, input/output redirection, and conditional and background execution.

Format

PIPE ***command-sequence*** **[*separator command-sequence*]...**

Parameters

command-sequence

A DCL command, a pipeline, or a subshell:

DCL command

A DCL command string, which can include qualifiers, parameters, keywords, and values.

Pipeline

A pipeline is a sequence of pipeline-segment commands connected by **pipes**, represented by the vertical-bar (|) separator. A pipeline-segment command is a DCL command that appears in a pipeline. The pipe connects the SYS$OUTPUT of one pipeline-segment command to the SYS$INPUT of the next command. The format of a pipeline is as follows:

```
pipeline-segment-command | pipeline-segment-command [|...]
```

Subshell

A subshell is one or more command sequences separated by separators and enclosed in parentheses. The format of a subshell is as follows:

```
(command-sequence [separator command-sequence]...)
```

Input/output redirection is allowed in a command sequence. The command before an angle bracket (> or <) redefines its SYS$INPUT, SYS$OUTPUT, or SYS$ERROR during execution. You cannot use angle brackets (<>) to represent a directory specification in a **PIPE** command because the **PIPE** command interprets angle brackets as input/output redirection syntax.

separator

Determines the processing action of the command sequences specified in a **PIPE** command. The valid **PIPE** separators are described in *Table 1, "PIPE Command Separators"*.

Table 1. PIPE Command Separators

Separator	Action
|	Key pipe separator. The pipe connects the SYS$OUTPUT of one pipeline-segment command to the SYS$INPUT of the next command.
;	Sequential execution. The command sequence following the semicolon (;) is executed after the preceding command sequence is completed. You must precede this separator with a blank space; otherwise, it is parsed as the Record Management System (RMS) file specification version number delimiter.

Separator	Action
&&	Conditional execution (upon success). The command sequence following the double ampersand (&&) is executed only if the preceding command sequence succeeds.
||	Conditional execution (upon failure). The command sequence following the double vertical bar (||) is executed only if the preceding command sequence fails.
&	Background execution. All command sequences that precede the ampersand (&) are executed asynchronously in a subprocess environment. The & separator is similar to the **SPAWN/NOWAIT** command. Note that any ampersand that precedes a character string without spaces in between is parsed as a conventional DCL symbol substitution expression rather than the background execution syntax.
@TEE	Command file, TEE.COM. Used for redirecting output to two targets (for example, one output is directed to the next stage in pipeline, and the other to a file). See the Examples section for an example of how to use TEE.COM.

In a **PIPE** command line, the "&" has the highest precedence, followed by "|", ";", "&&", and "||", which have equal precedence.

Description

The **PIPE** command allows you to perform UNIX-style command processing by executing multiple DCL commands in a single command line. You can use the **PIPE** command to execute DCL commands in a number of ways:

Multiple command execution

Multiple DCL commands are specified in a single **PIPE** command and executed sequentially. The syntax for multiple command execution is as follows:

```
PIPE   command-sequence ; command-sequence [; command-sequences]...
```

Conditional command execution

A command sequence is executed conditionally depending on the execution result of the preceding command sequence.

Using the following form, *command-sequence2* only executes if *command-sequence1* succeeds:

```
PIPE   command-sequence1   &&   command-sequence2
```

Using the following form, *command-sequence2* only executes if *command-sequence1* fails:

```
PIPE   command-sequence1   ||   command-sequence2
```

Pipeline command execution

A pipeline is formed by connecting DCL commands with pipes as follows:

```
PIPE pipeline-segment-command | pipeline-segment-command [|...]
```

Each pipeline-segment command runs in a separate subprocess with its SYS$OUTPUT connected to the SYS$INPUT of the next pipeline-segment command. These subprocesses execute in parallel;

however, they are synchronized to the extent that each pipeline-segment command, except the first, reads the standard output of its predecessor as its standard input. A pipeline finishes execution when the last pipeline-segment command is done.

It is very common to use filter applications in a pipeline. A filter application is a program that takes data from SYS$INPUT, transforms it in a specific way, and writes it to SYS$OUTPUT.

Subshell execution

Command sequences can be executed in a subprocess environment by using the subshell execution form:

```
PIPE   ( command-sequence [separator command-sequence]... )
```

The command sequences in a subshell are executed in a subprocess environment. DCL waits for the subshell to complete before executing the next command sequence. The ( ) separator is similar to the **SPAWN/WAIT** command.

Background execution

Command sequences can be executed in a subprocess environment by using the following form:

```
PIPE  command-sequence [ separator command-sequence]...  &
```

DCL does not wait for the command sequences to finish. Control passes back to DCL once the background subprocess is created.

Input/output redirection

A command sequence can redirect its SYS$INPUT, SYS$OUTPUT, or SYS$ERROR to a file during execution of the command as follows:

To redirect SYS$INPUT:

```
PIPE    command-sequence < redirected-input-file
```

To redirect SYS$OUTPUT:

```
PIPE    command-sequence > redirected-output-file
```

To redirect SYS$ERROR:

```
PIPE    command-sequence 2> redirected-error-file
```

A pipeline-segment command can also redirect its SYS$INPUT, SYS$OUTPUT, or SYS$ERROR; however, SYS$OUTPUT redirection is allowed only for the last pipeline-segment command, and SYS$INPUT redirection is allowed only for the first pipeline-segment command.

You can interrupt a **PIPE** command by pressing **Ctrl/Y**. If the **PIPE** command is executing in a pipeline or a subshell command sequence, the command sequence and the **PIPE** command are deleted. In this case, a **CONTINUE** command entered immediately after the interrupt will not resume the execution of the **PIPE** command.

If the **PIPE** command is executing a command sequence other than a subshell or a pipeline command sequence, DCL behaves as if the command sequence were entered as a DCL command without the **PIPE** command verb and interrupted by **Ctrl/Y**. See the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#_46_CTRLYINTERRUPTS] for more information on the **Ctrl/Y** interrupt.

Each command sequence sets the global symbol $STATUS with a returned value after it finishes execution. The return status of the **PIPE** command is the return status of the *last* command performed in the last segment. If all segments fail with some kind of error and the last segment returns with success, the status returned to DCL is the success.

When a **PIPE** command is executed in a command procedure with the ON condition processing, the conditional execution of command sequences (&&, ||) takes precedence over the action previously specified by the ON condition statement.

DCL Command Restrictions

The **PIPE** command creates a special execution context for its command sequences. The following DCL commands either do not work or exhibit new behavior in this context:

- **PIPE** – Nested **PIPE** commands in the same command procedure level are not allowed. There can only be one **PIPE** command context for each command procedure level; however, nested **PIPE** commands at different procedure levels are allowed. For example:

```
$ TYPE FOO.COM
$ ! FOO.COM
$   :
$ PIPE   ...
$ :
$
$ PIPE    @FOO.COM ; ...
```

  In this example, the **PIPE** command inside `FOO.COM` is allowed because it is executed at a different command procedure level.

- **GOTO** and **EXIT** – These two commands, when executed as **PIPE** command sequences, delete the **PIPE** command context before the **GOTO** or **EXIT** command is executed. Any command sequences following these two commands in a **PIPE** command are flushed.

- **STOP** – The **STOP** command, when executed after a **PIPE** command is interrupted by **Ctrl/Y**, deletes the **PIPE** command context.

- **THEN**, **ELSE**, **ENDIF**, **SUBROUTINE**, **ENDSUBROUTINE**, **RETURN**, and DCL labels – These commands cannot execute as **PIPE** command sequences because it is not possible to realize their functions in a **PIPE** command context.

Improving Subprocess Performance

A **PIPE** command can generate a number of subprocesses during execution. Often, the applications invoked by command sequences do not depend on the process logical names and symbol names. In this case, the spawning of subprocesses can be accelerated by using the **/NOLOGICAL_NAMES** and **/NOSYMBOLS** qualifiers, which suppress the passing of process logical names and symbols to the subprocesses created by the **PIPE** command.

Input/Output Redirection

DCL users can use the **DEFINE** or **ASSIGN** command to redirect SYS$INPUT, SYS$OUTPUT, or SYS$ERROR. Such redirection can be created as either the user-mode (using the **/USER_MODE** qualifier) or supervisor-mode (using the **/SUPERVISOR_MODE** qualifier) redirection. A user-mode redirection only affects the environment of the next user-mode image.

In a **PIPE** command, redirection can be achieved by using the redirection syntax. A **PIPE** command redirection is quite different from that created by the **DEFINE** or **ASSIGN** command, as follows:

- Redirections are created in supervisor mode. This means that both user-mode applications and DCL commands are affected by the redirections.
- The redirected environment only applies to the command sequence or the pipeline-segment command that specifies the redirection syntax. After the execution of the command sequence or pipeline-segment command, the original process input/output environment (that is, SYS$INPUT, SYS$OUTPUT, and SYS$ERROR) is restored before command execution continues.

When SYS$OUTPUT is redirected, the redirected output file is always created, whether or not the command sequence actually writes to SYS$OUTPUT. If a version of a file with the same name as the redirected output file already exists, a new version of that file is created. This behavior is the same as using the **DEFINE** or **ASSIGN** command to redefine SYS$OUTPUT in supervisor mode. Note that the redirected file is created before the command sequence is executed. If the redirected file is also used in the command sequence, the operation may fail, as in the following example:

```
$ TYPE FOO.COM
$ ! FOO.COM
$ :
$ PIPE ...
$ :
$
$ PIPE @FOO.COM ; ...
```

In this example, a new version of TRANS.LOG is created and opened for write access; the **SEARCH** command then tries to get read access to the most recent version of TRANS.LOG instead of the expected previous version.

When SYS$ERROR is redirected, the redirected error file is only created when the command sequence actually writes to the SYS$ERROR during execution, and there is no existing file with the same name as the redirected error file. If a file with the same name as the redirected error file already exists, that file is opened as the redirected error file. The error output generated by this command sequence is then appended to the end of the redirected error file. This behavior is the same as using the **DEFINE** or **ASSIGN** command to redefine SYS$ERROR in supervisor mode.

Pipelines and TEEs

This section describes aspects of DCL that function differently in the context of a pipeline.

Some of the following constructs are used in the implementation of a TEE.

Using SYS$COMMAND

The SYS$COMMAND of a subprocess is normally the same as its SYS$INPUT (if no command procedures are involved). In a pipeline, however, the SYS$COMMAND of a subprocess is set to the SYS$COMMAND of the parent process instead of to the preceding pipe (which is the SYS$INPUT of the pipeline-segment command).

Using TEEs and SYS$PIPE

In most cases, input from the pipe can be obtained by reading the data from SYS$INPUT; however, when a command procedure is invoked as a pipeline segment command, SYS$INPUT is redirected to the command procedure file. To obtain data from the pipe inside a command procedure, the logical SYS$PIPE can be used.

The following is an example of a pipeline DCL application TEE.COM:

```
$ ! TEE.COM - command procedure to display/log data flowing through
$ ! a pipeline
$ ! Usage: @TEE log-file
$
$ OPEN/WRITE tee_file 'P1'
$ LOOP:
$ READ/END_OF_FILE=EXIT SYS$PIPE LINE
$ WRITE SYS$OUTPUT LINE ! Send it out to the next stage of the pipeline
$ WRITE tee_file LINE ! Log output to the log file
$ GOTO LOOP
$ EXIT:
$ CLOSE tee_file
$ EXIT
```

The **PIPE** command to use TEE.COM can be:

```
$ PIPE SHOW SYSTEM | @TEE showsys.log | SEARCH SYS$INPUT LEF
```

The command procedure TEE.COM is used to log the data flowing through the pipeline. It reads in the data from SYS$PIPE instead of SYS$INPUT.

Image Verification in a Pipeline

In a pipeline, image verification is turned off by default, even when the command **SET VERIFY=IMAGE** is executed before the **PIPE** command is entered. This prevents duplication of data records going through the pipeline.

To turn on image verification in a pipeline, an explicit **SET VERIFY=IMAGE** command must precede the pipeline segment command. You can use a subshell to do this, as follows:

```
$ PIPE ... | (SET VERIFY=IMAGE ; ...) | ...
```

File Access Methods in a Pipeline

A pipeline segment command can only use the RMS sequential file access method to read and write to the pipes. Certain OpenVMS utilities may access their input and output files using methods other than sequential access. These operations are not supported in a pipeline, and will fail, as in the following example:

```
$ PIPE CC/NOOBJ/NOLIS TEST.C | SEARCH SYS$INPUT/WIND=(1,1) "%cc-w-"
%SEARCH-F-RFAERR, RMS error using RFA access
-RMS-F-RAC, invalid record access mode
```

In this example, the **/WINDOW** qualifier for the **SEARCH** command requires the relative file access method.

Qualifiers

/LOGICAL_NAMES (default)
/NOLOGICAL_NAMES

Copies process logical names and logical name tables to the subprocess of a command sequence. By default, all process logical names and logical name tables are copied to the subprocess except those explicitly marked CONFINE or created in executive or kernel mode.

/PRIVILEGES=CURRENT|AUTHORIZED

Determines whether the subprocess inherits the current process's current or authorized privileges as its authorized privileges. By default, the authorized privilege mask for the subprocess is taken from the current privileges of its creator. This corresponds to **/PRIVILEGES=CURRENT**. If the **/PRIVILEGES=AUTHORIZED** qualifier is specified, the subprocess's authorized privileges are taken from the creator's authorized privileges.

/SYMBOLS (default)
/NOSYMBOLS

Determines whether global and current procedure level local symbols (except $RESTART, $SEVERITY, and $STATUS) are passed to the subprocess. $RESTART, $SEVERITY, and $STATUS symbols are never passed to the subprocess.

/TRUSTED
/NOTRUSTED

Indicates that the **PIPE** command input originates in a trusted command procedure. **PIPE** commands are not allowed in CAPTIVE accounts. The **/TRUSTED** qualifier provides a way for properly written captive command procedures to perform **PIPE** operations when the command input originates in the captive command procedure where it can be trusted. For more information about trusted command procedures, see the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/].

Examples

1.
```
$ PIPE SHOW SYSTEM | SEARCH SYS$INPUT HIB
```

   This example uses the pipeline function to identify all hibernating processes on the system in one command.

2.
```
$ PIPE RUN TEST | SORT/SPECIFICATION=TEST.SRT SYS$INPUT SYS$OUTPUT -
_$ | DIFF SYS$INPUT  TEST.BENCHMARK
```

   This example uses the pipeline function to run a test, sort the result, and compare the result to the benchmark file in a single command without generating unnecessary intermediate files.

3.
```
$ PIPE  ( SET DEF WRK$:[WORK] ; RUN REPORT ) | MAIL SYS$INPUT SMITH
```

   This example shows one way a subshell can be specified as a pipe segment command in a pipeline.

4.
```
$ more :== TYPE/PAGE=SAVE SYS$INPUT
$ PIPE    ANA/RMS PAGE.TXT | more

Check RMS File Integrity                26-DEC-2001 16:12:00.06  Page 1
SYS$SYSDEVICE:[TEST]PAGE.TXT;2

FILE HEADER

    File Spec: SYS$SYSDEVICE:[TEST]PAGE.TXT;2
    File ID: (4135,58220,0)
    Owner UIC: [PIPE]
    Protection:  System: RWED, Owner: RWED, Group: RE, World:
    Creation Date:   26-NOV-2001 16:08:50.05
    Revision Date:   26-NOV-2001 16:09:09.06, Number: 1
    Expiration Date: none specified
```

```
    Backup Date:     none posted
    Contiguity Options:  none
    Performance Options: none
    Reliability Options: none
    Journaling Enabled:  none

RMS FILE ATTRIBUTES

RETURN/SPACE=More, PREV/NEXT=Scroll, INS/REM=Pan, SELECT=80/132, Q=Quit
```

This example shows the use of the **/PAGE** qualifier within a pipeline. The **/PAGE** function exists in a number of other DCL commands as well, and can be used similarly in conjunction with the **PIPE** command to form other useful tools.

5.
```
$ ! TEE.COM - command procedure to display/log data flowing through
$ !           a pipeline
$ ! Usage: @TEE log-file
$
$ OPEN/WRITE  tee_file 'P1'
$ LOOP:
$  READ/END_OF_FILE=EXIT  SYS$PIPE LINE
$  WRITE SYS$OUTPUT LINE ! Send it out to next stage of the pipeline
$  WRITE tee_file LINE  ! Log output to the log file
$  GOTO LOOP
$ EXIT:
$  CLOSE tee_file
$  EXIT
```

This is an example of a pipeline DCL application `TEE.COM`.

The **PIPE** command to use TEE.COM can be:

```
$ PIPE  SHOW SYSTEM | @TEE showsys.log | SEARCH SYS$INPUT LEF
```

The command procedure TEE.COM is used to log the data flowing through the pipeline. It reads in the data from SYS$PIPE instead of SYS$INPUT.

6.
```
$ CD_WORK :== PIPE   SAVE_DIR=F$DIRECTORY() ; SET DEFAULT FOO:[WORK]
$ BACK  :== SET DEF 'SAVE_DIR'
$
$ CD_WORK  ! Switch to working directory
$     :
$     :
$ BACK     ! Switch back to home directory


$ GET_RECORD :== PIPE READ/END_OF_FILE=CLEANUP IN RECORD ; -
            F$EDIT(RECORD, "COMPRESS, TRIM")
$
$ OPEN IN EMPLOYEE.DAT
$ LOOP:
$ GET_RECORD
$    :
$    :
$ GOTO LOOP
$
$ CLEAN_UP:
$    :
```

This example shows two simple uses of multiple commands with symbol definitions to build useful tools in command procedures.

7. `$ PIPE cc foo.c && link foo`

   If the compilation does not generate any error, the object file is linked to produce an executable image. If the program compilation generates an error, the linking step is skipped.

8. ```
   $
   $ PIPE RUN COLLECT_DATA.EXE || GOTO CLEAN_UP
   $   :
   $  :
   $ EXIT
   $
   $ CLEAN_UP:
   $ :
   $  :
   ```

   Using conditional command execution, it is easy to set up simple error handling control flow in a command procedure. If the image `COLLECT_DATA` fails, control is directed to `CLEAN_UP`.

9. `$ PIPE COPY LARGE_FILE.DAT REMOTE"user password"::[DESTINATION]*.*  &`

   This **PIPE** command creates a background process to handle the copying of the large file.

10. `$ PIPE (SET DEF [.DATA_DIR] ; BACKUP  DATA.SAV/SAV [...]) ; RUN FOO`

    The subshell command sequence is done in a subprocess. This means that changing a process-specific characteristic (for example, the default directory) will not affect the current process after the subshell is finished. In this example, the save set is restored in a subdirectory to provide the necessary data to run the program `FOO`.

11. ```
    $ RSH 0 PIPE SH DEC/FI | SH LOG/PR

    LNM$PROCESS_TABLE

      "SYS$COMMAND" = "_NODE$MPA24:"
      "SYS$DISK" = "SYS$SYSDEVICE:"
      "SYS$ERROR" = "_BG9930:"
      "SYS$INPUT" [super] = "_BG9930:"
      "SYS$INPUT" [exec] = "_NODE$MPA24:"
      "SYS$OUTPUT" [super] = "_BG9930:"
      "SYS$OUTPUT" [exec] = "_BG9930:"
      "SYS$PIPE" = "_NODE$MPA24:"
      "TT" = "_MPA24:"
    ```

    This example shows how `SYS$PIPE` and `SYS$OUTPUT` might differ.

PPPD

PPPD — Invokes the Point-to-Point Protocol utility (PPPD) that you can use to initiate and manage an Internet Protocol (IP) network connection over an asynchronous, serial data line.

Format

PPPD [*subcommand*]...

Description

PPPD extends the networking capability of OpenVMS Alpha by enabling you to do the following:

- Establish temporary, high-speed network connections between remote hosts. This includes both *dial-in* capability from a remote host to an OpenVMS Alpha host and *dial-out* capability from an OpenVMS Alpha host to a remote system or server box that supports the Point-to-Point Protocol (PPP).
- Establish permanent, low-speed network connections between local hosts, such as between a laptop computer and an Alpha workstation connected by a serial data line.
- Set and display communication characteristics, such as address compression, flow control, and line speed.

Note

This utility is enabled by your TCP/IP software during the network registration process. If you receive one of the following error messages, contact your system administrator to verify whether PPPD is currently available on your network.

```
%PPPD-E-PPPNOTAVAIL, point-to-point driver is not installed
%PPPD-E-NOTREG, network protocol has not been registered
```

For information about network registration, see the **SET NETWORK** command and see the *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/].

For a complete description of PPPD, see the relevant section in the *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/#d0e8479].

For detailed information about the asynchronous (ASN) and PPP device drivers that support this utility, see the documentation contained in the files PPP_INTERFACES.PS and PPP_INTERFACES.TXT located in the SYS$SYSROOT:[SYSHLP.EXAMPLES.PPPD.DOC] directory.

PRINT

PRINT — Queues one or more files for printing to an output queue. **Requires read (R) access to the file and submit (S) access to the queue. To specify functions unique to particular print symbionts, use the /PARAMETERS qualifier.**

Format

PRINT *filespec*[,...]

Parameter

***filespec*[,...]**

Specifies one or more files to be printed. The asterisk (*) and the percent sign (%) wildcard characters are allowed in the directory specification, file name, file type, and version number fields. The default file

type is that of the preceding file. If no previous file specification contains an explicit file type, the default file type is .LIS.

If you specify more than one file, separate the file specifications with either commas (,) or plus signs (+).

If you specify a node name, you must use the **/REMOTE** qualifier.

Description

The **PRINT** command places the specified files in an output queue for printing. By default, this queue is SYS$PRINT. All files queued by a single **PRINT** command are processed serially as one job. By default, the name of the print job is the name of the first file specified in the **PRINT** command.

The system assigns a unique **entry number** to each print job in the queue. When you enter the **PRINT** command, by default, the system displays the job name, the queue name, the entry number, and the job status.

The system automatically creates or updates the local symbol $ENTRY when a **PRINT** or **SUBMIT** command is completed successfully. The value of $ENTRY is a string that identifies the entry number of the most recently queued job. If you want to refer to a job's entry number later, store the value of $ENTRY in another symbol.

After you queue a print job, the version of the file submitted is printed, even if a newer version of the file is created before the print job runs. Also, another file with the same name and version number as the file queued cannot be substituted for the file that was queued.

Qualifiers

/AFTER=*time*
/NOAFTER

Holds the job until the specified time. The time can be specified as absolute time or a combination of absolute and delta times. If the specified time has passed, the job is queued for printing immediately.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/BACKUP
/NOBACKUP

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/BACKUP** qualifier selects files according to the dates of their most recent backups. This qualifier is incompatible with the **/CREATED**, **/EXPIRED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/BEFORE[=*time*]
/NOBEFORE

Selects only those files dated prior to the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the **/BEFORE** qualifier to indicate the time attribute to be used as the basis for selection: **/BACKUP**, **/CREATED** (default), **/EXPIRED**, or **/MODIFIED**.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/BURST[=*keyword*]
/NOBURST

Positional qualifier.

Controls whether two file flag pages with a burst bar between them are printed preceding a file. If the **/BURST** qualifier is specified between the **PRINT** command and the file specifications, it can take either of the following keywords:

ALL	Prints the flag pages and a burst bar before each file in the job.
ONE	Prints the flag pages and a burst bar before the first file in the job.

If you want the **/BURST** qualifier to apply to individual files in a multi-file job, place the qualifier directly after each file that you want to have the flag pages and a burst bar.

Use the **/[NO]BURST** qualifier to override the **/DEFAULT** options that have been set for the output queue you are using. The **/[NO]BURST** qualifier does not override the **/SEPARATE** options set for the queue.

When you specify the **/BURST** qualifier for a file, the **/[NO]FLAG** qualifier does not add or subtract a flag page from the two flag pages that are printed preceding a file.

/BY_OWNER[=*uic*]
/NOBY_OWNER

Selects only those files whose owner user identification code (UIC) matches the specified owner UIC. The default UIC is that of the current process.

Specify the UIC by using standard UIC format as described in the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#UIC_SECTION].

/CHARACTERISTICS= (*characteristic*[,...])

Specifies the name or number of one or more characteristics to be associated with the job. Characteristics can refer to such things as color of ink. If you specify only one characteristic, you can omit the parentheses.

A characteristic's number must range from 0 to 127. To see which characteristics have been defined for your system, use the **SHOW QUEUE/CHARACTERISTICS** command. To see which characteristics are associated with a particular queue, use the **SHOW QUEUE/FULL** command.

A print job can be processed on an execution queue if the job's characteristics are a subset of the queue's characteristics. However, if any of the characteristics associated with the job are not associated with the queue, the job remains pending until one or more of the following occurs:

- The characteristics specified with the queue are changed to make the job's characteristics a subset of the queue's characteristics (using, for example, the **SET QUEUE/CHARACTERISTICS** command).

- The characteristics specified with the job are changed to make the job's characteristics a subset of the queue's characteristics (using, for example, the **SET ENTRY/CHARACTERISTICS** command).
- The job is moved to a queue on which all the job's characteristics have been specified (using, for example, the **SET ENTRY/REQUEUE** command).
- The job is deleted (using, for example, the **DELETE/ENTRY** command).

/CONFIRM
/NOCONFIRM (default)

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

YES	NO	QUIT
TRUE	FALSE	**Ctrl/Z**
1	0	ALL
	Return	

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), but these abbreviations must be unique. Affirmative answers are YES, TRUE, and 1. Negative answers include: NO, FALSE, 0, and pressing **Return**. Entering QUIT or pressing **Ctrl/Z** indicates that you want to stop processing the command at that point. When you respond by entering ALL, the command continues to process, but no further prompts are given. If you type a response other than one of those in the list, DCL issues an error message and redisplays the prompt.

/COPIES=*n*

Positional qualifier.

Specifies the number of copies to print. The value of the parameter *n* can be from 1 to 255 and defaults to 1. If you place the **/COPIES** qualifier after the **PRINT** command name, each file in the parameter list is printed the specified number of times. If you specify the **/COPIES** qualifier following a file specification, only that file is printed the specified number of times.

/CREATED (default)
/NOCREATED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/CREATED** qualifier selects files based on their dates of creation. This qualifier is incompatible with the **/BACKUP**, **/EXPIRED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/DELETE
/NODELETE (default)

Positional qualifier.

Controls whether files are deleted after printing. If you place the **/DELETE** qualifier after the **PRINT** command name, all specified files are deleted. If you specify the **/DELETE** qualifier after a file specification, only that file is deleted after it is printed.

The protection applied to the file must allow delete (D) access for the life of the job. You need to have delete access when you submit the job and delete access when the system deletes your file at the end of the job.

/DEVICE=*queue-name*[:]

Places the print job in the specified queue (rather than the default queue SYS$PRINT). This qualifier is synonymous with the **/QUEUE** qualifier, except that the **/DEVICE** qualifier is reserved for special use by VSI. Its usage, therefore, is not recommended.

/EXCLUDE=(*filespec*[,...])
/NOEXCLUDE

Excludes the specified files from the print operation. You can include a directory but not a device in the file specification. The asterisk (*) and the percent sign (%) wildcard characters are allowed in the file specification; however, you cannot use relative version numbers to exclude a specific version. If you specify only one file, you can omit the parentheses.

/EXPIRED
/NOEXPIRED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/EXPIRED** qualifier selects files according to their expiration dates. (The expiration date is set with the **SET FILE/EXPIRATION_DATE** command.) The **/EXPIRED** qualifier is incompatible with the **/BACKUP**, **/CREATED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/FEED
/NOFEED

Positional qualifier.

Controls whether form feeds are inserted into the print job when the printer reaches the bottom margin of the form in use. You can suppress this automatic form feed (without affecting any of the other carriage control functions that are in place) by using the **/NOFEED** qualifier. The **/[NO]FEED** qualifier does not affect user-formatted files and can be used to override the installation-defined defaults that have been set for the output queue you are using.

/FLAG[=*keyword*]
/NOFLAG

Positional qualifier.

Controls whether a file flag page is printed preceding a file. The flag page contains the name of the user submitting the job, the job entry number, and other information about the file being printed. If the **/FLAG** qualifier is positioned between the **PRINT** command and the file specifications, it can take either of the following keywords:

Keyword	Description
ALL	Prints a file flag page before each file in the job.
ONE	Prints a file flag page before the first file in the job.

If you want the **/FLAG** qualifier to apply to individual files in a multi-file job, place the qualifier directly after each file that you want to have a flag page.

Use the **/[NO]FLAG** qualifier to override the **/DEFAULT=[NO]FLAG** option that has been set for the output queue you are using. The **/[NO]FLAG** qualifier does not override the **/SEPARATE=[NO]FLAG** option set for the queue.

When you specify the **/BURST** qualifier for a file, the **/[NO]FLAG** qualifier does not add or subtract a flag page from the two flag pages that are printed preceding the file.

/FORM=*form*

Specifies the name or number of the form to be associated with the print job. If you omit the **/FORM** qualifier, the default form for the execution queue is associated with the job.

Forms have attributes such as print image width and length or paper stock, which the print symbiont associates with a job when the job is processed. To see which forms have been defined for your system, use the **SHOW QUEUE/FORM** command. To find out which form is mounted currently on a particular queue and which form is specified as that queue's default form, use the **SHOW QUEUE/FULL** command.

The stock of the form associated with the job must match the stock of the form mounted on the execution queue on which you want the job to be processed. If the stocks do not match, the job remains pending until one or more of the following occurs:

- A form with the same stock as the job's form is mounted on the queue (using, for example, the **SET QUEUE/FORM_MOUNTED** command).
- A form with the same stock as the queue's mounted form is specified with the job (using, for example, the **SET ENTRY/FORM** command).
- The job is moved to a queue on which the stock of the mounted form matches the stock of the job's form (using, for example, the **SET ENTRY/REQUEUE** command).
- The job is deleted (using, for example, the **DELETE/ENTRY** command).

/HEADER
/NOHEADER (default)

Positional qualifier.

Controls whether a heading line is printed at the top of each page.

/HOLD
/NOHOLD (default)

Controls whether a job is available for printing immediately. The **/HOLD** qualifier holds the job until released by a **SET ENTRY/RELEASE** or **SET ENTRY/NOHOLD** command.

/IDENTIFY (default)
/NOIDENTIFY

Displays the job name, queue name, entry number, and status of the job when it is queued.

/JOB_COUNT=*n*

Prints the job *n* times. The value of the parameter *n* can be from 1 to 255 and defaults to 1.

/LOWERCASE
/NOLOWERCASE (default)

Indicates whether the print job must be printed on a printer that can print both lowercase and uppercase letters. The **/NOLOWERCASE** qualifier means that files can be printed on printers supporting only uppercase letters. If all available printers can print both uppercase and lowercase letters, you do not need to specify the **/LOWERCASE** qualifier.

/MODIFIED
/NOMODIFIED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/MODIFIED** qualifier selects files according to the dates on which they were last modified. This qualifier is incompatible with the **/BACKUP**, **/CREATED**, and **/EXPIRED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time modifiers, the default is the **/CREATED** qualifier.

/NAME=*job-name*

Names the job. The name consists of 1 to 39 alphanumeric characters. If characters other than alphanumerics, underscores (_), or dollar signs ($) are used in the name, enclose the name in quotation marks (" "). The default is the name of the first (or only) file in the job. The job name appears in the **SHOW ENTRY** and **SHOW QUEUE** command display and is printed on the flag page for the job.

/NOTE=*string*

Specifies a message string of up to 255 characters to appear on the flag page of the job.

/NOTIFY
/NONOTIFY (default)

Controls whether a message is broadcast to your terminal session when the job is printed or aborted.

/OPERATOR=*string*

Specifies a message of up to 255 characters to be sent to the operator when the job begins to print.

/PAGES=([*lowlim*,]*uplim*)

Positional qualifier.

Specifies the number of pages to print for the specified job. You can use the **/PAGES** qualifier to print portions of long files. By default, all pages of the file are printed.

The *lowlim* specifier refers to the first page in the group of pages that you want printed for that file. If you omit the *lowlim* specifier, the printing starts on the first page of the file.

The *uplim* specifier refers to the last page of the file that you want printed. If you want to print to the end of the file, but do not know how many pages the file contains, use quotation marks (" ") as the *uplim* specifier. You can omit the parentheses if you are including only a specific value for the *uplim* specifier. For example, **/PAGES=10** prints the first ten pages of the file; **/PAGES=(5,10)** prints pages 5 through 10; **/PAGES=(5,"")** starts printing at page 5 in the file and continues until the end of the file is reached.

/PARAMETERS= (*parameter*[(=*value*[,...])] [,...])

Specifies from 1 to 8 optional parameters that are unique to a particular print symbiont. Each parameter and its values are passed as a single text string directly to the print symbiont: logical names and DCL symbols are not translated by the **PRINT** command.

Each parameter can take from 1 to 8 values for a total of 255 characters. If you specify only one parameter or parameter value, you can omit the parentheses. Enclose parameter values containing lowercase letters or non-alphanumeric characters (such as spaces, parentheses, or commas) in quotation marks.

You can specify the **/PARAMETERS** qualifier only once in each **PRINT** command. For example, in the command **PRINT /PARAMETERS=(JJJ,KK) /PARAMETERS=(LL,MMM)**, the JJJ and KK are ignored because the second occurrence of the **/PARAMETERS** qualifier supersedes the first occurrence.

For information on parameters available for printing to PostScript printers using *DECprint Supervisor* software (DCPS), see the DCL Help topic PRINT_Parameter.

/PASSALL
/NOPASSALL (default)

Positional qualifier.

Specifies whether the symbiont bypasses all formatting and sends the output QIO to the driver with format suppressed. All qualifiers affecting formatting, as well as the **/HEADER**, **/PAGES**, and **/SETUP** qualifiers, are ignored.

If the **/PASSALL** qualifier is placed between the **PRINT** command and any file specifications, all files are printed in PASSALL mode. To specify the **/PASSALL** qualifier for only some files in the job, place the qualifier after each file that you want printed in PASSALL mode.

/PRIORITY=*n*

Requires OPER (operator) or ALTPRI (alter priority) privilege to raise the priority above the queue's maximum scheduling priority.

Specifies the job-scheduling priority of the print job. The value of the parameter *n* can be from 0 through 255, where 0 is the lowest priority and 255 is the highest. The default value of *n* is the value of the system parameter DEFQUEPRI. No privilege is needed to set the priority lower than the queue's maximum scheduling priority.

/QUEUE=*queue-name*[:]

Queues the job to the specified output queue. The default queue is SYS$PRINT. This qualifier is synonymous with the **/DEVICE** qualifier.

/REMOTE

Queues the job to SYS$PRINT on the remote node specified in the file specification; the file *must* exist on the remote node. When you use the **/REMOTE** qualifier, you *must* include the node name in the file specification.

You can specify only the following qualifiers with the **/REMOTE** qualifier:

/BACKUP

/BEFORE
/BY_OWNER
/CONFIRM
/CREATED
/EXCLUDE
/EXPIRED
/MODIFIED
/SINCE

Note that, unlike the printing on the local node, multiple files queued by a single **PRINT/REMOTE** command are considered separate jobs.

/RESTART (default)
/NORESTART

Indicates whether a job restarts after a system failure or after a **STOP/QUEUE/REQUEUE** command.

/RETAIN=*option*

Specifies the circumstances under which you want your jobs to be retained in a queue. When a job is retained in the queue, you can issue the **SHOW QUEUE** command after the job completes to see the status of the job. Without job retention, no record of a job is left in a queue after a job completes.

Use the following options to specify job retention:

- ALWAYS – Holds the job in the queue regardless of the job's completion status.
- DEFAULT – Holds the job in the queue as specified by the queue's retention policy.
- ERROR – Holds the job in the queue only if the job completes unsuccessfully.
- UNTIL=*time-value* – Holds the job in the queue for the specified length of time, regardless of the job's completion status.

Note

You cannot specify a **/NORETAIN** qualifier with the **PRINT** command (as system managers can with the commands **INITIALIZE/QUEUE**, **START/QUEUE**, and **SET QUEUE**); however, you can specify **/RETAIN=DEFAULT** with the **PRINT** command. The default option holds the job in the queue as specified by the queue's retention policy. If the system manager has not specified retention for the queue, the job is not retained.

How Job Retention Is Determined

Although you can specify job retention options for your own jobs, the job retention option you specify may be overridden by the job retention option of the queue on which your job executed. If you submit or print a job to a generic queue, the generic queue's job retention setting may also override the job retention option you specify. This section describes how job retention is determined.

An execution queue's job retention setting takes precedence over a generic queue's job retention setting; however, if the job's completion status does not match the job retention setting (if any) on the execution queue, then the generic queue's job retention setting attempts to control job retention. If the job's completion status does not match the job retention setting (if any) on the generic queue,

then the user-specified job retention setting is used. Jobs submitted directly to execution queues are not affected by job retention settings on generic queues.

If the execution queue's retention setting applies, the job is retained on the execution queue. Likewise, if the generic queue's retention setting applies, the job is retained on the generic queue. If the user-specified setting applies, the job is retained in the queue to which it was submitted.

The following example illustrates how the queue manager determines job retention:

Suppose you submit a job to a generic queue and specify **/RETAIN=ALWAYS**, and the job completes successfully.

First, the queue manager compares the job's completion status to the execution queue's retention setting. Suppose the queue is set with **/RETAIN=ERROR** (retains only jobs that complete unsuccessfully). The job is not retained in the execution queue because the error condition was not met.

The queue manager then compares the job's completion status to the generic queue's retention setting. Suppose the generic queue has no retention setting. The queue manager's comparison again fails to retain the job.

Finally, the queue manager compares the job's completion status to the retention setting you specified for the job. This comparison reveals that the job should be retained. Because the user-specified setting leads the queue manager to retain the job, the job is held in the queue to which the job was submitted – in this case, the generic queue.

For more information on types of queues, see the **INITIALIZE/QUEUE** command in the *VSI OpenVMS DCL Dictionary: A-M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_119]. For more information on setting retention options for queues, see the **START/QUEUE** or **SET QUEUE** command, or the **INITIALIZE/QUEUE** command in the *VSI OpenVMS DCL Dictionary: A-M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_119].

Timed Retention

Timed retention, which you specify using the **UNTIL=*time-value*** option, allows you to retain a job in the queue only as long as you need it. This eliminates the need to delete the job from the queue later.

For example, the following command retains the print job `MYFILE` in the queue until `7:31` on `December 14`, when the job will automatically be deleted from the queue.

```
$ PRINT/RETAIN=UNTIL=14-DEC-2001:07:31:0.0 MYFILE.DAT
```

However, depending on the queue's job retention policy, the job might be retained indefinitely. The job retention policy set on the queue takes precedence over the user-specified job retention setting. Because system managers cannot specify timed job retention for a queue, any jobs retained as a result of a queue's setting are retained indefinitely.

If you specify the **/RETAIN=UNTIL=*time-value*** option, you must supply a time value. The time value is first interpreted as a delta time, then as a combination time, and finally as absolute time. If you specify a delta time, the delta begins when the job completes. For example, if you specify **PRINT/RETAIN=UNTIL="+3:00"**, the job will be retained for three hours after the job completes. For information on specifying time values, see the relevant section in

the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT].

/SETUP=*module*[,...]

Extracts the specified modules from the device control library (containing escape sequence modules for programmable printers) and copies the modules to the printer before a file is printed. By default, no device control modules are copied.

Note that the module names are not checked for validity until the time that the file is actually printed; therefore, **PRINT/SETUP** is susceptible to typing errors and other mistakes. It is recommended only for experimental setups.

For production setups, see the **/SETUP** qualifier for the **DEFINE/FORM** command in the *VSI OpenVMS DCL Dictionary: A–M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_59].

/SINCE[=*time*]
/NOSINCE

Selects only those files dated on or after the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, JOB_LOGIN, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the **/SINCE** qualifier to indicate the time attribute to be used as the basis for selection: **/BACKUP**, **/CREATED** (default), **/EXPIRED**, or **/MODIFIED**.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/SPACE
/NOSPACE (default)

Positional qualifier.

Controls whether print job output is double-spaced. The default is single-spaced output.

/STYLE=*keyword*

Specifies the file name format for display purposes.

The valid keywords for this qualifier are CONDENSED and EXPANDED. Descriptions are as follows:

Keyword	Description
CONDENSED (default)	Displays the file name representation of what is generated to fit into a 255-length character string. This file name may contain a DID or FID abbreviation in the file specification.
EXPANDED	Displays the file name representation of what is stored on disk. This file name does not contain any DID or FID abbreviations.

The keywords CONDENSED and EXPANDED are mutually exclusive. This qualifier specifies which file name format is displayed in the output message, along with the confirmation if requested.

File errors are displayed with the CONDENSED file specification unless the EXPANDED keyword is specified.

See the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#FILE_ACCESS] for more information.

/SYMLINK=*keyword*

The valid keywords for this qualifier are [NO]WILDCARD and [NO]ELLIPSIS. Descriptions are as follows:

Keyword	Description
WILDCARD	Indicates that symlinks are enabled during wildcard searches.
NOWILDCARD	Indicates that symlinks are disabled during directory wildcard searches.
ELLIPSIS	Equivalent to WILDCARD (included for command symmetry).
NOELLIPSIS	Indicates that symlinks are matched for all wildcard fields except for ellipsis.

If the file named in the **PRINT** command is a symlink, the command operates on the symlink target.

/TRAILER[=*keyword*]
/NOTRAILER

Positional qualifier.

Controls whether a file trailer page is printed at the end of a file. The trailer page displays the job entry number as well as information about the user submitting the job and the files being printed. If the **/TRAILER** qualifier is positioned between the **PRINT** command and the file specifications, it can take either of the following keywords:

Keyword	Description
ALL	Prints a file trailer page after each file in the job.
ONE	Prints a file trailer page after the last file in the job.

If you want the **/TRAILER** qualifier to apply to individual files in a multi-file job, place the qualifier directly after each file that you want to have a trailer page.

Use the **/[NO]TRAILER** qualifier to override the **/DEFAULT=[NO]TRAILER** option that has been set for the output queue you are using. The **/[NO]TRAILER** qualifier does not override the **/SEPARATE=[NO]TRAILER** option set for the queue.

/USER=*username*

Requires CMKRNL (change mode to kernel) privilege and read (R) and write (W) access to the user authorization file (UAF).

Allows you to print a job on behalf of another user. The print job runs exactly as if that user had submitted it. The print job runs under that user's user name and user identification code (UIC), and accounting information is logged to that user's account. By default, the user identification comes

from the requesting process. The user name qualifier value can be any user name that is valid on your system. The name must be 1 to 12 alphanumeric characters.

Examples

1.
```
$ PRINT/QUEUE=LPB0/COPIES=10/AFTER=20   RESUME
  Job RESUME (queue LPB0, entry 239) holding until 14-DEC-2001 20:00
```

   The **PRINT** command in this example queues 10 copies of the file RESUME.LIS to printer LPB0, but requests that the copies not be printed until after 8:00 P.M.

2.
```
$ PRINT  THETA.TXT + BETA/FLAG + GAMMA/FLAG + *.LIS/FLAG
  Job THETA (queue SYS$PRINT, entry 237) pending
```

   The **PRINT** command in this example submits the files THETA.TXT, BETA.TXT, GAMMA.TXT, and the highest versions of all files with the file type .LIS as a single print job. Flag pages separate the individual files. Notice that the file type for BETA and GAMMA is .TXT, the file type of the first file in the list.

3.
```
$ PRINT/LOWERCASE   THETA.TXT/COPIES=2, -
_$ BETA.DOC/COPIES=3
Job THETA (queue SYS$PRINT, entry 240) pending
```

   The print job queued by the **PRINT** command in this example consists of two copies of THETA.TXT followed by three copies of BETA.DOC. This job must be printed on a printer that can print lowercase letters. If no such printer is available, the job waits in the queue.

4.
```
$ PRINT/JOB_COUNT=3   THETA.TXT,BETA/NOIDENTIFY
```

   The **PRINT** command in this example concatenates the files THETA.TXT and BETA.TXT into a single print job and prints three copies of the job. The **/NOIDENTIFY** qualifier requests that the job entry number and queue name not be displayed.

5.
```
$ COPY REPORT.MEM NODE3::
$ PRINT/REMOTE NODE3::REPORT.MEM
```

   In this example, the two commands are entered at a node other than NODE3. The **COPY** command copies the file REPORT.MEM from the current node to the NODE3 node. The **PRINT** command queues the file REPORT.MEM located on the NODE3 node for printing at the NODE3 node. The job entry number and queue name are not displayed when the **/REMOTE** qualifier is used.

6.
```
$ PRINT/HOLD   MASTER.DOC
  Job MASTER (queue SYS$PRINT, entry 540) holding
  .
  .
  .
$ SET ENTRY 540/RELEASE
```

   The **PRINT** command in this example queues a copy of the file MASTER.DOC to the default printer in a hold status. Later, the **SET ENTRY** command releases the hold status on the file and makes it available for printing.

7.
```
$ PRINT TEST.TXT
Job TEST.TXT (queue SYS$PRINT, entry 867) started on LPA0
$ SHOW ENTRY '$ENTRY'
Entry     Jobname     Username     Blocks    Status
```

```
-----     -------     --------     ------    ------
  867     TEST.TXT    Nisslert        135    Printing
          Submitted  10-DEC-2001 11:07 /FORM=DEFAULT /PRIORITY=100
          File: $5$DUA174:[NISSLERT]TEST.TXT;3
```

The **PRINT** command in this system example queues a copy of the file TEST.TXT to the default printer. It also creates the local symbol $ENTRY to store the job entry number. The **SHOW ENTRY** command requests a display of the entry using the symbol $ENTRY to identify it.

8.
```
$ PRINT/RETAIN=UNTIL=1 MYFILE.DAT
   .
   .
   .
$ SHOW QUEUE DOC$LN03
Server queue DOC$LN03, stopped, on NEWTON::, mounted form DEFAULT
Entry    Jobname    Username  Blocks  Status
-----    -------    --------  ------  ------
  436    MYFILE     MIGDAL         8  Retained until 11-DEC-2001 16:56
      %JBC-F-JOBABORT, job aborted during execution
        Completed  11-DEC-2001 15:56 on queue DOC$LN03
```

The **SHOW QUEUE** display in this system example includes the date and time at which a retained job completed and the queue on which it executed. The user set job retention to be one hour after job completion. Depending on the queue's job retention policy, the job might be deleted from the queue at 16:56.

9.
```
$ PRINT /PARAMETERS=AAA=123
```

This command passes the parameter AAA=123 to the print symbiont. The interpretation of the parameter is dependent on the symbiont.

10.
```
$ PRINT /PARAMETERS=(ddd,"e,(F=(3,4),g),h)",iii)
```

This command passes three parameters to the print symbiont: DDD, "e,(F=(3,4),g),h)", and III. The use of the quotes around the second parameter allows the use of nested parentheses, commas, and the preservation of alphabetic case. This parameter value is perfectly acceptable to the **PRINT** command, even though it is apt to subsequently be rejected by the print symbiont.

11.
```
$ PRINT /PARAMETERS=("N,O,P,Q,R,S,T",U,V)
```

This command passes three parameters to the print symbiont: "N,O,P,Q,R,S,T", U, and V. Depending on the parsing capabilities of the symbiont and the length of the parameters (each parameter is limited to 255 characters), this may be a workaround to the limitation of eight parameters.

PRODUCT

PRODUCT — Invokes the POLYCENTER Software Installation utility to install one or more software products, remove one or more software products, or perform other related operations. For a complete description of the POLYCENTER Software Installation utility and the use of the **PRODUCT** command, see the relevant section in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#_6017INST_UPG] and the *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/#d0e9706] or online help.

Format

PRODUCT [*subcommand product-name* [/*qualifiers*]]

PURGE

PURGE — Deletes all but the highest numbered versions of the specified files.

Format

PURGE [*filespec*[,...]]

Parameters

***filespec*[,...]**

Specifies one or more files to be purged. If you specify more than one file, separate the file specifications with either commas (,) or plus signs (+). The asterisk (*) and the percent sign (%) wildcard characters are allowed in the directory, file name, and file type fields; however, no version number can be specified. As a default, the **PURGE** command purges all files in the current directory. There are no file name or file type defaults with the **PURGE** command. If an input file specification is a symbolic link, the symbolic link itself is purged. Because only one version of a symbolic link can exist, this command has no effect on that file.

Description

The **PURGE** command deletes earlier versions of files. The **PURGE** command never deletes all versions of any file. By default, the **PURGE** command keeps only the highest version of a file. If you do not include a file specification with the **PURGE** command, all files in the current directory are affected by the purge.

Qualifiers

/BACKUP

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/BACKUP** qualifier selects files according to the dates of their most recent backups. This qualifier is incompatible with the **/CREATED**, **/EXPIRED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/BEFORE[=*time*]

Selects only those files dated prior to the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the **/BEFORE** qualifier to indicate the time attribute to be used as the basis for selection: **/BACKUP**, **/CREATED** (default), **/EXPIRED**, or **/MODIFIED**.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/BY_OWNER[=*uic*]

Selects only those files whose owner user identification code (UIC) matches the specified owner UIC. The default UIC is that of the current process.

/CONFIRM
/NOCONFIRM (default)

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

YES	NO	QUIT
TRUE	FALSE	**Ctrl/Z**
1	0	ALL
	Return	

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), but these abbreviations must be unique. Affirmative answers are YES, TRUE, and 1. Negative answers include: NO, FALSE, 0, and pressing **Return**. Entering QUIT or pressing **Ctrl/Z** indicates that you want to stop processing the command at that point. When you respond by entering ALL, the command continues to process, but no further prompts are given. If you type a response other than one of those in the list, DCL issues an error message and redisplays the prompt.

/CREATED (default)

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/CREATED** qualifier selects files based on their dates of creation. This qualifier is incompatible with the **/BACKUP**, **/EXPIRED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/ERASE
/NOERASE (default)

Erases the specified files from the disk so that the purged data no longer exists physically on the deallocated disk blocks.

When you delete a file, the area in which the file was stored is returned to the system for future use. The data that was stored in that location still exists in the system until new data is written over it. When the **/ERASE** qualifier is specified, the storage location is overwritten with a system-specified pattern so that the data no longer exists.

/EXCLUDE=(*filespec*[,...])

Excludes the specified files from the purge operation. You can include a directory but not a device in the file specification. The asterisk (*) and the percent sign (%) wildcard characters are allowed in the file specification; however, you cannot use relative version numbers to exclude a specific version. If you specify only one file, you can omit the parentheses.

/EXPIRED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/EXPIRED** qualifier selects files according to their expiration dates. (The expiration date is set with the **SET FILE/EXPIRATION_DATE** command.) The **/EXPIRED** qualifier is incompatible with the **/BACKUP**, **/CREATED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/GRAND_TOTAL (Alpha/IA-64 only)

Displays the total number of files and blocks or bytes deleted. The display is shown as blocks or bytes depending on the current default setting. You can use **SHOW PROCESS/UNITS** to display the current default. To change the default, execute the DCL command **SET PROCESS/UNITS=BYTES** or **SET PROCESS/UNITS=BLOCKS**.

/IGNORE=INTERLOCK (Alpha/IA-64 only)

Allows you to mark a write-accessed file for deletion. This removes the file name entry, and the file is deleted when it is closed by the final user.

/KEEP=*number-of-versions*

Specifies the maximum number of versions of the specified files to be retained in the directory. If you do not include the **/KEEP** qualifier, all but the highest numbered version of the specified files are deleted from the directory.

/LOG
/NOLOG (default)

Controls whether file specifications are displayed as the files are deleted.

/MODIFIED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/MODIFIED** qualifier selects files according to the dates on which they were last modified. This qualifier is incompatible with the **/BACKUP**, **/CREATED**, and **/EXPIRED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time modifiers, the default is the **/CREATED** qualifier.

/SINCE[=*time*]

Selects only those files dated on or after the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, JOB_LOGIN, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the **/SINCE** qualifier to indicate the time attribute to be used as the basis for selection: **/BACKUP**, **/CREATED** (default), **/EXPIRED**, or **/MODIFIED**.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/STYLE=*keyword*

Specifies the file name format for display purposes while purging files.

The valid keywords for this qualifier are CONDENSED and EXPANDED. Descriptions are as follows:

Keyword	Description
CONDENSED (default)	Displays the file name representation of what is generated to fit into a 255-length character string. This file name may contain a DID or FID abbreviation in the file specification.
EXPANDED	Displays the file name representation of what is stored on disk. This file name does not contain any DID or FID abbreviations.

The keywords CONDENSED and EXPANDED are mutually exclusive. This qualifier specifies which file name format is displayed in the output message, along with the confirmation if requested.

File errors are displayed with the CONDENSED file specification unless the EXPANDED keyword is specified.

See the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#FILE_ACCESS] for more information.

Examples

1. `$ PURGE`

   The **PURGE** command in this example deletes all but the highest numbered version of all files in the default directory.

2. `$ PURGE *.COM`

   The **PURGE** command in this example deletes all but the highest numbered version of each file with a file type of `.COM`.

3. `$ PURGE/KEEP=3  [WILDER.JOB308]ACCOUNT.COB`

   The **PURGE** command in this example deletes all but the three highest numbered versions of the file `ACCOUNT.COB` in the subdirectory `[WILDER.JOB308]`.

4. `$ PURGE/ERASE/SINCE=YESTERDAY [.MEMOS]`

   The **PURGE** command in this example purges all files in the `MEMOS` subdirectory that have been created or modified since yesterday and erases the storage locations so that the purged data no longer exists.

5. ```
   $ PURGE [BROD.TESTFILES]/LOG
   %PURGE-I-FILPURG, DISK1:[BROD.TESTFILES]AVE.OBJ;1 deleted (3 blocks)
   %PURGE-I-FILPURG, DISK1:[BROD.TESTFILES]BACK.OBJ;2 deleted (5 blocks)
   %PURGE-I-TOTAL, 2 files deleted (8 blocks)
   ```

   The **PURGE** command in this example purges all files cataloged in the subdirectory named `[BROD.TESTFILES]`. The **/LOG** qualifier requests the **PURGE** command to display the specification of each file it has deleted as well as the total number of files that have been deleted.

6. ```
   $ PURGE/GRAND_TOTAL STATUS.TXT
   %PURGE-I-TOTAL, 6 files deleted (600KB)
   ```

The output display in this example shows that 6 files were deleted for a total of 600KB. The process is currently set to display file sizes in bytes. To change future displays to show blocks, use the **SET PROCESS/UNITS=BLOCKS** command.

7. `$ PURGE/KEEP=2 TAMPA::DISK1:[EXAMPLE]*.LIS`

   The **PURGE** command in this example deletes all but the two highest numbered versions of each file with the file type .LIS in the directory EXAMPLE on remote node TAMPA.

READ

READ — Reads a single record from a specified input file and assigns the record's contents to a specified symbol name.

Format

READ ***logical-name*****[:]** ***symbol-name***

Parameters

logical-name**[:]**

Specifies the logical name of the input file from which a record is to be read. Use the logical name assigned by the **OPEN** command when the file was opened. (The **OPEN** command assigns a logical name to a file and places the name in the process logical name table.)

In addition, you can specify the process-permanent files identified by the logical names SYS$INPUT, SYS$OUTPUT, SYS$ERROR, and SYS$COMMAND.

symbol-name

Specifies the name of a symbol to be equated to the contents of the record. The name must be 1 to 255 alphanumeric characters and must start with an alphabetic letter, an underscore (_), or a dollar sign ($).

When you specify a symbol name for the **READ** command, the command interpreter places the symbol name in the local symbol table for the current command level. If the symbol has already been defined, the READ command redefines it to the new value being read.

Description

The **READ** command can read data from sequential, relative, or indexed files. After each record is read from the specified file, the **READ** command positions the record pointer at the next record in the file; however, if you are reading an indexed file, you can use the **/INDEX** and **/KEY** qualifiers to read records randomly.

The maximum size of any record that can be read in a single **READ** command is 2048 bytes. The reading of a 2048-byte record from a remote file opened by DCL requires that the system default network block count be increased to a minimum value of 25 (DCL **SET RMS_DEFAULT/NETWORK=25**).

To read a file, the file must be opened by using the **/READ** qualifier with the **OPEN** command. The process-permanent files identified by the logical names SYS$INPUT, SYS$OUTPUT, SYS$ERROR, and SYS$COMMAND do not have to be opened explicitly to be read.

If the **READ** command is executed interactively and the logical name is specified as one of the process-permanent files, SYS$INPUT, SYS$OUTPUT, SYS$COMMAND, or SYS$ERROR, the command interpreter prompts for input data. The **READ** command accepts data exactly as you enter it. The **READ** command does not convert characters to uppercase, remove extra spaces and tabs, or remove quotation marks (" "). Also, the READ command does not perform symbol substitution. See the **/PROMPT** qualifier for more information on issuing prompts with the **READ** command.

Qualifiers

/DELETE

Deletes a record from an indexed file after it has been read. An indexed file must be opened with the **/READ** and **/WRITE** qualifiers in order to use the **READ/DELETE** command.

/END_OF_FILE=*label*

Transfers control to the location specified by the *label* keyword (in the current command procedure) when the end of the file is reached. When the last record in the file is read, the OpenVMS Record Management Services (RMS) returns an error condition indicating the end-of-file (EOF). If the **/END_OF_FILE** qualifier is specified, the command interpreter transfers control to the command line at the specified label.

If the **/END_OF_FILE** qualifier is not specified, control is given to the error label specified with the **/ERROR** qualifier when the end of the file is reached. If neither the **/ERROR** nor the **/END_OF_FILE** qualifier is specified, then the current ON condition action is taken.

/ERROR=*label*

Transfers control to the location specified by the *label* keyword (in the current command procedure) when a read error occurs. If no error routine is specified and an error occurs during the reading of the file, the current ON condition action is taken.

Overrides any ON condition action specified.

If an error occurs and the target label is successfully given control, the reserved global symbol $STATUS retains the error code.

/INDEX=*n*

Specifies the index (*n*) to be used to look up keys when reading an indexed file.

If you do not specify the **/INDEX** qualifier, the most recent **/INDEX** qualifier value is used. If a previous value was not specified, the primary index is used (**/INDEX=0**).

/KEY=*string*

Reads a record with the key that matches the specified character string. Binary and integer keys are not allowed. This qualifier, when used together with the **/INDEX** qualifier, allows you random access to indexed files.

Key matches are made by comparing the characters in the **/KEY** string to characters in the record key.

To read records at random in an indexed file, you must specify the **/KEY** qualifier. Once a record is read randomly, all subsequent reads without the **/KEY** qualifier access records in the indexed file sequentially.

/MATCH=*option*

Specifies the key match algorithm to be used when searching for matching keys. Specify one of the following options:

Option	Description
EQ	Selects keys equal to the match value (default).
GE	Selects keys greater than or equal to the match value.
GT	Selects keys greater than the match value.
LE	Selects keys less than or equal to the match value.
LT	Selects keys less than the match value.

If you are reading indexed files and you do not use the **/MATCH** qualifier, the default is **/MATCH=EQ**.

/NOLOCK

Specifies that the record to be read not be locked and enables a record to be read that has been locked by other accessors.

By default, records are locked as they are read and unlocked on the next I/O operation on the file.

/PROMPT=*string*

Specifies an alternate prompt string to be displayed when reading from the terminal. The default prompt string is DATA:.

/TIME_OUT=*n*
/NOTIME_OUT (default)

Specifies the number of seconds after which the **READ** command is terminated if no input is received. If you enter the **/TIME_OUT** qualifier, you must specify a value from 0 to 255.

If you enter both the **/ERROR=*label*** and **/TIME_OUT** qualifiers, and the time limit expires, the error branch is taken.

/WAIT

Sets RAB$V_WAT to make a process wait for a record in a file. Can be used in combination with **/TIME_OUT** to restrict how long the process should wait before timing out upon failure to find the record.

Examples

1.
```
$ OPEN IN NAMES.DAT
$ LOOP:
$ READ/END_OF_FILE=ENDIT IN NAME
    .
    .
    .
$ GOTO LOOP
$ ENDIT:
$ CLOSE IN
```

The **OPEN** command opens the file NAMES.DAT for input and assigns it the logical name of IN. The **READ** command reads records from the file IN and places the contents into the symbol NAME. The **READ** command specifies the label ENDIT to receive control when the last record in the file has been read. The procedure loops until all records in the file have been processed.

2.
```
$ READ/ERROR=READERR/END_OF_FILE=OKAY  MSGFILE  CODE
   .
   .
   .
$ READERR:
$ CLOSE MSGFILE
   .
   .
   .
$ OKAY:
$ CLOSE MSGFILE
$ EXIT
```

The **READ** command reads records from the file MSGFILE and places the contents into the symbol CODE. The **READ** command also uses the **/ERROR** and **/END_OF_FILE** qualifiers to specify labels to receive control at the end-of-file (EOF) and on error conditions. At the EOF, control is transferred to the label OKAY. On other read errors, control is transferred to the READERR label.

3.
```
$ READ SYS$COMMAND  DATA_LINE
$ WRITE  OUTPUT_FILE  DATA_LINE
   .
   .
   .
```

The **READ** command requests data from the current SYS$COMMAND device. If the command procedure containing these lines is executed interactively, the command issues a prompt to the terminal, accepts a line of data, and equates the data entered to the symbol name DATA_LINE.

Then the **WRITE** command writes the value of the symbol DATA_LINE to the file identified by the logical name OUTPUT_FILE.

4.
```
$ OPEN/READ INPUT_FILE TRNTO::INVENTORY.DAT
$ OPEN/APPEND OUTPUT_FILE RECEIVE.DAT
$ READ INPUT_FILE DATA_LINE
$ WRITE OUTPUT_FILE DATA_LINE
```

The **OPEN/READ** command opens the file INVENTORY.DAT at the remote node TRNTO for reading and assigns it the logical name INPUT_FILE. The **OPEN/APPEND** command opens the file RECEIVE.DAT in the current default directory. The **READ** command requests data from the file INVENTORY.DAT at the remote node TRNTO. The **WRITE** command writes the value of the symbol DATA_LINE to the end of the local file RECEIVE.DAT.

RECALL

RECALL — Displays up to 254 previously entered commands on the screen for subsequent execution.

Format

RECALL [*command-specifier*]

Parameter

command-specifier

Specifies the number or the first several characters of the command you want to recall. You cannot use any of the qualifiers in conjunction with a command-specifier.

The specified characters should be unique. If they are not unique, the **RECALL** command displays the most recently entered command line that matches those characters. For example, suppose you enter a **SHOW STATUS** command and later enter a **SHOW TIME** command. If you then type **RECALL S**, the **SHOW TIME** command is recalled. You must type **RECALL SHOW S** to recall the **SHOW STATUS** command. The number of the command can be from 1 to 254 (where 1 is the last command entered).

The **RECALL** command itself is never assigned a number. If no command specifier is entered, the **RECALL** command recalls the most recently entered command. You can use the **/ALL** qualifier to have the system display all the commands in the recall buffer, along with their command numbers, to verify the number of the command you want to recall.

Description

When you enter commands to the system, they are stored in a recall buffer for later use with the **RECALL** command. Input to the **INQUIRE** command in command procedures is also placed in the recall buffer. The **RECALL** command itself is never stored in the recall buffer. You can use continuation characters with the commands to be recalled, but only 255 characters can be read at a time.

The recall buffer can hold up to 254 commands. The size of the recall buffer is 4 KB, which will store approximately 80 commands at 48 characters each.

If the length of stored commands makes it necessary, one or more of the oldest commands will be removed from the recall buffer to make room for the most recent command.

When you use the **RECALL** command, the system displays the command but does not process it. If you want it 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.

Qualifiers

/ALL [*command-specifier*]

Displays all the commands (and their numbers) available for recall. Do not use the **/ALL** qualifier with the other **RECALL** qualifiers.

To display multiple occurrences of a specific command in the recall buffer, specify the command after the **/ALL** qualifier.

Note

The **RECALL** command is never stored in the recall buffer.

/ERASE

Erases the contents of the recall buffer.

/INPUT=*filespec*

Causes each line of the input file to be added to the recall buffer. If a file type is not specified with the file name, .LIS is the default. The file specification cannot include the asterisk (*) and the percent sign (%) wildcard characters.

If the **/ERASE** qualifier is also specified, the contents of the recall buffer are erased before the contents of the specified file are read into the recall buffer.

The **/INPUT** qualifier is incompatible with the **/ALL** and **/PAGE** qualifiers.

/OUTPUT=*filespec*

Specifies the name of the output file where the contents of the recall buffer are written. If you do not specify a file type, the default is .LIS. The file specification cannot include the asterisk (*) and the percent sign (%) wildcard characters.

If the **/ERASE** qualifier is also specified, the contents of the recall buffer are erased after the contents of the recall buffer are written to the specified file.

The **/OUTPUT** qualifier is incompatible with the **/ALL** and **/PAGE** qualifiers.

/PAGE
/NOPAGE (default)

Displays all the commands (and their numbers) available for recall one screen at a time. The **/NOPAGE** qualifier produces the same output as the **/ALL** qualifier. You can cancel the output display by pressing **Ctrl/Z**.

The **/PAGE** qualifier is incompatible with the **/INPUT** and **/OUTPUT** qualifiers.

/SEARCH *string*

Searches the recall buffer and displays all the commands (and their numbers) that contain the specified search string.

Examples

1.
```
$ RECALL T
```

The **RECALL** command in this example recalls the last command entered that begins with the letter T.

2.
```
$ SHOW DEFAULT
  DISK3:[SMITH]
$ DIRECTORY SEPT*
  %DIRECT-W-NOFILES, no files found
$ SET DEFAULT [SMITH.LETTERS]
$ RECALL/ALL
  1 SET DEFAULT [SMITH.LETTERS]
  2 DIRECTORY SEPT*
  3 SHOW DEFAULT
$ RECALL 2
$ DIRECTORY SEPT*
```

```
  %DIRECT-W-NOFILES, no files found
$ RECALL 2
$ SET DEFAULT [SMITH.LETTERS]
  <edit command line>
$ SET DEFAULT [SMITH.MEMOS]
$ RECALL 2
$ DIRECTORY SEPT*
```

This example starts with a **SHOW DEFAULT** and a **DIRECTORY** command. After not finding the file you want, enter the **SET DEFAULT** command to move to the LETTERS subdirectory. Next use the **RECALL/ALL** command to see the list of commands you have entered. Enter the **RECALL 2** command to repeat the **DIRECTORY** command in the LETTERS subdirectory. Because you still have not found the file you want, enter the **RECALL 2** command again to recall the **SET DEFAULT** command. (With the entry of the latest **DIRECTORY** command, **SET DEFAULT** becomes command 2 in the **RECALL** list.) Edit the command line so that the system sets the default to the MEMOS subdirectory. Finally, recall the **DIRECTORY** command to try once more to find the file.

3.
```
$ TYPE COMMANDS.LIS
DIRECTORY
RUN AUTHORIZE
SET PROCESS/PRIVILEGES=ALL
$ RECALL/INPUT=COMMANDS.LIS
$ RECALL/ALL
  1 SET PROCESS/PRIVILEGES=ALL
  2 RUN AUTHORIZE
  3 DIRECTORY
  4 TYPE COMMANDS.LIS
```

   This example shows the contents of the COMMANDS.LIS file with the **TYPE** command, then uses the **RECALL/INPUT=COMMANDS.LIS** command to place the commands into the recall buffer. Issuing the **RECALL/ALL** command subsequently shows the numbered display of commands in the recall buffer.

4.
```
$ RECALL/OUTPUT=NEW_COMMANDS.LIS
$ TYPE NEW_COMMANDS.LIS
TYPE COMMANDS.LIS
DIRECTORY
RUN AUTHORIZE
SET PROCESS/PRIVILEGES=ALL
```

   This example places the contents of the recall buffer (see the previous example) into the NEW_COMMANDS.LIS file. Note that they are not numbered and in the reverse sequence of the **RECALL/ALL** command output.

5.
```
$ RECALL/ALL SHOW
  1 show system
  4 show user peleg
  5 show intrusion
  6 show error
  7 show time
```

   This example displays all occurrences of the **SHOW** command in the recall buffer.

6.
```
$ RECALL/SEARCH devi
  1 dir sys$sysdevice:[000000]
  4 show device dk
```

This example displays all commands in the recall buffer that contain the string "devi".

RENAME

RENAME — Changes all or part of a file specification of an existing disk file or disk directory.

Format

RENAME *input-filespec*[,...] *output-filespec*

Parameters

***input-filespec*[,...]**

Specifies the name of one or more files whose specifications are to be changed. The asterisk (*) and the percent sign (%) wildcard characters are allowed in the directory specification, file name, file type, or version number fields of the file specification. When wildcard characters are used, all files whose specifications satisfy the wildcard fields are renamed. If an input file specification is a symbolic link, the symbolic link itself is renamed. If the output file specification is a symbolic link, the operation fails.

output-filespec

Provides the new file specification to be applied to the input file. The **RENAME** command uses the device, directory, file name, and file type of the input file as defaults for fields in the output file that are either unspecified, or are indicated by the asterisk (*) and the percent sign (%) wildcard characters. Wildcard characters in corresponding fields of the input and output file specification result in multiple rename operations.

The **RENAME** command supplies output file version numbers in the following ways:

- If the output file specification contains an explicit version number, that version number is used.
- If the output file specification contains an asterisk (*) or a percent sign (%) wildcard character as the version number, the version number of the input file is used.
- If the input file specification contains an asterisk (*) or a percent sign (%) wildcard character as the version number, the version number of each input file names a corresponding output file.
- If no file exists with the same file name and type as the output file, the new file is assigned a version number of 1.
- If a file already exists with the same file name and type as the output file, the next higher version number is used (unless the **/NONEWVERSION** qualifier is specified).

Description

The **RENAME** command changes the directory name, file name, file type, or version number of a file. The node and disk designation for the input file specification must be the same as that for the output file specification. In addition, you must have delete (D) access privileges to a file to rename the file.

You cannot rename a file across a network.

Qualifiers

/BACKUP

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/BACKUP** qualifier selects files according to the dates of their most recent backups. This qualifier is incompatible with the **/CREATED**, **/EXPIRED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/BEFORE[=*time*]

Selects only those files dated prior to the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the **/BEFORE** qualifier to indicate the time attribute to be used as the basis for selection: **/BACKUP**, **/CREATED** (default), **/EXPIRED**, or **/MODIFIED**.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/BY_OWNER[=*uic*]

Selects only those files whose owner user identification code (UIC) matches the specified owner UIC. The default UIC is that of the current process.

Specify the UIC by using standard UIC format as described in the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#UIC_SECTION].

/CONFIRM
/NOCONFIRM (default)

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

YES	NO	QUIT
TRUE	FALSE	**Ctrl/Z**
1	0	ALL
	Return	

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), but these abbreviations must be unique. Affirmative answers are YES, TRUE, and 1. Negative answers include: NO, FALSE, 0, and pressing **Return**. Entering QUIT or pressing **Ctrl/Z** indicates that you want to stop processing the command at that point. When you respond by entering ALL, the command continues to process, but no further prompts are given. If you type a response other than one of those in the list, DCL issues an error message and redisplays the prompt.

/CREATED (default)

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/CREATED** qualifier selects files based on their dates of creation. This qualifier is incompatible with the

/BACKUP, **/EXPIRED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/EXCLUDE=(*filespec*[,...])

Excludes the specified files from the rename operation. You can include a directory but not a device in the file specification. The asterisk (*) and the percent sign (%) wildcard characters are allowed in the file specification; however, you cannot use relative version numbers to exclude a specific version. If you specify only one file, you can omit the parentheses.

/EXPIRED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/EXPIRED** qualifier selects files according to their expiration dates. (The expiration date is set with the **SET FILE/EXPIRATION_DATE** command.) This qualifier is incompatible with the **/BACKUP**, **/CREATED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/INHERIT_SECURITY
/NOINHERIT_SECURITY (default)

Controls whether the security profile (UIC, protection code, and ACL) of the renamed file is altered. If you specify **/INHERIT_SECURITY**, a new security profile is generated, as if a new file had been created with the new file and directory names. See the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#FILE_TEMPLATE] or online help (Hints topic) for a description of how a profile is assigned.

/LOG
/NOLOG (default)

Displays the file specification of each file as it is renamed.

/MODIFIED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/MODIFIED** qualifier selects files according to the dates on which they were last modified. This qualifier is incompatible with the **/BACKUP**, **/CREATED**, and **/EXPIRED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time modifiers, the default is the **/CREATED** qualifier.

/NEW_VERSION (default)
/NONEW_VERSION

Assigns a new version number if an output file specification is the same as that of an existing file. The **/NONEW_VERSION** qualifier displays an error message if an output file specification is the same as that of an existing file. The asterisk (*) and the percent sign (%) wildcard characters appearing in the version field of an input or output file overrides these qualifiers.

/SINCE[=*time*]

Selects only those files dated on or after the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, JOB_LOGIN, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the **/SINCE** qualifier to indicate the time attribute to be used as the basis for selection: **/BACKUP**, **/CREATED** (default), **/EXPIRED**, or **/MODIFIED.**

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/STYLE=*keyword*

Specifies the file name format for display purposes.

The valid keywords for this qualifier are CONDENSED and EXPANDED. Descriptions are as follows:

Keyword	Description
CONDENSED (default)	Displays the file name representation of what is generated to fit into a 255-length character string. This file name may contain a DID or FID abbreviation in the file specification.
EXPANDED	Displays the file name representation of what is stored on disk. This file name does not contain any DID or FID abbreviations.

The keywords CONDENSED and EXPANDED are mutually exclusive. This qualifier specifies which file name format is displayed in the output message, along with the confirmation if requested.

File errors are displayed with the CONDENSED file specification unless the EXPANDED keyword is specified.

See the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#FILE_ACCESS] for more information.

/SYMLINK=*keyword*

The valid keywords for this qualifier are [NO]WILDCARD and [NO]ELLIPSIS. Descriptions are as follows:

Keyword	Description
WILDCARD	Indicates that symlinks are enabled during wildcard searches.
NOWILDCARD	Indicates that symlinks are disabled during directory wildcard searches.
ELLIPSIS	Equivalent to WILDCARD (included for command symmetry).
NOELLIPSIS	Indicates that symlinks are matched for all wildcard fields except for ellipsis.

If the file named in the **RENAME** command is a symlink, the command operates on the symlink itself.

Examples

1. `$ RENAME  AVERAGE.OBJ  OLDAVERAGE`

   The **RENAME** command in this example renames the highest existing version of the file `AVERAGE.OBJ` to `OLDAVERAGE.OBJ`. If no file named `OLDAVERAGE.OBJ` currently exists, the new file is assigned a version number 1.

2. `$ RENAME/NONEW_VERSION  SCANLINE.OBJ;2  BACKUP.OBJ`

The **RENAME** command in this example renames the file SCANLINE.OBJ;2 to BACKUP.OBJ;2. The **/NONEW_VERSION** qualifier ensures that, if BACKUP.OBJ;2 already exists, the **RENAME** command does not rename the file, but instead reports the error.

3. `$ RENAME  *.TXT;*  *.OLD;*`

   The **RENAME** command in this example renames all versions of all files with the file type TXT to have the file type OLD. The file names and version numbers are not changed.

4. `$ RENAME WATER.TXT [.MEMOS]`

   The **RENAME** command in this example changes the directory name of WATER.TXT from your default directory to the MEMOS subdirectory. (The **RENAME** command moves the file to another directory.)

5. `$ RENAME  [BORDERS.TESTFILES]SAVE.DAT  []TEST`

   The **RENAME** command in this example renames the file SAVE.DAT in the directory BORDERS.TESTFILES to TEST.DAT. The new file is moved to the current default directory.

6. `$ RENAME COEISABLAST.TXT COEIsABlast.txt`

   The **RENAME** command in this example changes the case of COEISABLAST.TXT to COEIsABlast.txt. You must be on an ODS-5 disk and previously have used the **SET PROCESS/CASE_LOOKUP=SENSITIVE** command for this **RENAME** command to work properly.

7. ```
   $ RENAME/LOG
   _From:      DATA.*,INFO.*
   _To:        NEW
   %RENAME-I-RENAMED, _DISK0:[SYSTEM]DATA.AAA;1 renamed to _DISK0:[SYSTEM]NEW.AAA;1
   %RENAME-I-RENAMED, _DISK0:[SYSTEM]DATA.BBB;1 renamed to _DISK0:[SYSTEM]NEW.BBB;1
   %RENAME-I-RENAMED, _DISK0:[SYSTEM]DATA.CCC;1 renamed to _DISK0:[SYSTEM]NEW.CCC;1
   %RENAME-I-RENAMED, _DISK0:[SYSTEM]INFO.001;1 renamed to _DISK0:[SYSTEM]NEW.001;1
   %RENAME-I-RENAMED, _DISK0:[SYSTEM]INFO.002;1 renamed to _DISK0:[SYSTEM]NEW.002;1
   %RENAME-I-RENAMED, _DISK0:[SYSTEM]INFO.003;1 renamed to _DISK0:[SYSTEM]NEW.003;1
   $
   ```

   In this example, three files exist with the file name of DATA, and three files have the file name of INFO. This **RENAME** command illustrates the asterisk (*) wildcard character in the input file names and the use of temporary default file types and version numbers on the output files. The result is the renaming of all six files as displayed by the **/LOG** qualifier.

8. `$ RENAME NODE1::DISK2:[SMITH]ASSEMSHT.EXE NODE1::DISK3:`
   `[JONES]ASSEMBLYSHEET.EXE`

   The **RENAME** command in this example renames the file ASSEMSHT.EXE in the SMITH directory on remote node NODE1 and disk DISK3 to ASSEMBLYSHEET.EXE in the JONES directory on the same remote node and disk. You can rename a file on another node and disk only if the new file resides on that same node and disk.

REPLY

REPLY — Broadcasts a message to a terminal or terminals. See the qualifier descriptions for restrictions.
   ```

Format

REPLY [*message-text*]

Parameter

message-text

Specifies the text of the message. The text must be 1 to 511 characters. Enclose the text in quotation marks (" ") if it contains spaces, special characters, or lowercase characters.

Description

All users with OPER (operator) privilege can use the **REPLY** command to communicate with system users. The **REPLY** command does the following:

- Displays messages at users' terminals
- Responds to user requests
- Responds to magnetic tape file system requests
- Enables and disables operator status on a terminal (if the operator communication process [OPCOM] is running)
- Closes the operator's log file and opens a new one (if OPCOM is running)

You must always use one or more qualifiers with the **REPLY** command in order for it to be meaningful. If you use the **REPLY** command without using any qualifiers, an error message is returned. When you use the **REPLY** command for any purpose other than displaying messages at users' terminals, you must also use the **/ENABLE=*keyword*** qualifier. See the description of the **/ENABLE** qualifier to determine the appropriate keyword (or keywords) for your purpose.

Displaying Messages at Users' Terminals

To contact one or more system users, the operator enters one of the following **REPLY** commands:

- **REPLY/ALL *message-text***
- **REPLY/TERMINAL=(*terminal-name*[,...]) *message-text***
- **REPLY/USERNAME[=(*username*[,...])] *message-text***

The **/ALL** qualifier sends a message to all terminals that are on line and are connected to the OpenVMS system or OpenVMS Cluster. Generally, when an important message is to be broadcast, such as information about a system shutdown, you should use the **/ALL** qualifier.

The **/TERMINAL** qualifier sends a message to one or more specific terminals on the system or cluster.

The **/USERNAME** qualifier sends a message to terminals at which one or more system or cluster users are logged in.

Note that the **/TO** qualifier is *not* used under these three circumstances, because the operator is not replying to a specific request from either the file system or a user.

To broadcast to a terminal other than your own, you must have OPER privilege. The **REPLY** command is not complete until all terminals you are broadcasting to have received the message.

Responding to User Requests

When a user enters the **REQUEST/REPLY** command, the process associated with the requesting user's terminal is put in a wait state until the operator responds by using one of the following **REPLY** commands:

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

The **/ABORT** qualifier indicates that the user's request has been canceled.

The **/PENDING** qualifier sends a message to the user and keeps the user's process in a wait state until the request can be fulfilled or aborted.

The **/TO** qualifier indicates that the user's request has been fulfilled.

When a user enters the **REQUEST/REPLY** command, the message is displayed at the system console terminal. For example:

```
%OPCOM, 24-DEC-2017 09:49:24.47, request 3, from user SYSTEM
_TTB6:, This is a sample request
```

The user cannot enter any further commands until the operator responds using the **/ABORT** or the **/TO** qualifier, or until the user aborts the request. If the operator does not respond and the user does not abort the request, the request is repeated at 5-minute intervals on the operator's terminal until the operator replies.

The **REPLY** command is an essential part of the procedures that operators must use in order for users to gain access to tape and disk volumes.

Responding to File System Requests

When a multi-volume tape volume reaches the end-of-tape (EOT) mark, the magnetic tape file system suspends processing and sends a message to the operator to mount the next tape. The operator responds using one of the following REPLY commands:

- **REPLY/TO=*identification-number label***
- **REPLY/INITIALIZE_TAPE=*identification-number label***
- **REPLY/BLANK_TAPE=*identification-number label***
- **REPLY/ABORT=*identification-number***

The **/TO** qualifier indicates that the file system request has been fulfilled. When the request from the magnetic tape file system specifies a volume label, the operator mounts the specified tape and enters the **REPLY/TO** command; however, if the file system requests a new volume, the operator can reuse a scratch volume by mounting it and by entering the **REPLY/INITIALIZE_TAPE** command. The operator also can mount a blank volume and then enter the **REPLY/BLANK_TAPE** command. In any

case, the operator may add the message "label" to the **REPLY** command to specify the volume's label. The quotation marks (" ") are required syntax.

If the request is "**REMOUNT**" or "**MOUNT NEW**", the label is required in the message text. If the request is "**MOUNT**", no label is needed.

The **/ABORT** qualifier indicates that the file system request has been canceled.

Enabling and Disabling Operator Status on a Terminal

Any terminal connected to the operating system can be established as an operator's terminal if OPCOM is running. When an operator who is logged in to an account with OPER (operator) privilege enters the **REPLY/ENABLE** command at the designated terminal, that terminal can be used to respond to user requests and to monitor device status. Such a terminal retains operator status until it is specifically disabled, or until the end of the current interactive session, if it was established as a temporary operator's terminal (see the description of the **/TEMPORARY** qualifier).

Operator messages are printed on the system console terminal unless that terminal is explicitly disabled as an operator's terminal.

When the operator enters the **REPLY/ENABLE** command, OPCOM confirms that the terminal has been enabled. For example:

```
$ REPLY/ENABLE

%OPCOM, 24-DEC-2001 10:22:19.75, operator status for operator OPA0
CENTRAL, PRINTER, TAPES, DISKS, DEVICES, CARDS, NETWORK, CLUSTER, LICENSE,
OPER1, OPER2, OPER3, OPER4, OPER5, OPER6, OPER7, OPER8, OPER9, OPER10,
OPER11, OPER12
```

When the operator enters the **REPLY/DISABLE** command, OPCOM uses the following message to confirm that the terminal is no longer an operator terminal:

```
%OPCOM, 24-DEC-2017 10:03:23.48, operator disabled, operator OPA0
```

To grant specific operator status on a particular terminal, the operator includes one or more keywords after the **/ENABLE** qualifier. For example, to establish a terminal as an operator terminal that can receive messages pertaining to mounting and dismounting tapes and disks, the operator enters the following:

```
$ REPLY/ENABLE=(DISKS,TAPES)

%OPCOM, 24-DEC-2001 10:04:00.18, operator enabled, operator OPA0
$
%OPCOM, 24-DEC-2001 10:04:00.47, operator status for operator OPA0
TAPES, DISKS
```

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

To discontinue specific operator status, the operator includes one or more keywords after the **/DISABLE** qualifier. For example, to inhibit an operator terminal from receiving messages pertaining to mounting and dismounting disks, the operator enters the following command:

```
$ REPLY/DISABLE=DISKS

%OPCOM, 24-DEC-2017 10:04:30.83, operator status for operator OPA0TAPES
```

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

When an operator disables operator status on all terminals, including the system console terminal, OPCOM records all subsequent messages in the operator log file, except user requests and messages requiring an operator reply.

```
%OPCOM-S-OPRNOTIF, operator notified, waiting. 10:06:03.25
%OPCOM-S-OPREPLY, %OPCOM 24-DEC-2001 10:06:03:25, no operator coverage
```

To determine the operator status and to obtain a list of pending requests for a particular terminal, the operator enters the following command:

```
$ REPLY/STATUS
```

This command also shows all outstanding requests for this operator.

Closing the Operator's Log File and Opening a New One

To close the current operator's log file and open anew one, the operator enters the **REPLY/LOG** command. If OPCOM is running, all subsequent messages are recorded in the new log file. To close the current log file without opening a new one, the operator enters the **REPLY/NOLOG** command. All subsequent messages are not recorded until the operator enters the **REPLY/LOG** command.

Qualifiers

/ABORT=*identification-number*

Sends a message to the user or magnetic tape file system corresponding to the unique identification number and cancels the request.

/ALL

Requires OPER (operator) privilege.

Broadcasts a message to all terminals that are attached to the system or cluster. These terminals must be turned on and have broadcast-message reception enabled. The **/ALL** qualifier is incompatible with the **/USERNAME** and **/TERMINAL** qualifiers.

/BELL

Rings a bell at the terminal receiving a message when entered with the **/ALL**, the **/TERMINAL**, or the **/USERNAME** qualifier; two bells when entered with the **/URGENT** qualifier; and three bells when entered with the **/SHUTDOWN** qualifier.

/BLANK_TAPE=*identification-number*

Requires VOLPRO (volume protection) privilege.

Sends a message to the magnetic tape file system indicated by the identification number to override the checking of volume label information. The volume label must be specified in the *message-text* parameter. The current terminal must be enabled as an operator terminal for TAPES.

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

Requires OPER (operator) privilege. Requires OPER and SECURITY privileges for security messages.

If the operator communication manager (OPCOM) is running, restores to normal(that is, non operator) status the terminal at which the command is entered. The **/DISABLE** qualifier cannot

be entered from a batch job. To restrict the types of messages displayed on an operator's terminal, specify one of the following keywords:

Keyword	Description
CARDS	Inhibits messages sent to the card readers.
CENTRAL	Inhibits messages sent to the central system operator.
CLUSTER	Inhibits messages from the connection manager pertaining to cluster state changes.
DEVICES	Inhibits messages pertaining to mounting disks.
DISKS	Inhibits messages pertaining to mounting and dismounting disk volumes.
LICENSE	Inhibits messages pertaining to software licenses.
NETWORK	Inhibits messages pertaining to networks; the keyword CENTRAL must also be specified to inhibit network messages.
OPER1 to OPER12	Inhibits messages sent to operators identified as OPER1 to OPER12.
PRINTER	Inhibits messages pertaining to print requests.
SECURITY	Inhibits messages pertaining to security events; requires SECURITY privilege.
TAPES	Inhibits messages pertaining to mounting and dismounting tape volumes.

When an operator logs out from a remote or a dial-in terminal, the operator terminal is automatically disabled.

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

Requires OPER (operator) privilege. Requires OPER and SECURITY privileges for security messages.

Designates as an operator's terminal the terminal at which the **REPLY** command is entered, if OPCOM is running. The **/ENABLE** qualifier cannot be entered from a batch job. To enable the following types of messages displayed on an operator's terminal, specify one of the following keywords:

Keyword	Description
CARDS	Displays messages sent to the card readers.
CENTRAL	Displays messages sent to the central system operator.
CLUSTER	Displays messages from the connection manager pertaining to cluster state changes.
DEVICES	Displays messages pertaining to mounting disks.
DISKS	Displays messages pertaining to mounting and dismounting disk volumes.
LICENSE	Displays messages pertaining to software licenses.
NETWORK	Displays messages pertaining to networks; the keyword CENTRAL must also be specified to inhibit network messages.
OPER1 to OPER12	Displays messages sent to operators identified as OPER1 to OPER12.

Keyword	Description
PRINTER	Displays messages pertaining to print requests.
SECURITY	Allows messages pertaining to security events; requires SECURITY privilege.
TAPES	Allows messages pertaining to mounting and dismounting tape volumes.

/INITIALIZE_TAPE=*identification-number*

Sends a message to the magnetic tape file system indicated by the identification number to initialize a magnetic tape volume. This qualifier can be used whenever the file system requests the mounting of a new volume. The system performs normal protection and expiration checks before initializing the volume. The current terminal must be enabled as an operator terminal for TAPES.

If the tape drive cannot read the volume, the mount fails and an error message is returned. Use the **/BLANK_TAPE** qualifier to override the checking of information on a volume label.

/LOG
/NOLOG

Requires OPER (operator) privilege.

Closes the current operator's log file and opens a new one if OPCOM is running. The **/NOLOG** qualifier closes the current log file, but does not open anew log file. The current terminal must be enabled as an operator terminal. The operator can then examine the contents of the previous log file.

/NODE[=(*node-name*[,...])]

Sends a message to the local cluster node only. The optional parameter list allows you to specify which nodes will receive the message. Default sends messages to all cluster nodes.

/NOTIFY (default)
/NONOTIFY

Sends a message describing success back to the originating terminal.

/PENDING=*identification-number*

Requires OPER privilege.

Sends a message to the user specified by the identification number and prevents the user from entering other commands until the operator fulfills or aborts the request. The current terminal must be enabled as an operator terminal.

/SHUTDOWN

Sends a message beginning "SHUTDOWN..."; if used with the **/BELL** qualifier, rings three bells at terminals receiving the message.

/STATUS

Requires OPER (operator) privilege.

Reports the current operator status and all outstanding user requests for the terminal from which this command was entered. The current terminal must be enabled as an operator terminal.

/TEMPORARY

Designates the terminal at which the command is entered to be an operator's terminal for the current interactive session only. This qualifier is meaningful only when used with the **/ENABLE** qualifier.

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

Requires OPER (operator) privilege.

Broadcasts the message to specified terminals, where the *terminal-name* parameter is the device name of the terminal. Incompatible with the **/ALL** and **/USERNAME** qualifiers.

/TO=*identification-number*

Requires OPER (operator) privilege.

Sends a message to the user or file system specified by the identification number and completes the request. The current terminal must be enabled as an operator terminal.

Note that you can also use a variation of the **REPLY/TO** command in response to a **MOUNT/ASSIST** command where you redirect the mount operation to another device. Whenever you must substitute a device, load the user's volume on the alternate device and prepare the device for connection before entering the **REPLY** command. Use the following syntax:

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

You can abbreviate the word `SUBSTITUTE` to `S` and can use uppercase or lowercase characters. After a space, use the remainder of the *message-text* space to name the substituted device.

/URGENT

Sends a message beginning "URGENT..."; if used with the **/BELL** qualifier, rings two bells at terminals receiving the message.

/USERNAME[=(*username*[,...])]

Requires OPER (operator) privilege.

Broadcasts a message to all terminals at which users are logged in to the system (or OpenVMS Cluster), or only to the terminals of the specified users. Incompatible with the **/ALL** and **/TERMINAL** qualifiers.

/WAIT

Sends a message synchronously and then waits. The default is to send a message to OPCOM, which does the actual I/O. On a cluster, the message is sent to the local node.

Examples

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

The **REPLY** command in this example broadcasts a message to all terminals on the system. When the message appears at the user's terminal, it is prefixed with the terminal name, the user name of the sender, and (when DECnet-Plus or DECnet for OpenVMS is installed) the node name. The bell sounds at the terminal as the message is displayed.

2.
```
$ REPLY/ENABLE=DISKS
%OPCOM, 24-DEC-2001, 10:17:09.02, operator enabled, operator OPA0
$
%OPCOM, 24-DEC-2001 10:17:10.30, operator status for operator OPA0
DISKS
```

   The **REPLY/ENABLE** command in this example designates the terminal OPA0 as an operator terminal that can receive messages pertaining to mounting and dismounting disks. The OPCOM message confirms that terminal OPA0 is established as an operator's terminal.

3.
```
%OPCOM, 24-DEC-2001 10:19:33.21, request 5, from user SYSTEM
OPA0, Please mount OPGUIDE on DKA3:
$ REPLY/PENDING=5 "YOU'LL HAVE TO WAIT... -
_$ THERE ARE SEVERAL REQUESTS BEFORE YOURS"
    .
    .
    .
$ REPLY/TO=5
24-DEC-2001 10:20:25.50, request 5 completed by operator OPA0
```

   In this example the OPCOM message indicates that a user wants the operator to place the disk volume labeled OPGUIDE on the disk drive DKA3 and prepare the device for connection. The **REPLY/PENDING** command indicates that the operator can perform the task but not immediately; the **/PENDING** qualifier prevents the user from entering other commands until the operator fulfills or aborts the request. After mounting the disk on the drive the operator sends a message indicating that the request has been fulfilled. When no message is specified, OPCOM sends a standard message indicating that the task has been performed.

4.
```
%%%%%%%%%%  OPCOM, 24-DEC-2001 10:20:50.39  %%%%%%%%%%
request 5 from user ROBINSON
Please mount volume GRAPHIC_FILES in device _DUA11:
Shelf 4 - slot B
$ REPLY/TO=5 "SUBSTITUTE  DUA4"
```

   The **REPLY/TO** command with the SUBSTITUTE syntax in this example is used in response to a **MOUNT/ASSIST** command entered by user ROBINSON. The **MOUNT** device is switched to DUA4, and the logical name specified by the user in the **MOUNT** command is defined with an equivalence name of DUA4 rather than the drive originally specified.

5.
```
$ REPLY/STATUS
%OPCOM, 24-DEC-2001 10:20:50.39, operator status for operator OPA0
DISKS
```

   The **REPLY/STATUS** command in this example requests that the operator terminal status for terminal OPA0 be displayed. The response from OPCOM indicates that terminal OPA0 is enabled to receive messages from disk devices.

6.
```
$ REPLY/BELL/TERMINAL=TTC1: "YOUR FILE HAS COMPLETED PRINTING. BOB S."
```

   The **REPLY** command in this example sends a message to the user logged in at terminal TTC1. When the message displays, a bell rings at that terminal.

7.
```
$ REPLY/ENABLE
%OPCOM, 24-DEC-2001 10:22:19.75, operator status for operator OPA0
CENTRAL, PRINTER, TAPES, DISKS, DEVICES, CARDS, NETWORK, CLUSTER,
LICENSE, OPER11, OPER12
```

```
    .
    .
    .
$ REPLY/DISABLE=(PRINTER, TAPES)
%OPCOM, 24-DEC-2001 10:22:26.07, operator disabled, operator OPA0
```

The **REPLY/ENABLE** command in this example designates terminal OPA0 to receive messages from all facilities. Later, the **REPLY/DISABLE** command selectively disables OPA0 from receiving messages pertaining to print devices and tapes.

REQUEST

REQUEST — Displays a message at a system operator's terminal and optionally requests a reply. All messages are logged at the operator's console and in the operator's log file, if that file is initialized. To use this command, you must start the operator communication process (OPCOM) at boot time by specifying the DCL command **@SYS$SYSTEM:STARTUP OPCOM** in the site-specific startup command file, SYS$MANAGER:SYSTARTUP.COM.

Format

REQUEST *message-text*

Parameter

message-text

Specifies the text of the message to be displayed. The string can be up to 128 characters. If the string contains spaces, special characters, or lower-case characters, enclose it in quotation marks (" ").

Description

When you use the **REQUEST** command to send a message to an operator, the message is displayed at the operator terminals specified with the **/TO** qualifier.

If you specify the **/REPLY** qualifier, the message is assigned an identification number, so the operator can respond to the message. The system displays the following message:

```
%OPCOM-S-OPRNOTIF, operator notified, waiting...hh:mm:ss
```

When the operator responds to your request, the system displays a message such as the following:

```
%OPCOM-S-OPREPLY, message text entered by operator
```

If you request a reply, you cannot enter any commands until the operator responds. If you press **Ctrl/C**, the system displays the following message:

```
REQUEST - Enter message or cancel with ^Z
REQUEST - Message?
```

At this time, you can either enter another message, or press **Ctrl/Z** to cancel the request. If you enter another message, that message is sent to the operator, and you must continue to wait for a reply.

All messages are logged at the central operator's console and in the system operator's log file, if that file is initialized.

Qualifiers

/REPLY

Requests a reply to the message and issues a unique identification number to which the operator sends the response. The system displays a message that the operator has been notified; you cannot enter any commands until the operator responds. If you press **Ctrl/C** before the operator responds, you can then enter another message to the operator, or press **Ctrl/Z** to cancel the request.

/TO=(*operator*[,...])

Specifies one or more operators to whom you want to send the message. Possible keywords are as follows:

Keyword	Description
CARDS	Sends the message to operators designated to respond to card reader requests.
CENTRAL	Sends the message to the central system operator.
CLUSTER	Sends the message to operators designated to respond to cluster-related requests.
DEVICES	Sends the message to operators who mount and dismount disks.
DISKS	Sends the message to operators who mount and dismount disk volumes.
NETWORK	Sends the message to the network operator.
OPER1 to OPER12	Sends the message to operators identified as OPER1 to OPER12.
PRINTER	Sends the message to operators designated to handle print requests.
SECURITY	Sends the message to operators designated to respond to security-related requests.
TAPES	Sends the message to operators designated to mount and dismount tape volumes.

Examples

1.
```
$ PRINT/COPIES=2/QUEUE=LQ_PRINT  REPORT.OUT/FORM=LETTER
  Job REPRT (queue LQA1, entry 401) pending
$ REQUEST/REPLY/TO=PRINTER -
_$ "Have queued job 401 as FORM=LETTER;  can you print it?"
%OPCOM-S-OPRNOTIF, operator notified, waiting...10:42:16.10
%OPCOM-S-OPREPLY, AFTER 11:00
 14-DEC-2001 10:25:32.40, request 3 completed by operator OPA0
```

   In this example the **PRINT** command requests that multiple copies of a file be printed using a special paper (**/FORM=LETTER**). After queueing the job to the printer, the **REQUEST** command sends a message to the system operator.

   The operator sends a reply after completing the request.

2.
```
$ REQUEST/REPLY  "Are you there?"
%OPCOM-S-OPRNOTIF, operator notified, waiting...14:54:30.33
```

```
Ctrl/C
REQUEST-Enter message or cancel request with ^Z
REQUEST-Message? Ctrl/Z
%OPCOM-S-OPRNOTIF, operator notified, waiting... 14:59:01.38
%OPCOM-F-RQSTCAN, request was canceled
```

In this example the **REQUEST** command issues a message and requests a response. When no operator replies to the question, **Ctrl/C** is used to interrupt the request; then **Ctrl/Z** is used to cancel it.

RETURN

RETURN — Terminates a **GOSUB** subroutine procedure and returns control to the command following the calling **GOSUB** command.

Format

RETURN [*status-code*]

Parameter

status-code

Defines a longword (integer) value or expression equivalent to an integer value that gives the exit status of the subroutine by defining a numeric value for the reserved global symbol $STATUS. The value can be tested by the next outer command level. The low-order 3 bits of the longword integer value change the value of the reserved global symbol $SEVERITY. If you specify a status code, DCL interprets the code as a condition code. Note that even numeric values produce warning, error, and fatal error messages, and that odd numeric values produce either no message or a success or informational message.

If you do not specify a status code, the current value of $STATUS is saved. When control returns to the outer command level, $STATUS contains the status of the most recently executed command or program.

Description

The **RETURN** command terminates the **GOSUB** subroutine and returns control back to the command following the calling **GOSUB** command.

When a DCL command, user program, or command procedure completes execution, the command interpreter saves the condition code value in the global symbol$STATUS. The system maintains this value in hexadecimal. If a RETURN command does not explicitly set a value for $STATUS, the command interpreter uses the current value of $STATUS to determine the error status.

The low-order 3 bits of the status value contained in $STATUS represent the severity of the condition. The reserved global symbol $SEVERITY contains this portion of the condition code. Severity values range from 0 to 4, as shown in the following table:

Value	Severity
0	Warning
1	Success

Value	Severity
2	Error
3	Information
4	Severe (fatal) error

Note that the success and information codes have odd numeric values, and that warning and error codes have even numeric values.

Example

```
$ SHOW TIME
  15-DEC-2001 14:25:42
$ GOSUB SYMBOL
$ EXIT
$ SYMBOL:
$     SHOW SYMBOL RED
      RED = "SET DEFAULT [LOWE.DCL]"
$     RETURN 1
```

The **GOSUB** command transfers control to the subroutine labeled `SYMBOL`. After the subroutine is executed, the **RETURN** command transfers control back to the command following the calling **GOSUB** statement, giving `$STATUS` and `$SEVERITY` a value of `1`. The procedure then exits.

RUN (Image)

RUN (Image) — Executes an image within the context of your process. You can abbreviate the **RUN** command to a single letter, R.

Format

RUN *filespec*

Parameter

filespec

Specifies an executable image to be executed. The file type defaults to .EXE. The asterisk (*) and the percent sign (%) wildcard characters are not allowed.

Description

This command executes an image within the context of your process. You can abbreviate the **RUN** command to a single letter, R.

When working with installed images, if you specify an image name in the command line with an explicit version number (or a semicolon), the image runs with current process privileges. If you do not specify an explicit version number (or semicolon), the image runs with any privileges with which it was installed. See the relevant section in the *VSI OpenVMS System Management Utilities Reference Manual, Volume I: A–L* [https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#INSTALL_CH] for more information about the Install utility.

If you invoke an image locally, you must have read (R) or execute (E) access to the image. If you have DECnet software installed and want to execute an image over the network, you must have read (R) access to the image.

If you are invoking an image that requires one or more parameters, you must use the Automatic Foreign Command format or the Foreign Command format. For additional information, see the string assignment statement (:=) in the *VSI OpenVMS DCL Dictionary: A–M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_2] and see the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/].

Qualifier

/DEBUG
/NODEBUG

Executes the image under control of the debugger. The default is the **/DEBUG** qualifier if the image is linked with the **/DEBUG** qualifier and the **/NODEBUG** qualifier if the image is linked without the **/DEBUG** qualifier. The **/DEBUG** qualifier is invalid if the image is linked with the **/NOTRACEBACK** qualifier. The **/NODEBUG** qualifier overrides the effect of the **LINK/DEBUG** command. If the image was linked with the **/TRACEBACK** qualifier, traceback reporting is performed when an error occurs.

If the image was not linked with the debugger, you can specify the **/DEBUG** qualifier to request the debugger at execution time; however, if the **/NOTRACEBACK** qualifier was specified when the image was linked, the **/DEBUG** qualifier is invalid.

For a complete description of the OpenVMS Debugger, see the *VSI OpenVMS Debugger Manual* [https://docs.vmssoftware.com/vsi-openvms-debugger-manual/].

To get help on debugger commands from the DCL level, type the following command:

```
$ HELP/LIBRARY=SYS$HELP:DBG$HELP DEBUG
```

Examples

1.
```
$ RUN LIBRA.EXE
```

   The image `LIBRA.EXE` starts executing in the process. If the image `LIBRA` has been installed with amplified privileges, it runs with those privileges because you have not explicitly specified a version number or a semicolon. Alternatively, the image `LIBRA.EXE` still runs with its amplified privileges, if you enter the **RUN** command as follows:

```
$ RUN LIBRA.EXE
```

2.
```
$ MACRO/ENABLE=DEBUG ORION
$ LINK/DEBUG ORION
$ RUN ORION

 OpenVMS I64 DEBUG64 Version V8.3-014

%DEBUG-I-INITIAL, Language:IMACRO, Module:ORION
DBG>
    .
    .
    .
$ RUN/NODEBUG ORION
```

A program is compiled, linked, and run with the debugger. Subsequently, a **RUN/NODEBUG** command requests that the debugger, which is present in the image, not issue a prompt. If an error occurs while the image executes, the debugger can perform traceback and report on the error.

3. `$ RUN AQUARIUS.EXE;1`

   The image `AQUARIUS.EXE` starts executing in the process. If the image `AQUARIUS.EXE` has been installed with amplified privileges, it does not run with those privileges because you have specified a version number. Instead, the image runs with current process privileges only. When you specify a version number (or even just a semicolon), the image activator does not search its list of special images that have been installed with privileges. The process `AQUARIUS` still runs with only normal process privileges if you enter the **RUN** command as follows:

   ```
   $ RUN AQUARIUS.EXE;
   ```

   In this case, however, the highest version of the image `AQUARIUS` runs.

RUN (Process)

RUN (Process) — Creates a subprocess or a detached process to run an image and deletes the process when the image completes execution. A subprocess is created if any of the qualifiers except the **/UIC** or the **/DETACHED** qualifier is specified. A detached process is created if the **/UIC** or the **/DETACHED** qualifier is specified and you have the IMPERSONATE user privilege.

Format

RUN *filespec*

Parameter

filespec

Specifies the file name of an executable image to be executed in a separate process. The default file type is .EXE. The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification.

Description

The **RUN** command creates a process to execute the specified image. If you specify the **/UIC** or the **/DETACHED** qualifier, the **RUN** command creates a detached process; otherwise, the RUN command creates a subprocess.

When you specify any qualifiers with the **RUN** command, the **RUN** command creates a process and displays the process identification (PID) code in SYS$OUTPUT. The newly created process executes the image named in the file specification. When the image has finished executing, the system deletes the process that was running that image.

By default, the **RUN** command creates a subprocess with the same user identification code (UIC), current disk and directory defaults, privileges, and priority as the current process.

If the detached process terminates unexpectedly and you want to find out why, you can use the Accounting utility to display the final exit status of the process. For more information, see the relevant

section in the *VSI OpenVMS System Management Utilities Reference Manual, Volume I: A–L* [https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#ACC_U].

Both the **/DETACHED** and the **/UIC** qualifiers request the **RUN** command to create a detached process. You must have the user privilege IMPERSONATE or CMKRNL to create a detached process with a different UIC. When you create a detached process, the resource quotas are established as follows:

- If you do not specify a value for a particular quota, that quota is given the value of the corresponding PQL_D* (default) system parameter.
- If you specify a value for a particular quota, that value is compared with the quotas for the creator process (maximum) and the system parameters PQL_M* (minimum). If the value you specify is within the allowed range, it is used for the new process. If you specify a value greater than the creator process's quota, the creator process's quota is used. If the specified value is less than the corresponding system parameter, the system parameter value is substituted.

However, if you have the IMPERSONATE or CMKRNL privilege, you can specify any quotas for the detached process.

Input, Output, and Error Streams

Use the following qualifiers to assign equivalence names for the logical names SYS$INPUT, SYS$OUTPUT, and SYS$ERROR for the process:

/INPUT
/OUTPUT
/ERROR

The equivalence names you specify for these process-permanent files are interpreted within the context of the process you are creating. For example, file type defaults, and logical name use and translation are image- and language-dependent.

Defining Process Attributes

Use the following qualifiers to override the default attributes for a process:

/ACCOUNTING
/DUMP
/PRIORITY
/PRIVILEGES
/PROCESS_NAME
/SERVICE_FAILURE
/SWAPPING

Assigning Resource Quotas

When you enter a **RUN** command to create a process, you can define quotas to restrict the amount of various system resources available to the process. The following resource quota is deductible when you create a subprocess; that is, the value you specify is subtracted from your current quota and given to the subprocess:

Qualifier	Quota
/TIME_LIMIT	CPUTIME

The quota amount is returned to your current process when the subprocess is deleted.

The system defines minimum values for each specifiable quota. If you specify a quota that is below the minimum, or if you specify a deductible quota that reduces your current quota below the minimum, the **RUN** command cannot create the process. To determine your current quotas, enter the **SHOWPROCESS/QUOTAS** command.

You can also specify limits for nondeductible quotas. Nondeductible quotas are established and maintained separately for each process and subprocess. The following qualifiers specify nondeductible quotas:

Qualifier	Quota
/AST_LIMIT	ASTLM
/EXTENT	WSEXTENT
/IO_BUFFERED	BIOLM
/IO_DIRECT	DIOLM
/MAXIMUM_WORKING_SET	WSQUOTA
/WORKING_SET	WSDEFAULT

A third type of quota treatment is pooling. Pooled quotas are established when a detached process is created. They are shared by that process and all its descendent subprocesses. Charges against pooled quota values are subtracted from the current available totals as they are used and are added back to the total when they are not being used. The following qualifiers specify pooled quotas:

Qualifier	Quota
/BUFFER_LIMIT	BYTLM
/ENQUEUE_LIMIT	ENQLM
/FILE_LIMIT	FILLM
/PAGE_FILE	PGFLQUOTA
/QUEUE_LIMIT	TQELM
/SUBPROCESS_LIMIT	PRCLM

Hibernation and Scheduled Wakeup Requests

Use the following qualifiers to schedule execution of the image:

/DELAY
/INTERVAL
/SCHEDULE

If you specify any of these qualifiers, the **RUN** command creates the process and places it in hibernation. The process cannot execute the image until it is awakened. Time values specified with these three qualifiers control when the process is awakened to execute the specified image.

You can schedule wakeup requests for a specified delta time (**/DELAY** qualifier) or absolute time (**/SCHEDULE** qualifier). You can also schedule wakeup requests for recurrent intervals with the **/INTERVAL** qualifier. If you specify an interval time, the created process is awakened to execute the specified image at fixed time intervals. If the image terminates normally (for example, by a RET

instruction), the process returns to a state of hibernation, awaiting the next scheduled wakeup time and user-mode exit handlers are not called. At the next wakeup time, the image is recalled at its entry point; the image is not reactivated. If the image terminates abnormally, or by an **$EXIT** command, or by a **$FORCEX** command, the process does not return to hibernation, further scheduling requests are terminated, user-mode exit handlers are called, the image exits, and the created process is deleted.

Use the **/PROCESS_NAME** qualifier to give the created process a name. You can use this process name in a subsequent **STOP** or **CANCEL** command. A **STOP** command terminates execution of the image in the process and causes the process to be deleted. The **CANCEL** command cancels wakeup requests that are scheduled but have not yet been delivered.

Qualifiers

/ACCOUNTING (default)
/NOACCOUNTING

Requires ACNT (accounting) privilege to use the **/NOACCOUNTING** qualifier.

The **/NOACCOUNTING** qualifier stops the current accounting file tracking the resources used by the created process.

The **/ACCOUNTING** qualifier (the default) has no effect.

/AST_LIMIT=*quota*

Specifies the maximum number of asynchronous system traps (ASTs) that the created process can have outstanding.

If you do specify an AST limit quota, the default quota established at system generation time is used. The minimum required for any process to execute is 2.

The AST limit quota is nondeductible.

/AUTHORIZE
/NOAUTHORIZE (default)

Requires IMPERSONATE privilege.

When the image to be executed is the system login image (LOGINOUT.EXE), this qualifier searches the user authorization file (UAF) to validate a detached process. The **/NOAUTHORIZE** qualifier creates a detached process that runs under the control of the command interpreter.

When you specify the **/AUTHORIZE** qualifier, quotas are derived from the user authorization file (UAF) record of the process' owner. Any qualifiers to the RUN command that specify other quotas are ignored in favor of the UAF quotas.

When you specify the **/NOAUTHORIZE** qualifier, quotas are derived from the system parameters that set process quota default limits (parameters prefixed with PQL_D).

Specify the **/AUTHORIZE** qualifier if you want the login image to check the UAF whenever a detached process is created. The process-permanent files specified by the **/INPUT** and **/OUTPUT** qualifiers are made available to the command interpreter for input and output.

/BUFFER_LIMIT=*quota*

Specifies the maximum amount of memory, in bytes, that the process can use for buffered I/O operations or for temporary mailbox creation.

If you do not specify a buffered I/O quota, the default value established at system generation time is used. The minimum amount required for any process to execute is 1024 bytes.

The buffer limit quota is pooled.

/DELAY=*delta-time*

Places the created process in hibernation and awakens it after a specified time interval.

Specify the delta time according to the rules described for entering delta times in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DELTA_TIME] or the online help topic Date.

If you specify both the **/DELAY** and **/INTERVAL** qualifiers, the first wakeup request occurs at the time specified by the **/DELAY** qualifier. All subsequent wakeup requests occur at the interval specified by the **/INTERVAL** qualifier.

/DETACHED

Creates a detached process with the same user identification code (UIC) as the current process. (To create a detached process with a different UIC, use the **/UIC** qualifier.) By default, the created process is not a detached process.

By default, resource quotas for the detached process are limited by the quotas of the creator process (maximum) and the system parameters PQL_M*(minimum). The IMPERSONATE or CMKRNL privilege allows you to specify any quotas for the detached process that exceed the normal range. Unless you have the IMPERSONATE or CMKRNL privilege, the maximum number of detached processes that you can create is limited to the quota defined by MAX_DETACH in your UAF.

/DUMP
/NODUMP (default)

When an image terminates because of an unhandled error, the **/DUMP** qualifier causes the contents of the address space to be written to the file named *image-name*.DMP in the process's default device and directory. You can then use the Analyze/Process_Dump utility to analyze the dump.

/ENQUEUE_LIMIT=*quota*

Specifies the maximum number of locks that a process can have outstanding at any one time.

The default quota is that established at system generation time. The minimum required for any process to operate is 2.

/ERROR=*filespec*

Defines an equivalence name string of 1 to 63 alphanumeric characters for the logical device name SYS$ERROR. The logical name and equivalence name are placed in the process logical name table for the created process. (The **/ERROR** qualifier is ignored if you are running SYS$SYSTEM:LOGINOUT.)

/EXTENT=*quota*

Specifies the maximum size to which the image being executed in the process can increase its physical memory size.

The default quota is that established at system generation time. The minimum value required for any process to execute is 10 pages.

Specify the value of *n* as a number of 512-byte pagelets on Alpha. Note that the operating system rounds up this value to the nearest CPU-specific page so that actual amount of physical memory allowed may be larger than the specified amount on Alpha.

The extent quota is nondeductible.

/FILE_LIMIT=*quota*

Specifies the maximum number of files that a process can have open at any onetime.

The default quota is the quota established at system generation time. The minimum amount required for any process to execute is 2.

The file limit quota is pooled.

/INPUT=*filespec*

Defines an equivalence name string of 1 to 63 characters for SYS$INPUT. The logical name and equivalence name are placed in the process logical name table for the created process.

/INTERVAL=*delta-time*

Requests that the created process be placed in hibernation and be awakened at regularly scheduled intervals.

Specify the delta time according to the rules described in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DELTA_TIME] or the online help topic Date.

If you specify the **/DELAY** or the **/SCHEDULE** qualifier with the **/INTERVAL** qualifier, the first wakeup request occurs at the time specified by the **/DELAY** or the **/SCHEDULE** qualifier; all subsequent wakeup requests occur at intervals specified by the **/INTERVAL** qualifier. If you specify neither the **/DELAY** nor the **/SCHEDULE** qualifier with the **/INTERVAL** qualifier, the first wakeup request occurs immediately by default.

/IO_BUFFERED=*quota*

Specifies the maximum number of system-buffered I/O operations that the created process can have outstanding at any one time.

The default quota is the quota established at system generation time. The minimum required for any process to execute is 2.

The buffered I/O quota is nondeductible.

/IO_DIRECT=*quota*

Specifies the maximum number of direct I/O operations that the created process can have outstanding at any one time.

The default quota is the quota established at system generation time. The minimum required for any process to execute is 2.

The direct I/O quota is nondeductible.

/JOB_TABLE_QUOTA=*quota*

Allows you to specify a quota for a detached process's jobwide logical name table.

A value of 0 has a special meaning. It means that the table, for all practical purposes, has infinite quota because its quota is pooled with that of its parent table, the system directory table.

Note that the **/JOB_TABLE_QUOTA** qualifier is relevant only for detached processes. If the **/JOB_TABLE_QUOTA** qualifier is specified in a RUN command that results in the creation of a subprocess, it is ignored.

/KERNEL_THREAD_LIMIT=*n*

Sets the limit of the number of kernel threads that can be created in the new process. If the value specified is higher than the SYSGEN parameter MULTITHREAD, an error message is returned. Specifying the value 0 makes the process use the systemwide limit set by the MULTITHREAD parameter.

/MAILBOX=*unit*

Specifies the unit number of a mailbox to receive a termination message when the created process is deleted. If no mailbox is specified, the creating process receives no notification when the subprocess or detached process has been deleted.

/MAXIMUM_WORKING_SET=*quota*

Specifies the maximum size (up to 64K pages) to which the image being executed in the process can increase its working set size. An image can increase its working set size by calling the $ADJWSL (Adjust Working Set Limit) system service.

The default quota is the quota established at system generation time. The minimum value required for any process to execute is 10 pages.

The maximum working set quota is nondeductible.

/ON=*node_name*

Specifies an OpenVMS Cluster node on which a detached process is to be created. Quotas are determined by comparing the values you specify with the **RUN** command with the quotas of the creator process (maximum) and the PQL_M* system parameters on the target node (minimum). To specify quotas outside that range, you must have IMPERSONATE or CMKRNL privilege. Omitted quota values default to the values of the PQL_D* system parameters on the target node.

The qualifier value *node_name* is a 1- to 6-character string containing the SCS node name of the requested node.

For example, to create a process named `BAR` on node `FOO` that runs `MY_PROG.EXE`, enter the following command:

```
$ RUN $10$DKB100:[SMITH]MY_PROG.EXE /DETACH /ON="FOO"/PROCESS_NAME="BAR"
```

Note that the disk containing the image must be mounted on the specified node. Because the disk might not be mounted on the node on which the command is entered, the **RUN** command processor does not check whether the image exists. Consequently, the command can complete without error even though the created process aborts immediately because the image file cannot be found.

/OUTPUT=*filespec*

Defines an equivalence name string of 1 to 63 characters for the logical device name SYS $OUTPUT. Both the equivalence name and the logical name are placed in the process logical name table for the created process.

/PAGE_FILE=*quota*

Specifies the maximum number of pagelets that can be allocated in the paging file for the process. The paging file quota is the amount of secondary storage available during execution of the image.

The default quota is the quota established at system generation time. The minimum value required for a process to execute is 256 pagelets. The paging file quota is pooled.

/PRIORITY=*n*

Requires ALTPRI (alter priority) privilege to set the priority higher than your current process.

Specifies the base priority at which the created process executes.

On Alpha, the value of parameter *n* is a decimal 0 to 63, where 63 is the highest priority and zero is the lowest. Normal priorities range from 0 to 15; real-time priorities range from 16 to 63.

The default priority is that of the current process.

/PRIVILEGES=(*privilege*[,...])

Requires SETPRV (set privilege) privilege to specify privileges that you do not have.

Defines user privileges for the created process. You can extend any privilege you possess to a process you create. By default, the created process has the same privileges as its creator. If you specify only one privilege, you can omit the parentheses.

For a list of process privileges, see the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#PRIVILEGE_SUBJ].

You can also use the keyword NOSAME as the *privilege* parameter. If you specify **`/PRIVILEGES=NOSAME`**, the created process has no privileges.

/PROCESS_NAME=*process-name*

Specifies a name of 1 to 15 characters for the created process. The process name is implicitly qualified by the group number of the process's user identification code (UIC). By default, the name is null.

/QUEUE_LIMIT=*quota*

Specifies the maximum number of timer queue entries that the created process can have outstanding at any one time. This number includes timer requests and scheduled wakeup requests.

The default quota is the quota established at system generation time. A process does not require any timer queue quota in order to execute.

The timer queue entry quota is pooled.

/RESOURCE_WAIT (default)
/NORESOURCE_WAIT

Places the created process in a wait state when a resource required for a particular function is not available.

If you specify the **`/NORESOURCE_WAIT`** qualifier, the process receives an error status code when a resource is unavailable.

/SCHEDULE=*absolute-time*

Places the created process in hibernation and awakens it at the specified time.

Specify the absolute time value according to the rules given for entering absolute time values in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#ABSOLUTE_TIME] or the online help topic Date.

/SERVICE_FAILURE
/NOSERVICE_FAILURE (default)

Enables or disables an exception condition notification if an error occurs during a system service request. By default, an error status code is returned to the process.

If you specify the **/SERVICE_FAILURE** qualifier and an error occurs during a system service request, the process encounters an exception condition.

/SSLOG_ENABLE[=COUNT=*n*] [,FLAGS=[NO]ARG]

Valid on Alpha and IA-64 systems only.

Requires CMEXEC, CMKRNL, or SETPRV privilege to log argument values. The SYSGEN parameter SYSSER_LOGGING must be enabled or the command will fail.

Creates a process with system service logging enabled.

Keywords are as follows:

Keyword	Description
COUNT= *n*	Specifies how many P2-space buffers to log. (Default: 2)
FLAGS=[NO]ARG	Specifies whether or not service argument values are to be logged. The default is ARG, which requires privileges. If the value is ARG but you lack privilege, no argument values are logged.

When enabling SSLOG for a process, you can specify the number of buffers to be used for logging. Buffers are allocated in P2 space and are charged against the process's paging file quota. Each buffer is $65{,}024_{10}$ bytes or $FE00_{16}$ bytes. The buffer space remains allocated and the quota charged until the process is deleted.

Before you delete the process, stop the logging and close the log file by executing the **SET PROCESS/SSLOG=STATE=UNLOAD** command. The log file does not close automatically.

To analyze the log file, use the DCL command **ANALYZE/SSLOG**, which is described in online help and in the relevant section in the *VSI OpenVMS System Analysis Tools Manual* [https://docs.vmssoftware.com/vsi-openvms-system-analysis-tools-manual/#d0e28672].

/SUBPROCESS_LIMIT=*quota*

Specifies the maximum number of subprocesses that the created process is allowed to create.

The default quota is the quota established at system generation time. A process does not require any subprocess quota in order to execute.

The subprocess limit quota is pooled.

/SWAPPING (default)
/NOSWAPPING

Requires PSWAPM (process swap mode) privilege to inhibit process swapping.

Permits the process to be swapped. The default allows a process to be swapped from the balance set in physical memory to allow other processes to execute.

With **/NOSWAPPING** in effect, the process is not swapped out of the balance set when it is in a wait state. By default, a process may be swapped out of the balance set whenever it is in a wait state.

/TIME_LIMIT=*limit*

Specifies the maximum amount of CPU time (in delta time) a created process can use. CPU time is allocated to the created process in units of 10 milliseconds. When it has exhausted its CPU time limit quota, the created process is deleted.

If this quota is not specified and the created process is a detached process, the detached process receives a default value of zero, that is, unlimited CPU time.

If this quota is not specified and the created process is a subprocess, the subprocess receives half the CPU time limit quota of the creating process.

If this quota is specified as zero, the created process has unlimited CPU time providing that the creating process also has unlimited CPU time. If, however, the creating process does not have unlimited CPU time, the created process receives half the CPU time limit quota of the creating process.

The CPU time limit quota is a consumable quota; that is, the amount of CPU time used by the created process is not returned to the creating process when the created process is deleted.

If you restrict CPU time for a process, specify the time limit according to the rules for specifying delta time values, as described in the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DELTA_TIME] or the online help topic Date.

/TRUSTED

Specifies that the created process is part of the Trusted Computing Base (TCB) and performs its own auditing. The **/DETACH** qualifier is required as well as the IMPERSONATE privilege.

/UIC=*uic*

Specifies that the created process be a detached process and assigns it a user identification code (UIC). Specify the UIC by using standard UIC format as described in the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#UIC_SECTION].

/WORKING_SET=*default*

Specifies the number of pages in the working set of the created process.

The default working set size is the size established at system generation time. The minimum number of pages required for a process to execute is 10 pages. The value specified cannot be greater than the quota specified with **/MAXIMUM_WORKING_SET**.

The maximum working set quota is nondeductible.

Examples

1.
```
$ RUN/PROCESS_NAME=SUBA   SCANLINE
%RUN-S-PROC_ID, identification of created process is 00010044.
```

   In this example, the **RUN** command creates a subprocess named SUBA to run the image SCANLINE.EXE. The system gives the subprocess an identification number of 00010044.

2.
```
$ RUN/DELAY=3:30/OUTPUT=BALANCE.OUT BALANCE
```

   In this example, the **RUN** command creates a subprocess to run the image BALANCE.EXE 3 hours and 30 minutes from now; output is written to the file BALANCE.OUT.

3.
```
$ RUN/INTERVAL=1:40/PROCESS_NAME=STAT   STATCHK
%RUN-S-PROC_ID, identification of created process is 00050023
    .
    .
    .
$ CANCEL STAT
```

   In this example, the **RUN** command creates a subprocess named STAT to execute the image STATCHK.EXE. The process is scheduled to execute the image at intervals of 1 hour and 40 minutes. The process hibernates; however, because neither the **/DELAY** nor the **/SCHEDULE** qualifier is specified, the first wakeup request occurs immediately.

   The **CANCEL** command subsequently cancels the wakeup requests posted by the **/INTERVAL** qualifier. If the process is currently executing the image, it completes the execution and hibernates.

4.
```
$ RUN/PROCESS_NAME=LYRA  LYRA -
_$ /OUTPUT=_TTB3: -
_$ /ERROR=_TTB3:
%RUN-S-PROC_ID, identification of created process is 000A002F
```

   In this example, the **RUN** command creates a subprocess named LYRA to execute the image LYRA.EXE. The **/OUTPUT** and **/ERROR** qualifiers assign equivalences to the logical names SYS$OUTPUT and SYS$ERROR for the subprocess. Any messages the subprocess writes to its default output devices are displayed on the terminal TTB3.

5.
```
$ RUN/UIC=[100,4]/PRIVILEGES=(SAME,NOPSWAPM) -
_$ /NORESOURCE_WAIT   OVERSEER
%RUN-S-PROC_ID, identification of created process is 0001002C
```

   In this example, the **RUN** command creates a detached process to execute under the UIC [100,4]. The image OVERSEER.EXE is executed. The **RUN** command gives the process all the privileges of the current process, except the ability to alter its swap mode. The **/NORESOURCE_WAIT** qualifier disables resource wait mode for the process.

RUNOFF

RUNOFF — Invokes the DIGITAL Standard Runoff (DSR) text formatter to format one or more ASCII files. Creates formatted files from source DSR (.RNO) files, unformatted table of contents (.RNT) files, and unformatted index (.RNX) files. Optionally creates intermediate (.BRN) files for input to

RUNOFF/CONTENTS and **RUNOFF/INDEX** commands. For more information about the **RUNOFF**, **RUNOFF/CONTENTS**, and **RUNOFF/INDEX** commands and for a description of the DSR formatter, see the *VSI OpenVMS DIGITAL Standard Runoff Reference Manual* [https://docs.vmssoftware.com/digital-standard-runoff-reference-manual/] or online help.

Format

RUNOFF ***filespec*****[,...]**

SEARCH

SEARCH — Searches one or more files for the specified strings and displays the lines containing those strings.

Format

SEARCH ***filespec*****[,...]** ***search-string*****[,...]**

Parameters

filespec**[,...]**

Specifies one or more files to be searched. You must specify at least one file name. If you specify more than one file name, separate the file specifications with commas (,).

You can use the asterisk (*) and the percent sign (%) wildcard characters in the file specification.

search-string**[,...]**

Specifies the character string to be located in the specified files. Enclose strings containing lowercase letters, blanks, or other non-alphanumeric characters (including spaces) in quotation marks (" ").

You can use the **/MATCH** and **/EXACT** qualifiers to alter the way that **SEARCH** matches search strings.

Description

The **SEARCH** command searches through files for specific character strings; all lines containing occurrences of the strings are displayed. Use the **SEARCH** qualifiers to tailor the search operation to your specific needs.

The **SEARCH** command opens the file with shared read (R) and write (W) access. Therefore, any file that has its attributes set to shared write is searched even if it is currently opened by other users.

Qualifiers

/BACKUP

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/BACKUP** qualifier selects files according to the dates of their most recent backups. This qualifier is

incompatible with the **/CREATED**, **/EXPIRED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/BEFORE[=*time*]

Selects only those files dated prior to the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the **/BEFORE** qualifier to indicate the time attribute to be used as the basis for selection: **/BACKUP**, **/CREATED** (default), **/EXPIRED**, or **/MODIFIED**.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/BY_OWNER[=*uic*]

Selects only those files whose owner user identification code (UIC) matches the specified owner UIC. The default UIC is that of the current process.

Specify the UIC by using standard UIC format as described in the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#UIC_SECTION].

/CONFIRM
/NOCONFIRM (default)

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

YES	NO	QUIT
TRUE	FALSE	**Ctrl/Z**
1	0	ALL
	Return	

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), but these abbreviations must be unique. Affirmative answers are YES, TRUE, and 1. Negative answers include: NO, FALSE, 0, and pressing **Return**. Entering QUIT or pressing **Ctrl/Z** indicates that you want to stop processing the command at that point. When you respond by entering ALL, the command continues to process, but no further prompts are given. If you type a response other than one of those in the list, DCL issues an error message and redisplays the prompt.

/CREATED (default)

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/CREATED** qualifier selects files based on their dates of creation. This qualifier is incompatible with the **/BACKUP**, **/EXPIRED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/EXACT
/NOEXACT (default)

Controls whether the **SEARCH** command matches the search string exactly or treats uppercase and lowercase letters as equivalents. By default, **SEARCH** ignores case differences in letters.

Specifying the **/EXACT** qualifier causes the system to use less CPU time;therefore, if you are sure of the case of the letters in the string, it is more efficient to use the **/EXACT** qualifier.

/EXCLUDE=(*filespec*[,...])

Excludes the specified files from the search operation. You can include a directory but not a device in the file specification. The asterisk (*) and the percent sign (%) wildcard characters are allowed in the file specification; however, you cannot use relative version numbers to exclude a specific version. If you specify only one file, you can omit the parentheses.

/EXPIRED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/EXPIRED** qualifier selects files according to their expiration dates. (The expiration date is set with the **SET FILE/EXPIRATION_DATE** command.) The **/EXPIRED** qualifier is incompatible with the **/BACKUP**, **/CREATED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/FORMAT=*option*

Formats output in one of the following five ways:

Option	Description
DUMP	Displays all control characters (including <HT>, <CR>, and <LF>) and non-printable characters as ANSI mnemonics.
NOFF	Replaces control characters in text with ANSI mnemonics (for example, **Ctrl/C** is replaced with <ETX>). The terminal formatting characters <HT>, <CR>, <LF>, <VT> are passed without change. Form feed characters are replaced with <FF>.
NONULLS	Same as DUMP, but removes all null characters from the input file before reformatting. (In dump mode, the null character is displayed as <NUL>.) NONULLS is convenient when you are searching binary format files, such as EXE or OBJ files, that generally contain many zero bytes.
PASSALL	Moves control and non-printable characters to the output device without translating them. The terminal driver cannot send 8-bit characters to the terminal unless **SET TERMINAL/EIGHT_BIT** is already in effect. You can use **/FORMAT=PASSALL** whenever you do not want the **SEARCH** command to substitute the ANSI mnemonic for control characters (for example, <BEL> for **Ctrl/G**).
TEXT	Replaces control characters in text with ANSI mnemonics (for example, **Ctrl/C** is replaced with <ETX>). The terminal formatting characters <HT>, <CR>, <LF>, <VT>, and <FF> are passed without change. TEXT is the default format.

/HEADING (default)
/NOHEADING

Includes file names in the output file and displays a line of 30 asterisks (*) as a window separator between groups of lines that belong to different files. With the default heading format, file names are printed only when more than one file is specified or when the asterisk (*) and the percent sign (%) wildcard characters are used.

The **/WINDOW** qualifier displays a line of 15 asterisks to separate each window within a file.

/HIGHLIGHT[=*keyword*] (default)
/NOHIGHLIGHT

You can use one of the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting on ANSI video terminals with advanced video; REVERSE is the default highlighting on ANSI video terminals without advanced video.

For hardcopy printing, you can use the HARDCOPY=OVERSTRIKE and HARDCOPY=UNDERLINE keywords. This specifies that the strings should be highlighted in a manner suitable for most hardcopy printers. With overstrike highlighting, matched strings are double-printed, so that they appear darker. The matched strings are underlined with the underscore character.

Hardcopy printing is accomplished by adding a carriage return and spacing back over the line to overprint the string or underlines. Note that this can as much as double the length of the line, and perhaps lead to truncation if the device buffer size is too small.

VSI recommends that you use the **/HIGHLIGHT=UNDERLINE** qualifier with the LN01 printer rather than using the **/HIGHLIGHT=HARDCOPY=UNDERLINE** qualifier. The LN01 printer ignores OVERSTRIKE highlighting.

VSI recommends that you use either the **/HIGHLIGHT=BOLD** or the **/HIGHLIGHT=UNDERLINE** qualifier with the LN03 printer rather than using the **/HIGHLIGHT=HARDCOPY=UNDERLINE** qualifier. The LN03 printer ignores OVERSTRIKE highlighting.

Note

You cannot specify **/WILDCARD_MATCHING** with **/HIGHLIGHT**. An error will occur.

/KEY=(POSITION=*n*,SIZE=*n*)

Searches the records of a file (beginning at the specified position)for the length of the specified size.

You can specify the POSITION keyword value as 1 to 32,767. The first byte in a record is considered position 1.

Note

The **/KEY** qualifier must appear before the file name, or after the specified search string. If the qualifier is placed between these parameters, it is ignored.

/LIMIT=*n* (Alpha/IA-64 only)

Limits the number of matches displayed to the number specified by *n*.

/LOG
/NOLOG (default)

Outputs a message to the current SYS$OUTPUT device for each file searched. The message includes the file name, the number of records, and the number of matches for each file searched.

/MATCH=*option*

Interprets and matches multiple search strings in one of the following ways:

Option	Description
AND	A match occurs only if the record contains all the strings.
EQV	A match occurs if none or all of the search strings are in the record.
NOR	A match occurs only if the record contains none of the strings.
NAND	A match occurs only if the record does not contain all of the strings.
OR	A match occurs if the record contains any of the strings.
XOR	A match occurs if any of the search strings are in the record but not if all or none of them are in the record.

When only one search string is specified, the OR and AND options produce identical results. Similarly, NOR and NAND produce identical results for a single search string. If you specify none of these options, the default is **/MATCH=OR**.

/MODIFIED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/MODIFIED** qualifier selects files according to the dates on which they were last modified. This qualifier is incompatible with the **/BACKUP**, **/CREATED**, and **/EXPIRED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time modifiers, the default is the **/CREATED** qualifier.

/NUMBERS
/NONUMBERS (default)

Controls whether the source line number is displayed at the left margin of each line in the output.

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls whether the results of the search are output to a specified file. The output is sent to the current default output device (SYS$OUTPUT) if you omit the **/OUTPUT** qualifier or omit the file specification with the qualifier. The **/NOOUTPUT** qualifier means that no matching records are output as a result of the **SEARCH** command.

/PAGE[=*keyword*]
/NOPAGE (default)

Controls the display of information on the screen.

You can use the following keywords with the **/PAGE** qualifier:

Keyword	Description
CLEAR_SCREEN	Displays information one page at a time.
SCROLL	Displays information on a continuous stream.
SAVE[= *n*]	Enables screen navigation of information, where *n* is the number of pages to store.

The **/PAGE=SAVE** qualifier allows you to navigate through screens of information. The **/PAGE=SAVE** qualifier stores up to 5 screens of up to 255 columns of information. When you use the **/PAGE=SAVE** qualifier, you can use the following keys to navigate through the information:

Key Sequence	Description
Up arrow key, **Ctrl/B**	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.
Right arrow key	Scroll right one column.
Insert Here (**E2**)	Scroll right one half screen.
Remove (**E3**)	Scroll left one half screen.
Select (**E4**)	Toggle 80/132 column mode.
Prev Screen (**E5**)	Get the previous page of information.
Next Screen (**E6**), **Return**, **Enter**, **Space**	Get the next page of information.
F10, **Ctrl/Z**	Exit. (Some utilities define these differently.)
Help (**F15**)	Display utility help text.
Do (**F16**)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.

The **/PAGE** qualifier is not compatible with the **/OUTPUT** qualifier.

/REMAINING
/NOREMAINING (default)

Includes in the output all records from the first matched record to the end of the file. This qualifier overrides the value *n2* in the **/WINDOW** qualifier, but allows the qualifier **/WINDOW=*n1***.

/SINCE[=*time*]

Selects only those files dated on or after the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, JOB_LOGIN, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the **/SINCE** qualifier to indicate the time attribute to be used as the basis for selection: **/BACKUP**, **/CREATED** (default), **/EXPIRED**, or **/MODIFIED**.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/SKIP=*n* (Alpha/IA-64 only)

Skips the first *n* matches found before outputting match information.

/STATISTICS[=(*keyword*,...)]
/NOSTATISTICS[=(*keyword*,...)] (default)

Controls whether the following statistics about the search are displayed:

- Number of files searched
- Number of records searched
- Number of characters searched
- Number of records matched
- Number of lines printed
- Buffered I/O count
- Direct I/O count
- Number of page faults
- Elapsed CPU time
- Elapsed time

You can use the following keywords with the **/STATISTICS** qualifier:

Keyword	Description	
OUTPUT	Writes the statistics output lines into the output file in addition to the standard SYS$OUTPUT device.	
SYMBOLS	Define symbols to hold statistic values. The symbol names are as follows:	
	SEARCH$CHARACTERS_ SEARCHED	Displays the number of characters searched.
	SEARCH$CPU_TIME	Displays the total amount of CPU time that was charged to the process for the command during the time it took to complete the search.
	SEARCH$ELAPSED_TIME	Displays the total amount of wall clock time that it took to complete the search.
	SEARCH$FILES_SEARCHED	Displays the number of files searched.
	SEARCH$LINES_PRINTED	Displays the number of lines printed.
	SEARCH$RECORDS_ MATCHED	Displays the number of records matched.
	SEARCH$RECORDS_ SEARCHED	Displays the number of records searched.

Using the **/STATISTICS** qualifier with no keywords is the equivalent of **/STATISTICS=(OUTPUT,SYMBOLS)**.

/STYLE=*keyword*

Specifies the file name format for display purposes.

The valid keywords for this qualifier are CONDENSED and EXPANDED. Descriptions are as follows:

Keyword	Description
CONDENSED (default)	Displays the file name representation of what is generated to fit into a 255-length character string. This file name may contain a DID or FID abbreviation in the file specification.
EXPANDED	Displays the file name representation of what is stored on disk. This file name does not contain any DID or FID abbreviations.

The keywords CONDENSED and EXPANDED are mutually exclusive. This qualifier specifies which file name format is displayed in the output message, along with the confirmation if requested.

File errors are displayed with the CONDENSED file specification unless the EXPANDED keyword is specified.

See the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#FILE_ACCESS] for more information.

/SYMLINK=*keyword*

The valid keywords for this qualifier are [NO]WILDCARD and [NO]ELLIPSIS. Descriptions are as follows:

Keyword	Description
WILDCARD	Indicates that symlinks are enabled during wildcard searches.
NOWILDCARD	Indicates that symlinks are disabled during directory wildcard searches.
ELLIPSIS	Equivalent to WILDCARD (included for command symmetry).
NOELLIPSIS	Indicates that symlinks are matched for all wildcard fields except for ellipsis.

If the file named in the **SEARCH** command is a symlink, the command operates on the symlink target.

/WARNINGS[=*keyword*] (default)
/NOWARNINGS[=*keyword*]

Enables or disables warning messages.

Specifying one of the keywords listed below with **/WARNINGS** enables the corresponding message type and disables all other types. Specifying a keyword with **/NOWARNINGS** disables the corresponding message type and enables all other types. The following keywords are supported:

Keyword	Message Type
NOMATCHES	No matches were found.
TRUNCATE	The file record is too large and has been truncated.

Keyword	Message Type
NULLFILE	The file is empty (NULL).

/WILDCARD_MATCHING[= *keyword*] (Alpha/IA-64 only)

Specifies that the following characters in the search string are to be interpreted as wildcard characters:

asterisk (*) – Accept any number of unspecified characters in this position when searching for a match.
percent sign (%) – Accept any single character in this position when searching for a match.

The optional keyword can be either of the following:

RELAXED (default) – Automatically appends asterisks to the beginning and end of the string to be searched.
STRICT – Performs a narrow search based on the exact search string with no appended asterisks. An entire record must match the criteria specified for the wildcarded search.

Note

You cannot specify **/WILDCARD_MATCHING** with **/HIGHLIGHT**. An error will occur.

/WINDOW[=(*n1*,*n2*)]
/NOWINDOW (default)

Specifies the number of lines to be displayed with the search string.

If you specify *n1* and *n2*, the **/WINDOW** qualifier displays *n1* lines above the search string, the search string, and *n2* lines below the search string. Either of these numbers can be zero.

If you specify the **/WINDOW** qualifier without the values *n1* and *n2,* two lines above the search string, the search string, and the two lines below the search string are included in the output.

If you specify the **/WINDOW** qualifier with a single number (*n1*), *n1* specifies the number of lines to display including the search string. Half the lines precede the matched search string and half follow it.(If *n1* is even, one line is added to the lines following the matched search string.)

For example, if you specify **/WINDOW=10**, nine additional lines are listed along with the line containing the search string. Four lines are listed above the line containing the search string and five lines are listed below it, for a total of 10 lines.

If you specify **/WINDOW=0**, the file name of each file containing a match (but no records) is included in the output. This specification creates a file (using the **/OUTPUT** qualifier) that can be inserted into a command file to manipulate the files containing matches.

If you omit the **/WINDOW** qualifier, only the line containing a match is displayed.

The **/WINDOW** qualifier displays a line of 30 asterisks to separate each window within a file.

/WRAP
/NOWRAP (default)

Use with the **/PAGE=SAVE** qualifier to limit the number of columns to the width of the screen and to 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 and can be seen when you use the scrolling (left and right) features provided by the **/PAGE=SAVE** qualifier.

Examples

1. `$ SEARCH CABLE.MEM,JOYNER.MEM "MANUAL TITLE"`

   This command searches the files `CABLE.MEM` and `JOYNER.MEM` for occurrences of the character string `MANUAL TITLE`. Each line containing the string is displayed at the terminal. It is necessary to enclose the string in quotation marks because it contains a space character.

2. `$ SEARCH/OUTPUT=RESULTS.DAT/WINDOW=9 DISLIST.MEM NAME`

   The **SEARCH** command searches the file `DISLIST.MEM` for occurrences of the character string `NAME` and sends the output to the file `RESULTS.DAT`. The four lines preceding and following each occurrence of `NAME` are included in the output.

3. `$ SEARCH/OUTPUT=ALLSUB.COM/WINDOW=5000 *.COM SUBMIT`

   The **SEARCH** command searches all command files in the current directory for the string `SUBMIT`. If a match is found, **SEARCH** effectively copies the entire command file to the output file, because the window is so large.

4. ```
   $ SEARCH/OUTPUT=COLUMBUS.OH/WINDOW=(3,0)/NOHEAD/MATCH=AND -
   _$ *.DAT COLUMBUS,OH
   ```

   The **SEARCH** command searches all files of type `DAT` for lines containing both `COLUMBUS` and `OH`. When a match is found, the three previous lines (containing blank line, name, and street address) are copied to the new file. The new file `COLUMBUS.OH` is ready to use, because it does not contain headings and window separators.

5. ```
   $ SEARCH/OUTPUT=SWAP.LIS/FORMAT=PASSALL/NUMBERS/EXACT -
   _$ /WINDOW=10000 SWAP.PAS SWAP
   ```

   This **SEARCH** command produces a listing file with the line numbers at the left margin. The **/FORMAT=PASSALL** qualifier is specified so that form-feed characters in the source are passed through. The **/EXACT** qualifier is specified for efficiency (because it is known that the name `SWAP` in the program statement is always in uppercase). The **/WINDOW** qualifier is entered so that the entire file is copied to the output file `SWAP.LIS`.

6. `$ SEARCH/REMAINING CABLE.LOG FORTRAN`

   The **SEARCH** command displays all the lines in the `CABLE.LOG` file that follow the first occurrence of the string `FORTRAN`.

7. `$ SEARCH OMAHA::DISK1:[EXP]SUB.DAT,DATA.LIS VMS`

   The **SEARCH** command searches through the files `SUB.DAT` and `DATA.LIS` at remote node `OMAHA` for all occurrences of the string `VMS`. The list of all records containing the string `VMS` is displayed at the local terminal.

SET

SET — Defines or changes the session, batch job, or system values or characteristics. See the Description section of each command for details.

Format

SET *option*

Description

The **SET** command options are described individually in this manual. *Table 2, "SET Command Options"* lists all the **SET** command options, including those generally reserved for use by system operators and managers:

Table 2. SET Command Options

Option	Description
ACCOUNTING	Controls the current accounting file.
AUDIT	Provides the management interface to the security auditing system.
BOOTBLOCK (IA-64 only)	Initializes the boot block on the target device.
BROADCAST	Determines which messages will be broadcast to SYS$OUTPUT.
CACHE/RESET	Resets systemwide I/O caching statistics for the extended file cache (XFC).
CARD_READER	Defines the default ASCII translation mode for a card reader.
CLUSTER/EXPECTED_VOTES	Sets the total expected votes in the OpenVMS Cluster to a value that you specify or, if no value is specified, sets the total votes to a value determined by the system.
COMMAND	Adds commands that are defined in a command description file to your process command set or a command tables file.
CONTROL	Enables or disables interrupts caused by **Ctrl/T** or **Ctrl/Y**.
CPU	Changes the user capabilities associated with the specified CPUs.
DAY	Overrides the default day type specified in the user authorization file (UAF).
DEFAULT	Establishes a device and directory as the current default for file specifications.
DEVICE	Defines device characteristics.
DEVICE/SERVED	Lets you make a disk on a local node available to all the nodes on an OpenVMS Cluster.
DIRECTORY	Modifies the characteristics of one or more directories.
DISPLAY	Redirects the output of a DECwindows application.
DUMP_OPTIONS (x86-64 Only)	Updates the list of alternate devices where system crash dumps can be written.
ENTRY	Changes the current status or attributes of a job not currently executing in a queue.

Option	Description
FILE	Modifies the characteristics of one or more files.
HOST	Connects your terminal (through the current host processor) to another processor, called the remote processor.
HOST/DTE	Connects your system to a remote system by way of an outgoing terminal line.
HOST/DUP	Connects your terminal to a storage controller through the appropriate bus for that controller.
HOST/HSC	Connects your terminal to a remote HSC50 disk and tape controller through the computer interconnect (CI) bus.
HOST/LAT	Connects your terminal to a specified service available in the local area network (LAN), establishing one session for communication between your terminal and that service.
HOST/RLOGIN	Allows you to log in to a remote host over a TCP/IP connection and start an interactive terminal session by accessing the RLOGIN application.
HOST/TELNET	Connects you to a remote host over a TCP/IP connection by invoking the TELNET application.
HOST/TN3270	Connects you to a remote IBM host over a TCP/IP connection, causing the local keyboard to emulate an IBM 3279-class terminal keyboard by invoking the TN3270 terminal emulator.
IMAGE	Modifies or restores the image attributes of an OpenVMS ELF format image file.
KEY	Changes the current keypad state setting.
LOGINS	Allows or disallows users to log in to the system.
MAGTAPE	Defines characteristics of a magnetic tape device.
MESSAGE	Overrides or supplements system messages.
NETWORK	Registers the attributes of a network service.
ON	Controls whether the command interpreter checks for an error condition following the execution of commands in a command procedure.
OUTPUT_RATE	Sets the rate at which output is written to a batch job log file.
PASSWORD	Lets users change their own passwords; lets system managers change the system password.
PREFERRED_PATH	Specifies a particular host or preferred path to access a specific MSCP class disk or TMSCP class tape device.
PREFIX	Allows you to set a prefix control string for verified command lines.
PRINTER	Defines printer characteristics.
PROCESS	Defines execution characteristics of the current process.

Option	Description
PROMPT	Defines the DCL prompt.
PROTECTION/DEFAULT	Establishes the default protection to be applied to all files subsequently created.
QUEUE	Changes the current status or attributes of the specified queue.
RESTART_VALUE	Establishes a test value for restarting portions of batch jobs.
RIGHTS_LIST	Lets users modify the process rights list; lets privileged users modify the system rights list.
RMS_DEFAULT	Provides default multiblock and multibuffer count values to be used by RMS for file operations.
ROOT	Defines the POSIX root for the system.
SECURITY	Modifies the security profile of an object.
SERVER ACME_SERVER	Controls starting, stopping, and restarting of the ACME server, which controls authentication policies for $ACM system service clients.
SERVER REGISTRY_SERVER (Alpha/IA-64 only)	Controls starting, stopping, and restarting of the Registry server, which provides access to the Registry database for $REGISTRY system service clients.
SERVER SECURITY_SERVER	Controls starting, stopping, and restarting of the security server, which maintains information stored in the system intrusion and proxy databases.
SHADOW	Changes the characteristics of shadow sets created using Volume Shadowing for OpenVMS.
SYMBOL	Controls access to local and global symbols in command procedures.
TERMINAL	Defines terminal characteristics.
TIME	Resets the system clock to the specified value.
VERIFY	Controls whether the command interpreter displays lines in command procedures as it executes them.
VOLUME	Modifies the characteristics of one or more Files-11 volumes.
WORKING_SET	Changes the current working set limit or quota.

SET ACCOUNTING

SET ACCOUNTING — Controls the current accounting file. **Requires OPER (operator) privilege.**

Format

SET ACCOUNTING

Parameters

None.

Description

Each node on your system has its own current accounting file. You can control what resources this file tracks, and start up a new version of this file using the **SET ACCOUNTING** command.

There are two occasions when the resources used by a process are not tracked, despite the **SET ACCOUNTING** command:

- When you use the **RUN** (Process) command with the **/NOACCOUNTING** qualifier.
- When you use the $CREPRC system service with the PRC$M_NOACNT status flag.

Similarly, there is one occasion when the resources used by an image are always tracked, despite the **SET ACCOUNTING** command:

- When you install an image using the **/ACCOUNTING** qualifier of the Install utility.

For more information on how to use the **SET ACCOUNTING** command, see the relevant section in the *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/#_6017ACCOUNTING].

Qualifiers

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

Prevents the tracking of the resources specified by the keywords.

Table 3, "SET ACCOUNTING Keywords for Resource Types" lists the keywords you can use to specify the type of resource.

Table 3. SET ACCOUNTING Keywords for Resource Types

Keyword	Type of Resource
IMAGE	Resources used by an image
LOGIN_FAILURE	Resources used by an unsuccessful attempt to log in
MESSAGE	Unformatted record written to the accounting file by a call to the $SNDJBC system service
PRINT	Resources used by a print job
PROCESS	Resources used by a process

You do not need to stop the tracking of all processes and images. You can prevent resources being tracked for specific types of process and for images running in these types of process.

Table 4, "SET ACCOUNTING Keywords for Process Types" lists the keywords you can use to specify the type of process.

Table 4. SET ACCOUNTING Keywords for Process Types

Keyword	Type of Process
BATCH	Batch process
DETACHED	Detached process
INTERACTIVE	Interactive process
NETWORK	Network process
SUBPROCESS	Subprocess (the parent process can be a batch, detached, network, or interactive process)

If the system is no longer tracking any resources, **/DISABLE** closes the current accounting file.

If you use the **/DISABLE** qualifier and omit the keywords, the current accounting file does not track any resources, and the system closes the file.

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

Enables the tracking of the specified resources, and opens the current accounting file if it is not already open. The **/ENABLE** qualifier uses the same keywords as the **/DISABLE** qualifier.

Use the keywords shown in *Table 3, "SET ACCOUNTING Keywords for Resource Types"* to specify the types of resource that you want the local node to track in its current accounting file.

If the resources used by processes or images are being tracked, you can use the keywords shown in *Table 4, "SET ACCOUNTING Keywords for Process Types"* to enable the tracking of these resources for specified types of process and for images running in those types of process.

If you use the **/ENABLE** qualifier and omit the keywords, the current accounting file tracks all resources.

/LOG

Writes information to the current SYS$OUTPUT device as the command executes.

/NEW_FILE

Closes the current accounting file, and starts up a new version of it.

The name of the new file depends on whether the logical name ACCOUNTNG is defined in your system logical name table.

If this logical name is not defined, the **SET ACCOUNTING** command opens the file SYS$MANAGER:ACCOUNTNG.DAT.

If this logical name is defined, the command opens the file that this logical name points to. If you omit the directory, SYS$MANAGER is the default, and if you omit the file type, .DAT is the default.

The **/NEW_FILE** qualifier writes a record to the end of the old file that contains a forward pointer to the new file, and a record to the beginning of the new file that contains a backward pointer to the old file. These records contain the names of the new and old files respectively.

Examples

1. `$ SET ACCOUNTING /DISABLE /ENABLE=(PROCESS,BATCH,INTERACTIVE)`

```
$ SET ACCOUNTING /ENABLE=IMAGE
```

This example tells the system to track the resources used only by batch and interactive processes, and by images running in batch and interactive processes. It illustrates the cumulative effect of **/ENABLE** and **/DISABLE** qualifiers, and of **SET ACCOUNTING** commands.

The **/DISABLE** qualifier prevents the tracking of all resources. The **/ENABLE** qualifier then tells the system to track the resources used by batch and interactive processes. The second **SET ACCOUNTING** command tells the system to track the resources used by images.

2.
```
$ SET ACCOUNTING /NEW_FILE
$ RENAME SYS$MANAGER:ACCOUNTNG.DAT;-1 WEEK_24_RESOURCES.DAT
```

This example closes the current accounting file, opens a new version of it, and changes the name of the old file to `WEEK_24_RESOURCES.DAT`.

SET AUDIT

SET AUDIT — Provides the management interface to the security auditing system. Requires the SECURITY privilege.

Format

SET AUDIT/*qualifier*

Parameters

None.

Description

The **SET AUDIT** command and the **SHOW AUDIT** command provide the management interface to the security auditing system.

The **SET AUDIT** command enables or disables security auditing. In addition, you use the command to do the following:

- Select categories of events to audit
- Change the operational characteristics of the audit server
- Establish the location of the audit journal and the security archive file
- Control the monitoring of disk resources

Values set by the command are saved so it is unnecessary to set them each time the system starts up. Commands for event definition, resource monitoring, and starting a new log apply clusterwide, while other commands apply only to the local node.

Security auditing features require a certain amount of system overhead; therefore, you should be careful to select the features that will provide the most benefit in your work environment. Enable only the auditing of information that you know you will examine and analyze regularly. Any other collection of data is likely to be wasteful. For further information about auditing, see the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#AUDITING_CHAPTER].

There are five categories of qualifiers, grouped by task, for the **SET AUDIT** command:

Task	Qualifiers	Requirements
Define auditing events	/AUDIT /ALARM /CLASS /ENABLE /DISABLE	Specify whether you are defining alarms (**/ALARM**), audits (**/AUDIT**), or both. Also specify whether you are enabling (**/ENABLE**) or disabling (**/DISABLE**) the reporting of the event.
Define auditing log file	/DESTINATION /JOURNAL /VERIFY	Requires both the **/DESTINATION** and **/JOURNAL** qualifiers.
Define operational characteristics of the audit server and a listener mailbox (if any)	/INTERVAL /LISTENER /SERVER /VERIFY	None.
Define secondary log file	/ARCHIVE /DESTINATION /VERIFY	None.
Define resource monitoring defaults	/BACKLOG /EXCLUDE /JOURNAL /RESOURCE /THRESHOLD /VERIFY	With the **/RESOURCE** or **/THRESHOLD** qualifier, include the **/JOURNAL** qualifier.

Qualifiers

/ALARM

Makes the command apply to alarms, which are messages displayed on an operator terminal. See the description of the DCL command **REPLY/ENABLE** for details on how to enable terminals to display security messages.

/ARCHIVE=[*keyword*,...]

Specifies which classes of audit event messages are written to the security archive file. Specify one or more of the following keywords:

Keyword	Description
NONE	Disables archiving on the system.
[NO]ALL (default)	Enables or disables archiving of all system security events. By default, no events are archived.
SYSTEM_ALARM	Enables archiving of all security alarm events.
SYSTEM_AUDIT	Enables archiving of all security audit events.

Archiving should be run on only one node in an OpenVMS Cluster with its own audit server database because multiple nodes will try to open the audit file exclusively.

/AUDIT

Makes the command apply to audits, which are messages recorded in the system security audit log file.

/BACKLOG=[*keyword*[,...]]

Specifies the thresholds for suspending a process that has exceeded the process message limit. The thresholds include the total number of messages in memory and the number belonging to the particular process. To prevent a process from being suspended, use the **/EXCLUDE** qualifier. Specify the following keywords:

Keyword	Description
TOTAL=(*n1*,*n2*,*n3*)	Thresholds at which flow control is initiated and accelerated; see description below.
PROCESS=(*p1*,*p2*)	Thresholds at which process submissions are controlled.

Total Messages	Default	Process Messages	Default	Action Taken
N1	100	P1	5	When there are 100 messages in memory, the audit server suspends any process that has submitted 5 or more messages until all messages are written to disk.
N2	200	P2	2	When there are 200 messages in memory, the audit server suspends any process that has submitted 2 or more messages until all messages are written to disk.
N3	300			Any process with messages in memory is suspended until all messages are written to disk.

/CLASS=*class*

Specifies the class of the object whose auditing attributes are to be modified. If **/CLASS** is not specified, the command assumes the class is FILE. Specify one of the following keywords with the **/CLASS** qualifier:

CAPABILITY
COMMON_EVENT_CLUSTER
DEVICE
FILE
GROUP_GLOBAL_SECTION
LOGICAL_NAME_TABLE
QUEUE
RESOURCE_DOMAIN
SECURITY_CLASS
SYSTEM_GLOBAL_SECTION
VOLUME

/DESTINATION=*filespec*

When changing the destination of event messages, specifies the new location of the system security audit log file. The device, if part of the file specification, must be a disk. The **/DESTINATION** qualifier requires the **/JOURNAL** qualifier in this case.

Once you have relocated the log file, execute the command **SET AUDIT/SERVER=NEW_LOG** to let all the nodes in the cluster know of the new location. The previous audit log file is closed and all subsequent audit event messages generated throughout the cluster are sent to the new audit log file.

When used with **/ARCHIVE**, specifies the name of the archive log file. Events can be archived to a local or remote file on any file-structured disk device. For example, you can use an archive file to redirect event messages from a satellite to a larger node in the cluster.

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

Disables alarms or audits for the specified events. To disable all system events and file access events, specify the keyword ALL. You must specify at least one of the keywords. For a list of the keywords to use with the **/DISABLE** qualifier, see the **/ENABLE** qualifier description. You must also specify either the **/ALARM** or **/AUDIT** qualifier, or both, when you use the **/DISABLE** qualifier.

Note

In processing the **SET AUDIT** command, the system processes the **/DISABLE** qualifier last. If you specify both the **/ENABLE** and **/DISABLE** qualifiers for items in the same class on the same command line, the **/DISABLE** qualifier disables any enabled items. VSI recommends that you use separate lines for commands containing the **/ENABLE** and **/DISABLE** qualifiers.

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

Enables alarms or audits for the specified events. To enable all system events and file access events, specify the keyword ALL. You must specify at least one keyword. You must also specify either the **/ALARM** or **/AUDIT** qualifier, or both, when you use the **/ENABLE** qualifier.

The keywords that you can specify with either the **/ENABLE** or the **/DISABLE** qualifier are as follows:

Keyword	Description
ACCESS=(*condition*[:*access*[,...]][,...])	Specifies access events for all objects in a class. (To audit a single object, use an auditing ACE and enable the access control list (ACL) category.) VSI recommends that when you enable auditing conditionally, you enable it for all possible forms of access because the system can check access rights at several points during an operation. (For example, a FAILURE might occur on a read or write access check.) See the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#SYS_ACC_PWD] for information about the various types of access permitted on each class. (For example, the Access keyword, CREATE, is not defined for FILE objects.)

<table>
<tr><th>Keyword</th><th colspan="2">Description</th></tr>
<tr><td rowspan="20"></td><td colspan="2"></td></tr>
<tr><td>Condition Keyword</td><td>Description</td></tr>
<tr><td>ALL</td><td>All object access</td></tr>
<tr><td>BYPASS</td><td>Successful object access due to the use of the BYPASS privilege</td></tr>
<tr><td>FAILURE</td><td>Unsuccessful object access</td></tr>
<tr><td>GRPPRV</td><td>Successful object access due to the use of the group privilege (GRPPRV)</td></tr>
<tr><td>READALL</td><td>Successful object access due to the use of the READALL privilege</td></tr>
<tr><td>SUCCESS</td><td>Successful object access</td></tr>
<tr><td>SYSPRV</td><td>Successful object access due to the use of the system privilege (SYSPRV)</td></tr>
<tr><td>Access Keyword</td><td>Description</td></tr>
<tr><td>ALL</td><td>All types of access</td></tr>
<tr><td>ASSOCIATE</td><td>Associate access</td></tr>
<tr><td>CONTROL</td><td>Control access to examine or change security characteristics</td></tr>
<tr><td>CREATE</td><td>Create access. To audit create events for files, use the CREATE keyword.</td></tr>
<tr><td>DELETE</td><td>Delete access</td></tr>
<tr><td>EXECUTE</td><td>Execute access</td></tr>
<tr><td>LOCK</td><td>Lock access</td></tr>
<tr><td>LOGICAL</td><td>Logical I/O access</td></tr>
<tr><td>MANAGE</td><td>Manage access</td></tr>
<tr><td>PHYSICAL</td><td>Physical I/O access</td></tr>
<tr><td>READ</td><td>Read access</td></tr>
<tr><td>SUBMIT</td><td>Submit access</td></tr>
<tr><td>WRITE</td><td>Write access</td></tr>
<tr><td>ACL</td><td colspan="2">Specifies an event requested by an audit or alarm ACE in the access control list (ACL) of an object. To audit all objects of a class, use the ACCESS keyword.</td></tr>
<tr><td>ALL</td><td colspan="2">Specifies all system events and file access events. It does not enable access events for object classes other than FILE.</td></tr>
</table>

<table>
<tr><th>Keyword</th><th colspan="2">Description</th></tr>
<tr><td>AUDIT=keyword</td><td colspan="2">Specifies events within the auditing subsystem. Only one keyword is currently defined.</td></tr>
<tr><td></td><td>Keyword</td><td>Description</td></tr>
<tr><td></td><td>ILLFORMED</td><td>Specifies ill-formed events from internal calls (identified by NSA$M_INTERNAL) to $AUDIT_EVENT, $CHECK_PRIVILEGE, $CHKPRO, or $CHECK_ACCESS system services. An ill-formed event is caused by an incomplete or syntactically incorrect argument being supplied to one of these system services by a piece of privileged code.</td></tr>
<tr><td>AUTHORIZATION</td><td colspan="2">Specifies the modification of any portion of the system user authorization file (SYSUAF), network proxy authorization file (NETPROXY), or the rights list (RIGHTLIST) (including password changes made through the AUTHORIZE, SET PASSWORD, or LOGINOUT commands or the $SETUAI system service).</td></tr>
<tr><td>BREAKIN=(keyword [,...])</td><td colspan="2">Specifies the occurrence of one or more classes of break-in attempts, as specified by one or more of the following keywords:

ALL
DETACHED
DIALUP
LOCAL
NETWORK
REMOTE</td></tr>
<tr><td>CONNECTION</td><td colspan="2">Specifies a logical link connection or termination through DECnet-Plus, DECnet Phase IV, DECwindows, $IPC, or SYSMAN.</td></tr>
<tr><td>CREATE</td><td colspan="2">Specifies the creation of an object. Requires the /CLASS qualifier if it is not a file.</td></tr>
<tr><td>DEACCESS</td><td colspan="2">Specifies deaccess from an object. Requires the /CLASS qualifier if it is not a file.</td></tr>
<tr><td>DELETE</td><td colspan="2">Specifies the deletion of an object. Requires the /CLASS=DEVICE qualifier.</td></tr>
<tr><td>IDENTIFIER</td><td colspan="2">Specifies that the use of identifiers as privileges should be audited. For further information, see the VSI OpenVMS Guide to System Security [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/].</td></tr>
</table>

Keyword	Description	
INSTALL	Specifies modifications made to the known file list through the INSTALL utility.	
LOGFAILURE= (*keyword*[,...])	Specifies the occurrence of one or more classes of login failures, as specified by the following keywords:	
	ALL	All possible types of login failures
	BATCH	Batch process login failure
	DETACHED	Detached process login failure
	DIALUP	Dialup interactive login failure
	LOCAL	Local interactive login failure
	NETWORK	Network server task login failure
	REMOTE	Interactive login failure from another network node, for example, with a **SET HOST** command
	SERVER	Server or TCB-based login failure.
	SUBPROCESS	Subprocess login failure
LOGIN= (*keyword*[,...])	Specifies the occurrence of one or more classes of login attempts, as specified by the following keywords. See the LOGFAILURE keyword for further description.	
	ALL DETACHED LOCAL REMOTE SUBPROCESS	BATCH DIALUP NETWORK SERVER
LOGOUT= (*keyword*[,...])	Specifies the occurrence of one or more classes of logouts, as specified by the following keywords. See the LOGFAILURE keyword for further description.	
	ALL DETACHED LOCAL REMOTE SUBPROCESS	BATCH DIALUP NETWORK SERVER
MOUNT	Specifies a mount or dismount operation.	
NCP	Specifies access to the network configuration database, using the network control program (NCP).	
PRIVILEGE= (*keyword*[,...])	Specifies successful or unsuccessful use of privilege, as specified by the following keywords: FAILURE [:*privilege*(,...)] --- Unsuccessful use of privilege SUCCESS [:*privilege*(,...)] --- Successful use of privilege	

Keyword	Description	
	For a listing of privileges, see the online help for the DCL command **SET PROCESS/PRIVILEGES**.	
PROCESS= (*keyword*[,...])	Specifies the use of one or more of the process control system services, as specified by the following keywords:	
	ALL	Use of any of the process control system services
	CREPRC	All use of $CREPRC
	DELPRC	All use of $DELPRC
	SCHDWK	Privileged use of $SCHDWK
	CANWAK	Privileged use of $CANWAK
	WAKE	Privileged use of $WAKE
	SUSPND	Privileged use of $SUSPND
	RESUME	Privileged use of $RESUME
	GRANTID	Privileged use of $GRANTID
	REVOKID	Privileged use of $REVOKID
	GETJPI	Privileged use of $GETJPI
	FORCEX	Privileged use of $FORCEX
	SETPRI	Privileged use of $SETPRI
	Privileged use of a process control system service means the caller used GROUP or WORLD privilege to affect the target process.	
SYSGEN	Specifies the modification of a system parameter with the OpenVMS System Generation utility.	
TIME	Specifies the modification of system time.	

/EXCLUDE=*process-id*
/NOEXCLUDE=*process-id*

Adds a process identification (PID) to the audit server's process exclusion list. The process exclusion list contains those processes that will not be suspended by the audit server if a resource exhaustion reaches the action threshold. By default, real-time processes and all of the following processes are included in the process exclusion list and are never suspended:

CACHE_SERVER
CLUSTER_SERVER
CONFIGURE
DFS$COM_ACP
DNS$ADVER
IPCACP
JOB_CONTROL
NETACP
NET$ACP
OPCOM

REMACP
SHADOW_SERVER
SMISERVER
SWAPPER
TP_SERVER
VWS$DISPLAYMGR
VWS$EMULATORS

Use the **SET AUDIT/NOEXCLUDE** command to remove a process from the process exclusion list; however, processes listed above cannot be removed from the exclusion list. Also note that PIDs are not automatically removed from the process exclusion list when processes log out of the system.

/INTERVAL=(*keyword*[,...])

Specifies the delta times to be used for regular audit server operations. For information about specifying delta times, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT].

The following table describes keywords for the **/INTERVAL** qualifier:

Keyword	Description
ARCHIVE_FLUSH=*time*	Specifies the interval at which data collected by the audit server is written to the archive file. The default is 1 minute.
JOURNAL_FLUSH=*time*	Specifies the interval at which data collected by the audit server is written to the audit log file. The default is 5 minutes.
RESOURCE_MONITOR=*time*	Specifies the interval at which the audit server retries log file allocation or access. This interval applies whenever free space in the log file is below either the warning or action thresholds, or when the volume holding the log file is inaccessible. The default interval is 5 minutes.
RESUME_SCAN=*time*	Specifies the interval at which the audit server reviews an existing resource exhaustion condition. The default is 15 minutes.

/JOURNAL[=*journal-name*]

Specifies the name of the audit journal; the name defaults to SECURITY. (Currently, there is only one journal.)

The **/JOURNAL** qualifier is required when redefining the audit log file or when specifying resource monitoring characteristics with the **/RESOURCE** or the **/THRESHOLD** qualifier.

/LISTENER=*device*
/NOLISTENER

Specifies the name of a mailbox device to which the audit server sends a binary copy of all security audit event messages. Users can create such a mailbox to process system security events as they occur. For a description of the message formats written to the listener mailbox, see the Audit Analysis Utility documentation in the *VSI OpenVMS System Management Utilities Reference Manual, Volume I: A–L* [https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#AUDITING_PART].

Use the **SET AUDIT/NOLISTENER** command to disable a listener device.

/RESOURCE=*keyword*[,...]

Enables or disables the monitoring of disk volumes to ensure adequate space for audit journal entries; it also specifies the monitoring method to use. The **/JOURNAL** qualifier is required. For more information about resource monitoring, see the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#d0e17705].

Keyword	Description
DISABLE	Disables monitoring on the disk volume containing the audit journal.
ENABLE	Enables resource monitoring on the disk volume containing the audit journal.

/SERVER=*keyword*[,...]

Modifies audit server characteristics. The following table describes keywords for the **/SERVER** qualifier:

Keyword	Description
EXIT	Initiates an audit server shutdown. This is the only method for removing the audit server process from the system; the audit server cannot be deleted or suspended.
FINAL_ACTION=*action*	Specifies the action the audit server should take when it runs out of memory and cannot buffer messages. (For more information, see the discussion of message flow control in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#d0e17455].) Specify one of the following actions: CRASH – Crash the system if the audit server runs out of memory. IGNORE_NEW – Ignore new event messages until memory is available. New event messages are lost but event messages in memory are saved. PURGE_OLD (default) – Remove old event messages until memory is available for the most current messages.
FLUSH	Copies all buffered audit and archive records to the security audit log file and security archive file, respectively.
INITIATE	Enables auditing during system startup. Ordinarily, auditing is started from VMS$LPBEGIN in STARTUP.COM but, if a site redefines the logical name SYS$AUDIT_SERVER_INHIBIT, the OpenVMS system waits for a **SET AUDIT/SERVER=INITIATE** command before enabling auditing.
NEW_LOG	Creates a new clusterwide audit log file. Typically, this is used daily to generate a new version of the audit log file. The following sequence of commands can be used to reset the space monitoring thresholds and then to recreate the auditing log, thereby creating a smaller log file:

Keyword	Description

	`$ SET AUDIT /JOURNAL=SECURITY /THRESHOLD=`
	` WARN=200`
	`$ SET AUDIT /SERVER=NEW_LOG`
	
	By default, the size of the new auditing log file is based on the size of the previous auditing logs.

RESUME	Requests the audit server process to resume normal activity on the system, if adequate disk space is available. Normally, once the resource monitoring action threshold has been reached, the audit server process suspends most system activity and waits 15 minutes before attempting to resume normal system activity.
START	Starts the audit server process on the system. In order to fully enable the auditing subsystem, the **SET AUDIT/SERVER=INITIATE** command must be used after the **SET AUDIT/SERVER=START** command has completed. Use the following command procedure to start the audit server: `SYS$SYSTEM:STARTUP AUDIT_SERVER`

/THRESHOLD=*type*=*value*

Specifies threshold values used in monitoring available space in the audit log file. The auditing system issues advisory messages to central and security operators whenever free space in the audit log file falls below the WARNING threshold. The auditing system suspends processes that generate audit events when free disk space is below the action threshold. (See the **/RESOURCE** qualifier). The **/JOURNAL** qualifier is required.

The following table lists the types of thresholds:

Keyword	Description
WARNING=*value*	Specifies the threshold at which the audit server notifies all security operator terminals that resources are getting low.
ACTION=*value*	Specifies the threshold at which the audit server starts suspending processes that are generating audit records. (Certain processes are immune to this: see the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/].

The following table lists the default warning and action values for each monitoring mode:

Mode	Warning	Action
Blocks	100	25
Delta time	2 0:00:00	0 0:30:00

/VERIFY

Do not return the dollar sign ($) prompt until the audit server completes the command. Associated qualifiers determine which of the following actions occur:

- Redefinition of auditing events
- Redefinition of the audit log file or the archive file
- Modification of the audit server's operational characteristics
- Modification of resource monitoring attributes

If you do not want to wait for the command to complete, specify **/NOVERIFY**.

Examples

1.
```
$ SET AUDIT/AUDIT/ENABLE= -
_$ (CREATE,ACCESS=(SYSPRV,BYPASS),DEACCESS)/CLASS=FILE
$ SHOW AUDIT/AUDIT
System security audits currently enabled for:
.
.
.

  FILE access:
    Failure:     read,write,execute,delete,control
    SYSPRV:      read,write,execute,delete,control
    BYPASS:      read,write,execute,delete,control
    Other:       create,deaccess
```

The **SET AUDIT** command in this example enables auditing of file creation and file deaccess; it also enables auditing for any file access done by using either SYSPRV or BYPASS privilege.

2.
```
$ SET AUDIT/JOURNAL=SECURITY/DESTINATION=AUDIT$:[AUDIT]TURIN
$ SET AUDIT/SERVER=NEW
$ SHOW AUDIT/JOURNAL
List of audit journals:
  Journal name:           SECURITY
  Journal owner:          (system audit journal)
  Destination:            AUDIT$:[AUDIT]TURIN.AUDIT$JOURNAL
```

The **SET AUDIT** command in this example demonstrates how to switch to a new journal.

3.
```
$ SET AUDIT/SERVER=FINAL=CRASH
$ SHOW AUDIT/SERVER
Security auditing server characteristics:
  Database version:       4.4
  Backlog (total):        100, 200, 300
  Backlog (process):      5, 2
  Server processing intervals:
    Archive flush:        0 00:01:00.00
    Journal flush:        0 00:05:00.00
    Resource scan:        0 00:05:00.00
  Final resource action:  crash system
```

The **SET AUDIT** command in this example changes the audit server's final action setting so the system crashes when the audit server runs out of memory.

4.
```
$ SET AUDIT/ARCHIVE/DESTINATION=SYS$SPECIFIC:[SYSMGR]TURIN-ARCHIVE
$ SHOW AUDIT/ARCHIVE
Security archiving information:
```

```
 Archiving events:    system audits
 Archive destination: SYS$SPECIFIC:[SYSMGR]TURIN-ARCHIVE.AUDIT$JOURNAL
```

The **SET AUDIT** command in this example enables a node-specific archive file.

5.
```
$ SET AUDIT/JOURNAL/RESOURCE=ENABLE
$ SHOW AUDIT/JOURNAL
List of audit journals:
  Journal name:          SECURITY
  Journal owner:         (system audit journal)
  Destination:           SYS$COMMON:[SYSMGR]SECURITY.AUDIT$JOURNAL
  Monitoring:            enabled
    Warning thresholds,  Block count:   100   Duration:  2 00:00:00.0
    Action thresholds,   Block count:    25   Duration:  0 00:30:00.0
```

The **SET AUDIT** command in this example enables disk monitoring and switches the mode so the disk space is monitored in terms of time rather than free blocks.

SET BOOTBLOCK (IA-64 Only)

SET BOOTBLOCK — Initializes the boot block on the target device.

Format

SET BOOTBLOCK [*boot-file*]

Parameters

boot-file

Specifies the file name of the boot file for the target disk.

Architecture-specific defaults are applied for the boot file. For IA-64, the default is SYS$SYSROOT:[SYS$LDR]SYS$EFI.SYS (SYS$EFI).

The boot file must be contiguous. If the target boot file is not contiguous, use the DCL command **COPY/CONTIGUOUS** or another functionally similar mechanism to recreate a contiguous version of the boot file.

The boot file must also be marked NOMOVE (using the DCL command **SET FILE/NOMOVE**) to avoid a bootstrap failure that could result from the normal and expected operations of disk defragmentation tools.

Description

The **SET BOOTBLOCK** command writes a boot block onto the specified disk.

As an alternative to using the **SET BOOTBLOCK** command, you can use **RUN** to invoke the SETBOOT utility, which then prompts you for all required input.

Qualifier

/BLOCK_SIZE=512 (default)
/BLOCK_SIZE=2048

Specifies the target block size for the bootstrap device, in bytes.

The 512-byte block size applies for most disk devices. The 2048-byte block size typically is used only when writing the boot block to specific OpenVMS IA-64 CD and DVD devices.

/Integrity servers

Specifies OpenVMS IA-64 as the target architecture for the boot block. The default boot file for OpenVMS IA-64 is SYS$SYSROOT:[SYS$LDR]SYS$EFI.SYS.

/PRESERVE=*keyword*
/PRESERVE=SIGNATURE

/PRESERVE=SIGNATURE maintains the existing GUID disk signature value. The default is to generate a new signature for the target volume.

SIGNATURE is the only *keyword* currently supported.

SET BROADCAST

SET BROADCAST — Enables you to selectively screen out various kinds of messages from being broadcast to your terminal.

Format

SET BROADCAST=(*class-name*[,...])

Parameter

***class-name*[,...]**

Specifies the class of message that you want to enable or disable for broadcast to your terminal. If you specify only one class, you can omit the parentheses. The class names are as follows:

Class Name	Description
ALL	Enables all message classes.
[NO]DCL	Specifies **Ctrl/T** and **SPAWN/NOTIFY** messages.
[NO]GENERAL	Specifies all normal **REPLY** messages or messages from $BRDCST.
[NO]MAIL	Specifies notification of mail.
NONE	Disables all message classes.
[NO]OPCOM	Specifies messages issued by the operator communication process (OPCOM).
[NO]PHONE	Specifies messages from the Phone utility.
[NO]QUEUE	Specifies messages referring to print or batch jobs issued by the queue manager.
[NO]SHUTDOWN	Specifies messages issued from the **REPLY/SHUTDOWN** command.
[NO]URGENT	Specifies messages issued from the **REPLY/URGENT** command.
[NO]USER1 to [NO]USER16	Specifies messages from specific user groups. (For information on setting up user-written broadcast messages, see the description of the

Class Name	Description
	$BRKTHRU system service in the *VSI OpenVMS System Services Reference Manual: A–GETUAI* [https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-a-getuai/#JUN_72].)

Description

The **SET BROADCAST** command enables you to receive certain kinds of messages at your terminal, but not others. By default, you receive all messages at your terminal (**SET BROADCAST=ALL**). **SET BROADCAST=NONE** screens out all messages.

After you have used the **SET BROADCAST** command to screen out some classes of messages (for example, **SET BROADCAST=NOPHONE**), you can use the command to restore that class (**SET BROADCAST=PHONE**).

Use the **SHOW BROADCAST** command to see which message classes are currently being screened out.

Note

SYS$COMMAND must be a terminal.

Examples

1. ```
   $ SET BROADCAST=(NOMAIL, NOPHONE)
       .
       .
       .
   $ SET BROADCAST=MAIL
   ```

   In this example, the first **SET BROADCAST** command screens out all mail and phone messages. Later the second **SET BROADCAST** command restores mail messages. Phone messages are still screened.

2. ```
   $ SET BROADCAST=NONE
       .
       .
       .
   $ SET BROADCAST=(SHUTDOWN, URGENT, DCL, OPCOM)
   ```

   In this example, the first **SET BROADCAST** command screens out all messages. Later the second **SET BROADCAST** command restores SHUTDOWN, URGENT, DCL, and OPCOM messages. General, phone, mail, queue, and user messages are still screened.

SET CACHE/RESET

SET CACHE/RESET — Resets systemwide I/O caching statistics for the extended file cache (XFC). **OPER privilege is required to execute this command.**

Format

SET CACHE/RESET

Parameters

None.

Description

The **SET CACHE/RESET** command resets several of the systemwide I/O caching statistics that are displayed by the **SHOW MEMORY/CACHE** and **SDA SHOW MEMORY** commands. Statistics that are reset include: I/O counts, hit rates, hit counts, reads bypassing cache, and the read/write ratio.

A timestamp displays the time the cache was last reset, thus subsequent **SHOW MEMORY/CACHE** commands display statistics from this time.

Example

```
$ SHOW MEMORY/CACHE
              System Memory Resources on 22-AUG-2017 11:22:22.50

Extended File Cache  (Time of last reset: 16-AUG-2017 11:16:24.96)
 Allocated (Mbytes)     151.39   Maximum size (Mbytes)    1024.00
 Free (Mbytes)            0.21   Minimum size (Mbytes)       0.23
 In use (Mbytes)        151.18   Write hit rate                 0%
 Read hit rate              97%  Write I/O count             5321
 Read I/O count         208052   Write hit count                0
 Read hit count         203761   Writes bypassing cache         0
 Reads bypassing cache    1787   Vols in Full XFC mode          0
 Files cached open         316   Vols in VIOC Compatible mode   1
 Files cached closed       300   Vols in No Caching mode        0
 Read/Write ratio           97%  Vols in Perm. No Caching mode  0
```

This example shows the cache statistics prior to issuing the **SET CACHE/RESET** command.

```
$ SET CACHE/RESET
```

This command resets the cache statistics and updates the timestamp.

```
$ SHOW MEMORY/CACHE
              System Memory Resources on 22-AUG-2017 11:27:41.11

Extended File Cache  (Time of last reset: 22-AUG-2017 11:27:37.76)
 Allocated (Mbytes)     151.39   Maximum size (Mbytes)    1024.00
 Free (Mbytes)            0.21   Minimum size (Mbytes)       0.23
 In use (Mbytes)        151.18   Write hit rate                 0%
 Read hit rate             100%  Write I/O count                0
 Read I/O count              9   Write hit count                0
 Read hit count              9   Writes bypassing cache         0
 Reads bypassing cache       0   Vols in Full XFC mode          0
 Files cached open         316   Vols in VIOC Compatible mode   1
 Files cached closed       300   Vols in No Caching mode        0
 Read/Write ratio          100%  Vols in Perm. No Caching mode  0
```

This example shows the reset cache statistics and the modified timestamp.

SET CARD_READER

SET CARD_READER — Defines the default translation mode for cards read from a card reader. All subsequent input read from the specified card reader is converted using the specified mode.

Format

SET CARD_READER *device-name*[:]

Parameter

***device-name*[:]**

Specifies the name of the card reader for which the translation mode is to beset. The device must not be currently allocated to any other user.

Description

When the system is bootstrapped, the translation mode for cards read into all card readers is set at 029. If you do not specify either of the command qualifiers, the **SET CARD_READER** command has no effect; that is, the current translation mode for the device remains the same.

Qualifiers

/026

Sets the card reader for cards punched on an 026 punch.

/029

Sets the card reader for cards punched on an 029 punch.

/LOG
/NOLOG (default)

Controls whether log information is displayed at the terminal to confirm that the card reader is set.

Example

```
$ ALLOCATE CR:
  _CRA0: ALLOCATED
$ SET CARD_READER CRA0:/029
$ COPY  CRA0: [PEARLMAN.DATAFILES]CARDS.DAT
```

The **ALLOCATE** command requests the allocation of a card reader by specifying the generic device name. When the **ALLOCATE** command displays the name of the device, the **SET CARD_READE**R command sets the translation mode at `029`. Then the **COPY** command copies all the cards read by the card reader `CRA0` into the file `CARDS.DAT` in the directory `[PEARLMAN.DATAFILES]`.

SET CLUSTER/EXPECTED_VOTES

SET CLUSTER/EXPECTED_VOTES — Sets the total expected votes in the OpenVMS Cluster to a value that you specify or, if no value is specified, sets the total votes to a value determined by the system. **Requires OPER (operator) privilege.**

Format

SET CLUSTER/EXPECTED_VOTES[=*value*]

Parameter

value

Specifies the total number of expected votes in the cluster.

Description

The **SET CLUSTER/EXPECTED_VOTES** command enables you to adjust the total number of expected votes in the cluster. Set this value equal to the number of votes contributed by each node plus the number of votes contributed by the cluster quorum disk. The system will automatically calculate the value of the cluster quorum from the total number of expected votes in the cluster.

You can specify the expected total votes value as part of the **SET CLUSTER/EXPECTED_VOTES** command string. If you enter the command without specifying a value for expected votes, the system calculates the value for you, using the following formula:

```
EXPECTED_VOTES=(NODE1_VOTES+NODE2_VOTES+...)+QUORUM_DISK_VOTES
```

NODE *n*_VOTES is the value of the system parameter VOTES for each node in the cluster and QUORUM_DISK_VOTES is the value of the system parameter QDSKVOTES.

When you enter the **SET CLUSTER/EXPECTED_VOTES** command without specifying a value, the system assumes that all nodes that are expected to be in the cluster are currently members.

In general, you use the **SET CLUSTER/EXPECTED_VOTES** command only when a node is leaving the cluster for an extended period of time. Under normal circumstances, quorum is not reduced when a node leaves the cluster, because it is assumed that the node may be rebooted and rejoin the cluster. If a node is removed and is unable to rejoin the cluster within a reasonable period of time (for example, if a node crashes due to a hardware problem and cannot rejoin the cluster for several days), the quorum for the cluster can safely be reduced by lowering the total expected votes until that node rejoins.

The purpose of a cluster quorum is to eliminate any possibility of the cluster partitioning into separate clusters and simultaneously accessing the same resources (such as disks). If the sum of the votes for all members of the cluster is smaller than the cluster quorum, all nodes will block activity until new nodes join to increase the vote total. Lowering the quorum value (by reducing the value of the total expected votes) when one or more nodes leave the cluster for long periods of time reduces this possibility.

Note that no matter what value you specify for the **SET CLUSTER/EXPECTED_VOTES** command, you cannot increase quorum to a value that is greater than the number of the votes present, nor can you reduce quorum to a value that is half or fewer of the votes present.

When you enter the **SET CLUSTER/EXPECTED_VOTES** command, either with or without an expected votes value specified, the system responds with a message indicating the new value that was actually set. Note that you need to enter this command only on one node in the cluster, because the new value for total expected votes is propagated through the cluster. This new expected votes value should then be stored in the system parameter EXPECTED_VOTES on each node, so that it remains in effect after the nodes reboot.

When a node that was previously a member of the cluster is ready to rejoin, you should increase the system parameter EXPECTED_VOTES to its original value before bringing the node back to the cluster. Note that you do not need to use the **SET CLUSTER/EXPECTED_VOTES** command to increase the number of expected votes, because the expected votes value will be increased automatically when the node rejoins the cluster.

Examples

1. `$ SET CLUSTER/EXPECTED_VOTES`

   The **SET CLUSTER** command in this example instructs the system to calculate the total expected votes value for you, because no value is specified as part of the command string. The system uses the NODE*n*_VOTES + QUORUM_DISK_VOTES formula.

2. `$ SET CLUSTER/EXPECTED_VOTES=9`

   The **SET CLUSTER** command in this example sets the total expected votes to `9`, which is the value specified in the command string.

SET COMMAND

SET COMMAND — Invokes the Command Definition Utility, which adds commands to your process command table or to a specified command table file. For a complete description of the Command Definition Utility, see the relevant section in the *VSI OpenVMS Command Definition, Librarian, and Message Utilities Manual* [https://docs.vmssoftware.com/vsi-openvms-command-definition-librarian-and-message-utilities/#CD_CHAP].

Format

SET COMMAND [*filespec*[,...]]

Parameter

[*filespec*[,...]]

Specifies the name of one or more command definition files. If you specify more than one name, separate them with commas. The default file type is .CLD. See the relevant section in the *VSI OpenVMS Command Definition, Librarian, and Message Utilities Manual* [https://docs.vmssoftware.com/vsi-openvms-command-definition-librarian-and-message-utilities/#CDU_WRITING_DEFINITION_FILE] for more information about writing a command definition file.

The asterisk (*) and the percent sign (%) wildcard characters are allowed in the file specification.

Qualifiers

/DELETE=(*verb*[,...])

Specifies **/DELETE** mode to delete verbs from the command table you are modifying.

You can use the **/DELETE** qualifier to delete a verb in either your process command table or in a command table file specified with the **/TABLE** qualifier. If you do not use the **/TABLE** qualifier to specify an alternate command table, the default is to delete verbs from your process command table. If you do not use the **/OUTPUT** qualifier to specify an output file, the default is to return the modified command table to your process.

The verb specifies a verb to be deleted. If you specify two or more verbs, separate them with commas and enclose the list in parentheses. If you specify only one verb, you can omit the parentheses.

You cannot use the **/LISTING**, **/OBJECT**, or **/REPLACE** qualifiers in **/DELETE** mode.

/LISTING[=*filespec*]
/NOLISTING

Controls whether an output listing is created and optionally provides an output file specification for the listing file. A listing file contains a listing of the command definitions along with any error messages. The listing file is similar to a compiler listing.

If you specify the **/LISTING** qualifier and omit the file specification, output is written to the default device and directory; the listing file will have the same name as the first command definition file and a file type of .LIS.

You can use the **/LISTING** qualifier only in **/OBJECT** or **/REPLACE** mode; you cannot create a listing in **/DELETE** mode. In **/OBJECT** and **/REPLACE** modes, the default is **/NOLISTING**.

/OBJECT[=*filespec*]

Specifies **/OBJECT** mode to create an object module from a command definition file and optionally provides an object file specification. You cannot use the **/OBJECT** qualifier to create an object module from a command definition that contains the IMAGE keyword.

You can specify only one command definition file when you use **SET COMMAND/OBJECT**.

If you specify the **/OBJECT** qualifier and omit the file specification, output is written to the default device and directory; the object file will have the same name as the input file and a file type .OBJ.

You cannot use the **/DELETE**, **/OUTPUT**, **/REPLACE**, or **/TABLE** qualifiers in **/OBJECT** mode.

Note

When compiling very large command definition (CLD) files, you may encounter the following error:

```
%CDU-F-INTNODESPACE, Internal error: node space exhausted
```

The default value of virtual memory allocated by the **SET COMMAND/OBJECT** command is 384KB. This value may not be sufficient to process large CLD files. To increase this size, use the CDU$TABLE_SPACE logical name to specify a larger value.

For example:

```
$ DEFINE CDU$TABLE_SPACE 2048
```

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the modified command table should be placed. If you provide an output file specification, the modified command table is written to the specified file. If you do not provide an output file specification, the edited command table replaces your process table. The **/NOOUTPUT** qualifier indicates that no output is to be generated.

You can specify an output file specification with the **/OUTPUT** qualifier only when you also use the **/TABLE=*filespec*** qualifier to provide the input tables. The default file type is .EXE.

You can use the **/OUTPUT** qualifier only in **/DELETE** or **/REPLACE** mode; you cannot use the **/OUTPUT** qualifier in **/OBJECT** mode.

In **/DELETE** and **/REPLACE** mode, the default is **/OUTPUT** with no file specification.

/REPLACE

Specifies **/REPLACE** mode to add or replace verbs in the command table you are modifying.

You can use the **/REPLACE** qualifier to modify either the process command table or a command table file specified with the **/TABLE** qualifier. If you do not use the **/TABLE** qualifier to specify an alternate command table, the default is to modify your process command table. If you do not use the **/OUTPUT** qualifier to specify an output file, the default is to return the modified command table to your process.

You cannot use the **/OBJECT** or **/DELETE** qualifiers in **/REPLACE** mode.

If you do not explicitly specify **/DELETE**, **/OBJECT**, or **/REPLACE** mode, the default is **/REPLACE**.

/TABLE=[*filespec*]

Specifies the command table that is to be modified. If you specify the **/TABLE** qualifier and omit the file specification, the current process command table is modified. If you include a file specification, the specified command table is modified. The default file type is .EXE.

If you use the **/TABLE** qualifier to provide an input command table file, you should also use the **/OUTPUT** qualifier to provide an output table file; otherwise, the modified command table will be written to your process and will replace your process command table.

You can only use the **/TABLE** qualifier in **/DELETE** or **/REPLACE** mode; you cannot use the **/TABLE** qualifier in **/OBJECT** mode.

In **/REPLACE** and **/DELETE** mode, the default is **/TABLE** with no input file specification.

Examples

1. `$ SET COMMAND SNAG`

   This command adds the commands in `SNAG.CLD` to the command table in the current process.

2. `$ SET COMMAND/OBJECT SNAG`

   This command creates an object file using the commands in `SNAG.CLD`. This object file can then be linked with other object files to process application defined commands.

3. `$ SET COMMAND/TABLE=MYTAB/OUTPUT=MYCLI SNAG`

   This command adds the commands in `SNAG.CLD` to the command table `MYTAB.EXE` and writes the updated table to `MYCLI.EXE`.

   When you use the **/TABLE** qualifier to specify an input command table, you should use the **/OUTPUT** qualifier to specify an output file; otherwise, your process command table will be overwritten.

4. `$ SET COMMAND/DELETE=HOLD`

   This command deletes the definition for the verb `HOLD` from your process command table. The modified command table is returned to your process.

SET CONTROL

SET CONTROL — Enables or disables the **Ctrl/Y** or the **Ctrl/T** function. Pressing **Ctrl/Y** interrupts a command and returns you to the DCL command level. Pressing **Ctrl/T** momentarily interrupts a command to print a line of statistics. **`SET CONTROL=T` requires that `SET TERMINAL/BROADCAST` be set for the information to be displayed at your terminal.**

Format

`SET CONTROL[=(T,Y)]`

`SET NOCONTROL[=(T,Y)]`

Parameter

`(T,Y)`

Specifies that T (**Ctrl/T**) or Y (**Ctrl/Y**) be enabled or disabled. If you specify only one character, you can omit the parentheses. If you do not specify either T or Y, Y is the default.

Description

The **Ctrl/Y** function provides a general-purpose escape from the current operation. The **Ctrl/Y** function can generally be used during an interactive terminal session to interrupt the current command, command procedure, or program image.

The **`SET NOCONTROL=Y`** command can be used for special application programs. When the **`SET NOCONTROL=Y`** command is executed in a system-specified command procedure for a particular user at login, that user can communicate only with the application program that controls the terminal.

When you press **Ctrl/Y** and **`SET NOCONTROL=Y`** is in effect, the INTERRUPT message is displayed, but no interruption takes place.

Note

Note that DCL maintains a pending **Ctrl/Y** asynchronous system trap (AST) to the terminal driver. This affects captive command procedures when using the **`SET HOST`** command. For more information, see the description of the **`SET HOST`** command.

`SET NOCONTROL=Y` also disables the **Ctrl/C** cancel function for all commands and programs that do not have special action routines responding to the **Ctrl/C** function.

The **Ctrl/T** function displays a single line of statistical information about the current process. When you press **Ctrl/T** during an interactive terminal session, it momentarily interrupts the current command, command procedure, or image to display statistics. The statistical information includes the node and user names, the current time, the current process, CPU usage, number of page faults, level of I/O activity, and memory usage. For example:

```
BOSTON::SMITH  16:21:04 EDT       CPU=00:00:03.33 PF=778 IO=296 MEM=277
```

When **`SET NOCONTROL=T`** (the default) is in effect, pressing **Ctrl/T** does not cause any statistics to be displayed.

Customizing the Output of Ctrl/T

By defining the new symbol DCL$CTRLT, users can augment the traditional **Ctrl/T** output with the text defined for DCL$CTRLT. This technique can be used to indicate progress in user applications or for debugging purposes.

Examples

1.
```
$ SET NOCONTROL=Y
```

   The **SET CONTROL** command in this example disables the **Ctrl/Y** function as well as most **Ctrl/C** functions.

2.
```
$ SET CONTROL=T
```

   The **SET CONTROL** command in this example enables the **Ctrl/T** function.

3.
```
$ SET NOCONTROL=(T,Y)
```

   The **SET CONTROL** command in this example disables both the **Ctrl/T** and **Ctrl/Y** functions.

4.
```
$ Ctrl/T
NODE22::SMITH 16:21:04 (DCL) CPU=00:03:29.39 PF=14802 IO=18652 MEM=68
$ SET NOCONTROL=T
$ Ctrl/T
```

   As shown in this example, when you press **Ctrl/T**, the system displays the appropriate information. The **SET NOCONTROL=T** command disables the **Ctrl/T** function. Now when you press **Ctrl/T**, no information is displayed.

5.
```
$ TYPE CTRLT_LOOP.COM
$ inner=0
$ outer=0
$ loop:
$ loop1:
$ if inner .gt.  20000 then goto end_loop1
$ inner=inner+1
$ dcl$ctrlt=F$FAO("Inner loop count is !SL Outer loop count is !
SL" ,inner,outer)
$ goto loop1
$ end_loop1:
$ inner=0
$ outer=outer+1
$ goto loop
$ @CTRLT_LOOP

<<Press Ctrl/T>>
NODE1::JSMITH 10:46:37 (DCL) CPU=00:03:42.68 PF=13453 IO=6743 MEM=187
Inner loop count is 12306
Outer loop count is 0
NODE1::JSMITH 10:46:43 (DCL) CPU=00:03:49.19 PF=13455 IO=6744 MEM=187
Inner loop count is 19200
Outer loop count is 2
  .
  .
  .
<<Press Ctrl/C>> <<--- To exit the loop.
```

```
$ SET NOCONTROL=T
$ @CTRLT_LOOP
<<Press Ctrl/T>>
<<Nothing is displayed as CTRL/T is disabled.>>
<<Press Ctrl/C>> <<--- To exit the loop.
```

This example demonstrates the use of DCL$CTRLT within a command procedure. The command procedure is running in a loop that updates the symbol DCL$CTRLT to indicate the number of loop iterations executed so far.

SET CPU

SET CPU — Changes the user capabilities associated with the specified CPUs. **Requires ALTPRI and WORLD privileges.**

Format

SET CPU [*cpu-id*[, ...]]

Parameter

cpu-id

Specifies a decimal value representing the identity of a processor in an OpenVMS multiprocessing system. On an Alpha 7000 system, the CPU is the backplane slot number of the processor.

Qualifiers

/ALL

Applies the specified operation to all CPUs in the active set.

/ASSIGN [=*option*] (Alpha/IA-64 only)

Assigns the CPU to the current instance if no option is specified.

If an option is specified, the CPU is assigned to that resource.

Option	Description
$$PARTITION	The current soft partition of the OpenVMS instance.
$$HARD_PARTITION	The hard partition of the OpenVMS instance.

Supported only on AlphaServer systems that support partitioning.

/AUTO_START (Alpha/IA-64 only)
/NOAUTO_START

Sets or clears the instance-specific autostart flag for the specified CPUs.

When autostart is enabled, that CPU will join the OpenVMS active set when it is assigned or migrated into the partition. The CPU will also autostart if a power-up transition is completed while the CPU is owned by the issuing instance.

Supported only on AlphaServer systems that support partitioning.

/CAPABILITY
/NOCAPABILITY

Allows bits in the process user capability mask to be set or cleared individually, in groups, or all at once.

Note

The **SET CPU/[NO]CAPABILITY** command will fail if there is an active process that requires a capability that is not available on any remaining active CPU.

Specifying the **/CAPABILITY** qualifier has no direct effect, but merely indicates the target of the operations specified by the following secondary qualifiers:

Qualifier	Description
/SET=(*n*[,...])	Sets all user capabilities defined by the position values *n*, where *n* has the range of 1 to 16.
/CLEAR=(*n*[,...])	Clears all user capabilities defined by the position values *n*, where *n* has the range of 1 to 16.

The secondary qualifiers can all be used at once as long as the user capability bits defined in the **/SET** and **/CLEAR** parameters do not overlap.

The **/NOCAPABILITY** qualifier clears all user capability bits.

/CLEAR=(*n*[,...])

Clears all user capabilities defined by the position values *n* specified by the **/CAPABILITY** qualifier.

/FAILOVER=*option* (Alpha/IA-64 only)
/NOFAILOVER

Establishes instance-specific failover relationships for each CPU in the instance's potential set.

When the instance crashes, CPUs with a failover target other than the current instance will be assigned or migrated to that target.

Specifying the **/FAILOVER** qualifier has no direct effect; however, it indicates the target of the operations specified by the following options:

Option	Description
instance_name	The name of any valid running instance in the current hard partition.
partitionID	The numeric ID of any partition(reflected in the configuration tree) in the current hard partition. An operating system instance is not required to be running with this identifier.
$$PARTITION	Assigns a resource to the current partition of the current instance.
$$HARD_PARTITION	Assigns a resource to the hard partition node of the configuration tree. This makes the CPU available to multiple soft partitions below it.

The **/FAILOVER** qualifier generates a resource assignment only when the current instance crashes.

Supported only on AlphaServer systems that support partitioning.

/GLOBAL

Modifies the global cell SCH$GL_DEFAULT_CPU_CAP. This global cell is used to initialize the user capability mask of CPUs that are brought into the active set for the first time.

/MIGRATE=*option* (Alpha/IA-64 only)

Transfers ownership of the CPU from the current instance to another soft partition.

Option	Description
instance_name	The name of any valid running instance in the current hard partition.
partitionID	The numeric ID of any partition(reflected in the configuration tree) in the current hard partition. An operating system instance is not required to be running with this identifier.

Supported only on AlphaServer systems that support partitioning.

/POWER=*option* (Alpha/IA-64 only)

Turns the power on or off in one or more CPU slots. Valid options are ON and OFF.

Supported only on AlphaServer GS series systems.

/OVERRIDE_CHECKS (Alpha/IA-64 only)

Directs the **SET/CPU** command to bypass a series of checks that determine whether the specified processor is eligible for removal from the active set.

/REFRESH (Alpha/IA-64 only)

Examines and updates the OpenVMS context for the specified CPU or CPUs, using the hardware configuration tree.

/SET=(*n*[,...])

Sets all user capabilities defined by the position values *n* specified by the **/CAPABILITY** qualifier.

/START (Alpha/IA-64 only)

Initiates a request for the specified CPU to join the OpenVMS active set if it is not already there.

Examples

1. `$ SET CPU/CAPABILITY/GLOBAL/SET=(2,4)`

   This command sets user capabilities `2` and `4` in the global cell SCH$GL_DEFAULT_CPU_CAP.

2. `$ SET CPU/CAPABILITY/GLOBAL/SET=(2,4) 0`

This command sets user capabilities 2 and 4 in the global cell SCH$GL_DEFAULT_CPU_CAP, as well in the active CPU 0.

3.
```
$ SET CPU/CAPABILITY/SET=(1,2) 0
$ SHOW CPU/FULL
SYSTEM: EUROS, HPE RX2600  (900MHZ/1.5MB)

  SMP EXECLET   = 3 : ENABLED : STREAMLINED.
  CONFIG TREE   = VERSION 6
  PRIMARY CPU   = 0
  HWRPB CPUS    = 2
  PAGE SIZE     = 8192
  REVISION CODE =
  SERIAL NUMBER = US30464615
  DEFAULT CPU CAPABILITIES:
        SYSTEM: QUORUM RUN
  DEFAULT PROCESS CAPABILITIES:
        SYSTEM: QUORUM RUN

CPU 0    STATE: RUN               CPUDB: 8901C000     HANDLE: 00005D70
         OWNER: 000004C8         CURRENT: 000004C8     PARTITION 0
 (EUROS)
        CHGCNT:        1           STATE: PRESENT, PRIMARY,
 REASSIGNABLE
       PROCESS: TEST11               PID: 20201185
  CAPABILITIES:
        SYSTEM: PRIMARY QUORUM RUN RAD0
          USER: USR1 USR2
  SLOT CONTEXT: 901EB000
     CPU     -  STATE.........: RC, PA, PP, CV, PV, PMV, PL
                TYPE..........: ITANIUM MAJOR = 31,  MINOR = 0
                SPEED.........: 900 MHZ
                LID...........: 00000000
                VARIATION.....: IEEE FP, PRIMARY ELIGIBLE
                SERIAL NUMBER..: 0001A968930A1A25
                REVISION.....:
                HALT REQUEST...: 0
                SOFTWARE COMP..: 7.0
     PALCODE -  REVISION CODE..: 1.1
                COMPATIBILITY..: 0
                MAX SHARED CPUS: 0
  BINDINGS:     * NONE *
  FASTPATH:
        FGB0
        PKA0
        PEA0
        EIA0
  FEATURES:
     AUTOSTART - ENABLED.
     FASTPATH  - SELECTION ENABLED AS PREFERRED CPU.

CPU 1    STATE: RUN               CPUDB: 89163480     HANDLE: 00005E80
         OWNER: 000004C8         CURRENT: 000004C8     PARTITION 0
 (EUROS)
        CHGCNT:        1           STATE: PRESENT, REASSIGNABLE
       PROCESS: * NONE *
  CAPABILITIES:
```

```
         SYSTEM: QUORUM RUN RAD0
   SLOT CONTEXT: 901EC000
      CPU     -  STATE.....: RC, PA, PP, CV, PV, PMV, PL
                 TYPE.....: ITANIUM MAJOR = 31,  MINOR = 0
                 SPEED.....: 900 MHZ
                 LID.....: 01000000
                 VARIATION.....: IEEE FP
                 SERIAL NUMBER..: 000363B4B0C5C862
                 REVISION.....:
                 HALT REQUEST...: 0
                 SOFTWARE COMP..: 7.0
      PALCODE -  REVISION CODE..: 1.1
                 COMPATIBILITY..: 0
                 MAX SHARED CPUS: 0
   BINDINGS:     * NONE *
   FASTPATH:
         FGA0
         EWA0
         PKB0
         BG0
   FEATURES:
      AUTOSTART - ENABLED.
      FASTPATH  - SELECTION ENABLED AS PREFERRED CPU.

$ SET CPU/CAPABILITY/CLEAR=1 0

$ SHOW CPU/FULL
SYSTEM: EUROS, HPE RX2600  (900MHZ/1.5MB)

SMP EXECLET   = 3 : ENABLED : STREAMLINED.
  CONFIG TREE   = VERSION 6
  PRIMARY CPU   = 0
  HWRPB CPUS    = 2
  PAGE SIZE     = 8192
  REVISION CODE =
  SERIAL NUMBER = US30464615
  DEFAULT CPU CAPABILITIES:
         SYSTEM: QUORUM RUN
  DEFAULT PROCESS CAPABILITIES:
         SYSTEM: QUORUM RUN
CPU 0    STATE: RUN               CPUDB: 8901C000     HANDLE: 00005D70
          OWNER: 000004C8         CURRENT: 000004C8     PARTITION 0
 (EUROS)
         CHGCNT:        1           STATE: PRESENT, PRIMARY,
 REASSIGNABLE
       PROCESS: TEST11               PID: 20201185
  CAPABILITIES:
         SYSTEM: PRIMARY QUORUM RUN RAD0
           USER: USR2
  SLOT CONTEXT: 901EB000
      CPU     -  STATE.....: RC, PA, PP, CV, PV, PMV, PL
                 TYPE.....: ITANIUM MAJOR = 31,  MINOR = 0
                 SPEED.....: 900 MHZ
                 LID.....: 00000000
                 VARIATION.....: IEEE FP, PRIMARY ELIGIBLE
                 SERIAL NUMBER..: 0001A968930A1A25
                 REVISION.....:
                 HALT REQUEST...: 0
```

```
                  SOFTWARE COMP..: 7.0
      PALCODE -   REVISION CODE..: 1.1
                  COMPATIBILITY..: 0
                  MAX SHARED CPUS: 0
    BINDINGS:     * NONE *
    FASTPATH:
          FGB0
          PKA0
          PEA0
          EIA0
    FEATURES:
       AUTOSTART - ENABLED.
       FASTPATH  - SELECTION ENABLED AS PREFERRED CPU.

  CPU 1    STATE: RUN               CPUDB: 89163480     HANDLE: 00005E80
           OWNER: 000004C8         CURRENT: 000004C8     PARTITION 0
   (EUROS)
          CHGCNT:        1           STATE: PRESENT, REASSIGNABLE
         PROCESS: * NONE *
    CAPABILITIES:
          SYSTEM: QUORUM RUN RAD0
    SLOT CONTEXT: 901EC000
       CPU     -  STATE.........: RC, PA, PP, CV, PV, PMV, PL
                  TYPE..........: ITANIUM MAJOR = 31,  MINOR = 0
                  SPEED.........: 900 MHZ
                  LID...........: 01000000
                  VARIATION.....: IEEE FP
                  SERIAL NUMBER..: 000363B4B0C5C862
                  REVISION.....:
                  HALT REQUEST...: 0
                  SOFTWARE COMP..: 7.0
       PALCODE -  REVISION CODE..: 1.1
                  COMPATIBILITY..: 0
                  MAX SHARED CPUS: 0
    BINDINGS:     * NONE *
    FASTPATH:
          FGA0
          EWA0
          PKB0
          BG0
    FEATURES:
       AUTOSTART - ENABLED.
       FASTPATH  - SELECTION ENABLED AS PREFERRED CPU.
  $
```

In this example, the first **SET CPU** command sets user capabilities `1` and `2` for CPU `0`. The second **SET CPU** command clears capability `1`.

4. `$ SET CPU/REFRESH/ALL`

   This command updates the OpenVMS context in this instance for all CPUs physically residing in the current hard partition.

5. `$ SET CPU/FAILOVER=WFGLXA 12`

   This command sets CPU `12` to failover to another instance (`WFGLXA`), if the current instance crashes.

6. `$ SET CPU/FAILOVER=$$HARD 13`

   This command assigns CPU `13` to hard partition ownership if the current instance crashes. Assignment to the hard partition makes the CPU visible and available to other instances for assignment.

7. `$ SET CPU/ASSIGN=$$HARD 14`

   This command assigns CPU `14`, currently owned by the instance, to hard partition ownership. Assignment to the hard partition makes the CPU visible and available to other instances for assignment.

8. `$ SET CPU/ASSIGN 15`

   This command assigns CPU `15`, currently owned by the hard partition, to this instance. If the CPU is powered, in the potential set, and autostart enabled, then it will automatically join the active set.

SET DAY

SET DAY — Sets the default day type specified in the user authorization file (UAF) for the current day. **Requires OPER (operator) privilege.**

Format

SET DAY

Parameters

None.

Description

The **SET DAY** command overrides the default primary and secondary day types that are used to control daily logins. These default day types are defined by the system manager in the user authorization file (UAF).

The **SET DAY** command is useful when you need to override the day type because of a change in the work days of a particular week. For example, Monday, which is normally a work day, is occasionally a holiday. You can use the **SET DAY** command to override the normal day type for Monday and set it to a different day type.

Qualifiers

/DEFAULT

Overrides any previous **SET DAY** specification and specifies that the normal UAF defaults are to be used to determine today's day type.

/LOG
/NOLOG (default)

Controls whether log information is displayed at the terminal to confirm that the new **SET DAY** information has been set.

/PRIMARY

Sets today until midnight to a primary day.

/SECONDARY

Sets today until midnight to a secondary day.

Examples

1. `$ SET DAY/PRIMARY`

   The **SET DAY** command in this example overrides the current default day type and sets today until midnight to a primary day.

2. `$ SET DAY/DEFAULT`

   The **SET DAY** command in this example overrides the previous **SET DAY** command and sets today's day type to the UAF-defined default.

SET DEFAULT

SET DEFAULT — Sets your default device and directory specifications. The new default is applied to all subsequent file specifications that do not explicitly include a device or directory name. When you change the default device assignment, the system equates the specified device with the logical name SYS$DISK.

Format

SET DEFAULT [*device-name*[:]][*directory-spec*]

Parameter

***device-name*[:]**

Specifies the name of the device you want to go to.

Note

The **SET DEFAULT** command will accept a node name with the device name(that is, node-name::device-name). However, the results of operations using a default with a node name are unpredictable. Some utilities and applications do not allow a node name to be specified in a file specification and will fail if the default file specification contains a node name.

directory-spec

Specifies the name of the directory you want to go to. A directory name must be enclosed in brackets ([ ]). Use the minus sign (-) to specify the next higher directory from the current default.

On Alpha systems, if the length of the resulting default directory specification exceeds 255 characters, the **SET DEFAULT** attempts to replace the default directory string with a generated directory

specification. If this happens, then in addition to the normal syntax check, the entire path to that specification, including the device, is verified and must exist for the call to succeed.

On Alpha systems, if the process parse style is set to extended, then the directory specification may include a directory ID (DID).

You must specify either the *device-name* parameter or the *directory-spec* parameter. If you specify only the device name, the current directory is the default for the *directory-spec* parameter. If you specify only the directory name, the current device is the default for the *device-name* parameter.

You can use a logical name but it must constitute at least the device part of the specification. When you use a search list logical name as the parameter, the logical name is not translated by the **SET DEFAULT** command. Instead, the **SET DEFAULT** command retains the logical name so that OpenVMS Record Management Services (RMS) is able to access the entire search list. If you enter the **SHOW DEFAULT** command, the search list logical name is displayed as the default device, along with its equivalence strings.

Examples

1. `$ SET DEFAULT [TANNER]`

   The **SET DEFAULT** command in this example changes the default directory to `[TANNER]`. The default disk device does not change.

2. `$ SET DEFAULT $FLOPPY1:[MOREAU.MEMOS]`

   The **SET DEFAULT** command in this example sets your default to the `MOREAU.MEMOS` subdirectory on `$FLOPPY1`.

3. `$ SET DEFAULT $FLOPPY1:`

   The **SET DEFAULT** command in this example sets the default device to `$FLOPPY1`. The directory name does not change.

4. `$ SET DEFAULT [-]`

   The **SET DEFAULT** command in this example changes the default directory to the parent directory of the one you are currently in. For example, if the current directory is $FLOPPY1:[NIELSEN.MEMOS], this command sets your default to $FLOPPY1:[NIELSEN]. If you are in $FLOPPY1:[NIELSEN], this command sets your default to the master directory on the disk---$FLOPPY1:[000000].

5. ```
   $ SAVEDEF = F$ENVIRONMENT("DEFAULT")
   $ SET DEFAULT [122001.ZAMORA.APP10]    .    .    .
   $ SET DEFAULT  'SAVEDEF'
   ```

   The command procedure in this example uses the `F$ENVIRONMENT` lexical function to save the current default directory in the symbol named `SAVEDEF`. The **SET DEFAULT** command changes the default directory `122001.ZAMORA.APP10`. Later, the symbol `SAVEDEF` is used to restore the original default directory.

6. ```
   $ SHOW DEFAULT
   WORK:[TOP]
   $ DEFINE X WORK:[TOP.SUB1],WORK:[TOP.SUB2]
   $ SET DEFAULT X
   $ SHOW DEFAULT
   X:[TOP]
   ```

```
 =   WORK:[TOP.SUB1]
 =   WORK:[TOP.SUB2]
$ DIRECTORY
Directory WORK:[TOP.SUB1]
KUDOS.TMP;1
Total of 1 file.
Directory WORK:[TOP.SUB2]
KUDOS.TMP;1
Total of 1 file.
Grand total of 2 directories, 2 files.
$ DIRECTORY []
Directory WORK:[TOP]
KUDOS.TMP;1           NETSERVER.LOG;2
Total of 2 files.
```

In this example, the default directory is `WORK:[TOP].X` is then defined to be a search list consisting of two subdirectories. When the **SETDEFAULT X** command is entered, the search list (`X`) is equated with the logical name SYS$DISK and is entered into the disk field. The subsequent **SHOW DEFAULT** command shows both the search list and the current default directory, followed by the expanded search list.

If a **DIRECTORY** command is entered, the directories searched are those contained in the logical name `X`; however, if the current default directory specification ([]) is explicitly entered, the current default directory, rather than SYS$DISK, is searched.

7.
```
$ SET PROCESS/PARSE_STYLE=EXTENDED
$ SET DEFAULT WORK:[11,1,0]
$ SHOW DEFAULT
  WORK:[11,1,0]
$ DIRECTORY
Directory WORK:[11,1,0]
X.X;1
Total of 1 file.
```

On Alpha systems, the **SET DEFAULT** command in this example uses a directory ID (DID) in the directory specification.

8.
```
$ SET PROCESS/PARSE_STYLE=EXTENDED
$ SET DEFAULT -
  WORK:[.LONG_DIRECTORY_NAME_AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA]
$ SET DEFAULT -
  WORK:[.LONG_SUBDIRECTORY_NAME_BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB]
$ SET DEFAULT -
  WORK:[.LONG_SUBDIRECTORY_NAME_CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC]
$ SET DEFAULT -
  WORK:[.LONG_SUBDIRECTORY_NAME_DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD]
$ SET DEFAULT -
  WORK:[.LONG_SUBDIRECTORY_NAME_EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE]
$ SHOW DEFAULT
  MDA0:[15,1,0]
$ SET DEFAULT [-]
%RMS-F-DIR, error in directory name
$ SET DEFAULT [.TEST]
$ SHOW DEFAULT
  MDA0:[15,1,0.TEST]
```

On Alpha systems, because the total length of the default directory string would exceed 255 characters in this example, a Directory ID is used in the directory specification. When the default directory string is replaced with a Directory ID, it is not possible to use the minus sign to specify the next higher directory.

SET DEVICE

SET DEVICE — Establishes a print device or terminal as a spooled device or establishes the operational status for a device.

Format

SET DEVICE [*device-name*[:]]

Parameter

***device-name*[:]**

Specifies the name of the device whose spooling or operational status is to change. The device must be a print device or a terminal if you want to change the spooling status; the device must be a disk or magnetic tape if you want to change the operational status.

When you specify qualifiers that act on shadow sets, the device might be a shadow set member (*ddcu*:) or the whole shadow set (DSA*nnnn*:), as noted in the qualifier format. Depending on the qualifier, if no device name is supplied, the qualifier applies to *all* shadow sets mounted on the node from which the command is executed.

Description

Requires OPER (operator) privilege.

PHY_IO (physical I/O) privilege is required when specifying the Alpha-only qualifiers that change the characteristics of shadow sets. These qualifiers also can be specified using the **SET SHADOW** command, which provides additional options for controlling shadow sets. Qualifier settings for shadow sets remain in effect until they are changed using either **SET DEVICE** or **SET SHADOW** or until the shadow set or shadow set member is dismounted. If the shadow set is remounted or a shadow set member is returned to the shadow set from which it was dismounted, the settings must be specified again. For more information about volume shadowing, see the *VSI OpenVMS Volume Shadowing Guide* [https://docs.vmssoftware.com/vsi-openvms-volume-shadowing-guide/].

When you specify any of the following qualifiers, the operational status of the drive is changed: **/ABORT_VIRTUAL_UNIT**, **/AVAILABLE**, **/COPY_SOURCE**, **/DUAL_PORT**, **/ENABLE**, **/ERROR_LOGGING**, **/FORCE_REMOVAL**, **/MEMBER_TIMEOUT**, **/MVTIMEOUT**, **/POLL**, **/READ_COST**, **/SITE**, or **/SWITCH**.

When you specify the **/SPOOLED** qualifier, program output that uses OpenVMS Record Management Services (RMS) or file control system (FCS) and that specifies the print device name is written onto an intermediate disk rather than written directly to the print device or terminal.

Qualifiers

/ABORT_VIRTUAL_UNIT DSA*nnnn*: (Alpha/IA-64 only)

Aborts mount verification on a shadow set.

Use this qualifier when you know that the unit cannot be recovered. When you use this qualifier, the shadow set must be in mount verification. The shadow set aborts mount verification immediately

on the node from which the command is issued. If the shadow set is not in mount verification, this command returns the error

```
%SYSTEM-E-UNSUPPORTED, unsupported operation or function.
```

After this command completes, the shadow set still must be dismounted. Use the following command to dismount the shadow set:

```
$ DISMOUNT/ABORT DSAnnnn:
```

/AVAILABLE
/NOAVAILABLE

Controls whether the specified disk or magnetic tape is to be considered available.
You must dismount the specified disk or magnetic tape before entering the **SET DEVICE/[NO]AVAILABLE** command. If you specify the **/NOAVAILABLE** qualifier, any attempt to allocate or mount the specified disk or magnetic tape is prevented. Devices are automatically set **/AVAILABLE** when brought online even if the device had been previously set **/NOAVAILABLE**.

/COPY_SOURCE {*ddcu*: | DSA*nnnn*:} (Alpha/IA-64 only)

Specifies which source member of a shadow set will be used as the source for read data during full copy operations when a third member is added to the shadow set. This qualifier affects only copy operations that do not use disk copy data (DCD) commands.

Some storage controllers, such as the HSG80, have a read-ahead cache, which significantly improves single-disk read performance. Copy operations normally alternate reads between the two source members, which effectively nullifies the benefits of the read-ahead cache. This qualifier lets you force all reads from a single, specified source member for a copy operation.

If only a shadow set (DSA*nnnn*:) is specified, all reads for full copy operations will be performed from the disk that is the current "master" member, regardless of physical location of the disk.

If a shadow set member (*ddcu*:) is specified, that member will be used as the source of all copy operations. This allows you to choose a local source member, rather than a remote master member.

/DUAL_PORT
/NODUAL_PORT

Controls whether the port seize logic in the device driver of the specified disk is to be enabled. This qualifier should be used only on disks that contain a dual port kit and have been dismounted.

/ENABLE
/NOENABLE

Enables or disables a path to a multipath device; the affected path is the one named in the **/PATH** qualifier. All paths are initially enabled; the current path cannot be disabled.

/ERROR_LOGGING
/NOERROR_LOGGING

Controls whether device errors are logged in the error log file. When you specify the **/ERROR_LOGGING** qualifier, all error messages reported by the device on which error logging is

enabled are recorded in the error log file. Use the **SHOW DEVICE/FULL** command to find out the current status.

/FORCE_REMOVAL *ddcu*: (Alpha/IA-64 only)

Expels the specified shadow set member from the shadow set.

If connectivity to a device has been lost and the shadow set is in mount verification, this qualifier causes the member to be expelled from the shadow set immediately.

If the shadow set is not currently in mount verification, no immediate action is taken. If connectivity to a device has been lost but the shadow set is not in mount verification, this qualifier lets you flag the member to be expelled from the shadow set, as soon as it does enter mount verification.

The specified device must be a member of a shadow set that is mounted on the node where the command is issued.

/LOG
/NOLOG (default)

Controls whether log information is displayed at the terminal.

/MEMBER_TIMEOUT=*n* *ddcu*: (Alpha/IA-64 only)

Specifies the timeout value to be used for a shadow set member.

The value supplied by this qualifier overrides the system parameter SHADOW_MBR_TMO for this specific device. Each member of a shadow set can be assigned a different MEMBER_TIMEOUT value.

The valid range for *n* is 1 to 16777215 seconds.

The specified device must be a member of a shadow set that is mounted on the node where the command is issued.

/MV_INTVL=*n*

Overrides the systemwide setting of the MVSUPMSG_INTVL system parameter with the value specified by *n* (seconds) for the specified device only. If **/MV_INTVL=0**, the value for MVSUPMSG_INTVL is used instead. If MVSUPMSG_INTVL=0 or MVSUPMSG_NUM=0, all mount verifications are announced. For more information about MVSUPMSG_INTVL, see the relevant section in the *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/#SYS_PARAMS_SEC].

/MV_NUM=*n*

Overrides the systemwide setting of the MVSUPMSG_NUM system parameter with the value specified by `n` (seconds) for the specified device only. If **/MV_NUM=0**, the value for MVSUPMSG_NUM is used instead. If MVSUPMSG_NUM=0 or MVSUPMSG_INTVL=0, all mount verifications are announced. For more information about MVSUPMSG_NUM, see the relevant section in the *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/#SYS_PARAMS_SEC].

/MVTIMEOUT=*n* DSA*nnnn*: (Alpha/IA-64 only)

Specifies the mount verification timeout value to be used for the shadow set specified by its virtual unit name (DSA*nnnn*:).

The value supplied by this qualifier overrides the value specified by the system parameter MVTIMEOUT for this specific shadow set.

Note

You cannot change the value of MVTIMEOUT for a system disk. Any attempt to do so results in an error.

The valid range for *n* is 1 to 16777215 seconds.

The specified shadow set must be mounted on the node where the command is issued.

/PATH=*path-description-string*

(For multipath devices only) Specifies a particular route to the device. The path description string identifies a path from the host to the device (for example, PGA0.2343-A855-BC99-876F or PKC0.5). Wildcard characters have no special meaning. The path description can be abbreviated to the minimum unique string.

Note that this qualifier does not imply that any action be taken. To take action, you must also specify one of the following qualifiers: **/(NO)ENABLE**, **/(NO)POLL**, or **/SWITCH**.

/POLL
/NOPOLL

Specifies that the path specified by the **/PATH** qualifier is to be polled for continued availability as a path switch destination. This does not affect any attempt to switch to this path. Its purpose is only to inform an operator that a path may or may not be viable as a path switch destination. By default, all device paths are enabled for polling when they are discovered. Note that this qualifier applies only to multipath devices.

/PREFERRED_CPUS=*cpu-a*[,*cpu-b*,*cpu-c*,...]
/NOPREFERRED_CPUS

Allows you to specify a CPU or a set of candidate CPUs from which the operating system chooses the CPU to assign to the Fast Path port. The chosen CPU is called the preferred CPU for this Fast Path port. The Fast Path port's interrupt I/O completion processing and I/O initiation processing are performed on this preferred CPU.

In addition to selecting the preferred CPU, the User Preferred CPU is set for this port. Setting the User Preferred CPU prevents the port from being reassigned to another CPU unless the User preferred CPU is being stopped. The qualifier can be negated. When the **/NOPREFERRED_CPUS** qualifier is specified, the User Preferred CPU is cleared for the port, but it remains a Fast Path port, and the current preferred CPU is not changed.

If both **/PREFERRED_CPUS** and **/NOPREFERRED_CPUS** are specified on the same command line, **/NOPREFERRED_CPUS** is ignored.

If Fast Path is enabled, you can use the DCL command, **SHOW DEVICE/FULL**, to display the current preferred CPU and the User Preferred CPU.

For more information about using Fast Path features to improve I/O performance, see the relevant section in the *VSI OpenVMS I/O User's Reference Manual* [https://docs.vmssoftware.com/vsi-openvms-io-user-s-reference-manual/#d0e29396].

/READ_COST=*n* {*ddcu*: | DSA*nnnn*:} (Alpha/IA-64 only)

Enables you to modify the default cost assigned to each shadow set member (*ddcu*:).By modifying the assignments, you can bias the reads in favor of one member of a two-member shadow set, or, in the case of three-member shadow sets, in favor of one or two members of the set over the remaining members. The device specified must be a shadow set or member of a shadow set that is mounted on the node where the command is issued.

The valid range for the specified cost is 1 to 65,535 units.

The shadowing driver assigns default READ_COST values to shadow set members when each member is initially mounted. The default value depends on the device type and its configuration relative to the system mounting it. The following list of device types is ordered by the default READ_COST assignments, from the lowest cost to the highest cost:

- DECram device
- Directly connected device in the same physical location
- Directly connected device in a remote location
- DECram served device
- Default value for other served devices

The value supplied by the **/READ_COST** qualifier overrides the default assignment. The shadowing driver adds the value of the current queue depth of the shadow set member to the READ_COST value and then reads from the member with the lowest value.

Different systems in the cluster can assign different costs to each shadow set member.

When this qualifier specifies a shadow set (DSA*nnnn*:) instead of a shadow set member, the **/READ_COST** qualifier switches the read cost setting for *all* shadow set members back to the default read cost settings established automatically by the shadowing software. The specified shadow set must be mounted on the node where the command is issued. In this case, the value specified for the cost can be any number because it will be ignored.

If the **/SITE** command qualifier has been specified, the shadowing driver takes site values into account when it assigns default READ_COST values. In order for the shadowing software to determine whether a device is in the category of "directly connected device in a remote location", the **/SITE** command qualifier must have been applied to *both* the shadow set and the shadow set member.

Reads requested for a shadow set from a system at site 1 are performed from a shadow set member that is also at site 1. Reads requested for the same shadow set from site 2 can read from the member located at site 2.

/RESET=(*keyword*[,*keyword*])

Requires CMKRNL and DIAGNOSE privileges.

Resets either or both the error count and the operation count for a device. You must specify one or both of the following keywords:

Keyword	Description
ERROR_COUNT	Resets the error count.
OPERATION_COUNT	Resets the operation count.

When either count is reset, a message is sent to the error log. The message indicates that the count was reset and includes the value of the counter before the reset.

/SITE=*n* {DSA*nnnn*: | *ddcu*:} (Alpha/IA-64 only)

Indicates to the shadowing driver the site location of the specified shadow set (DSA*nnnn*:) or shadow set member (*ddcu*:). If the member is removed from the shadow set and later is returned, this qualifier must be specified again.

The SHADOW_SITE_ID system parameter defines the default site location of the shadow set. You can override the default location of the shadow set with the **/SITE** qualifier.

The valid range for the site location, represented by *n*, is 1 through 255.

After you apply this qualifier, the setting remains in effect until you change it either with this command or with the **SET SHADOW/SITE** command.

This qualifier can improve read performance because the member that is physically local to the system will be the preferred disk from which to read, provided that you specify the **/SITE** qualifier for each shadow set member and for the shadow set. (In a Fibre Channel configuration, shadow set members at different sites are directly attached to the system. For the Volume Shadowing and cluster software, there is no distinction between local and remote in multiple-site Fibre Channel configurations.)

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

Controls whether files are spooled to an intermediate disk.

The queue name indicates the printer queue to which a file is queued. If a queue name is not supplied, the default is the name of either the printer or terminal.

The intermediate disk name identifies the disk to which the spooled files are written. If the intermediate disk name is not supplied, the default is SYS$DISK (the current default disk). The intermediate disk must be mounted before files can be written to it.

Once the device has been set spooled to a disk, that disk cannot be dismounted until the spooled device is set to **/NOSPOOLED**. All channels must be deassigned from a print device before its spooling characteristics can be changed. Also, the queue that is assigned to the device must be stopped.

/SWITCH

Attempts to switch from the current path of a multipath device to the path named in the **/PATH** qualifier. The path switch is attempted only if the path has not been disabled by **SET DEVICE/NOENABLE**. The path switch is initiated but not completed before returning control

to the user; use **SHOW DEVICE/MULTIPATH** to verify that the path switching has completed successfully.

This command requires the OPER privilege. Additionally, if the device is currently allocated by another process, as tape devices often are, the SHARE privilege is needed.

Examples

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

   In this example, the **/SPOOLED** qualifier requests that the printer queue `LPA0` be spooled to an intermediate disk before files directed to the disk are printed. Because no intermediate disk was specified, the intermediate disk defaults to SYS$DISK.

2. `$ SET DEVICE/ERROR_LOGGING  DKB2:`

   The **SET DEVICE** command in this example requests that all device errors reported on device `DKB2` be logged in the error log file.

3. `$ SET DEVICE/NOAVAILABLE DRA0:`

   The **SET DEVICE** command in this example prevents any attempt to mount a disk on device `DRA0`.

4. `$ SET DEVICE/DUAL_PORT DRA0:`

   The **SET DEVICE** command in this example enables the dual port seize logic in device `DRA0`.

5. `$ SET DEVICE/PREFERRED_CPU=1 PNA0`

   The Fast Path port `PNA0` is assigned to CPU `1`, and the User Preferred CPU is set to `1`.

6. `$ SET DEVICE/NOPREFERRED_CPU  PNA0`

   The port `PNA0` will have the User Preferred CPU cleared, but will remain a Fast Path port, with CPU `1` as its current preferred CPU.

7. `$ SET DEVICE/PREFERRED_CPU=(0,1,2)  PNA0`

   The operating system selects CPU `0`, `1`, or `2` as the preferred CPU for Fast Path port `PNA0`, and sets its User Preferred CPU to the selected CPU.

8. `$ SET DEVICE $1$DGA1001:/PATH=PGB0.0123-4567-89AB-CDEF/ENABLE`

   In this example, a request is made to enable path `PGB0.0123-4567-89AB-CDEF` as a switch path destination. No error is indicated if the path was in the enabled state prior to the request. Had the **/NOENABLE** option been used, the path would have been removed from switch path consideration.

9. `$ SET DEVICE $1$DGA1001:/PATH=PGB0.0123-4567-89AB-CDEF/POLL`

   In this example, a request is made to enable path `PGB0.0123-4567-89AB-CDEF` for polling of its continued availability as a path switch destination. Had the **/NOPOLL** option been used, the path would have been removed from path availability polling consideration.

10. `$ SET DEVICE $1$DGA1001:/PATH=PGB0.0123-4567-89AB-CDEF/SWITCH`

In this example, a request is made to switch from the current path to path `PGB0.0123-4567-89AB-CDEF`. Should the path fail, the switch algorithm of the unspecified path is used.

11.
```
$ ! Command sequence used on system 1 at site 1:
$ MOUNT/SYSTEM DSA0:/SHADOW=($1$DGA0:,$1$DGA1:) TEST
$ SET DEVICE/SITE=1  DSA0:
$ !
$ ! Command sequence used on system 2 at site 2:
$ MOUNT/SYSTEM DSA0:/SHADOW=($1$DGA0:,$1$DGA1:) TEST
$ SET DEVICE/SITE=2  DSA0:
$ !
$ ! Commands used on both system 1 and system 2:
$ SET DEVICE/SITE=1  $1$DGA0:
$ SET DEVICE/SITE=2  $1$DGA1:
```

In this example, the system at `site 1` will perform all reads from `$1$DGA0`, which is the physically local disk, while the system at `site 2` will perform all reads from its physically local disk, `$1$DGA1`.

SET DEVICE/SERVED

SET DEVICE/SERVED — Makes a disk on a local node available to all the nodes in an OpenVMS Cluster. **Applies only to OpenVMS Cluster environments.**

Format

SET DEVICE/SERVED *node-name$ddcu*:

Parameter

***node-name$ddcu*:**

Specifies the device name of the device that you want to make available to the cluster.

Description

The **SET DEVICE/SERVED** command is used in conjunction with the mass storage control protocol (MSCP) server to make a disk on a local node available to all nodes on the cluster. The local node must be a member of an OpenVMS Cluster, and the local MSCP server must have been invoked by the System Generation utility (SYSGEN).

You cannot use the **SET DEVICE/SERVED** command under the following conditions:

- In service of a Phase II shadow set virtual unit
- On devices that are already mounted
- On system disks
- On quorum disks

The **SET DEVICE/SERVED** command string can be included as part of the local startup command file and entered before the Mount utility mounts the disk to be served (made available to the entire cluster).

Example

```
$ SET DEVICE/SERVED DRA4:
```

The **SET DEVICE/SERVED** command in this example instructs the MSCP server to make the disk device DRA4 on your local node available to all other processors on your cluster.

SET DIRECTORY

SET DIRECTORY — Modifies the characteristics of one or more directories. **See the qualifier descriptions for restrictions.**

Format

SET DIRECTORY [*device-name*[:]]*directory-spec*[,...]

Parameters

***device-name*[:]**

Specifies the device on which the directory that you want to modify is located. The device name parameter is optional.

***directory-spec*[,...]**

Specifies one or more directories to be modified. If you specify more than one directory, separate the directory specifications with commas. The asterisk (*) and the percent sign (%) wildcard characters are allowed.

Qualifiers

/BACKUP

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/BACKUP** qualifier selects directories according to the dates of their most recent backups. This qualifier is incompatible with the **/CREATED**, **/EXPIRED**, and **/MODIFIED** qualifiers, which also allow you to select directories according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/BEFORE[=*time*]

Selects only those directories dated prior to the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the **/BEFORE** qualifier to indicate the time attribute to be used as the basis for selection: **/BACKUP**, **/CREATED** (default), **/EXPIRED**, or **/MODIFIED**.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/BY_OWNER[=*uic*]

Selects only those directories whose owner user identification code (UIC) matches the specified owner UIC. The default UIC is that of the current process.

Specify the UIC by using standard UIC format as described in the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#UIC_SECTION].

The **/OWNER_UIC** qualifier requires control access, a system UIC, or privilege.

/CONFIRM
/NOCONFIRM (default)

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

YES	NO	QUIT
TRUE	FALSE	**Ctrl/Z**
1	0	ALL
	Return	

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), but these abbreviations must be unique. Affirmative answers are YES, TRUE, and 1. Negative answers include: NO, FALSE, 0, and pressing **Return**. Entering QUIT or pressing **Ctrl/Z** indicates that you want to stop processing the command at that point. When you respond by entering ALL, the command continues to process, but no further prompts are given. If you type a response other than one of those in the list, DCL issues an error message and redisplays the prompt.

/CREATED (default)

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/CREATED** qualifier selects directories based on their dates of creation. This qualifier is incompatible with the **/BACKUP**, **/EXPIRED**, and **/MODIFIED** qualifiers, which also allow you to select directories according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/EXCLUDE=(*filespec*[,...])

Excludes the specified directories from the SET DIRECTORY operation. You can include a directory but not a device in the file specification. The asterisk (*) and the percent sign (%) wildcard characters are allowed in the file specification; however, you cannot use relative version numbers to exclude a specific version. If you specify only one directory, you can omit the parentheses.

/EXPIRED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/EXPIRED** qualifier selects directories according to their expiration dates. (The expiration date is set with the **SET FILE/EXPIRATION_DATE** command.) The **/EXPIRED** qualifier is incompatible with the **/BACKUP**, **/CREATED**, and **/MODIFIED** qualifiers, which also allow you to select directories

according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/LOG
/NOLOG (default)

Controls whether the system displays the directory specification of each directory that is modified as the command executes.

/MODIFIED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/MODIFIED** qualifier selects directories according to the dates on which they were last modified. This qualifier is incompatible with the **/BACKUP**, **/CREATED**, and **/EXPIRED** qualifiers, which also allow you to select directories according to time attributes. If you specify none of these four time modifiers, the default is the **/CREATED** qualifier.

/OWNER_UIC[=*uic*]

Requires control access, a system UIC, or privilege.

Specifies an owner UIC for the directory. The default UIC is that of the current process.

/SINCE[=*time*]

Selects only those directories dated on or after the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, JOB_LOGIN, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the **/SINCE** qualifier to indicate the time attribute to be used as the basis for selection: **/BACKUP**, **/CREATED** (default), **/EXPIRED**, or **/MODIFIED**.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/STYLE=*keyword*

Specifies the file name format for display purposes.

The valid keywords for this qualifier are CONDENSED and EXPANDED. Descriptions are as follows:

Keyword	Description
CONDENSED (default)	Displays the file name representation of what is generated to fit into a 255-length character string. This file name may contain a DID or FID abbreviation in the file specification.
EXPANDED	Displays the file name representation of what is stored on disk. This file name does not contain any DID or FID abbreviations.

The keywords CONDENSED and EXPANDED are mutually exclusive. This qualifier specifies which file name format is displayed in the output message, along with the confirmation if requested.

File errors are displayed with the CONDENSED file specification unless the EXPANDED keyword is specified.

See the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#FILE_ACCESS] for more information.

/SYMLINK=*keyword*

The valid keywords for this qualifier are [NO]WILDCARD and [NO]ELLIPSIS. Descriptions are as follows:

Keyword	Description
WILDCARD	Indicates that symlinks are enabled during wildcard searches.
NOWILDCARD	Indicates that symlinks are disabled during directory wildcard searches.
ELLIPSIS	Equivalent to WILDCARD (included for command symmetry).
NOELLIPSIS	Indicates that symlinks are matched for all wildcard fields except for ellipsis.

If the file named in the **SET DIRECTORY** command is a symlink, the command operates on the symlink target.

/VERSION_LIMIT[=*n*]

Sets the maximum number of versions that files in a specified directory can have. If you do not set a version limit, a value of 0 is used, indicating that the number of file versions is limited only to the Files-11 architectural limit of 32,767. If you change the version limit for a directory, the new version limit applies only to files created after the change has been made. When creating a file, if the total number of versions of that file name exceeds the specified version limit, then the file with the lowest version number is deleted from the directory without notification to the user.

The version limit set on a directory has no effect on the version limit set on a particular file in that directory. To set a version limit on a particular file in a directory, use the **SET FILE/VERSION_LIMIT[=*n*]** command.

To view the version limit on a directory, use the **DIRECTORY/FULL** command on a directory file and look at the File Attributes field of the output.

Examples

1. `$ SET DIRECTORY/VERSION_LIMIT=5/CONFIRM [VERCAMMEN...]`

   The **SET DIRECTORY** command in this example sets a version limit of `5` for all files in the `VERCAMMEN` directory (the root directory) and all subdirectories of `[VERCAMMEN]`. The **/CONFIRM** qualifier requests that you confirm whether the specified directory should actually be modified. Note that this version limit applies only to files created after the command is entered.

2. `$ SET DIRECTORY/VERSION_LIMIT=8 [.JAIMON...]`

   In this example, the **SET DIRECTORY** command sets a version limit of `8` for all files in the `JAIMON` subdirectory and any of its subdirectories (for example, `[.JAIMON.ADRIAN]`). The setting for these specific directories overrides any version limit that might be set for the root directory and its other subdirectories.

3. `$ SET DIRECTORY/OWNER_UIC=[360,020] [WOERTMAN],[MOSER]`

The **SET DIRECTORY** command in this example modifies both the [WOERTMAN] and [MOSER] directories, changing their owner UICs. Using the **/OWNER_UIC** qualifier requires SYSPRV (system privilege).

SET DISPLAY

SET DISPLAY — Directs the output of an OpenVMS DECwindows application. Output can be directed from any processor running a DECwindows application, including workstations, to any DECwindows workstation. Both source and destination nodes must be part of the same network.

Format

SET DISPLAY [*display-device*]

Parameter

display-device

Specifies a logical name for the workstation display you are creating or modifying. If you are directing application output to multiple workstation displays, you can use different logical names to point to each display. If you do not specify a *display-device* parameter, the logical name DECW$DISPLAY is used. This means that, by default, application output will be displayed on the workstation display device referred to by DECW$DISPLAY.

By entering the command **SHOW DISPLAY**, you can see the default workstation node where applications will be displayed, as well as related device, screen, and transport information.

If you specified your own logical name in the **SET DISPLAY/CREATE** command, include that logical name in the **SHOW DISPLAY** command.

Description

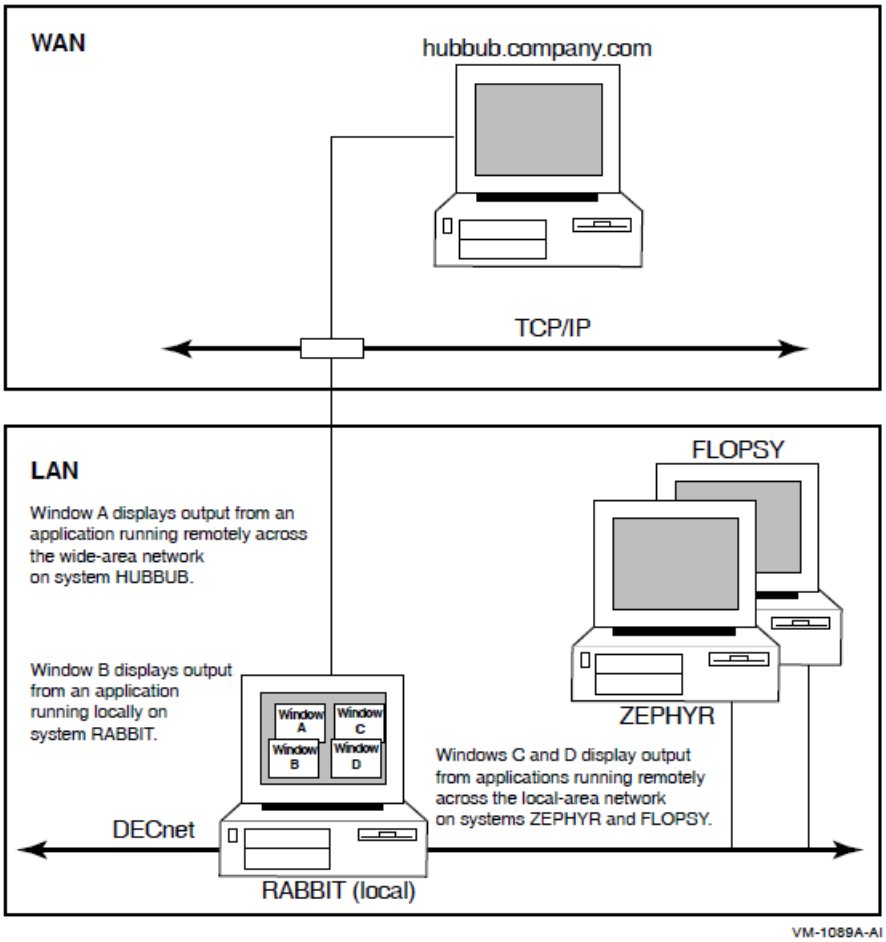
DECwindows gives you the ability to run applications across a network. The **SET DISPLAY** command enables you to direct the output of client applications across the network to an alternate workstation (X display server) for viewing. For example, the **SET DISPLAY** command allows you to:

- Redirect the output from local client applications to a remote workstation for display.
- Redirect the output from remote client applications to your local workstation for display. Although the application runs on another processor, it looks the same as any other application running locally on your workstation.

By running applications on a remote processor for local display on your workstation monitor, you can take advantage of larger computers that might be better suited to a specific computing task. By default, applications running on your workstation are displayed on your workstation. While DECwindows must be available on both nodes, only the display (X server) node must be a workstation.

The **SET DISPLAY** command affects only those applications running from the process from which you enter the command. This means that although you might be running one application on your workstation and displaying it on another workstation, you can continue to run applications on your workstation for display in other windows on your workstation as illustrated in *Figure 1, "Running Remote and Local Applications"*.

Figure 1. Running Remote and Local Applications



Qualifiers to the **SET DISPLAY** command enable you to create the display device and manage aspects of the network connection between the client application and the X display server, as described in the following sections.

Redirecting Application or Display Output

To direct application output to another DECwindows workstation, enter the **SET DISPLAY** command in the following form:

```
SET DISPLAY/CREATE/NODE=workstation_display/TRANSPORT=transport_name
```

If you do not specify the **/TRANSPORT** qualifier, the value defined by the logical DECW$SETDISPLAY_DEFAULT_TRANSPORT is used. If this logical is not defined, the value of DECNET is used as the default.

Subsequently, you can redirect the display to other workstation devices by entering a DCL command in the following form:

```
SET DISPLAY/NODE=workstation_display
```

Specifying Workstation Access

By default, you are authorized to run client applications locally on your workstation (X display server). You cannot log into another node and direct applications to display on your workstation unless you explicitly authorize yourself to do so. This prevents unauthorized users on other nodes in the network from directing output to or receiving input from your workstation without your specific permission.

Using the DECwindows Session Manager, you or a privileged user can grant other users running remote applications access to your workstation using one of the following access control schemes:

- User-based access control
- Token-based access control (Alpha/IA-64 only)
    - Magic Cookie (based on the MIT-MAGIC-COOKIE-1 protocol)
    - Kerberos (based on the MIT-KERBEROS-5 protocol)

Once access is granted, authorized users can use the **SET DISPLAY** command to create a display device that accesses the remote workstation, or more specifically, the remote X display server.

When using a display device to access a remote X display server, make sure that the node or host name you specify in the **SET DISPLAY** command represents a workstation on which you are authorized to display applications. For example, if you specify **SET DISPLAY/CREATE/NODE=HUBBUB** from node ZEPHYR, and are not authorized to display applications on the workstation HUBBUB, an error message is displayed indicating that you are not authorized to use the device.

On Alpha and IA-64 systems, the **/GENERATE**, **/REVOKE**, and **/[NO]XAUTHORITY** qualifiers to **SET DISPLAY** enable you to specify additional authorization information when directing client applications to a workstation that uses a token-based access control scheme.

Note

The MIT-KERBEROS-5 protocol does not currently support key generation, as defined by the SECURITY extension. Do not use the **/GENERATE** and **/REVOKE** qualifiers when connecting to an X display server protected by Kerberos access control.

For information about displaying applications on other workstation nodes, using the DECwindows Session Manager to authorize remote connections, and supported access control schemes see the DECwindows documentation.

Using Proxy Servers (Alpha/IA-64 only)

Some system configurations use Low-Bandwidth X (LBX) proxy servers to manage communication between client applications and one or more X display servers. With DECwindows, an LBX proxy server configuration can be categorized as either standalone or managed. Managed configurations use a proxy manager application to further filter and direct incoming connection requests by client applications.

The **/PROXY** qualifier specifies that a proxy manager application be used to locate a managed proxy server. The **/PMPORT** and **/PMTRANSPORT** qualifiers identify the port and network transport for the proxy manager. The **/[NO]LBXAUTHENTICATE** and **/LBXDATA** qualifiers specify the authorization protocol used between the proxy server and the X display server.

For information about using and configuring an LBX proxy server or a proxy manager, see the DECwindows documentation.

Specifying Named Properties (Alpha/IA-64 only)

Named property values are designed to store configuration information associated with the display device, such as the network address of a session manager. By using the **/VALUE** qualifier, you can set or modify one or more values associated with the display device.

See also the description of the **SHOW DISPLAY** command for more information.

Specifying the Device Mode

By using one of the following qualifiers: **/EXECUTIVE_MODE**, **/SUPERVISOR_MODE**, or **/USER_MODE**, you can specify the mode of the device you are creating. The default is the **/SUPERVISOR_MODE** qualifier.

See also the description of the **SHOW DISPLAY** command for more information.

Qualifiers

/CREATE

Creates the workstation display device (WSA *n*:) on which a DECwindows application is displayed. You must specify the **/CREATE** qualifier the first time you use the **SET DISPLAY** command, but you need not respecify it if you continue to redirect output from applications to other workstations with subsequent **SET DISPLAY** commands.

When you specify the **/CREATE** qualifier without the **/NODE** qualifier, the workstation device defaults to the current node. Additionally, if you omit the **/TRANSPORT** qualifier, the transport defaults to either the value specified by the DECW$SETDISPLAY_DEFAULT_TRANSPORT logical, if defined, or to DECNET.

/DELETE

Cancels the redirected display by deassigning the logical name DECW$DISPLAY and deleting the created display device. If you specified a logical name as the *display-device* parameter with the **SET DISPLAY/CREATE** command, entering the **SET DISPLAY/DELETE *display-device*** command cancels the redirected display by deassigning the logical name you specified and deleting the created display device.

The DECwindows Session Manager defines DECW$DISPLAY in your job logical name table when you open a terminal (DECterm) window. When you redirect application output to another workstation with the **SET DISPLAY/CREATE** command, an additional DECW$DISPLAY logical name is defined in your process logical name table. This definition supersedes the definition in the job logical name table. Output from applications run from the process in which you executed the **SET DISPLAY/CREATE** command will be displayed on the workstation referred to by the definition of DECW$DISPLAY in the process logical name table. Enter the **SHOW DISPLAY** command to see where this application will be displayed. To see whether multiple definitions for DECW$DISPLAY exist, enter the command **SHOW LOGICAL** DECW$DISPLAY.

If DECW$DISPLAY is still defined (for example, in the job logical name table) after you specify the **/DELETE** qualifier, any DECwindows applications run from this process will be displayed on the workstation device to which output is now directed. Enter the **SHOW DISPLAY** command if you are unsure of the node to which DECW$DISPLAY refers.

Caution

If you modify or delete the definition of DECW$DISPLAY from the job logical name table, you will be unable to start another session. Be careful not to specify the **/DELETE** qualifier without having first redirected the display with the **SET DISPLAY/CREATE** command.

You cannot specify the **/DELETE** and **/CREATE** qualifiers on the same command line.

/EXECUTIVE_MODE

Creates an executive-mode device and assigns the logical name DECW$DISPLAY to point to it. This qualifier must be used with the **/CREATE** qualifier. Devices created with the **/EXECUTIVE_MODE** qualifier are deleted only in the following situations:

- They are explicitly deleted with the **/DELETE** qualifier.
- The system is rebooted.

To modify or delete executive-mode devices, you must have SYSNAM (system logical name) privilege.

/GENERATE[=(*keyword*[=*value*][,...]) (Alpha/IA-64 only)

Requires a trusted network connection to an X display server and the SECURITY extension must be enabled on that server.

Instructs the display device to connect to the X display server and generate anew authorization key (consisting of the authorization protocol name and a magic cookie). This authorization key is then stored in the X authority file currently in use. If you do not want to overwrite any previously existing keys for this display device, use the **/XAUTHORITY** qualifier to write the generated key to an alternate X authority file.

Note

The MIT-KERBEROS-5 protocol does not currently support key generation, as defined by the SECURITY extension. Do not use the **/GENERATE** qualifier when connected to an X display server protected by Kerberos access control.

Each generated key follows the default settings unless otherwise specified. This means that, by default, each generated authorization key:

- Follows the MIT-MAGIC-COOKIE-1 protocol
- Grants an untrusted connection to the X server
- Is not associated with a particular application group
- Maintains a timeout period of 60 seconds before which it must be used(otherwise, it is automatically revoked)

Possible keyword values are as follows:

Keyword	Description
TRUSTED	Specifies that the key grant unrestricted access to the X server. This keyword permits a client application to perform all actions.
UNTRUSTED	Specifies that the key grant restricted access to the X server. This keyword restricts client applications to a subset of X server extensions and prevents access to windows other than those created by the requesting application.

Keyword	Description
TIMEOUT=*seconds*	Specifies the number of seconds after which the key is revoked automatically if not in use. The timeout period begins immediately, and refreshes once all connections using this key have closed.
NOTIMEOUT	Specifies that the key has no expiration and must be explicitly revoked.
GROUP=*groupid*	Specifies that the key is associated with a particular application group and is to be revoked if that group is deleted. Although the use of application groups is supported by DECwindows Motif via the XC-APPGROUP extension, DECwindows Motif does not currently supply any utilities or desktop applications to create these groups. To create an application group, you must use an X Window System application or utility external to DECwindows Motif.
PROTOCOL=*name*	Specifies the protocol to be used to authenticate the key. Valid values are server dependent. For the DECwindows X display server, the only valid value for the *name* parameter is MIT-MAGIC-COOKIE-1. If you are connecting to a third-party X server, the names of any additional authorization protocols on that X server that support the SECURITY extension and key generation are also valid.

The keyword sets **TRUSTED/UNTRUSTED** and **TIMEOUT/NOTIMEOUT** are mutually exclusive. Do not enter both values on the same command line.

/LBXAUTHENTICATE=*protocol-name* (Alpha/IA-64 only)
/NOLBXAUTHENTICATE (Alpha/IA-64 only)

Specifies the authorization protocol used to grant the LBX proxy server access to the X display server. This qualifier is used with the **/PROXY** and **/CREATE** qualifiers when creating an initial LBX proxy server connection. If a proxy server connection already exists, these qualifiers are ignored. If you do not specify **/LBXAUTHENTICATE**, the authorization protocol is obtained from the current X authority file. If you specify **/NOLBXAUTHENTICATE**, the proxy server defaults to the X authority file of the user who started the proxy manager application.

Valid values for the *protocol-name* parameter are MIT-MAGIC-COOKIE-1 and MIT-KERBEROS-5. If you supply a *protocol-name* value of MIT-MAGIC-COOKIE-1, you must also specify the value of the magic cookie using the **/LBXDATA** qualifier.

/LBXDATA=*value* (Alpha/IA-64 only)

Specifies a data value for the authorization protocol used between the LBX proxy server and the X display server. This qualifier must be used with the **/LBXAUTHENTICATE**, **/PROXY**, and **/CREATE** qualifiers.

When you issue a
SET DISPLAY/CREATE/PROXY/LBXAUTHENTICATE=MIT-MAGIC-COOKIE-1
command, use this qualifier to provide a hexadecimal magic cookie value to the MIT-MAGIC-COOKIE-1 protocol.

/NODE=*workstation_display*

Defines the workstation on which you want to display DECwindows applications. The node name you provide cannot be a cluster alias (a name that represents multiple nodes configured in an OpenVMS Cluster), but must instead identify an actual node.

You must create a workstation display device with the **/CREATE** qualifier before you can redirect the output from applications to other workstations. Do not enter the **SET DISPLAY/NODE=*workstation_display*** command without having previously specified the **/CREATE** qualifier.

When you specify the **/CREATE** qualifier without the **/NODE** qualifier, the workstation defaults to the current node.

Make sure that you are authorized to display applications on the workstation you specify. For more information about using the DECwindows Session Manager to authorize yourself to display applications from other nodes, see the DECwindows documentation.

Each node, both source and destination, must be defined in the other's network node database. For example, to display applications on node HUBBUB from ZEPHYR, HUBBUB must be entered in ZEPHYR's network node database, and ZEPHYR must be entered in HUBBUB's network node database. In addition, users on ZEPHYR must be authorized in the DECwindows Session Manager to display applications on HUBBUB. For information about entering nodes in a network node database, see the DECnet-Plus for OpenVMS documentation.

/PMPORT=[*port-number*] (Alpha/IA-64 only)

Specifies the port number on which the proxy manager is listening. This qualifier is used with the **/PROXY** and **/CREATE** qualifiers. The default port is 6500.

/PMTRANSPORT=*transport-name* (Alpha/IA-64 only)

Specifies the network transport used to connect to the proxy manager. This qualifier is used with the **/PROXY** and **/CREATE** qualifiers.

Valid values for the *transport-name* parameter are the same as those listed for the **/TRANSPORT** qualifier. If you do not specify a transport, then the network transport specified by the **/TRANSPORT** qualifier is used.

/PROXY[=*node*] (Alpha/IA-64 only)

Specifies that a proxy manager application should be used to locate an LBX proxy server, which serves as an intermediary for processing client and server connections for a display device. Use an LBX proxy server when connecting to a remote node located over a low-bandwidth network connection, such as the Internet. The proxy manager should run on the local network. This qualifier is used with the **/CREATE** qualifier.

When you create a display device using the **SET DISPLAY/CREATE/PROXY** command, you can use that device to connect to an X display server exactly the same as you would with any other display device. The client connects to the LBX proxy server returned by the proxy manager, which in turn connects to the X display server.

The value of the **/PROXY** qualifier specifies the node on which the proxy manager is running. The proxy manager is responsible for starting or locating a proxy server to provide access to the X display server specified by the transport, node, and server values. The proxy manager returns the transport, node, and server values for the selected proxy server to the display device. If you do not specify a value for the *node* parameter, the default location is the current node.

The **/PMPORT** and **/PMTRANSPORT** qualifiers can be used to further specify the proxy manager's location.

/QUOTA=(*quota-name=value*[,...]) (Alpha/IA-64 only)

Requires SYSPRV (system privilege) privilege or the DECW$WS_QUOTA rights identifier. If using an executive mode device, SYSNAM (system name) privilege is also required.

Modifies the value of the name count or data space quota associated with the display device. These quotas influence the amount of non-paged pool available for storing named properties.

Possible values for the *quota-name* parameter are as follows:

- NAME_COUNT – The name count quota restricts the total number of named properties associated with the device. The default name count quota is the value of the DECW$WS_DEFAULT_NAME_COUNT logical, or 12 if the logical is not defined.
- DATA_SPACE – The data space quota restricts the total amount of space available (in bytes) for storing property values. Note that this space includes some data overhead and may not total the exact length of the values. The default data space quota is the value of the DECW$WS_DEFAULT_DATA_SPACE logical, or 10752 if the logical is not defined.

/REVOKE (Alpha/IA-64 only)

Requires a trusted network connection to an X display server and that the SECURITY extension be enabled on that server.

Revokes an authorization key created with the **/GENERATE** qualifier.

If you specify the **/REVOKE** qualifier, the revoke operation is performed before the other actions on the **SET DISPLAY** command. For example, you can use both the **/REVOKE** and **/GENERATE** qualifiers on the same command line, in any order, to revoke one key and generate another.

Do not use with the **/CREATE** qualifier.

/SCREEN=*screen-number*

Defines the screen to be associated with this display device. In some hardware configurations, the system supports more than one screen. In such a configuration, you can specify the screen to which the application is directed.

/SERVER=*server-number*

Defines the server to be associated with the display device indicated by the *display-device* parameter to the **SET DEVICE** command. Using the transport mechanism, the server passes data between the user and the application. In some hardware configurations, the system supports more than one server. In such a configuration, you can specify the server to which the application is directed.

On Alpha and IA-64 systems in LBX proxy server configurations, this qualifier specifies either an X display server or a standalone LBX proxy server. Standalone LBX proxy servers are not managed by the proxy manager and are manually assigned a server number on startup.

If you do not specify the *display-device* parameter, the **SET DISPLAY/SERVER** command associates the server with the display device indicated by the logical name DECW$DISPLAY.

/SUPERVISOR_MODE (default)

Creates a supervisor-mode device and assigns the logical name DECW$DISPLAY to point to it. This qualifier must be used with the **/CREATE** qualifier. When the user logs out, the device is deleted.

/TRANSPORT=*transport-name*

Defines the transport mechanism, for example, DECnet or local, that passes information between the application and the workstation. The transport mechanism is used to send input from the user to the application and output from the application to the display device. In proxy manager configurations, the value specified for this qualifier also acts as a default value for the **/PMTRANSPORT** qualifier.

The *transport-name* parameter can be one of the following values:

- TCPIP or DECNET

  Specifies the appropriate network transport for applications running and displaying on different nodes. You can specify the default transport value used with the **SET DISPLAY/CREATE** command by defining the DECW$SETDISPLAY_DEFAULT_TRANSPORT logical in the job, process, or system table. The value of this logical is then used each time you enter the **SET DISPLAY/CREATE** command without the **/TRANSPORT** qualifier. If you specify neither the qualifier nor the logical, the value of DECNET is used as default.

- LOCAL

  Optimizes the performance of applications running and displaying on the same node.

On Alpha and IA-64 systems, the valid set of values for the *transport-name* parameter is further determined by the DECwindows logical DECW$IPV6_SUPPORT. This logical specifies which Internet Protocol version is supported: IPv4 or IPv6. See the DECwindows Motif documentation for more information about the DECW$IPV6_SUPPORT logical and support for the IPv6 protocol.

When using a proxy manager on Alpha and IA-64 systems, this qualifier specifies the transport that the selected LBX proxy server should use to connect to the X server.

When using a standalone proxy server (specified by the **/SERVER** qualifier), this qualifier specifies the transport that the client should use to connect to the proxy server.

/USER_MODE

Creates a user-mode display device and assigns the logical name DECW$DISPLAY to point to it. This qualifier must be used with the **/CREATE** qualifier. The lifetime of a user-mode device is one DECwindows image: when the next DECwindows image exits, the device is deleted.

/VALUE=(NAME=*property-name*,*action*[=*data*]) (Alpha/IA-64 only)

Sets, deletes, or modifies a named property value associated with the display device. Named property values are intended to store configuration information associated with the display device, such as the network address of a session manager.

Use the following options to specify the type of action to perform:

- DELETE – Deletes the specified property value.

- SET=*data* – Sets the specified property to the supplied value.

- APPEND=*data* – If the property exists, adds the supplied value to the end of list of values associated with the property. If the property does not exist, it is created and set to the supplied value.

Use the **SHOW DISPLAY/VALUES** command to display all existing named properties and their current values. Property names are restricted to 31 characters. Names that include a dollar sign ($) are reserved by VSI for use with OpenVMS products and facilities. For example, names starting with the string DECW$ are reserved for DECwindows Motif.

/XAUTHORITY=[*filespec*] (Alpha/IA-64 only)
/NOXAUTHORITY (Alpha/IA-64 only)

Directs the display device to use a specific X authority file when connecting the X display server. The X authority file contains data that grants or restricts access to the X display server depending on the authorization protocol in use. If a file specification is not entered with this qualifier, the default of SYS$LOGIN:DECW$XAUTHORITY.DECW$XAUTH is used. If the file specification is entered and the file does not currently exist, it is created automatically when an entry is made into the file.

Note that with this qualifier, all X authority file specifications resolve to a physical device (rather than a logical device) in order to remain available to other detached processes. If you specify a search list, and the file is not found, the X authority file specification resolves to the first directory in the search list.

Use **/NOXAUTHORITY** to clear the setting and return to the file referenced by the DECW$XAUTHORITY logical or the default file if the logical is not defined.

Examples

1.
```
$ SHOW DISPLAY
  Device:     WSA1: [super]
  Node:       0
  Transport:  LOCAL
  Server:     0
  Screen:     0

$ SET DISPLAY/CREATE/NODE=ZEPHYR/EXECUTIVE_MODE
$ SHOW DISPLAY
  Device:     WSA2:  [exec]
  Node:       ZEPHYR
  Transport:  DECNET
  Server:     0
  Screen:     0
$ SPAWN/NOWAIT/INPUT=NL: RUN SYS$SYSTEM:DECW$CLOCK

$ SET DISPLAY/DELETE

$ SHOW DISPLAY
  Device:     WSA1:  [super]
  Node:       0
  Transport:  LOCAL
  Server:     0
  Screen:     0
```

   In this example, you are logged in to your workstation (device `WSA1:`), here referred to as node `0`. (0 is the standard shorthand notation for representing your node.) You want to run the DECwindows Clock on your workstation and display it on another workstation, `ZEPHYR`.

   Assuming you are authorized to display applications on `ZEPHYR`, you redirect the application's output to `ZEPHYR` with the **SET DISPLAY** command. The device created on `ZEPHYR` is an

executive-mode device. You enter the **SHOW DISPLAY** command to verify the location of the redirected display. You then run `Clock`. When you finish running `Clock`, you disable the redirected display by entering the **SET DISPLAY/DELETE** command. Finally, you enter the **SHOW DISPLAY** command to verify that any applications subsequently run on your node will also be displayed there.

Note that a new workstation display device, `WSA2`, is created when you enter the **SET DISPLAY/CREATE** command. When you cancel the redirected display with the **SET DISPLAY/DELETE** command, application output is once again displayed on the workstation display device referred to by `WSA1`.

2.
```
$ SET DISPLAY/CREATE/NODE=FLOPSY RABBIT
$ SHOW DISPLAY RABBIT

 Device:    WSA2:  [super]
 Node:      FLOPSY
 Transport: DECNET
 Server:    0
 Screen:    0

$ RUN/DETACHED/OUTPUT=WSA2: SYS$SYSTEM:DECW$CLOCK

$ SET DISPLAY/CREATE/NODE=ZEPHYR ZNODE
$ SHOW DISPLAY ZNODE

 Device:    WSA3:  [super]
 Node:      ZEPHYR
 Transport: DECNET
 Server:    0
 Screen:    0

$ RUN/DETACHED/OUTPUT=WSA3: SYS$SYSTEM:DECW$CALENDAR

$ RUN SYS$SYSTEM:DECW$BOOKREADER
$ SHOW DISPLAY

 Device:    WSA1:  [super]
 Node:      0
 Transport: LOCAL
 Server:    0
 Screen:    0
```

In this example, you are logged in to your node (device `WSA1:`), and want to direct the output from applications to several workstation displays in the same session. By specifying different logical names in the **SET DISPLAY** command, you can redirect the output without changing the logical name definition for `DECW$DISPLAY`. This allows you to display the output from most applications on your default display but occasionally display output on another workstation. You can also continue to run and display applications on your node. In this example, `Clock` is displayed on node `FLOPSY`, `Calendar` is displayed on node `ZEPHYR`, and `Book reader` is displayed on your workstation.

Note that to run your applications with the DCL command **RUN/DETACHED**, you must use the device name that equates to the logical display device name you specified in the **SET DISPLAY** command. Use the **SHOW DISPLAY** command to obtain this device name.

3.
```
$ SET DISPLAY/CREATE/GENERATE=TRUSTED -
_$ /XAUTHORITY=XAUTHORITY_TEMP/NODE=ZEPHYR
$ PIPE SHOW DISPLAY/EXTR | RSH FLOPSY "XAUTH NMERGE SYS$INPUT"
```

```
$ SET HOST FLOPSY
$ SET DISPLAY/NODE=ZEPHYR
$ RUN SYS$SYSTEM:DECW$CLOCK
```

In this example, you are logged into your workstation (`WSA1:`) and want to run the DECwindows `Clock` application on a remote node (`FLOPSY`) and display it on another workstation (`ZEPHYR`). Both systems are using Magic Cookie authorization to control access to the X display server.

Assuming you are already authorized to connect to the server on node `ZEPHYR`, you create a display device that connects to `ZEPHYR` and generates a new authorization key. This key grants trusted connections to the server on `ZEPHYR` within 60 seconds of key generation. To restrict trusted access to the server, the key is stored in a new X authority file, `XAUTHORITY_TEMP.DECW$XAUTH`.

The authorization key is then extracted and copied to `FLOPSY`, and merged with other entries in your X authority file on that system. You then set host to node `FLOPSY` and set display to node `ZEPHYR`. When you run the clock application from `FLOPSY`, it connects to the server on `ZEPHYR` and is allowed access as specified by the generated authorization key.

4.
```
$ SET DISPLAY/CREATE/VALUE= -
_$ (NAME=DECW$SESSION_MANAGER,SET="tcpip/zephyr:9510")
$ SHOW DISPLAY/SYMBOLS/ALL
    Device:    WSA23:  [super]
    Node:      0
    Transport: DECNET
    Server:    0
    Screen:    0
  User-defined values:
    "DECW$SESSION_MANAGER" = "tcpip/zephyr:9510"
$ SHOW SYMBOL DECW$DISPLAY__DECW$SESSION_MANAGER
  DECW$DISPLAY__DECW$SESSION_MANAGER == "tcpip/zephyr:9510"
$ SET DISPLAY UNTRUSTED/CREATE/XAUTHORITY=TEMP/GENERATE -
_$ /VALUE=(NAME=DECW$SESSION_MANAGER,-)
_$ SET="'DECW$DISPLAY__DECW$SESSION_MANAGER'"
```

In this example, you create a display device, and set the `DECW$SESSION_MANAGER` property to the network address of a session manager on remote node `ZEPHYR` using port number `9510`. The **`SHOW DISPLAY/SYMBOLS`** command then defines a DCL symbol for the port value. The symbol is then used to set the port value for a new display device.

5.
```
$ SET DISPLAY/CREATE/PROXY=GATEWY/NODE=hubbub.company.com -
_$ /PMTRANSPORT=DECNET/NOLBXAUTHENTICATE/TRANSPORT=TCPIP
$ SHOW DISPLAY
    Device:    WSA23:  [super]
    Node:      HUBBUB.COMPANY.COM
    Transport: TCPIP
    Server:    0
    Screen:    0
  Connection will use:
    LBX proxy on node:  gatewy.company.com
    Transport to proxy: TCPIP
    Server number:      63
$ RUN SYS$SYSTEM:DECW$CLOCK
```

In this example, you are logged into a local workstation (`FLOPSY`) and want to connect to an external host (`hubbub.company.com`), which is outside of your company's firewall. You use the proxy manager on a gateway host (`GATEWY`) to start the LBX proxy server, connect to the external host, and display the DECwindows `Clock` application.

Note that DECnet is used within the local area network (LAN) to communicate to the proxy manager, and TCP/IP is used by the proxy server to communicate to the X display server on the external host. When using a managed proxy server, the proxy server chooses the transport for the client-to-proxy server connection.

SET DUMP_OPTIONS (x86-64 Only)

SET DUMP_OPTIONS (x86-64 Only) — Updates the list of alternate devices where system crash dumps can be written.

Format

SET DUMP_OPTIONS /DEVICE=(*device*[,...]) [/LOG]

Parameters

None.

Description

This command does not create a dump file on any specified device; the recommended method is to use the **CREATE** command in the SYSGEN utility. If a specified device is not mounted or no dump file is found there, a warning will be output.

The updated list of devices is saved in system memory for immediate use, and in the file SYS$MANAGER:SYS$DUMP_CONFIG.DAT for subsequent system reboots.

Qualifiers

/DEVICE=(*device*[,...])
/NODEVICE

Specify the devices containing a system dump file (SYSDUMP.DMP) in the directory [SYS*n*.SYSEXE]. You can omit the parentheses when you specify only a single device. Only non-shadowed devices that are directly connected to the system can be specified (neither MSCP-served nor shadow set units can be used).

The previous contents of the list are replaced by the specified list. Use **/NODEVICE** to clear the list of alternate devices and only allow dumps to be written to the dump file on the system disk.

/LOG

Outputs informational messages.

Examples

1.
```
$ SET DUMP_OPTIONS /DEVICE=DKA100:
```

2.
```
$ SET DUMP_OPTIONS /DEVICE=$77$DKA200:/LOG
%SET-I-FILEUPDATED, SYS$MANAGER:SYS$DUMP_CONFIG.DAT updated
%SET-I-MEMUPDATED, memory settings updated
```

3. `$ SET DUMP_OPTIONS /DEVICE=($77$DKA200:,DKA100:)`

   In this example, `DKA100:` will only be used as a system dump device if `$77$DKA200:` is not accessible at the time of the system crash.

SET ENTRY

SET ENTRY — Changes the current status or attributes of a job not currently executing in a queue. **Requires delete (D) access to the job or manage(M) access to the queue.**

Format

SET ENTRY *entry-number*[,...]

Parameter

***entry-number*[,...]**

Specifies the entry number of the job you want to change. If you specify more than one entry, separate the entry numbers with commas (,).

The system assigns a unique entry number to each queued print or batch job in the system. By default, the **PRINT** and **SUBMIT** commands display the entry number when they successfully queue a job for processing. These commands also create or update the local symbol $ENTRY to reflect the entry number of the most recently queued job. To find a job's entry number, enter the **SHOW ENTRY** or the **SHOW QUEUE** command.

Description

The **SET ENTRY** command allows you to change the status or attributes of one or more jobs, as long as the jobs are not executing currently. You cannot affect individual files within a multifile job with the **SET ENTRY** command.

The qualifiers enable you to specify different attributes or delete attributes. Some qualifiers apply to both batch and print jobs. Others are restricted to either batch jobs or print jobs. The defaults for all the **SET ENTRY** qualifiers are the attributes and status that the job has before you enter the **SET ENTRY** command.

This command requires delete (D) access to the job, manage (M) access to the queue, or OPER (operator) privilege.

Qualifiers

/AFTER=*time*
/NOAFTER

Requests that the specified job be held until after a specific time. If the specified time has already passed, the job is scheduled for processing.

You can specify either absolute time or a combination of absolute and delta times. For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

Note

Releasing or rescheduling a job before its submission time is not supported for security reasons. Therefore, if you set the system time to a future time and submit a job on anode where the queue manager is running, you cannot release or reschedule that job back to the present time if you reset the system time.

For example, you can set the system time to a future time of January 1, 2020 and submit a job to run on January 9, 2020; however, if you set the system time back to the present time and try to release the job to the present time, the job will be released to January 1, 2020.

This situation impacts the following qualifiers: **/AFTER**, **/NOHOLD**, and **/RELEASE**.

To specify **/AFTER** for a job on hold, you must also specify **/NOHOLD** in order to cause the job to be held only until the specified time. Jobs can be released before the specified time by using the **/NOAFTER** or **/RELEASE** qualifier with the **SET ENTRY** command.

/BURST
/NOBURST

Controls whether two file flag pages with a burst bar between them are printed preceding each file in a job.

Use the **/[NO]BURST** qualifier to override the **/DEFAULT** qualifier options that have been set for the output queue you are using. The **/[NO]BURST** qualifier does not override the **/SEPARATE** qualifier options set for the queue.

When you specify the **/BURST** qualifier for a file, the **/[NO]FLAG** qualifier does not add or subtract a flag page from the two flag pages that are printed preceding a file.

/CHARACTERISTICS=(*characteristic*[,...])
/NOCHARACTERISTICS

Specifies the name or number of one or more characteristics to be associated with the job. Characteristics can refer to such things as color of ink. If you specify only one characteristic, you can omit the parentheses.

A characteristic's number must range from 0 to 127. To see which characteristics have been defined for your system, use the **SHOW QUEUE/CHARACTERISTICS** command. To see which characteristics are associated with a particular queue, use the **SHOW QUEUE/FULL** command.

A print job can be processed on an execution queue if none, some, or all of the characteristics associated with the queue also are associated with the job. That is, the job's characteristics must be a subset of the queue's characteristics. However, if any of the characteristics associated with the job are not associated with the queue, the job remains pending until one or more of the following occurs:

- The characteristics specified with the queue are changed to make the job's characteristics a subset of the queue's characteristics (using, for example, the **SET QUEUE/CHARACTERISTICS** command).
- The characteristics specified with the job are changed to make the job's characteristics a subset of the queue's characteristics (using, for example, the **SET ENTRY/CHARACTERISTICS** command).
- The job is moved to a queue on which all the job's characteristics have been specified (using, for example, the **SET ENTRY/REQUEUE** command).

- The job is deleted (using, for example, the **DELETE/ENTRY** command).

/CLI=*filename*

Specifies the name of a command language interpreter (CLI) to use in processing the batch job. The file name specifies that the CLI be SYS$SYSTEM:*filename*.EXE. If you do not specify the **/CLI** qualifier, the job is run by the CLI specified in the user authorization file (UAF), or whatever CLI was specified when the job was originally submitted to the queue.

/COPIES=*n*

Specifies the number of copies to print. The value of *n* can be any number from 1 to 255. When you use the **/COPIES** qualifier with the **SET ENTRY** command, the number of copies can apply only to the entire print job. You cannot use this qualifier to specify different numbers of copies for individual files within a multifile job.

/CPUTIME=*time*

Specifies a CPU time limit for the batch job. You can specify the *time* parameter as delta time, the value 0, INFINITE, or NONE. If the queue on which the job executes has a defined CPUMAXIMUM value, the smaller of the specified job and queue values is used. If the queue on which the job executes does not have a specified maximum CPU time limit, the smaller of the **SUBMIT** command and user authorization file (UAF) values is used. If the queue on which the job executes does not have a specified maximum CPU time limit and the UAF has a specified CPU time limit of NONE, either the value 0 or the keyword INFINITE allows unlimited CPU time. If you specify NONE, the specified queue or UAF value is used. CPU time values must be greater than or equal to the number specified by the system parameter PQL_MCPULM.

For more information on specifying CPU time limits, see the "CPU Time Limit Specifications and Actions" table for the **INITIALIZE/QUEUE** command in the *VSI OpenVMS DCL Dictionary: A-M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_119]. For information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/FEED
/NOFEED

Controls whether form feeds are inserted into the print job when the printer reaches the bottom margin of the form in use. You can suppress this automatic form feed (without affecting any of the other carriage control functions that are in place) by using the **/NOFEED** qualifier.

When you use the **/FEED** qualifier with the **SET ENTRY** command, the qualifier applies to all files in the print job. You cannot use this qualifier to specify form feeds for individual files within a multifile job.

/FLAG
/NOFLAG

Controls whether a flag page is printed preceding each file in a print job. The flag page contains the name of the user submitting the job, the job entry number, and other information about the file being printed.

Use the **/[NO]FLAG** qualifier to override the installation-defined defaults that have been set for the output queue you are using or to override the qualifier specified in the **PRINT** command that queued the job.

/FORM=*form*

Specifies the name or number of the form to be associated with the print job. If you omit the **/FORM** qualifier, the default form for the execution queue is associated with the job.

Forms have attributes such as print image width and length or paper stock, which the print symbiont associates with a job when the job is processed. To see which forms have been defined for your system, use the **SHOW QUEUE/FORM** command. To find out which form is mounted currently on a particular queue and which form is specified as that queue's default form, use the **SHOW QUEUE/FULL** command.

The stock of the form associated with the job must match the stock of the form mounted on the execution queue on which you want the job to be processed. If the stocks do not match, the job remains pending until one or more of the following occurs:

- A form with the same stock as the job's form is mounted on the queue(using, for example, the **SET QUEUE/FORM_MOUNTED** command).
- A form with the same stock as the queue's mounted form is specified with the job (using, for example, the **SET ENTRY/FORM** command).
- The job is moved to a queue on which the stock of the mounted form matches the stock of the job's form (using, for example, the **SET ENTRY/REQUEUE** command).
- The job is deleted (using, for example, the **DELETE/ENTRY** command).

/HEADER
/NOHEADER

Controls whether a heading line is printed at the top of each output page in a print job.

/HOLD
/NOHOLD

Controls whether the job is to be made available for processing or held for processing later. If you specify the **/HOLD** qualifier, the job is not released for processing until you enter **SET ENTRY/NOHOLD** or **SET ENTRY/RELEASE**.

You can use the **/NOHOLD** qualifier to release jobs that have been held for the following reasons:

- A job was submitted with the **/HOLD** qualifier.
- A completed job is being retained in a queue.
- A user-written symbiont has refused a job.

Note

Releasing or rescheduling a job before its submission time is not supported for security reasons. Therefore, if you set the system time to a future time and submit a job on anode where the queue manager is running, you cannot release or reschedule that job back to the present time if you reset the system time.

For example, you can set the system time to a future time of January 1, 2020 and submit a job to run on January 9, 2020; however, if you set the system time back to the present time and try to release the job to the present time, the job will be released to January 1, 2020.

This situation impacts the following qualifiers: **/AFTER**, **/NOHOLD**, and **/RELEASE**.

/JOB_COUNT=*n*

Requests that an entire print job be printed n times, where *n* is a decimal integer from 1 to 255. This qualifier overrides the **/JOB_COUNT** qualifier with the **PRINT** command.

/KEEP
/NOKEEP

Controls whether the batch job log file is deleted after it is printed.

/LOG_FILE[=*filespec*]
/NOLOG_FILE

Creates a log file with the specified file specification. You can specify a different device name, as long as the process executing the batch job has access to the device on which the log file will reside. Logical names in the file specification are translated in the context of the process that executes the **SET ENTRY** command.

If you omit the **/LOG_FILE** qualifier and specify the **/NAME** qualifier, the log file is written to a file having the same file name as that specified by the **/NAME** qualifier; the file type is .LOG. When you omit the **/LOG_FILE** qualifier, the job-name value used with the **/NAME** qualifier must be a valid file name.

/LOWERCASE
/NOLOWERCASE

Indicates whether the print job must be printed on a printer that can print both uppercase and lowercase letters. The **/NOLOWERCASE** qualifier means that files can be printed on printers that print only uppercase letters. If all available printers can print both uppercase and lowercase letters, you do not need to specify the **/LOWERCASE** qualifier.

/NAME=*job-name*

Names the job. The job name must be 1 to 39 alphanumeric characters. The **SHOW ENTRY** and **SHOW QUEUE** commands display the job name. For batch jobs, the job name is also used for the batch job log file. For print jobs, the job name is also used on the flag page of the printed output.

The default job name is the name of the first file in the job.

/NOCHECKPOINT

For a batch job, erases the value established by the most recently executed **SET RESTART_VALUE** command. For a print job, clears the stored checkpoint so that the job will restart from the beginning.

/NODELETE

Cancels file deletion for a job that was submitted with the **/DELETE** qualifier. If you did not specify the **/DELETE** qualifier when the job was originally submitted to the queue, you cannot use the SET ENTRY command to establish file deletion at a later time. You cannot use the **/NODELETE** qualifier to cancel deletion of individual files in a multifile job.

/NOTE=*string*

Specifies a message of up to 255 characters to appear on the flag page of the print job. Enclose messages containing lowercase letters, blanks, or other non alphanumeric characters (including spaces) in quotation marks (" ").

/NOTIFY
/NONOTIFY

Controls whether a message notifies you when your job has been completed or aborted. Notification is sent to any terminal session on the same OpenVMS Cluster system to which you are logged in.

/OPERATOR=*string*

Specifies a message string of up to 255 characters to be sent to the operator just before the print job begins to print. Enclose the message in quotation marks (" ") if it contains spaces, special characters, or lowercase characters.

/PAGES=([*lowlim*,]*uplim*)

Specifies the number of pages to print for the specified job. You can use the **/PAGES** qualifier to print portions of long files. By default, all pages of the file are printed. When you use the **/PAGES** qualifier with the **SET ENTRY** command, the qualifier can apply only to an entire job. You cannot use this qualifier to specify different numbers of pages to be printed for individual files within a multifile job.

The *lowlim* specifier refers to the first page of the file that you want to print. If you omit the *lowlim* specifier, the printing starts on the first page of the file.

The *uplim* specifier refers to the last page of the file that you want to print. When you want to print to the end of the file but do not know how many pages are in the file, use quotation marks (" ") as the *uplim* specifier.

You can omit the parentheses when you specify only a value for the *uplim* specifier. For example, **/PAGES=10** prints the first 10 pages of the file; **/PAGES=(5,10)** prints pages 5 to 10; **/PAGES=(5,"")** starts printing at page 5 and continues until the end of the file is reached.

/PARAMETERS=(*parameter*[,...])

Specifies from one to eight optional parameters to be passed to the job. Each parameter can have as many as 255 characters. If you specify only one parameter, you can omit the parentheses. To specify a parameter that contains any special characters or delimiters, enclose the parameter in quotation marks.

For batch jobs, the parameters define values to be equated to the symbols named P1 to P8 in each command procedure in the job. The symbols are local to the specified command procedures.

/PASSALL
/NOPASSALL

Specifies whether the symbiont bypasses all formatting of the print job and sends the output QIO to the driver with format suppressed. All qualifiers affecting formatting, as well as the **/HEADER**, **/PAGES**, and **/PAGE_SETUP** qualifiers, are ignored.

When you use the **/PASSALL** qualifier with the **SET ENTRY** command, the qualifier applies to the entire job. You cannot use this qualifier to specify PASSALL mode for individual files within a multifile job.

/PRINTER[=*queue-name*]
/NOPRINTER

Queues the batch job log file for printing when the job is completed. The default output queue for the log file is SYS$PRINT. The **/PRINTER** qualifier allows you to specify an output queue. The **/NOPRINTER** qualifier assumes the **/KEEP** qualifier.

/PRIORITY=*n*

Requires OPER (operator) or ALTPRI (alter priority) privilege to raise the priority above the value of the queue's maximum scheduling priority.

Specifies the job-scheduling priority of the job. The value of *n* is an integer in the range of 0 to 255, where 0 is the lowest priority and 255 is the highest.

The default value for the **/PRIORITY** qualifier is the value of the system parameter DEFQUEPRI. No privilege is needed to set the priority lower than the queue's maximum scheduling priority.

/RAD=*n*

Specifies the RAD number on which the submitted batch job is to execute.

When a job is submitted to a batch queue that does not have a RAD setting, the job will execute using the RAD specified. If the batch queue has a RAD setting, the job will execute using the RAD specified on the queue.

RAD is supported on AlphaServer GS series systems and, starting from OpenVMS Version 8.4, support is extended to NUMA-capable IA-64 systems.

/RELEASE

You can use the **/RELEASE** qualifier to release jobs that have been held for the following reasons:

- A job was submitted with the **/AFTER** qualifier.
- A job was submitted with the **/HOLD** qualifier.
- A completed job is being retained in a queue.
- A user-written symbiont has refused a job.

Note

Releasing or rescheduling a job before its submission time is not supported for security reasons. Therefore, if yous et the system time to a future time and submit a job on anode where the queue manager is running, you cannot release or reschedule that job back to the present time if you reset the system time.

For example, you can set the system time to a future time of January 1, 2020 and submit a job to run on January 9, 2020; however, if you set the system time back to the present time and try to release the job to the present time, the job will be released to January 1, 2020.

This situation impacts the following qualifiers: **/AFTER**, **/NOHOLD**, and **/RELEASE**.

/REQUEUE=*queue-name*[:]

Requests that the job be moved from the original queue to the specified queue.

/RESTART
/NORESTART

Specifies whether a batch or print job is restarted after a system failure or a **STOP/QUEUE/REQUEUE** command.

/RETAIN=*option*

Specifies the circumstances under which you want your jobs to be retained in a queue. When a job is retained in the queue, you can issue the **SHOW QUEUE** command after the job completes to see the status of the job. Without job retention, no record of a job is left in a queue after a job completes.

Use the following options to specify job retention:

- ALWAYS – Holds the job in the queue regardless of the job's completion status.
- DEFAULT – Holds the job in the queue as specified by the queue's retention policy.
- ERROR – Holds the job in the queue only if the job completes unsuccessfully.
- UNTIL=*time-value* – Holds the job in the queue for the specified length of time, regardless of the job's completion status.

Note

You cannot specify a **/NORETAIN** qualifier with the **SET ENTRY** command (as system managers can with the commands **INITIALIZE/QUEUE**, **START/QUEUE**, and **SET QUEUE**); however, you can specify **/RETAIN=DEFAULT** with the **SET ENTRY** command. The default option holds the job in the queue as specified by the queue's retention policy. If the system manager has not specified retention for the queue, the job is not retained.

How Job Retention Is Determined

Although you can specify job retention options for your own jobs, the job retention option you specify may be overridden by the job retention option of the queue on which your job executed. If you submit or print a job to a generic queue, the generic queue's job retention setting may also override the job retention option you specify. This section describes how job retention is determined.

An execution queue's job retention setting takes precedence over a generic queue's job retention setting; however, if the job's completion status does not match the job retention setting (if any) on the execution queue, then the generic queue's job retention setting attempts to control job retention. If the job's completion status does not match the job retention setting (if any)on the generic queue, then the user-specified job retention setting is used. Jobs submitted directly to execution queues are not affected by job retention settings on generic queues.

If the execution queue's retention setting applies, the job is retained on the execution queue. Likewise, if the generic queue's retention setting applies, the job is retained on the generic queue. If the user-specified setting applies, the job is retained in the queue to which it was submitted.

The following example illustrates how the queue manager determines job retention:

Suppose you submit a job to a generic queue and specify **/RETAIN=ALWAYS**, and the job completes successfully.

First, the queue manager compares the job's completion status to the execution queue's retention setting. Suppose the queue is set with **/RETAIN=ERROR** (retains only jobs that complete unsuccessfully). The job is not retained in the execution queue because the error condition was not met.

Second, the queue manager compares the job's completion status to the generic queue's retention setting. Suppose the generic queue has no retention setting. The queue manager's comparison again fails to retain the job.

Finally, the queue manager compares the job's completion status to the retention setting you specified for the job. This comparison reveals that the job should be retained. Because the user-specified setting leads the queue manager to retain the job, the job is held in the queue to which the job was submitted – in this case, the generic queue.

For more information on types of queues, see the **INITIALIZE/QUEUE** command in the *VSI OpenVMS DCL Dictionary: A-M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_119]. For more information on setting retention options for queues, see the **START/QUEUE** or **SET QUEUE** command, or the **INITIALIZE/QUEUE** command in the *VSI OpenVMS DCL Dictionary: A-M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_119].

Timed Retention

Timed retention, which you specify using the UNTIL=*time-value* option, allows you to retain a job in the queue only as long as you need it. This eliminates the need to delete the job from the queue later.

For example, the following command retains job 172 in the queue until 7:31 on April 19, when the job will automatically be deleted from the queue.

```
$ SET ENTRY/RETAIN=UNTIL=19-APR-2001:07:31:0.0 172
```

However, depending on the queue's job retention policy, the job might be retained indefinitely. The job retention policy set on the queue takes precedence over the user-specified job retention setting. Because system managers cannot specify timed job retention for a queue, any jobs retained as a result of a queue's setting are retained indefinitely.

If you specify the **/RETAIN=UNTIL=*time-value*** option, you must supply a time value. The time value is first interpreted as a delta time, then as a combination time, and finally as absolute time. If you specify a delta time, the delta begins when the job completes. For example, if you specify **SET ENTRY/RETAIN=UNTIL="+3:00"**, the job will be retained for three hours after the job completes. For information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT].

/SETUP=*module*[,...]

Extracts the specified modules from the device control library (containing escape sequence modules for programmable printers) and copies the modules to the printer before each file in a print job is printed.

When you use the **/SETUP** qualifier with the **SET ENTRY** command, the qualifier applies to the entire print job. You cannot use this qualifier to specify different setup modules for individual files within a multifile job.

/SPACE
/NOSPACE

Controls whether the output of a print job is double-spaced. Specifying the **/NOSPACE** qualifier causes the output to be single-spaced. When you use the **/SPACE** qualifier with the **SET ENTRY** command, the qualifier applies to the entire job. You cannot use this qualifier to specify different spacing for individual files within a multifile job.

/TRAILER
/NOTRAILER

Controls whether a trailer page is printed at the end of each file in a print job. The trailer page displays the entry number, as well as information about the user submitting the job and the files being printed.

When you use the **/TRAILER** qualifier with the **SET ENTRY** command, trailer pages are placed at the end of each file in a multifile job.

Use the **/[NO]TRAILER** qualifier to override the installation-defined defaults that have been set for the output queue you are using or the qualifier specified in the **PRINT** command that queued the job.

/WSDEFAULT=*n*

Defines, for a batch job, a working set default, which is the default number of physical pages that the job can use.

If the queue on which the job executes has a nonzero default working set, the smaller of the specified job and queue values is used. If the queue on which the job executes has a working set default of zero, the smaller of the specified job value and the value established in the user authorization file (UAF) is used.

Specify the value of *n* as a number of 512-byte pagelets on Alpha. Note that the operating system rounds up this value to the nearest CPU-specific page so that the actual amount of physical memory allowed may be larger than the specified amount on Alpha. If you specify zero or NONE, the specified queue or UAF value is used. Working set default values must range between the numbers specified by the system parameters PQL_MWSDEFAULT and WSMAX.

/WSEXTENT=*n*

Defines, for the batch job, a working set extent, which is the maximum amount of physical memory that the job can use. The job uses the maximum amount of physical memory only when the system has excess free pages. If the queue on which the job executes has a nonzero working set extent, the smaller of the specified job and queue values is used. If the queue on which the job executes has a working set extent of zero, the smaller of the specified job value and the value established in the user authorization file (UAF) is used.

Specify the value of *n* as a number of 512-byte pagelets on Alpha. Note that the operating system rounds up this value to the nearest CPU-specific page so that the actual amount of physical memory allowed may be larger than the specified amount on Alpha. If you specify zero or NONE, the specified queue or UAF value is used. Working set extent values must range between the numbers specified by the system parameters PQL_MWSEXTENT and WSMAX.

/WSQUOTA=*n*

Defines, for the batch job, a working set quota, which is the amount of physical memory that the job is guaranteed. If the queue on which the job executes has a nonzero working set quota, the smaller of the specified job and queue values is used. If the queue on which the job executes has a working set quota of zero, the smaller of the specified job value or the value established in the user authorization file (UAF) is used.

Specify the value of *n* as a number of 512-byte pagelets on Alpha. Note that the operating system rounds up this value to the nearest CPU-specific page so that the actual amount of physical memory allowed may be larger than the specified amount on Alpha. If you specify zero or NONE, the

specified queue or UAF value is used. Working set quota values must range between the numbers specified by the system parameters PQL_MWSQUOTA and WSMAX.

Examples

1.
```
$ PRINT/HOLD   MYFILE.DAT
  Job MYFILE (queue SYS$PRINT, entry 112) holding
$ SET ENTRY 112/RELEASE/JOB_COUNT=3
```

   The **PRINT** command in this example requests that the file MYFILE.DAT be queued to the system printer, but placed in a hold status. The **SET ENTRY** command releases the job for printing and requests that three copies of the job be printed.

2.
```
$ SUBMIT CLIMATE
  Job CLIMATE (queue SYS$BATCH, entry 211) pending
$ SET ENTRY 211/HOLD/NAME=TEMP
```

   The **SUBMIT** command in this example queues the command procedure CLIMATE.COM for processing as a batch job. The **SET ENTRY** command places the job in a hold state and changes the job name to TEMP, assuming that the job has not yet begun execution.

3.
```
(wide)
$ PRINT/FLAG=ALL/AFTER=20:00 MEMO.MEM, LETTER.MEM, REPORT.MEM/SPACE
  Job MEMO (queue SYS$PRINT, entry 172) holding until 20:00
$ SET ENTRY 172 /BURST/NOSPACE/HEADER
```

   The **PRINT** command in this example requests that three files be printed after 8:00 P.M. on the default printer with flag pages preceding each file. It also requests that the file REPORT.MEM be double-spaced. Later a **SET ENTRY** command calls for a burst page at the beginning of each file and requests that all files in the job be single-spaced. This command also requests that headers be printed on each page of each file in the job.

4.
```
$ SUBMIT/HOLD/QUEUE=ANYRADQ  TEST.COM
Job TEST (queue ANYRADQ, entry 23) holding

$ SET ENTRY 23 /RAD=1
$ SHOW ENTRY/FULL 23
 Entry  Jobname         Username     Blocks  Status
 -----  -------         --------     ------  ------
    23  TEST            SYSTEM               Holding
     On idle batch queue ANYRADQ
     Submitted 24-JUL-2001 14:19:37.44 /KEEP /NOPRINT /PRIORITY=100 /
RAD=0
     File: _$1$DKB200:[SWEENEY.CLIUTL]TEST.COM;1
```

   In this example, TEST.COM is submitted to the queue ANYRADQ, and the **SET ENTRY** command is used to set the RAD to 1.

SET FILE

SET FILE — Modifies the characteristics of one or more files. **See the qualifier descriptions for restrictions.**

Format

SET FILE *filespec*[,...]

Parameter

***filespec*[,...]**

Specifies one or more files to be modified. If you specify more than one file, separate the file specifications with commas (,). The asterisk (*) and the percent sign (%) wildcard characters are allowed.

Description

The **SET FILE** command modifies a number of file characteristics.

Qualifiers

/ATTRIBUTE=(*file-attribute*[,...])

Sets the attributes associated with a file. The following table lists possible keywords and the relationship to both ACP-QIO and OpenVMS RMS File attributes:

Keyword	**ACP-QIO File Attribute**[1]	**OpenVMS RMS File Attribute**[2]
ATTDATE=*date*	FAT$C_ATTDATE	XAB$Q_ATT
ACCDATE=*date*	FAT$C_ACCDATE	XAB$Q_ACC
BAKDATE=*date*	FAT$C_BAKDATE	XAB$Q_BDT
BKS:*value*	FAT$B_BKTSIZE=*byte*	FAB$B_BKS=*byte*
CBT	FH2$V_CONTIGB, FHC $V_CONTIGB	FABV_CBT, XABV_CBT
CREDATE=*date*	FAT$C_CREDATE	XAB$Q_CDT
DEQ:*value*	FAT$W_DEFEXT=*word*	FAB$W_DEQ=*word*
EBK:*value*	FAT$L_EFBLK=*longword*	XAB$L_EBK=*longword*
EXPDATE=*date*	FAT$C_EXPDATE	XAB$Q_EDT
FFB:*value*	FAT$W_FFBYTE=*word*	XAB$W_FFB=*word*
FSZ:*value*	FAT$B_VFCSIZE=*byte*	FAB$B_FSZ=*byte*
GBC:*value*	FAT$W_GBC=*word*	FAB$W_GBC=*word*
HBK:*value*	FAT$L_HIBLK=*longword*	XAB$L_HBK=*longword*
LRL:*value*	FAT$W_RSIZE=*word*	XAB$W_LRL=*word*
MODDATE=*date*	FAT$C_MODDATE	XAB$Q_MOD
MRS:*value*	FAT$W_MAXREC=*word*	FAB$W_MRS=*word*
ORG:IDX	FAT$V_FILEORG=FAT $C_INDEXED	FAB$B_ORG=FAB$C_IDX
ORG:REL	FAT$V_FILEORG=FAT $C_RELATIVE	FAB$B_ORG=FAB$C_REL
ORG:SEQ	FAT$V_FILEORG=FAT $C_SEQUENTIAL	FAB$B_ORG=FAB$C_SEQ

Keyword	ACP-QIO File Attribute[1]	OpenVMS RMS File Attribute[2]
RAT:BLK	FAT$B_RATTRIB=FAT $M_NOSPAN	FAB$B_RAT=FAB$M_BLK
RAT:CR	FAT$B_RATTRIB=FAT $M_IMPLIEDCC	FAB$B_RAT=FAB$M_CR
RAT:FTN	FAT$B_RATTRIB=FAT $M_FORTRANCC	FAB$B_RAT=FAB$M_FTN
RAT:MSB	FAT$B_RATTRIB=FAT $M_MSBVAR	FAB$B_RAT=FAB$M_MSB
RAT:NONE	FAT$B_RATTRIB=0	FAB$B_RAT=0
RAT:PRN	FAT$B_RATTRIB=FAT $M_PRINTCC	FAB$B_RAT=FAB$M_PRN
REVDATE=*date*	FAT$C_REVDATE	XAB$Q_RDT
RFM:FIX	FAT$V_RTYPE=FAT $C_FIXED	FAB$B_RFM=FAB$C_FIX
RFM:STM	FAT$V_RTYPE=FAT $C_STREAM	FAB$B_RFM=FAB$C_STM
RFM:STMCR	FAT$V_RTYPE=FAT $C_STREAMCR	FAB$B_RFM=FAB $C_STMCR
RFM:STMLF	FAT$V_RTYPE=FAT $C_STREAMLF	FAB$B_RFM=FAB $C_STMLF
RFM:UDF	FAT$V_RTYPE=FAT $C_UNDEFINED	FAB$B_RFM=FAB$C_UDF
RFM:VAR	FAT$V_RTYPE=FAT $C_VARIABLE	FAB$B_RFM=FAB$C_VAR
RFM:VFC	FAT$V_RTYPE=FAT$C_VFC	FAB$B_RFM=FAB$C_VFC
VRS:*value*	FAT$W_VERSIONS=*word*	XAB$W_VERLIMIT=*word*

[1]For further information, see the *VSI OpenVMS I/O User's Reference Manual* [https://docs.vmssoftware.com/vsi-openvms-io-user-s-reference-manual/].

[2]For further information, see the *VSI OpenVMS Record Management Services Reference Manual* [https://docs.vmssoftware.com/vsi-openvms-record-management-services-reference-manual/].

/BACKUP
/NOBACKUP

Specifies that the Backup utility (BACKUP) records the contents of the file. The **/NOBACKUP** qualifier causes BACKUP to record the attributes of the file but not its contents. This qualifier is valid only for Files-11 Structure On-Disk Level 2 and 5 files.

The **/NOBACKUP** qualifier is useful for saving files that contain unimportant data, such as SWAPFILES.

/BEFORE[=*time*]

Selects only those files dated prior to the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, LOGIN,

TODAY (default), TOMORROW, or YESTERDAY. Specify the **/CREATED** or the **/MODIFIED** qualifier to indicate the time attribute to be used as the basis for selection. The **/CREATED** qualifier is the default.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/BY_OWNER[=*uic*]

Selects only those files whose owner user identification code (UIC) matches the specified owner UIC. The default UIC is that of the current process.

Specify the UIC by using standard UIC format as described in the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#UIC_SECTION].

/CACHING_ATTRIBUTE=*keyword*

Use this qualifier to control which files are cached by the Extended File Cache. It sets the caching attribute for a file or directory in a Files-11 ODS-2 or ODS-5 volume.

The caching attribute of a file is the default caching option that is used by the Extended File Cache when an application accesses the file without specifying which caching option it wants to use.

The *keyword* can be either WRITETHROUGH or NO_CACHING. Use WRITETHROUGH for files that you want to be cached. Use NO_CACHING for files that you don't want to be cached.

The Extended File Cache does not cache directories. The caching attribute of a directory controls only how the caching attribute is inherited by new files and subdirectories created in the directory:

- When you create a new directory or file, it inherits its caching attribute from its parent directory.
- When you create a new version of an existing file, the new file inherits its caching attribute from the highest version of the existing file.

When you use the **INITIALIZE** command to create a new Files-11 volume, the caching attribute of its root directory (000000.DIR;1) is set to write-through. This means that by default, all the files and directories you create in the volume will inherit a caching attribute of write-through unless you use **SET FILE/CACHING_ATTRIBUTE**.

When you change the caching attribute of a directory, it does not affect the caching attribute of any existing files and subdirectories in the directory.

When you change the caching attribute of a file, it does not affect the type of caching being used by any applications that are currently accessing the file.

/CONFIRM
/NOCONFIRM (default)

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

YES NO QUIT

TRUE	FALSE	**Ctrl/Z**
1	0	ALL
	Return	

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), but these abbreviations must be unique. Affirmative answers are YES, TRUE, and 1. Negative answers include: NO, FALSE,0, and pressing **Return**. Entering QUIT or pressing **Ctrl/Z** indicates that you want to stop processing the command at that point. When you respond by entering ALL, the command continues to process, but no further prompts are given. If you type a response other than one of those in the list, DCL issues an error message and redisplays the prompt.

/CREATED (default)

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/CREATED** qualifier selects files based on their dates of creation. This qualifier is incompatible with the **/MODIFIED** qualifier, which also allows you to select files according to time attributes. The **/CREATED** qualifier is the default qualifier.

/DATA_CHECK[=([NO]READ,[NO]WRITE)]

Specifies whether a read data check (rereading each record), a write data check (reading each record after it is written), or a combination of the two is performed on the file during transfers. By default, a write data check is performed.

/END_OF_FILE

Resets the end-of-file (EOF) mark to the highest block allocated.

/ENTER=*new-filespec*

The *new-filespec* parameter is used to create either an alias or a hard link for the file specified in the **SET FILE** command. For detailed information about using hard links and aliases, see the relevant section in the *VSI OpenVMS System Manager's Manual* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#HARD_LINKS].

Normally you would use **/ENTER** to create an alias or a hard link in a directory different from the one where the original file name resides. If the names are not in different directories, you or another user could subsequently lose data during a delete or purge operation. The **DELETE** and **PURGE** commands and the file version limit feature can behave unpredictably if the original name and the new name are in the same directory.

To remove an alias or hard link, use the **SET FILE/REMOVE** command. Exercise caution when using the **DELETE** and **SET FILE/REMOVE** commands, or you could end up with either an inaccessible file that has no name or a name that does not refer to a file. Follow these guidelines to avoid such problems:

- Use **SET FILE/REMOVE** to remove an alias; do not use the **DELETE** command to remove an alias.
- Do not use **SET FILE/REMOVE** to remove the original file name.

If you do not follow these guidelines and encounter problems, use **ANALYZE/DISK/REPAIR** to move inaccessible files to the SYSLOST directory and remove names that no longer refer to files.

/ERASE_ON_DELETE

Specifies that the specified files are erased from the disk (not just written over) when the **DELETE** or **PURGE** command is issued for the files. See the **DELETE/ERASE** command in the *VSI OpenVMS DCL Dictionary: A–M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_61] for more information.

/EXCLUDE=(*filespec*[,...])

Excludes the specified file from the **SET FILE** operation. You can include a directory name but not a device name in the file specifications. The asterisk (*) and the percent sign (%) wildcard characters are allowed in the file specification; however, you cannot use relative version numbers to exclude a specific version. If you specify only one file, you can omit the parentheses.

/EXPIRATION_DATE=*date*
/NOEXPIRATION_DATE

Requires read (R), write (W), and control access. Being the owner of the file is one way to get control access.

Controls whether an expiration date is assigned to the specified files.

Specify the date according to the rules described in the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date. Absolute date keywords are allowed. If you specify zero as the date, today's date is used.

/EXTENSION[=*n*]

Sets the extend quantity default for the file. The value of the parameter *n* can range from 0 to 65,535. If you omit the value specification or specify a value of 0, OpenVMS Record Management Services (OpenVMS RMS) calculates its own value for the **/EXTENSION** qualifier.

See the **SET RMS_DEFAULT** command for a description of the **/EXTEND_QUANTITY** qualifier.

/GLOBAL_BUFFER[=*keyword*[=*n*]]
/NOGLOBAL_BUFFER

For OpenVMS versions prior to Version 8.3, sets the OpenVMS Record Management Services (OpenVMS RMS) global buffer count (the number of buffers that can be shared by processes accessing the file) for the specified files. The value *n* must be an integer in the range from 0 to 32,767. A value of 0 disables buffer sharing.

The **/SHARE** qualifier can be used to enable or disable global buffers on a file currently being accessed; however, any new global buffer settings will only be applied to new accessors of the file. If a file is already open with global buffers, any new number of global buffers will not take effect until the file is closed by all accessors of the file.

For OpenVMS V8.3 and later, sets the OpenVMS RMS global buffer count for the specified files. Note, you can specify only one type of global buffer qualifier in the same command string.

The *keyword* can be:

- COUNT=*n* – The value *n* sets the longword count of the number of global buffers. For indexed files, this keyword allows you to exceed the traditional maximum of 32767. For other files, the maximum of 32767 still applies.

- PERCENT=*p* – The value *p* expresses the size of the global cache as a percentage of the total number of used blocks currently used in the file.
- DEFAULT – Requests RMS at runtime to recalculate the global cache size based on an algorithm that makes use of two global buffer SYSGEN parameters, GB_CACHEALLMAX and GB_DEFPERCENT.

The following qualifiers can also be used with the **/SHARE** qualifier:

- **/GLOBAL_BUFFER=*n***
- **/GLOBAL_BUFFER=COUNT=*n***
- **/GLOBAL_BUFFER=PERCENT=*n***
- **/GLOBAL_BUFFER=DEFAULT**
- **/NOGLOBAL_BUFFER**
- **/[NO]STATISTICS**

/LOG
/NOLOG (default)

Displays the file specification of each file modified as the command executes.

/MODIFIED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/MODIFIED** qualifier selects files according to the dates on which they were last modified. This qualifier is incompatible with the **/CREATED** qualifier, which also allows you to select files according to time attributes. If you do not specify the **/MODIFIED** qualifier, the default is the **/CREATED** qualifier.

/MOVE
/NOMOVE

Controls whether movefile operations are enabled on the specified file.

When you create a file, movefile operations are enabled on that file. You should disable movefile operations on specialized files that are accessed other than through the XQP (such as files accessed through logical I/O to a disk).

Note that movefile operations are automatically disabled on critical system files. Do not enable movefile operations on these files.

/NODIRECTORY

Use with extreme caution. Requires SYSPRV (system privilege).

Removes the directory attributes of a file and allows you to delete the corrupted directory file even if other files are contained in the directory. When you delete a corrupted directory file, the files contained within it are lost.

Use **ANALYZE/DISK_STRUCTURE/REPAIR** to place the lost files in [SYSLOST]. You can then copy the lost files to a new directory. This qualifier is valid only for Files-11 On-Disk Structure Level 2 files. For more information about the Verify utility, see the relevant section

in the *VSI OpenVMS System Management Utilities Reference Manual, Volume I: A–L* [https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#_1_ERRORREPORTINGANDREPAIR].

/OWNER_UIC[=*uic*]

This qualifier has been superseded by the **SET SECURITY/OWNER** command.

/PROTECTION[=(*ownership*[:*access*][,...])]

This command has been superseded by the **SET SECURITY/PROTECTION** command.

/REMOVE

Use with caution.

Enables you to remove one of the names of a file that has more than one name, without deleting the file. If you have created an additional name for a file with the **/ENTER** qualifier of **SET FILE**, you can use the **/REMOVE** qualifier to remove either the original name or the alias. The file still exists and can be accessed by whatever name or names remain in effect.

However, if you accidentally remove the name of a file that has only one name, you cannot access that file with most DCL commands; use the **ANALYZE/DISK_STRUCTURE** utility to retrieve the file.

When **SET FILE/REMOVE** is executed on a symbolic link, the link is not removed. No error is displayed.

/SEMANTICS=*semantics-tag*
/NOSEMANTICS

Use the **/SEMANTICS** qualifier to create or change a semantics tag. Use the **/NOSEMANTICS** qualifier to remove a semantics tag from a file. For more information, see the *Guide to OpenVMS File Applications* [https://docs.vmssoftware.com/guide-to-openvms-file-applications/].

/SHARE

Allows you to enable or disable global buffers or statistics on a file currently being accessed by other users.

Requires SYSPRV privilege.

Only new accessors of the file acquire the new settings. For example, if a file is opened with no global buffers specified and the **SET FILE/GLOBAL=*n*/SHARE** command is issued, only new accessors of the file will use global buffers. If **/STATISTICS** is enabled on an active file, only operations performed by new accessors of the file are measured.

If a file is already open with global buffers, any new number of global buffers will not take effect until the file is closed by all accessors of the file.

The **/SHARE** qualifier is valid only with the following qualifiers:

- **/[NO]GLOBAL_BUFFER=*n***
- **/[NO]STATISTICS**

/SHELVABLE
/NOSHELVABLE

Controls whether the file is shelvable.

/SINCE[=*time*]

Selects only those files dated on or after the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, JOB_LOGIN, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify the **/CREATED** or the **/MODIFIED** qualifier to indicate the time attribute to be used as the basis for selection. The **/CREATED** qualifier is the default.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/STATISTICS
/NOSTATISTICS (default)

Enables the gathering of RMS statistics on the specified file. These statistics can then be viewed by using the Monitor utility, which is invoked with the DCL command **MONITOR**. The **SET FILE/STATISTICS** command applies an application ACE to the specified file. The ACE does not affect access control and is only meaningful to the application assigning it.

The **/SHARE** qualifier can be used to enable or disable statistics on a file currently being accessed. However, only statistics of new accessors of the file will be measured.

/STYLE=*keyword*

Specifies the file name format for display purposes.

The valid keywords for this qualifier are CONDENSED and EXPANDED. Descriptions are as follows:

Keyword	Description
CONDENSED (default)	Displays the file name representation of what is generated to fit into a 255-length character string. This file name may contain a DID or FID abbreviation in the file specification.
EXPANDED	Displays the file name representation of what is stored on disk. This file name does not contain any DID or FID abbreviations.

The keywords CONDENSED and EXPANDED are mutually exclusive. This qualifier specifies which file name format is displayed in the output message, along with the confirmation if requested.

File errors are displayed with the CONDENSED file specification unless the EXPANDED keyword is specified.

See the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#FILE_ACCESS] for more information.

/SYMLINK=*keyword*
/NOSYMLINK (default)

If an input file is a symbolic link, the file referred to by the symbolic link is the file that is set.

The **/SYMLINK** qualifier indicates that the symbolic link itself is set.

The valid keywords for this qualifier are [NO]WILDCARD, [NO]ELLIPSIS, and [NO]TARGET. Descriptions are as follows:

Keyword	Description
NOWILDCARD	Indicates that symlinks are disabled during directory wildcard searches.
WILDCARD	Indicates that symlinks are enabled during wildcard searches.
NOELLIPSIS	Indicates that symlinks are matched for all wildcard fields except for ellipsis.
ELLIPSIS	Equivalent to WILDCARD (included for command symmetry).
TARGET	Indicates that if the named file is a symlink, then the symlink is followed to operate on the symlink target.
NOTARGET	Indicates that the command operates on the named file whether it is an ordinary file or a symlink.

If the file named in the **SET FILE** command is a symlink, the command by default operates on the symlink target.

/TRUNCATE

Truncates the file at the end of the block containing the end-of-file (EOF)marker, that is, the qualifier releases allocated but unused blocks of the file.

/UNLOCK

Clears a file marked as deaccess locked. Deaccess locking is required by and used by those few applications that maintain their own locking and consistency, typically without the use of the OpenVMS distributed lock manager, and potentially also without the use of RMS. When an application using deaccess locking does not correctly deaccess the file (often due to an application or system failure), the file is marked as locked, and is thus inaccessible until the integrity of the contents of the file are verified and the **SET FILE/UNLOCK** command is used.

This command does not affect the state of files that are locked using RMS or the distributed lock manager.

For details on file deaccess locking, see the *VSI OpenVMS I/O User's Reference Manual* [https://docs.vmssoftware.com/vsi-openvms-io-user-s-reference-manual/], the ACP-QIO interface documentation, and specifically the FIB$V_DLOCK option available on the IO$_CREATE and IO $_ACCESS functions.

The **SET FILE/UNLOCK** command can clear the cause of the following error message:

```
%SYSTEM-W-FILELOCKED, file is deaccess locked
```

However, this command cannot resolve the cause of the error message:

```
%RMS-W-FLK, file currently locked by another user
```

/VERSION_LIMIT[=*n*]

Sets the maximum number of versions that a specified file can have in a directory. If you do not set a version limit, a value of 0 is used, indicating that the number of file versions is limited only to the Files-11 architectural limit of 32,767. When creating a file, if the total number of versions of that

file name exceeds the specified version limit, then the file with the lowest version number is deleted from the directory without notification to the user.

If you set the version limit to 3 when there are already five versions of that file in a directory, there will continue to be five versions of that file unless you specifically delete some or purge the directory. Once the number of file versions is equal to or less than the current version limit, this version limit is maintained.

The version limit applies to all existing versions of a specified file in a directory regardless of whether or not you specified any version in the command.

To view the version limit on a file, use the **DIRECTORY/FULL** command on a file name and look at the File Attributes field of the output or use the F$FILE_ATTRIBUTES(filename,"VERLIMIT") lexical function.

Examples

1. `$ SET FILE/EXPIRATION_DATE=19-DEC-2001:11:00 BATCH.COM;3`

   The **SET FILE** command requests that the expiration date of the file `BATCH.COM;3` be set to 11:00 A.M., December 19, 2001.

2. `$ SET FILE/BEFORE=31-DEC/ERASE_ON_DELETE PERSONNEL*.SAL`

   This **SET FILE** command calls for all files that match the file specification `PERSONNEL*.SAL` and are dated before December 31 of the current year to have their disk locations erased whenever one of them is deleted with commands such as **DELETE** or **PURGE**.

3. `$ SET FILE/OWNER_UIC=[360,020]/VERSION_LIMIT=100 MYFILE.DAT`

   The **SET FILE** command modifies the characteristics of the file `MYFILE.DAT`, changing the owner user identification code (*). You must have system privilege (SYSPRV) to change the owner UIC.

4. ```
   $ SET FILE/NOMOVE TEST.FDL
   $ DIRECTORY/FULL TEST.FDL

   Directory SYS$SYSDEVICE:[BERGMANN]

   TEST.FDL;1   File ID:  (10,8,0)
     .
     .
     .
   File attributes: Allocation: s, Extend: 0, Global buffer count: 0
      No version limit, MoveFile disabled
     .
     .
     .
   ```

   Movefile operations are disabled on the file `TEST.FDL`. A **DIRECTORY/FULL** command on the file `TEST.FDL` affirms that the file attribute Movefile is disabled.

5. ```
   $  SET FILE/ATTRIBUTES=ORG:SEQ -
   _$ TEST$:[DATA]SET_ATTRIBUTES.DATA_FILE/LOG
   %SET-I-MODIFIED, TEST$:[DATA]SET_ATTRIBUTES.DATA_FILE;1 MODIFIED
   ```

   The command, **SET FILE/ATTRIBUTES**, changes the file organization of the specified file.

6. ```
   $  SET FILE/PROTECTION=(S:RWE,O=RWE,G:RE,W:RE) TEMP.DIR
   $ DIRECTORY/PROTECTION TEMP.DIR

   Directory DKB0:[FULGHUM]

   TEMP.DIR;1            (RWE,RWE,RE,RE)
   ```

   This example sets the protection on the `TEMP.DIR` file with the **SET FILE** command and then displays the protection of the file with the **DIRECTORY** command.

7. ```
   $  SET FILE/SHARE/GLOBAL_BUFFER=5000/STATISTICS INVENTORY.IDX
   ```

   This example sets `5000` global buffers on the `INVENTORY.IDX` file and enables statistics. If the file is open and the **SET FILE** command is issued without the **/SHARE** qualifier, the following error is returned: SYSTEM-W-ACCONFLICT (file access conflict). The **/SHARE** qualifier allows the global buffers and statistics to be enabled on an open file; however, these settings only apply to new accessors of the file.

8. ```
   $ SET FILE/GLOBAL_BUFFER=100 NEWFILE.DAT
   $ SET FILE/GLOBAL_BUFFER=COUNT=100000 NEWFILE.DAT
   ```

   In a clustered environment with mixed OpenVMS versions, the same file can be opened on different nodes with different global buffer counts. For nodes prior to Version 8.3, use the old compatibility setting, and for Version 8.3 nodes and later use the new values.

9. ```
   $ DIRECTORY NEWFILE.TXT

   Directory WORK:[DOCUMENTS]

   NEWFILE.TXT;3
   NEWFILE.TXT;2
   NEWFILE.TXT;1

   $ SET FILE/VERSION_LIMIT=10/LOG NEWFILE.TXT;
   %SET-I-MODIFIED, WORK:[DOCUMENTS]NEWFILE.TXT;3 modified
   ```

   This example sets the version limit of 10 on all three existing versions of `NEWFILE.TXT`. Note that in this case, the **/LOG** qualifier shows only the highest version file specification as modified though the version limit has been applied to all file versions.

SET HOST

SET HOST — Connects your terminal (through the current host processor) to another processor, called the remote processor.

Format

SET HOST *node-name*

Parameter

node-name

Specifies the node name of the remote processor to which you will connect.

Description

The **SET HOST** command connects you to another processor on a network. The command requires that:

- **You must have an account on the remote system to log in.**
- **The NETMBX (network mailbox) privilege is set.**

(The **SHOW NETWORK** command lists the names of nodes accessible to your node.) Once the connection is made, the remote processor prompts for a user name and password. You must have an account on the remote processor to log in.

Once you have connected to the remote processor and have logged in, you can use DCL commands just as you would on your local processor. You can even use the **SET HOST** command to connect to another remote processor.

For example, if your local node is BOSTON, you can use the command **SET HOST ALBANY** to connect to the node ALBANY. You can then use the command **SET HOST AKRON** to connect (still through BOSTON and ALBANY) to the node AKRON.

Use the **LOGOUT** command to log out of the last processor you have logged into and return to the previous processor. For example, when you use the **LOGOUT** command, you have logged out of (and disconnected from) the processor at node AKRON, but you are still logged in (and connected) to the processor at ALBANY. A second **LOGOUT** command logs you out of node ALBANY, and disconnects you from it. A third **LOGOUT** command logs you out of the local processor, BOSTON.

You can also abort operations and return directly to the original host processor, if necessary. Press **Ctrl/Y** at least two times in rapid succession. the following message is displayed:

```
Are you repeating ^Y to abort the remote session?
```

If you respond Y or YES, control returns to the original node. Other responses, such as N or NO, do not abort the connection. This technique is useful when you want to exit quickly without entering a series of **LOGOUT** commands, or when part of the network becomes disconnected and you want to return to the host.

Note that **SET HOST**, unlike the OpenVMS terminal driver, buffers output data from an executing program. Buffering improves performance in wide-area networks; however, in the case of programs providing output *only*, the buffering causes a discrepancy between what is happening in the remote program and what is displayed on the local terminal. That is, a program might finish executing before you see the output on the local terminal.

This discrepancy can be particularly confusing when you use the **Ctrl/Y** or the **Ctrl/C** function, or out-of-band abort characters to abort the execution of a program. For example, when you press **Ctrl/Y** or **Ctrl/C** (or enter out-of-band abort characters) during the execution of a captive command procedure, **SET HOST** immediately stops the display on the local terminal. It also aborts the current read and write operations and any pending write operations, *including all buffered write operations*. Therefore, although it seems that the remote program aborts at the point in the program at which the display on the local terminal is stopped, the program might have executed beyond that point already – and might have finished executing – before you pressed **Ctrl/Y**.

Note that several **SET HOST** qualifiers, such as **/MOP**, **/VTP**, and **/X29**, are available only if DECnet-Plus is installed on your system. For information about using these qualifiers, type the following command:

```
$ HELP DECnet-Plus
```

Qualifiers

/APPLICATION_PROTOCOL[= *protocol*]

Specifies the protocol connection, CTERM (CTDRIVER) or RTERM (RTTDRIVER) on the specified node. CTERM protocol is the default if no connection is specified.

/BUFFER_SIZE=*n*

Changes the packet size of the protocol message sent between the terminal and the remote processor if a connection to the remote processor is already established. The default buffer size is 1010 bytes; however, the buffer size *n* can range from 140 bytes to 1024 bytes. The value of the parameter *n* is reset to 140 bytes if a value below 140 is specified; a value for *n* above 1024 bytes is reset to 1024.

You can force the host node to write to the terminal in smaller packets, thereby ensuring that write operations to the terminal are displayed at more frequent intervals, by setting *n* to a value just above the minimum of 140 bytes. On slow DECnet links, setting the buffer size to a smaller value may decrease pauses between write operations when large amounts of data are being scrolled to the screen.

/LOG[=*filespec*]
/NOLOG (default)

Controls whether a log file of the entire session is kept. If you use the **/LOG** qualifier without the file specification, the log information is stored in the file SETHOST.LOG.

/RESTORE
/NORESTORE (default)

Saves current terminal characteristics before a remote terminal session is begun and restores them when the remote session is terminated.

Examples

1.
```
$ SET HOST/APPLICATION_PROTOCOL=CTERM DAKOTA
```

The **/APPLICATION_PROTOCOL** qualifier specifies the CTERM protocol (the default) on node DAKOTA in this command line.

2.
```
$ SET HOST ITALIC
Username: GRESO
Password: <PASSWORD>
    Welcome to OpenVMS VAX Version 7.3 on node ITALIC
  .
  .
  .
$ LOGOUT
GRESO logged out at 19-DEC-2001 15:04:25.27
%REM-S-END, Control returned to node _CASLON::
```

In this example, the name of the local node is CASLON. This **SET HOST** command connects the user terminal to the processor at the network node named ITALIC. The remote processor

then prompts for user name and password. Use the normal login procedure to log in to the remote processor.

Once you are logged in at a remote node, you can use the **SET HOST** command to establish communication with another node. After logging in to node ITALIC, you could type **SET HOST BODONI**.

You would again be prompted for a user name and password. If you then supply a valid user name and password, you will be logged in to node BODONI.

Note that when you log out of node BODONI, control is returned to node ITALIC. You must log out of node ITALIC to return to your local node, CASLON.

3.
```
$  SET TERMINAL/WIDTH=80
$  SET HOST/RESTORE GENEVA
Username: Jones
Password: <PASSWORD>
$  SET TERMINAL/WIDTH=132
   .
   .
   .
$  logout
JONES   logged out at 19-DEC-2001 11:04:51.45

%REM-S-END, control returned to node _ORACLE
```

This example shows user JONES on node ORACLE logging in to remote node GENEVA and specifies that the original terminal screen width be restored to 80 characters when the remote session is terminated.

SET HOST/DTE

SET HOST/DTE — Connects your system to a remote system through an outgoing terminal line. **You must have an account on the remote system in order to log into that system after the connection is made. You must also have the ability to assign a channel to the terminal port specified. Your system manager can set the device protection on the terminal port to allow you access.**

Format

SET HOST/DTE *terminal-name*

Parameter

terminal-name

Specifies the name of an outgoing terminal line, which connects your system directly to another system or modem.

Description

The **SET HOST/DTE** command allows you to connect your terminal to another system. Once the connection is made, the remote system prompts for a user name and password. You must have an account on the remote system to log in.

Once you have logged in to the remote system, you can use DCL commands just as you would on your local system.

To log in on lines that expect a break rather than a carriage return, press **Ctrl/***break-character* to generate the break.

While connected to the remote system, you can issue configuration commands from DTE command mode. DTE commands enable you to configure a **SET HOST/DTE** session while it is in progress.

To issue DTE commands, you must get into DTE command mode by pressing **Ctrl/***command-character* By default, the command character is the at sign (@). Once you are in command mode, the DTEPAD prompt will be displayed on your screen.

Table 5, "SET HOST/DTE Session Commands" describes the DTE commands.

Table 5. SET HOST/DTE Session Commands

Command	Description
CLEAR	Disconnects your local system from DTEPAD. This command is equivalent to the **QUIT** command.
EXIT	Returns the session to emulation mode.
QUIT	Disconnects your local system from DTEPAD. This command is equivalent to the **CLEAR** command.
SAVE *filespec*	Saves information on the current configuration settings. The current settings are written to the specified file as a series of **SET DTE** commands. You can later use this file as an initialization file.
SEND BREAK	Sends a break to remote systems that expect a break rather than a carriage return.
SET DTE	Modifies configurable characteristics of DTEPAD while in a **SET HOST/DTE** session. You can use the **SET HOST/DTE** qualifiers in command mode.
SHOW DTE	Displays the current settings of all the configurable characteristics of DTEPAD. It also displays the number of bytes of data transmitted and received during the current **SET HOST/DTE** session and the number of errors detected by the terminal port.
SPAWN	Creates a subprocess of your current *local* process. The command is identical to the DCL **SPAWN** command except that the DTE **SPAWN** command has fewer qualifiers, as follows: **/INPUT=*filespec*** **/LOGICAL_NAMES** and **/NOLOGICAL_NAMES** **/OUTPUT=*filespec*** **/PROCESS=*subprocess-name*** **/SYMBOLS** and **/NOSYMBOLS** **/WAIT** and **/NOWAIT** See the **SPAWN** command for more information about these qualifiers.

To terminate a session, you should first log out of the remote system. Then quit DTEPAD by pressing **Ctrl/***escape-character*. By default, the escape character is a backslash (\). Alternatively, quit DTEPAD by entering the **QUIT** or the **CLEAR** command from DTE command mode.

Qualifiers

/BREAK=*break-character*

Selects the break character. The break character is used to generate a break on lines that expect a break rather than a carriage return. To generate a break, press **Ctrl/***break-character*.

The break character can be any ASCII character between @ and z, except C, M, Q, S, Y, or the left bracket ([). You cannot select a character currently defined as either the command character (see the description of the **/COMMAND=*command-character*** qualifier) or the escape character (see the description of the **/ESCAPE=*escape-character*** qualifier).

The ASCII characters between @ and z include the alphabetic characters in both upper and lower cases, the at sign (@), the backslash (\), the right bracket (]), the circumflex (^), the underscore (_), and the grave accent ( ).

By default, the break character is the right bracket (]).

/COMMAND=*command-character*

Selects the command character. Use the command character to access DTE command mode by pressing **Ctrl/***command-character*.

The command character can be any ASCII character between@ and z, except C, M, Q, S, Y, or the left bracket ([). You cannot select a character currently defined as either the break character (see the description of the **/BREAK=*break-character*** qualifier) or the escape character (see the description of the **/ESCAPE=*escape-character*** qualifier).

The ASCII characters between @ and z include the alphabetic characters in both upper and lower cases, the at sign (@), the backslash (\), the right bracket (]), the circumflex (^), the underscore (_), and the grave accent ( ).

By default, the command character is the at sign (@).

/DIAL=(NUMBER:*number* [,MODEM_TYPE:*modem-type*])

Allows a modem attached to the outgoing terminal line to be autodialed using the autodial protocol of that modem. The NUMBER keyword is the telephone number to be autodialed and is a required parameter.

Before you dial a new number, you must log out of the current remote system.

On Alpha, the MODEM_TYPE keyword defaults to DMCL (any modem that uses the DEC Modem Command Language).

Each modem type requires a specific modem dialer code. Check with your system manager to see which modem dialer codes are installed on your system.

In addition, the MODEM_TYPE keyword can be used to specify a modem type other than DF03, DF112, or DMCL. A template is provided for users interested in supporting other modems with autodial capabilities (see SYS$EXAMPLES:DTE_DF03.MAR).

/ECHO
/NOECHO (default)

Determines whether the terminal input is echoed by your local system. By default, all echoing is performed by the remote system.

/EIGHT_BIT (default)
/NOEIGHT_BIT

Determines whether the outgoing terminal line supports 8-bit or 7-bit characters. By default, 8-bit characters are supported. If you specify **/NOEIGHT_BIT**, then 7-bit characters are supported.

/ERROR_ACTION=CONTINUE (default)
/ERROR_ACTION=EXIT

Specifies the error action by using the EXIT or the CONTINUE keyword.

When an error is detected on the outgoing terminal line, the error is reported to the local system and an error message is displayed on your terminal.

If the error action is CONTINUE, then communication with the remote system continues uninterrupted.

If the error action is EXIT, then the local system immediately exits from the remote system.

/ESCAPE=*escape-character*

Selects the escape character. You can use the escape character to exit from DTEPAD by pressing **Ctrl/***escape-character*.

The escape character can be any ASCII character between @ and z, except C, M, Q, S, Y, or the left bracket ([). You cannot select a character currently defined as either the break character (see the description of the **/BREAK=*break-character*** qualifier) or the command character (see the description of the **/COMMAND=*command-character*** qualifier).

By default, the escape character is a backslash (\).

The ASCII characters between @ and z include the alphabetic characters in both upper and lower cases, the at sign (@), the backslash (\), the right bracket (]), the circumflex (^), the underscore (_), and the grave accent ( ).

/FLOW_CONTROL (default)
/NOFLOW_CONTROL

Determines whether XON/XOFF flow control is enabled. By default, flow control is enabled.

XON/XOFF flow control is a means of preventing data-overrun errors. Incoming data is stored in receive buffers; when these buffers are full, a signal is sent to the remote system to stop transmission. Once there is sufficient space in the receive buffers, another message is sent to restart transmission.

You should disable XON/XOFF flow control when the remote system has no XON/XOFF flow control.

/INIT[=*filespec*] (default)
/NOINIT

Sets the configurable characteristics of DTEPAD according to values contained in the specified initialization file.

If you use qualifiers in the command line to define the values of any of the configurable characteristics, these will supersede the values contained in the initialization file.

By default, DTEPAD tries to translate the logical name DTEPAD$INI in order to find the appropriate initialization file. If you use the **/INIT** qualifier and omit the file specification, DTEPAD translates the logical name DTEPAD$INI and finds the appropriate file. If DTEPAD$INI is not defined, then DTEPAD uses **/NOINIT** as the default.

An initialization file can contain any combination of any of the following DTE commands:

- **SAVE**
- **SEND BREAK**
- **SET DTE**
- **SPAWN**

The following is an example of an initialization file:

```
SET DTE/MAX_BUFFERS=150SET DTE/READ_DELAY=100SEND BREAK
```

/LOG[=*filespec*]
/NOLOG (default)

Controls whether a log file of the session is kept.

If you use the **/LOG** qualifier and omit the file specification, then the log information is written to the file DTEPAD.LOG.

When used to log a modem session, the log file contains any noise that occurred on the phone line. For example, typing a file in order to get it recorded in the log file could result in noise being recorded along with the file data. Therefore, the use of a log file is not recommended for the purpose of file transfers.

VSI recommends that you use asynchronous DECnet to transfer files.

/MAX_BUFFERS=*number-buffers*

Specifies the maximum number of receive buffers. Receive buffers are buffers used to receive incoming data from the modem port. They are allocated as they are required.

By default, the maximum number of receive buffers is 100. The minimum number you can specify is 20.

/PARITY=NONE (default)
/PARITY=ODD
/PARITY=EVEN

Selects parity on the outgoing terminal line.

/READ_DELAY=*delay*

Specifies the modem port read delay in milliseconds. The modem port read delay is the time interval during which data in the modem port is transferred into receive buffers at the terminal.

By default, the modem port read delay is 50 milliseconds. This is also the minimum value.

A long modem port read delay slows the rate at which data is displayed at your terminal, and also increases the risk of data-overrun errors; however, a longer read delay requires less CPU overhead.

/SPEED=(*output-rate*,*input-rate*)

Sets the baud rate at which the terminal receives and transmits data. If the input and output rates are the same, specify the qualifier as **/SPEED=*rate***.

Not all terminals support different input and output baud rates. For specific information on baud rates for your terminal, consult the manual for that terminal.

The default transmission rates are installation dependent.

The valid baud rates are as follows:

```
 50       150       1800       4800       38400
 75       300       2000       7200       57600
110       600       2400       9600       76800
134      1200       3600      19200      115200
```

If you select an invalid or unsupported speed, then the terminal line speed will remain set at its previous value.

Examples

1.
```
$ SET HOST/DTE TTA2:/DIAL=NUMBER:5551234
Username: SMITH
Password:
```

   This command connects the user terminal to the outgoing terminal line `TTA2:`, which is attached to a modem (type DF03 by default) set to autodial the phone number `555-1234`. The remote processor then prompts for user name and password. Use the normal login procedure to log in to the remote system.

2.
```
$ SET HOST/DTE/DIAL=(NUMBER:5551234#,MODEM_TYPE:DF112) TTA2:
Username: SMITH
Password:
```

   This command accomplishes the same thing as in the first example, except that it uses the `DF112` modem. The number sign (`#`) is required to activate the autodialer in the `DF112`.

3.
```
$ SET HOST/DTE/NOEIGHTBIT TTA2
```

   This command sets the outgoing terminal port, `TTA2`, to support 7-bit characters.

4.
```
$ SET HOST/DTE/NOFLOW_CONTROL TTA2
```

   This command disables the XON/XOFF flow control.

5.
```
$ SET HOST/DTE/ERROR_ACTION=EXIT TTA2
```

   This command sets the error action to **EXIT**.

6.
```
$ SET HOST/DTE/MAX_BUFFERS=150 TTA2
```

   This command sets the maximum number of receive buffers to `150`.

7.
```
$ SET HOST/DTE/ESCAPE=E TTA2
```

   This command defines the letter `E` as the escape character. Note that DTEPAD is not case sensitive.

8. `$ SET HOST/DTE/LOG TTA2`

   This command enables logging of the session. Since no log file is specified, the log is written to the default log file, DTEPAD.LOG.

9. `$ SET HOST/DTE/INIT=MYFILE.INI TTA2`

   This command uses the initialization file `MYFILE.INI`.

10. `$` **`Ctrl/@`**
    `DTEPAD>`

    This example shows you how to access DTE command mode by pressing **Ctrl/@**, where `@` is the command character.

11. `DTEPAD> SAVE MYFILE.INI`

    This command saves the information on the current configuration in the file `MYFILE.INI`.

12. `DTEPAD> SET DTE/BREAK=G`

    This command defines the letter `G` as the break character.

13. `DTEPAD> SET DTE/ECHO`

    This command sets the terminal port so that echoing is performed by the local system.

14. `DTEPAD> SET DTE/SPEED=4800`

    This command sets the input and output baud rates to `4800`.

15. `DTEPAD> SET DTE/LOG=myfile.log`

    This command directs logging to the file `MYFILE.LOG`.

16. ```
    DTEPAD> SHOW DTE
      Port TXA0              EIGHT_BIT ECHO
      Flow control           XON/XOFF
      Parity                 NONE
      Transmit Speed         4800
      Receive Speed          4800
      Error action           CONTINUE
      Break character        B
      Escape character       \
      Command character      @
      Maximum buffers        100
      Read delay             50 milliseconds
      Log file               MYFILE.INI
      Number dialed          12345678
      Modem type             DF03

      Bytes transmitted      75
      Bytes received         132
      Errors                 0
    ```

    This command displays all the current settings of the configurable characteristics, the number of bytes transmitted and received, and the number of errors detected.
    ```

SET HOST/DUP

SET HOST/DUP — Connects your terminal to a storage controller through the appropriate bus for that controller. The **/SERVER** and **/TASK** qualifiers are required. **For use only with storage controllers. Requires the DIAGNOSE privilege.**

Format

SET HOST/DUP/SERVER=*server-name*

/TASK=*task-name node-name*

Parameter

node-name

Specifies the node name of the storage controller.

Description

The **SET HOST/DUP** command creates a virtual terminal connection and executes a utility or diagnostic program on a storage controller that uses the Diagnostic and Utilities Protocol (DUP) Standard Dialogue.

Once the connection is established, operations are performed under the control of the utility or diagnostic program.

When the utility or diagnostic program terminates, control is returned to the local system.

To abort or prematurely terminate the connection and return control to the local system, press Ctrl/\.

Further information regarding the operation of the utility and diagnostic programs available on a particular controller may be obtained from the appropriate documentation set for that controller.

Preparing for Use

To use the **SET HOST/DUP** facility, you must first install FYDRIVER, the DUP class driver. To load FYDRIVER, add the following commands to the SYSTARTUP.COM command procedure in the SYS$MANAGER directory.

On Alpha and IA-64, specify the following commands:

```
$ RUN SYS$SYSTEM:SYSMAN
 SYSMAN>  IO CONNECT FYA0/NOADAPTER/DRIVER=SYS$FYDRIVER
```

This operation requires CMKRNL (change mode to kernel) privilege.

Qualifiers

/LOG[=*filespec*]
/NOLOG (default)

Controls whether a log file of the entire session is kept. If you use the **/LOG** qualifier without the file specification, the log information is stored in the file HSCPAD.LOG.

/SERVER=*server-name*

Specifies the server name for the target storage controller.

This qualifier is required.

/TASK=*task-name*

Specifies the utility or diagnostic name to be executed on the target storage controller under direction of the server.

This qualifier is required.

Example

```
$ SET HOST/DUP/SERVER=MSCP$DUP/TASK=DIRECT R2DH5Y
%HSCPAD-I-LOCPROGEXE, Local program executing - type ^\ to exit utility
```

The **SET HOST/DUP** command in this example connects the user terminal to the utility program called `DIRECT` executing on a storage controller named `R2DH5Y` under direction of the `MSCP$DUP` server.

SET HOST/HSC

SET HOST/HSC — Connects your terminal to an HSC disk and tape controller through the computer interconnect (CI) bus. **Used only with HSC controllers. Requires the DIAGNOSE privilege.**

Format

SET HOST/HSC *node-name*

Parameter

node-name

Specifies the node name of the HSC.

Description

The **SET HOST/HSC** command establishes a connection to an HSC disk and tape controller by way of the CI bus. (The **SHOW CLUSTER** command lists the names of HSC controllers that are accessible to your node.) No password is required to access the HSC; however, only **SHOW** commands are accepted when the HSC Secure/Enable switch is in the Secure position.

Once the connection is made to the HSC, operations can be performed as if you were attached to the local terminal of the HSC; however, access to ODT (Octal Debugging Tool) and offline diagnostics are not permitted.

Press **Ctrl/C** to obtain a prompt from the HSC before entering commands. To exit from the HSC and return to the local system, press **Ctrl/**.

A description of HSC commands and utilities can be obtained from the HSC documentation.

Preparing for Use

To use the **SET HOST/HSC** facility, you must first install FYDRIVER, which is the Diagnostic and Utilities Protocol (DUP) driver associated with the CI. To load FYDRIVER, add the following commands to the SYSTARTUP.COM command procedure in the SYS$MANAGER directory:

```
$ RUN SYS$SYSTEM:SYSGENCONNECT FYA0/NOADAPTER
```

This operation requires CMKRNL (change mode to kernel) privilege.

Qualifier

/LOG[=*filespec*]
/NOLOG (default)

Controls whether a log file of the entire session is kept. If you use the **/LOG** qualifier without the file specification, the log information is stored in the file HSCPAD.LOG.

Example

```
$ SET HOST/HSC HSC001
%HSCPAD-I-LOCPROGEXE, Local program executing - type ^\ to exit, ^Y for
 prompt
HSC>
```

This **SET HOST/HSC** command connects the user terminal to the HSC named HSC001.

SET HOST/LAT

SET HOST/LAT — Connects your terminal to a specified service available on the local area network (LAN), establishing one session for communication between your terminal and that service. **The service node that provides the service must be on the same extended LAN and must be running at least Version 5.0 of the LAT protocol.**

Format

SET HOST/LAT *service-name*

Parameter

service-name

Specifies the name of the service to which you want your terminal connected. A service is a resource on the LAN. A service often consists of all of the resources of a computer system. Other examples of services are a file storage system and an application program running on a computer system. A computer system that offers one or more services is called a service node.

If several service nodes offer the same service, and you do not specify the **/NODE=*node-name*** qualifier, your terminal connects to the service node that is the least busy.

To display a list of services on your LAN, use the LAT Control Program (LATCP) **SHOW SERVICES** command. See the relevant section in the *VSI OpenVMS System Management Utilities Reference Manual, Volume I: A–L* [https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#LATCP_PART].

Description

The **SET HOST/LAT** command allows you to connect your terminal to a specified service, establishing one LAT session for communication between your terminal and that service. (For an overview of the LAT protocol and its advantages, see the relevant section in the *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/#LAT_PROTOCOL_SEC].)

The service node that provides the service must be on the same extended LAN and must be running at least Version 5.0 of the LAT protocol.

Some services are password protected. You are prompted for a password unless you specify the password with the **/PASSWORD** qualifier.

Once the connection to the service is made, you can interact with the service as if your terminal is directly connected to it. Some services will prompt you. For example, if the service is an OpenVMS system, it prompts you for a user name and password. You must have an account on the service node to log in.

Press the disconnect character to end the LAT session and return to DCL command level on your local system. With some services, such as general time sharing services like OpenVMS, you can end the LAT session by logging out of the service. The default disconnect character is **Ctrl/**. Use the **/DISCONNECT** qualifier to change the default disconnect character.

Qualifiers

/AUTOCONNECT
/NOAUTOCONNECT

Specifies whether connection attempts should be retried automatically when a connection fails because a service is unknown or unavailable, or because a node is unknown or unreachable. Also specifies that reconnects should be attempted automatically if a service has disconnected abnormally. The default is **/NOAUTOCONNECT**.

/AUTOPROMPT (default)
/NOAUTOPROMPT

Causes an OpenVMS Username: prompt to appear with no user action when a **SET HOST/LAT** command is issued.

On a terminal server port, you can configure the port to have AUTOPROMPT disabled (**/NOAUTOPROMPT**) so that you are required to press Return when connecting to a node to get the Username: prompt; however, when connecting to a reverse LAT service, the AUTOPROMPT characteristic should be disabled.

/BREAK=*break-character*

Defines a character that generates a break on lines that expect a break rather than a carriage return. To generate a break, press the **Ctrl** and *break-character* keys together. The default *break-character* value is the tilde (~).

You can select any ASCII character between @ and Z, except C, M, Q, S, Y, and the left bracket ([). You cannot select a character that is already defined as the disconnect character.

/DESTINATION_PORT=*port-name*

Specifies the port on a node to which you want to connect. The **/NODE** qualifier is required when you specify the **/DESTINATION_PORT** qualifier. The port must be available and must offer the service you specify. OpenVMS and certain other LAT service node systems ignore the **/DESTINATION_PORT** qualifier.

/DIAL=(NUMBER:*number* [,MODEM_TYPE:*modem-type*])

Allows a modem attached to the outgoing terminal line to be autodialed using the autodial protocol of that modem. The NUMBER keyword is the telephone number to be autodialed and is a required parameter.

The MODEM_TYPE keyword is optional. It can be used to specify any of the following modem types:

- DMCL (any modem that uses the DEC Modem Command Language)
- DF03 (default)
- DF112

Each modem type requires a specific modem dialer code. Check with your system manager to see which modem dialer codes are installed on your system.

In addition, the MODEM_TYPE keyword can be used to specify a modem type other than DF03, DF112, or DMCL. A template is provided for users interested in supporting other modems with autodial capabilities (see SYS$EXAMPLES:DTE_DF03.MAR).

/DISCONNECT=*disconnect-character*

Defines the character that you can use to disconnect from a remote session. To generate a disconnect, press the **Ctrl** and *disconnect-character* keys together. The default *disconnect-character* is the backslash (\).

You can select any ASCII character from @ through Z, except C, M, Q, S, Y, and the left bracket ([). For example, if you specify **/DISCONNECT=A**, **Ctrl/A** will be the disconnect character. You cannot select a character that is already defined as the break character.

/EIGHT_BIT (default)
/NOEIGHT_BIT

Determines whether the outgoing terminal line supports 8-bit or 7-bit characters. By default, 8-bit characters are supported. If you specify **/NOEIGHT_BIT**, then 7-bit characters are supported.

Note

To change the number of bits per character on the remote terminal server port, that port must have the REMOTE MODIFICATION characteristic enabled.

/FRAME=*n*

The **/FRAME=*n*** qualifier allows a user making a LAT connection to a remote system to specify the number of data bits that the terminal driver expects for every character that is input or output. The value of *n* can be from 5 to 8. The default value depends on the settings for the terminal established

by the **/PARITY** and **/EIGHT_BIT** qualifiers. The following example specifies a character frame size of 7 bits per character:

```
$ SET HOST/LAT /FRAME=7 DIAL_OUT_SVC
```

/LOG[=*log-file*]

Logs all data that is delivered during the LAT session. If you do not specify a log file, the data is stored in the file SETHOST_LAT.LOG.

/NODE=*node-name*

Specifies the node that offers the service to which you want to connect. Failover is not performed if the connection fails.

/PASSWORD=*password*

Specifies the password required by a service that is password protected. If you do not specify the **/PASSWORD** qualifier when requesting a connection to a password-protected service, you are prompted for a password.

/QUEUE
/NOQUEUE (default)

When connecting to a reverse LAT service that is already in use (such as a dial out modem), you are notified that the service is in use and the **SET HOST/LAT** command terminates. However, LAT can allow incoming connections to be queued to any reverse LAT service that supports service queueing, which enables users to be placed on a queue when using reverse LAT services.

LAT reports your position in the queue. When the resource becomes available, you are immediately notified. You can cancel the queued connection by pressing **Ctrl/Y** and terminating the **SET HOST/LAT** command.

/SPEED=(*output-rate*,*input-rate*)

Sets the baud rate at which the terminal receives and transmits data. If the input and output rates are the same, specify the qualifier as **/SPEED=*rate***.

Not all terminals support different input and output baud rates. For specific information on baud rates for your terminal, consult the manual for that terminal.

The default transmission rates are installation dependent.

The valid values for input and output baud rates are as follows:

```
300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600
```

If you select an invalid or unsupported speed, the terminal line speed will remain set at its previous value.

Note

To change the speed on the terminal server port, the REMOTE MODIFICATION characteristic must be enabled on that terminal server port.

When initiating a connection, **SET HOST/LAT** may attempt to set terminal server port characteristics, such as the speed and the default speed for a LAT device. If the device connected to the terminal server port has a fixed speed (such as a dial-out modem) and you do not want the host

to try to change this, disable the REMOTE MODIFICATION characteristic on that port using the following command:

```
Local> DEFINE PORT x REMOTE MODIFICATION DISABLE
Local> LOGOUT PORT x
```

Use the following command for terminal servers that support the **CHANGE** command:

```
Local> CHANGE PORT x REMOTE MODIFICATION DISABLE
```

Examples

1.
```
$ SET HOST/LAT SORTER
%LAT-S-CONNECTED, session to SORTER established
%LAT-I-TODISCON, type ^\ to disconnect the session
Username: BARKER
Password:
  .
  .
  .
$ LOGOUT
BARKER logged out at 30-DEC-2001  11:04:51.45
%LAT-I-DISCONNECTED, session disconnected from SORTER
-LAT-I-END, control returned to node HOME
$
```

   This **SET HOST/LAT** command connects the user to the service SORTER, which is a computer system. The first message confirms that the user has been connected to that service. The second message informs the user how to disconnect the session. (The user can also disconnect the session by logging out from SORTER.) SORTER then prompts for the username and password. Use the normal login procedure to log in to the system. When the user logs out of the service SORTER, the terminal displays the DCL command prompt of the user's local processor system (HOME).

2.
```
$ SET HOST/LAT/DESTINATION_PORT=BOSTON -
_$ /NODE=STATE/DISCONNECT=F BUDGET
```

   This command connects the user's terminal to the service BUDGET that is offered on port BOSTON, on service node STATE. The user can disconnect the session by pressing **Ctrl/F**.

3.
```
$ SET HOST/LAT PURSE
Password:
```

   This command attempts to connect the user's terminal to the service PURSE. The service PURSE is password protected, so the user is prompted for a password. The user could have specified the password within the **SET HOST/LAT** command, as shown in the next example.

4.
```
$ SET HOST/LAT/PASSWORD=BEOR PURSE
```

   This command connects the user's terminal to the password-protected service PURSE. The password is BEOR.

SET HOST/RLOGIN

SET HOST/RLOGIN — Allows you to log in to a remote host over a TCP/IP connection and start an interactive terminal session by accessing the RLOGIN application.

Format

SET HOST/RLOGIN *IPhostname* | *IPaddress*

Note

You can specify the *IPhostname* or the *IPaddress*, but not both.

Parameter

IPhostname

Specifies the IP host name of the remote host.

IPaddress

Specifies an IP address of the remote host.

Description

The **SET HOST/RLOGIN** command allows you to log in to a remote host by invoking RLOGIN client software that runs on your system. Once you start the terminal session, you can enter commands interactively on the remote host. The parameter that specifies the remote host is either an IP host name or an IP address.

Qualifiers

/AUTHENTICATE

Specifies that Kerberos authentication should be used for acquiring access to the remote node.

/TERMINAL_TYPE=*type*

Sets the terminal emulator to one of the following terminal types: VT100, VT200, VT300, VT400, VT500.

/TRUNCATE_USERNAME

Specifies that the current user name should be truncated to 8 characters before attempting to connect to the remote node. The qualifier is required for communication with systems that limit the size of their login names to 8 characters. The **/TRUNCATE_USERNAME** qualifier is ignored if **/USERNAME** is specified.

/USERNAME=*username*

Specifies the user name for logging in to the remote node. The user name can been closed in quotes to preserve the case of the user name for case sensitive systems such as UNIX systems. If the **/USERNAME** qualifier is not specified, the default is the current user's user name.

Example

```
$ SET HOST/RLOGIN remotehst1
```

This example creates an `RLOGIN` connection to remote host `remotehst1` over a TCP/IP connection.

SET HOST/TELNET

SET HOST/TELNET — Connects you to a remote host over a TCP/IP connection by invoking the TELNET application.

Format

SET HOST/TELNET *IPhostname* | *IPaddress*

Note

You can specify the *IPhostname* or the *IPaddress*, but not both.

Parameters

IPhostname

Specifies the IP host name of the remote host.

IPaddress

Specifies an IP address of the remote host.

Description

The **SET HOST/TELNET** command allows you to connect to a server on a remote system by invoking the TELNET client software that runs on your system. The parameter that specifies the remote host is either an IP host name or an IP address.

Most of the attributes are negotiated with the remote node. Except for the **/AUTHENTICATE** qualifier, the qualifiers are used only in exception cases (for example, cases where a remote server does not support a negotiated parameter but requires a certain characteristic for the connection).

Qualifiers

/AUTHENTICATE

Specifies that Kerberos authentication should be used for acquiring access to the remote node.

/PORT=*port*

Specifies the remote TCP port to use. The default is 23.

/TERMINAL_TYPE=*type*

Sets the terminal emulator to one of the following terminal types: VT100, VT200, VT300, VT400, VT500.

Example

```
$ SET HOST/TELNET remotehst2
```

This example creates a `TELNET` connection to remote host `remotehst2` over a TCP/IP connection.

SET HOST/TN3270

SET HOST/TN3270 — Connects you to a remote IBM host over a TCP/IP connection, causing the local keyboard to emulate an IBM 3279-class terminal keyboard by invoking the TN3270 terminal emulator.

Format

SET HOST/TN3270 ***IPhostname | IPaddress***

Note

You can specify the *IPhostname* or the *IPaddress*, but not both.

Parameters

IPhostname

Specifies the IP host name of the remote host.

IPaddress

Specifies an IP address of the remote host.

Description

The **SET HOST/TN3270** command allows you to connect with a TELNET server on a remote IBM system by invoking the TN3270 terminal emulator TELNET client program on the local host. The parameter that specifies the remote host is either an IP host name or an IP address. The TN3270 client will automatically determine the terminal type (IBM-3278-2, IBM-3278-3, IBM-3278-4, or IBM-32798-5).

Qualifiers

/AUTHENTICATE

Specifies that Kerberos authentication should be used for acquiring access to the remote node.

/PORT=*port*

Specifies the remote TCP port to use. The default is 23.

/TERMINAL_TYPE

Sets the terminal emulator to one of the following terminal types: VT100, VT200, VT300, VT400, VT500.

Example

```
$ SET HOST/TN3270 remotehst3
```

This example creates a connection to a TELNET server on the remote IBM system `remotehst3` over a TCP/IP connection.

SET IMAGE

SET IMAGE — Modifies the current attributes for the specified image file. This command can be used to modify the initial values set by the OpenVMS IA-64 Linker for certain image attributes.

Format

SET IMAGE *image-filespec*

Parameters

image-filespec

Specifies the name of an OpenVMS IA-64 image file to modify. The command overlays the current image file; it does not create a new version of the image file.

The asterisk (*) and percent sign (%) wildcard characters are allowed in the file specification if the **/RESTORE** qualifier is used. If you omit the file type extension, the default file type .EXE is applied.

Description

Note

This command can be executed on an Alpha or IA-64 system, but it operates only on IA-64 images.

Qualifier

/FLAGS=(*keyword*[,...])

Specifies which image attribute flags to turn on or off. The original flags are set by the OpenVMS IA-64 Linker at image link time. The possible keywords are listed below with a brief description.

Important

Beware of modifying the flag values unless you are very knowledgeable about the internals of the image.

Keyword	Description
[NO]CALL_DEBUG	Call Debugger at startup.
[NO]DBG_IN_DSF	Debug records in debug symbol file.
[NO]DBG_IN_IMG	Debug records in image file.
[NO]EXE_INIT	Image has a pointer to EXE$INITIALIZE.
[NO]IMGSTA	Call SYS$IMGSTA.
[NO]INITIALIZE	Image has a pointer to LIB$INITIALIZE.
[NO]MAIN	Image has a main transfer address.

Keyword	Description
[NO]MKTHREADS	Enable multiple kernel thread use.
[NO]NOP0BUFS	No P0 buffers for RMS image I/O.
[NO]P0IMAGE	Image is loaded only to P0 space.
[NO]SIGNATURES	TIE Signatures are present.
[NO]TBK_IN_DSF	Traceback records in debug symbol file.
[NO]TBK_IN_IMG	Traceback records in image file.
[NO]UPCALLS	User thread upcalls are enabled.

/LOG

Displays a message to the output device indicating whether the command successfully modified the image.

/RESTORE

Restores the original image attributes of the file – that is, the initial attributes that were set by the OpenVMS IA-64 Linker. You cannot specify the **/FLAGS** or **/SUCCESS** qualifier when **/RESTORE** is used.

/SUCCESS

Modifies the image link completion code value to SUCCESS. The run-time behavior of the image remains the same as the original link.

Examples

1.
```
$ SHOW IMAGE WARNING
Show Image Version 1.2                    29-SEP-2004 10:15:52.61
WORK1:[SWEENEY.CLIUTL]WARNING.EXE;1
This is an OpenVMS IA64 (Elf format) executable image file
Image Identification Information
                  Image name: WARNING
    Global Symbol Table name: WARNING
   Image file identification: X-6
  Image build identification: <unavailable>
         Link identification: Linker T02-14
              Link Date/Time: 12-JUN-2004 10:15:16.46
             Patch Date/Time: Never
       Manipulation Date/Time: Never
  Image Dynamic Data Version: 1.2
Image Completion Code Value: WARNING  State: Original
Image Link Flags state: Original
                Image Link Flags
     ----------------------------------------
     CALL_DEBUG      : Call debugger
     IMGSTA          : Call SYS$IMGSTA
     MAIN            : Image has main transfer
     TBK_IN_IMG      : Traceback records in image file
     DBG_IN_IMG      : Debug records in image file
$!
$ SET IMAGE/SUCCESS WARNING.EXE
```

```
$ SHOW IMAGE WARNING.EXE
Show Image Version 1.2                        29-SEP-2004 11:41:54.93
WORK1:[SWEENEY.CLIUTL]WARNING.EXE;1
This is an OpenVMS IA64 (Elf format) executable image file
Image Identification Information
                  Image name: WARNING
    Global Symbol Table name: WARNING
   Image file identification: X-6
  Image build identification: <unavailable>
         Link identification: Linker T02-14
              Link Date/Time: 12-JUN-2004 10:02:53.51
             Patch Date/Time: Never
       Manipulation Date/Time: 29-SEP-2004 11:26:31.57
  Image Dynamic Data Version: 1.2
Image Completion Code Value: SUCCESS  State: Manipulated
Image Link Flags state: Original
               Image Link Flags
     ------------------------------------------
     CALL_DEBUG      : Call debugger
     IMGSTA          : Call SYS$IMGSTA
     MAIN            : Image has main transfer
     TBK_IN_IMG      : Traceback records in image file
     DBG_IN_IMG      : Debug records in image file
```

In this example the **SET IMAGE/SUCCESS** command is used to change the link completion code value in the image to SUCCESS. The output of the second **SHOW IMAGE** command displays that the image completion code value has been changed.

2.
```
$ SHOW IMAGE SYS$SYSTEM:DCE$DCED.EXE
Show Image Version 1.2                        29-SEP-2004 12:40:04.32
SYS$COMMON:[SYSEXE]DCE$DCED.EXE;199
This is an OpenVMS IA64 (Elf format) executable image file
Image Identification Information
                  Image name: DCE$DCED
    Global Symbol Table name: DCE$DCED
   Image file identification: DCE T3.2-040610
  Image build identification: <unavailable>
         Link identification: Linker I01-68
              Link Date/Time: 11-JUN-2004 01:50:11.71
             Patch Date/Time: Never
       Manipulation Date/Time: Never
  Image Dynamic Data Version: 1.1
Image Completion Code Value: SUCCESS  State: Original
Image Link Flags state: Original
               Image Link Flags
     ------------------------------------------
     MAIN            : Image has main transfer
$ SET IMAGE/FLAGS=(MKTH,UPCALL) SYS$SYSTEM:DCE$DCED.EXE
$!
$ SHOW IMAGE SYS$SYSTEM:DCE$DCED.EXE
Show Image Version 1.2                        29-SEP-2004 12:43:34.90
SYS$COMMON:[SYSEXE]DCE$DCED.EXE;1
This is an OpenVMS IA64 (Elf format) executable image file
Image Identification Information
                  Image name: DCE$DCED
    Global Symbol Table name: DCE$DCED
   Image file identification: DCE T3.2-040610
  Image build identification: <unavailable>
```

```
          Link identification: Linker I01-68
               Link Date/Time: 11-JUN-2004 01:50:11.71
              Patch Date/Time: Never
          Manipulation Date/Time: 29-SEP-2004 12:43:02.14
  Image Dynamic Data Version: 1.1
Image Link Flags state: Manipulated
     Current Image Flags        Original Link Flags
     -------------------        -------------------
     MKTHREADS                  MAIN
     UPCALLS
     MAIN
     Image Link Flag   Description
     ---------------   ------------------------------------
     MKTHREADS       : Multiple kernel threads enabled
     UPCALLS         : Upcalls enabled
     MAIN            : Image has main transfer
```

The **SET IMAGE** command in this example shows how to enable multiple kernel threads and thread upcalls in the image SYS$SYSTEM:DCE$DCED.EXE.

3.
```
$ SHOW IMAGE PATCH.EXE;61
Show Image                                  30-SEP-2004 09:06:39.57
WORK1:[SWEENEY.PATCH]PATCH.EXE;61
This is an OpenVMS IA64 (Elf format) executable image file

                  Image name: PATCH
    Global Symbol Table name: PATCH
              Link Date/Time: 23-SEP-2004 10:22:07.04
      Manipulation Date/Time: Never

            Link flags state: Original

          Link Flags
     ------------------
     CALL_DEBUG
     IMGSTA
     MAIN
     TBK_IN_IMG
     DBG_IN_IMG
$ DEFINE PATCH WORK1:[SWEENEY.PATCH]PATCH.EXE;61
$ PATCH X.LIS
         OpenVMS Integrity servers Debug64 Version 8.2

%DEBUG-I-INITIAL, Language: BLISS, Module: PATMAIN

DBG>Exit
$ SET IMAGE/FLAGS=NOCALL_DEBUG PATCH.EXE
$ PATCH X.LIS


  OpenVMS PATCH Version 8.2

%PATCH-I-NOGBL, some or all global symbols not accessible
%PATCH-I-NOLCL, image does not contain local symbols
PATCH>
```

This example demonstrates how to modify an image to stop the debugger from being called when the image is initiated. The image PATCH.EXE was linked with debugging enabled. The

SET IMAGE /FLAGS=NOCALL_DEBUG command allows the debugger to be bypassed when the image is run.

SET KEY

SET KEY — Sets and locks the key definition state for keys defined with the **DEFINE/KEY** command.

Format

SET KEY

Description

When you define keypad keys using the **DEFINE/KEY** command, you can as sign a specific state name to the key definition. If that state is not set when you press the key, the definition is not processed. Use the **SET KEY** command to change the current state to the appropriate state.

Qualifiers

/LOG (default)
/NOLOG

Controls whether the system displays a message indicating that the key state has been set.

/STATE=*state-name*
/NOSTATE

Specifies the name of the state. The state name can be any alphanumeric string. If you omit the **/STATE** qualifier or use the **/NOSTATE** qualifier, the current state is left unchanged. The default state is DEFAULT.

Example

```
$ SET KEY /STATE=EDITING
```

The **SET KEY** command in this example sets the key state to the state EDITING. You can now use the key definitions that were defined for the state EDITING.

SET LOGINS

SET LOGINS — Sets the interactive limit (number of interactive users allowed on the system), or displays the interactive limit and the current number of interactive users. **Requires OPER (operator) privilege to set the login limit.**

Format

SET LOGINS

Parameters

None.

Description

The SET LOGINS command is not retroactive. All users logged in to the system before you enter the **SET LOGINS** command are not affected by the command; however, once the limit you set is reached, no more users can log in to the system until someone else logs out. Users with the OPER privilege are not affected by the limit.

If you do not specify a parameter value with the **/INTERACTIVE=*n*** qualifier, the **SET LOGINS** command displays the following information:

```
Login quotas - Interactive limit=x, Current interactive value=y
```

The value *x* represents the current interactive limit, and the value *y* represents the number of users currently logged in to the system.

Qualifier

/INTERACTIVE[=*n*]

Establishes the number of interactive users allowed to gain access to the system. If the parameter *n* is specified, the interactive limit is set to the value *n*. If the parameter *n* is not specified, the **SET LOGINS** command displays the current interactive limit and the number of interactive users.

Examples

1.
```
$ SET LOGINS/INTERACTIVE=5
%SET-I-INTSET, login interactive limit=5, current interactive value=3
```

   In this example, the **SET LOGINS** command specifies that only five interactive users can be logged in to the system.

2.
```
$ SET LOGINS/INTERACTIVE
%SET-I-INTSET, login interactive limit=9, current interactive value=6
```

   When the **SET LOGINS** command is entered without a parameter, as shown in this example, the **/INTERACTIVE** qualifier requests that the current status of the login quotas be displayed. The message returned indicates that the maximum number of interactive users allowed on the system is `9` and that the number of interactive users currently logged in is `6`. No change is made.

SET MAGTAPE

SET MAGTAPE — Defines the default characteristics associated with a specific magnetic tape device for subsequent file operations. **The SET MAGTAPE command is valid for magnetic tape devices mounted with foreign volumes.**

Format

SET MAGTAPE *device-name*[:]

Parameter

***device-name*[:]**

Specifies the name of the magnetic tape device for which the characteristics are to be set. The device must not be currently allocated to any other user.

Qualifiers

/DENSITY=*density-value*

Specifies the default density for all write operations on the magnetic tape device when the volume is mounted as a foreign tape or as an unlabeled tape.

Valid tape density values are as follows:

Keyword	**Description**
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 keywords above are understood by TMSCP/TUDRIVER code prior to OpenVMS Version 7.2. The remaining keywords in this table are supported only on Alpha and IA-64 systems.	
TK85	DLT Tx85: 10625 BPI – Cmpt III (Alpha/IA-64 Only)
TK86	DLT Tx86: 10626 BPI – Cmpt III (Alpha/IA-64 Only)
TK87	DLT Tx87: 62500 BPI – Cmpt III (Alpha/IA-64 Only)
TK88	DLT Tx88: (Quantum 4000) – Cmpt IV (Alpha/IA-64 Only)
TK89	DLT Tx89: (Quantum 7000) – Cmpt IV (Alpha/IA-64 Only)
QIC	All QIC drives are drive-settable only (Alpha/IA-64 Only)
8200	Exa-Byte 8200 (Alpha/IA-64 Only)
8500	Exa-Byte 8500 (Alpha/IA-64 Only)
DDS1	Digital Data Storage 1 – 2G (Alpha/IA-64 Only)
DDS2	Digital Data Storage 2 – 4G (Alpha/IA-64 Only)
DDS3	Digital Data Storage 3 – 8-10G (Alpha/IA-64 Only)
DDS4	Digital Data Storage 4 (Alpha/IA-64 Only)
AIT1	Sony Advanced Intelligent Tape 1 (Alpha/IA-64 Only)
AIT2	Sony Advanced Intelligent Tape 2 (Alpha/IA-64 Only)
AIT3	Sony Advanced Intelligent Tape 3 (Alpha/IA-64 Only)

Keyword	Description
AIT4	Sony Advanced Intelligent Tape 4 (Alpha/IA-64 Only)
DLT8000	DLT 8000 (Alpha/IA-64 Only)
8900	Exabyte 8900 (Alpha/IA-64 Only)
SDLT	SuperDLT1 (Alpha/IA-64 Only)
SDLT320	SuperDLT320 (Alpha/IA-64 Only)

Note that tape density keywords cannot be abbreviated.

/END_OF_FILE

Writes a tape mark at the current position on the magnetic tape volume.

/FAST_SKIP=*option*

Requires PHY_IO privilege.

Allows you to skip by file mark or by record.

Note

This tape positioning qualifier is for use on local SCSI tape drives only.

Option	Description
PER_IO (default)	Allows a local MK device to use the skip-by-filemarks function. The tape drive must be able to do a SCSI READ POSITION command and report blank check at end-of-data. The IO$M_ALLOWFAST function modifier must be supplied with IO$_SKIPFILE. Otherwise, the tape will skip files using the skip-by-records function.
ALWAYS	Allows a local MK device to use the skip-by-filemarks function. The tape drive must be able to support the skip-by-filemarks function, and no modifications should be needed to the IO$_SKIPFILE function.
NEVER	Specifies that a local MK device skip only by records. If you use a utility that depends on the semantics of skipping with skip-records, you may also need to use this option because it causes BACKUP or COPY to use the previous positioning.

/LOG
/NOLOG

Displays information about the operations performed on the magnetic tape volume.

/LOGSOFT (default)
/NOLOGSOFT

Controls whether soft errors on the specified device are to be logged in the error log file. Soft errors are errors corrected by the hardware without software intervention. This qualifier affects only devices that support hardware error correction, such as the TU78 magnetic tape drive. When used with other devices, this qualifier has no effect.

/MEDIA_FORMAT= [NO]COMPACTION

Controls whether data records are automatically compacted and blocked together on a TA90E tape drive. Data compaction and record blocking increase the amount of data that can be stored on a single tape cartridge.

Note that once data compaction or noncompaction has been selected for a given cartridge, that same status applies to the entire cartridge.

/RETENSION

Moves a TZK10 tape cartridge to the end of the tape, and then back to the beginning of the tape. Use the **/RETENSION** qualifier on a regular basis to help maintain the integrity of TZK10 tape cartridges.

The **/RETENSION** qualifier must be used in conjunction with the **/REWIND** or **/UNLOAD** qualifier. The **/RETENSION** qualifier completes its action before **/REWIND** or **/UNLOAD**. Use **/RETENSION/REWIND** when you want the tape cartridge to remain loaded in the drive. Use **/RETENSION/UNLOAD** when you want to unload the tape cartridge after the retension operation.

This qualifier only affects TZK10 tape cartridge drives and has no effect on non-SCSI drives.

/REWIND

Requests that the volume on the specified device be rewound to the beginning of the magnetic tape.

/SKIP=*option*

Requests that the magnetic tape volume be positioned according to any of the following options:

Option	Description
BLOCK: *n*	Skips the specified number of blocks.
END_OF_TAPE	Positions the volume at the end-of-tape (EOT) mark.
FILES: *n*	Skips the specified number of tape marks (not files). The tape is positioned just past the *n* the tape mark. If *n* is negative, the tape is still positioned after the *n* the tape mark, not before it. If the tape is already positioned immediately after a tape mark, a skip of -1 results in no net movement. (The tape moves to the preceding mark, but is then repositioned to the end of the mark, where it began.) For more information on tape marks and files, see the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/].
RECORD: *n*	Skips the specified number of records up to 32767.

/UNLOAD

Requests that the volume on the specified device be rewound and unloaded.

Examples

1. `$ MOUNT MTB1:/FOREIGN`

```
$ SET MAGTAPE MTB1: /DENSITY=800
```

The **MOUNT** command in this example mounts a foreign tape on the device MTB1. The **SET MAGTAPE** command defines the density for writing the magnetic tape at 800 bpi.

2.
```
$ MOUNT MTA0:/FOREIGN
$ SET MAGTAPE MTA0:/SKIP=FILES:4
```

The **MOUNT** command in this example mounts a foreign tape on the device MTA0; the **SET MAGTAPE** command directs the magnetic tape position to skip four files.

SET MESSAGE

SET MESSAGE — Sets the format for system messages or specifies a process level message file. Lets you override or supplement the system messages.

Format

SET MESSAGE [*filespec*]

Parameter

filespec

Specifies the name of the process level message file. Messages in this file supersede messages for the same conditions in the system message file or in an existing process message file. The file type defaults to .EXE. The asterisk (*) and the percent sign (%) wildcard characters are not allowed. If you do not specify this parameter, the qualifiers apply to the system message file.

Description

The **SET MESSAGE** command specifies which message fields the OpenVMS system displays. The message format is as follows:

```
%FACILITY-L-IDENT, text
```

When a process is created, the default is to display all four message fields: facility, severity, identification, and text. To control which fields of a message are displayed, specify the appropriate qualifiers. For example, to omit the FACILITY field in message displays, specify **SET MESSAGE /NOFACILITY**.

By including the **SET MESSAGE** command in your login command file, you can select specific portions of the messages for your process.

Use the **SET MESSAGE** command, also, to override or supplement system messages. Whenever any software detects an error and invokes the $GETMSG system service, the message files are searched in the following order: image message sections first, process-permanent message files second, and the system message file last. Thus, with the **SET MESSAGE** command, you can introduce messages earlier in the searching order; you can either override or supplement the system messages. (Note that the new message definitions affect only your process.)

If a process-permanent message file exists when you specify the **SET MESSAGE** command with a file specification, the old file is removed and the new file is added.

The message definitions you specify must result from a successful compilation with the **MESSAGE** command. For full details on how to create your own messages with the Message utility, see the relevant section in the *VSI OpenVMS Command Definition, Librarian, and Message Utilities Manual* [https://docs.vmssoftware.com/vsi-openvms-command-definition-librarian-and-message-utilities/#MSG_CHAP].

Qualifiers

/DELETE

Removes any process-permanent message files currently in effect. Do not specify the *filespec* parameter with the **/DELETE** qualifier.

/FACILITY (default)
/NOFACILITY

Formats messages so that the facility name prefix appears.

/IDENTIFICATION (default)
/NOIDENTIFICATION

Formats messages so that the message identification prefix appears.

/SEVERITY (default)
/NOSEVERITY

Formats messages so that the severity level appears.

/TEXT (default)
/NOTEXT

Formats messages so that the message text appears.

Examples

1. ```
   $ TYPE XXX
   %TYPE-W-OPENIN, error opening DB1:[MARSHALL]XXX.LIS; as input
   -RMS-E-FNF, file not found
     .
     .
     .
   $  SET MESSAGE/NOIDENTIFICATION
     .
     .
     .
   $ TYPE XXX
   %TYPE-W, error opening DB1:[MARSHALL]XXX.LIS; as input
   -RMS-E, file not found
   ```

   When the first **TYPE** command is entered in this example, the error messages include all fields. Later, the **SET MESSAGE** command establishes that the IDENT portion (the abbreviation for the message text) is omitted in future messages.

2. ```
   $ SET MESSAGE NEWMSG
   ```

   The **SET MESSAGE** command in this example specifies that the message text in `NEWMSG.EXE` supplements the existing system messages.

SET NETWORK

SET NETWORK — Registers the attributes of a network service. **Requires SYSNAM (system name) privilege.**

Format

SET NETWORK ***network-service***

Parameter

network-service

Specifies the name of a network service.

Description

The **SET NETWORK** command allows you to control information about network services on an OpenVMS system. This information is contained in structures pointed to by executive mode logical names SYS$NET_SERVICES_1 to SYS$NET_SERVICES_12. Each structure defines a particular network service and contains the following information:

- Product name
- Manufacturer
- Type of network
- Node name
- Address(es)
- Interface(s) (OpenVMS device driver)
- Vendor-specific data
- Status
- Connections
- Counters
- Start command
- Stop command
- Path to the Point-to-Point Protocol utility (PPPD) shareable image

The following items are required for network registration:product name, manufacturer, type of network, node name, address. For each network service that supports the Point-to-Point Protocol (PPP), the file name of the PPPD shareable image must also be specified using the **/PPPD_CALLOUT** qualifier.

The **SET NETWORK** command allows you to perform the following actions on the information on network services:

- Add a service (use the **/REGISTER** qualifier)
- Delete a service (use the **/REMOVE** qualifier)
- Modify the information for a service (use the **/UPDATE** qualifier)

The **/REGISTER** qualifier is the default, if you do not specify an action.

Qualifiers

/ADDRESS=*address*[*es*]

Specifies the network address of the local node.

/CONNECTIONS=

Specifies a command to be executed to display the number of network connections.

/COUNTERS=

Specifies a command to be executed to display the number of network counters.

/DATA=

Specifies a character string provided by the network service vendor.

/INTERFACE=

Specifies the OpenVMS driver for the network service.

/MANUFACTURER=

Specifies the manufacturer of the network service.

/NETWORK_TYPE=

Specifies the protocol for the network service.

/NODE=

Specifies the name of the local node.

/PPPD_CALLOUT=*image-name*

Specifies the name of the protected shareable image activated by PPPD. This image contains the routine PPPD$OPEN_CONNECT, which notifies the network service that a new physical transport exists that supports PPP. If the image resides in the SYS$SHARE directory, enter the file name of the image without the suffix (such as, PPPD_CALLOUT instead of PPPD_CALLOUT.EXE); otherwise, enter the system logical that identifies the location of the image.

/REGISTER

Defines a new network service.

/REMOVE

Deletes a network service from the database.

/START=

Specifies an image to be executed when you issue the **START/NETWORK** command for the network service.

/STATUS=

Specifies a command to be executed to display additional status information.

/STOP=

Specifies an image to be executed when you issue the **STOP/NETWORK** command for the network service.

/UPDATE

Allows you to modify the specified data for the network service.

Examples

1.
```
$ SET NETWORK DECnet -
_$     /MANUFACTURER= "Hewlett-Packard Company" -
_$     /NODE="GALENA" -
_$     /ADDRESS="19.129" -
_$     /NETWORK_TYPE="DNA V" -
_$     /INTERFACE="net 0" -
_$     /DATA="Router: No" -
_$     /STATUS="mcr ncl show node 0 all" -
_$     /CONNECTIONS="mcr ncl show node 0 session control port * all" -
_$     /COUNTERS="mcr ncl show node 0 session control all counters" -
_$     /START="@sys$startup:net$startup" -
_$     /STOP="@sys$manager:net$shutdown"
```

   This command adds the network service DECnet-Plus (Phase V) to the database, with the specified characteristics.

2.
```
$ SET NETWORK "TCP/IP" /REGISTER -
_$ /MANUFACTURER="Hewlett-Packard Company" -
_$ /NODE="ipv6.ucx.mars.univers.com" -
_$ /ADDRESS="16.20.207.92" -
_$ /NETWORK_TYPE="TCP/IP" -
_$ /STATUS="TCPIP SHOW SERVICE" -
_$ /CONNECTIONS="TCPIP SHOW DEVICE" -
_$ /PPPD_CALLOUT="TCPIP$PPPD_CALLOUT"
```

   This command creates a new `TCP/IP` network service, adds it to the database, and enables the `PPPD` utility by supplying a logical name that identifies the location of the shareable image.

3.
```
$ SET NETWORK DECnet -
_$                /MANUFACTURER= "Hewlett-Packard Company" -
_$                /NODE="EMARET" -
_$                /ADDRESS="12.378" -
_$                /NETWORK_TYPE="DNA IV" -
_$                /INTERFACE="net 0" -
_$                /STATUS="mcr ncp show exec characteristics" -
_$                /CONNECTIONS="mcr ncp show known link" -
_$                /COUNTERS="mcr ncp show exec count" -
_$                /START="@sys$manager:startnet" -
```

```
_$                /STOP="mcr ncp set executor state shut"
```

This command adds the network service DECnet (Phase IV) to the database, with the specified characteristics.

SET ON

SET ON — Enables error checking by the command interpreter after the execution of each command in a command procedure. Specify **SET NOON** to disable error checking.

Format

SET [NO]ON

Parameters

None.

Description

During the execution of command procedures, the command interpreter normally checks the status code returned when a DCL command or program image completes and saves the numeric value of this code in the reserved symbol named $STATUS. The low-order 3 bits of this value are also saved in the reserved symbol $SEVERITY. Command procedure execution aborts when either an error or fatal error is detected.

Use the **SET NOON** command to override default error checking. When **SET NOON** is in effect, the command interpreter continues to place the status code value in $STATUS and the severity level in $SEVERITY, but does not perform any action based on the values. As a result, the command procedure continues to execute no matter how many errors are returned.

Exception

If you use a **CALL**, **GOSUB**, or **GOTO** command and specify a label that does not exist in the current command procedure, the procedure issues a warning message and exits.

The **SET ON** or **SET NOON** command applies only at the current command level. If you use the **SET NOON** command in a command procedure that executes another procedure, the default, **SET ON**, is established while the second procedure executes.

Example

```
$ SET NOON
$ DELETE  *.SAV;*
$ SET ON
$ COPY  *.OBJ  *.SAV
```

This command procedure routinely copies all object modules into new files with the file type .SAV. The **DELETE** command first deletes all existing files with the .SAV file type, if any. The **SET NOON** command ensures that the procedure continues executing even if there are no files with the .SAV file type in the current directory. Following the **DELETE** command, the **SET ON** command restores error checking. Then the **COPY** command makes copies of all existing files with .OBJ file type.

SET OUTPUT_RATE

SET OUTPUT_RATE — Sets the rate at which output is written to a batch job log file. **For use only within command procedures that are submitted as batch or detached jobs.**

Format

SET OUTPUT_RATE[=*delta-time*]

Parameter

delta-time

The time interval at which output is written from the output buffer to the batch job log file. If no delta time is specified, the information is written in the output buffer to the log file, but the output rate is not changed from the default of once per minute. Specify *delta-time* as [dddd-][hh:mm:ss.cc]. For more information on delta time, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

Description

When you submit a batch job, the output to be written to the log file is stored in an output buffer. Periodically, the buffer is flushed and its contents are written to the batch job log file. By default, the buffer is flushed once a minute; therefore, you can type the log file to determine how much of the job has completed while the job is still executing.

To change the default output rate, include the **SET OUTPUT_RATE** command in the command procedure that you are submitting as a batch job. When the **SET OUTPUT_RATE** command is executed within a batch job and a delta time is specified, DCL flushes the buffer, sets the default output rate, and starts a new output interval.

If the **SET OUTPUT_RATE** command is executed within a batch job and a delta time is not specified, DCL flushes the buffer but does not change the default output rate and does not start a new interval.

If you issue the **SET OUTPUT_RATE** command interactively, or within a command procedure that is executed interactively, then no action is performed.

Example

```
$ SET OUTPUT_RATE=:0:30
   .
   .
   .
```

This command, when executed within a batch job, changes the default output rate from once a minute to once every 30 seconds.

SET PASSWORD

SET PASSWORD — Changes a password or system password. Also, can establish a secondary password or system password, or remove a secondary password. **See the qualifier descriptions for restrictions.**

Format

SET PASSWORD

Parameters

None.

Description

All user accounts on a system have passwords. A password is required for logging in to the system.

To maintain secrecy, users should change their passwords from time to time. The **SET PASSWORD** command offers a means of making this change.

A system manager can control which users have the right to change their passwords, and can establish a minimum password length and the maximum period of time that a password can remain unchanged. The OpenVMS system automatically screens passwords against a dictionary and a history list to prevent the use of native language words or the reuse of old passwords. A system manager can add words readily associated with the site to the dictionary, thus disallowing them as passwords.

Systems can also have passwords (not to be confused with the password associated with the SYSTEM account). The system manager uses the **SET PASSWORD/SYSTEM** command to change the system password from time to time.

A user password can contain up to 32 alphanumeric and special characters. Unless your system manager has set the PWDMIX flag in your authorization account record, the only special characters permitted are the dollar sign ($) and underscore (_). Without the PWDMIX authorization, all lowercase characters are converted to uppercase before the password is encrypted. For example, "eagle" is the same as "EAGLE".

If you have PWDMIX authorization, you can specify uppercase and lowercase alphabetic characters, and you can use any special characters in the printable character set. However, avoid using characters that have special significance for layered products that accept passwords. For example, a double quotation mark (") would be unacceptable in a password in a DECnet access control string.

Note

Blank spaces are permissible within a password, but they are not considered part of the password, and OpenVMS ignores them. For example, "MY PASSWORD" is an acceptable password, but the system only records "MYPASSWORD." This means that "MYPA SSWORD" is also a valid password for the account in question.

A password that contains blank spaces will fail in contexts where spaces have meaning. For example, the syntax of an access control string assumes there is one and only one space, preceding the password, for example:

```
$ DIRECTORY JULY04"JEFFERSON PRESIDENT"::TEST.SDML
```

If you enter the following command, it will fail:

```
$ DIRECTORY JULY04"JEFFERSON PRE SIDENT"::TEST.SDML
```

Use the following procedure to change your password:

1. Enter the **SET PASSWORD** command.
2. The system prompts you for your current password. Enter your current password.
3. The system prompts you for a new password. Enter a new password, or press **Return** to disable your current password.
4. The system prompts you to verify the password. Enter the new password to verify. (If the two entries of the new password do not match, the password does not change.)

The following guidelines are recommended to minimize the chances of passwords being discovered by trial-and-error or by exhaustive search:

- Make passwords at least 6 characters long.
- Avoid names or words that are readily associated with you.
- Change your passwords at least once every month.

To ensure that the previous guidelines are met, use the **/GENERATE[=*value*]** qualifier. This qualifier generates random passwords of up to 12 characters in length. The system manager can require individual users to use generated passwords. For more information about this, see the description of the Authorize utility's **/GENERATE_PASSWORD** qualifier in the *VSI OpenVMS System Management Utilities Reference Manual, Volume I: A–L* [https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#AUTHORIZE_PART].

External Authentication

If external authentication is enabled on your system and user accounts in the SYSUAF file are marked for external authentication, those users are authenticated using their external user IDs and passwords.

When a user has logged in using external authentication, the **SET PASSWORD** command attempts to notify the external authentication service of any password change. The following messages are displayed when this takes place:

```
%SET-I-SNDEXTAUTH, Sending password request to external authenticator
%SET-I-TRYPWDSYNCH, Attempting password synchronization
```

If the external authentication service cannot be contacted or the password change is rejected, an error message is displayed to the user:

```
%SET-E-EXTPWDERR, Password could not be set by external authenticator
```

If the user was not externally authenticated but the EXTAUTH flag is set (the user logged in with **/LOCAL_PASSWORD**), the new password is only set locally in the SYSUAF file (normal OpenVMS password policy checks do not apply in this case).

Qualifiers

/ALGORITHM=*keyword*

Selects a specific password generation algorithm.

The **/ALGORITHM** qualifier requires the **/GENERATE** qualifier.

The valid keywords for this qualifier are the following:

- ALPHABETIC – Selects the alphabetic password generator. (Formerly the only password generator).
- MIXED_CHARACTER – Selects the mixed-character password generator. Mixed-character passwords are valid only for users with the PwdMix flag set in their user authorization records.

These keywords are mutually exclusive and cannot be negated.

/GENERATE[=*value*]

Generates a list of five random passwords. Press Return to repeat the procedure until a suitable password appears.

Value is a number from 1 to 64 that restricts the length of the password. For any value *n*, the **SET PASSWORD** command generates alphabetic passwords of from *n* to (*n*+2) characters long, but no longer than 64 characters.

If your system manager has established a minimum password length for your account, **SET PASSWORD/GENERATE=*n*** compares that length with the optional value specified with the **/GENERATE** qualifier, and uses the larger of the two values. If you do not specify a value with the **/GENERATE** qualifier, the account minimum length is used.

Note

If the **SET PASSWORD/GENERATE** command fails to work properly, consult your system manager to be sure that either the file SYS$LIBRARY:VMS$PASSWORD_DICTIONARY.DATA exists, or the logical name VMS$PASSWORD_DICTIONARY is correctly defined.

/SECONDARY

Creates or allows you to replace a secondary password. The procedure is the same as setting your primary password.

Once a secondary password has been established, you will receive two PASSWORD: prompts when logging in. The primary password should be typed in first, followed by the secondary password.

Secondary passwords make it possible to set up an account that requires two different people to access it. Each person knows one of the two passwords, and both passwords are required to log in successfully.

To remove your secondary password, press **Return** when **SET PASSWORD/SECONDARY** prompts you for a new password and verification. After you do this, you will receive a single PASSWORD: prompt when logging in. If you remove the secondary password, your system manager must restore it.

The **/SECONDARY** and **/SYSTEM** qualifiers are incompatible.

/SYSTEM

Requires the SECURITY privilege.

Changes the system password rather than a user password.

A system password can be from 0 to 32 alphanumeric characters. 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.

A system password is valid only for the node it is set on. In an OpenVMS Cluster, each node can have a different system password.

If a terminal line has the system password (SYSPWD) characteristic set, no terminal prompts are sent to that terminal until the system password is entered.

The **/SYSTEM** and **/SECONDARY** qualifiers are incompatible. For more information about the use of system passwords, see the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#SETTING_SYST_PWDS].

Example

```
$ SET PASSWORD
Old password: HONCHO
New password: BIG_ENCHILADA
Verification: BIG_ENCHILADA
```

In response to the **SET PASSWORD** command, the system first prompts for the old password and then for the new password. The system then prompts again for the new password to verify it. The password changes if the user is authorized to change this account's password, if the old password is given correctly, and if the new password is given identically twice; otherwise, an error message appears and the password remains unchanged.

In a real session, neither the old password nor the new password and its verification appear on the screen or paper.

SET PREFERRED_PATH

SET PREFERRED_PATH — Access to a MSCP/TMSCP class device may be available to an OpenVMS node from more than one host or storage controller. Use the **SET PREFERRED_PATH** command to specify a particular host or preferred path to access a specific MSCP class disk or TMSCP class tape device.

Format

SET PREFERRED_PATH *device-name*:

Parameter

***device-name*:**

Specifies the name of a MSCP class disk or TMSCP class tape device.

Description

If a preferred path is specified, the normal OpenVMS path selection process is modified to select the user-specified path over other available paths, assuming the user-specified path is available.

This command is only meaningful for MSCP/TMSCP class devices that have more than one path available.

Qualifiers

/HOST=*host_name*

Tells the MSCP/TMSCP class driver that you want this host to be the preferred access path to the specified device. The class driver remembers this host name until it is changed by taking one of the following actions:

- Issuing another **SET PREFERRED_PATH** command specifying a different host
- Using the **/NOHOST** qualifier
- Rebooting the system

Note that simply specifying a preferred path does not mean that it is immediately selected if the disk or tape device is currently using another path. Use the **/FORCE** qualifier along with the **/HOST** qualifier to force the preferred path to be used immediately.

The *host_name* is the name of the host that will be the preferred path to the disk or tape device.

/FORCE

Tells the class driver to initiate the path selection process immediately. If this qualifier is omitted when a new host name is specified, a switch from the current path to the new preferred path will not occur until some other event initiates the path selection process.

/NOHOST

Clears any previously defined preferred path assignment. Restores usual OpenVMS path selection behavior.

Examples

1. `$ SET PREFERRED_PATH $10$DUA10: /HOST=HSC014`

   Prior to issuing this command, the `$10$DUA10:` disk device has host `HSC015` as its primary path and host `HSC014` as its secondary path. Issuing this command selects host `HSC014` as the preferred path.

   Note that the preferred path has been recorded by the class driver; however, the disk will remain on the current path (`HSC015`) until the next time the path selection process is initiated.

2. `$ SET PREFERRED_PATH $10$DUA10: /HOST=HSC014 /FORCE`

   To make the path change occur immediately, include the **/FORCE** qualifier on the command line with the preferred path specification.

3. `$ SET PREFERRED_PATH $10$DUA10: /FORCE`

   Issue this command to initiate path selection processing if the specified device has a primary path that differs from the preferred path.

   If the preferred path is available, the device moves to that path.

4. `$ SET PREFERRED_PATH $10$DUA10: /NOHOST`

   Remove the specified preferred path by using the **/NOHOST** qualifier if the device should no longer have a preferred path.

SET PREFIX

SET PREFIX — Allows you to set a prefix control string for verified command lines.

Format

SET [NO]PREFIX *string*

Parameter

string

Specifies the FAO control string to be used in generating a prefix to a verified command line. The following rules apply:

- No more than 64 characters are allowed in the control string.
- The resulting string can be no longer than 64 characters.
- Basic formatting FAO directives can be used ("!/", "!_", "!^", "!!", "!%F", and "!*n**c").
- Time and date FAO directives can be used ("!%T" and "!%D").
- Repeat counts can be used ("!*n*(DD)").
- Output field length specifications can be used ("!lengthDD").
- Combination of repeat count and output field length can be used("!*n*(lengthDD)").
- FAO directives that require arguments will always receive a value of zero.

Description

The **SET PREFIX** command allows you to prefix verified command lines with a custom string. This string is a limited FAO control string that specifies date and time information as well as constant information and formatting controls (that is, tabs, form feeds, and so on). For example, this allows you to use a full date and time prefix (a time stamp) to identify batch runs and to verify that a batch job ran at the expected time.

See the documentation on the F$FAO lexical function in the *VSI OpenVMS DCL Dictionary: A–M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#d0e33542] for more information about FAO control strings.

The first line of a verified command is prefixed with the result of the control string. The control string is evaluated before the command itself is executed. Any continuation lines are prefixed with a blank string in order to make them flush with the first line of the command. Command input and output are not prefixed. The prefix control string can later be retrieved by using VERIFY_PREFIX with F$ENVIRONMENT.

Example

```
$ SET VERIFY
$ @TEST
```

```
$ SET DEFAULT SYS$LOGIN
$ SHOW DEFAULT
  USER$:[JENSEN]
$ SET PREFIX "(!5%T) "
$ @TEST
(17:52) $ SET DEFAULT SYS$LOGIN
(17:52) $ SHOW DEFAULT
  USER$:[JENSEN]
```

This example demonstrates the difference between having and not having a prefix for verification. The first command turns on verification. (Verification must be turned on to see the prefix.) The second command invokes a test procedure to show what the output looks like without a prefix. The third and fourth lines reflect the contents of the test procedure invoked in the preceding command. The third command sets the prefix to an FAO control string so that the first five characters of the standard time will be shown for each command. The last command invokes the test procedure again to demonstrate what the output looks like with a prefix.

SET PRINTER

SET PRINTER — Establishes the characteristics of a specific line printer. The default values listed for qualifiers to the **SET PRINTER** command are the defaults for an initially bootstrapped system. **Requires OPER (operator) privilege. If the printer is a spooled device, LOG_IO (logical I/O) privilege is required to modify its characteristics.**

Format

SET PRINTER *printer-name*[:]

Parameter

***printer-name*[:]**

Specifies the name of a line printer to set or modify its characteristics.

Qualifiers

/CR
/NOCR (default)

Controls whether the printer driver outputs a carriage return character. Use this qualifier for printers on which line feeds do not imply carriage returns.

Specify the **/NOCR** qualifier for printers where the line feed, form feed, vertical feed, and carriage return characters empty the printer buffer. The **/NOCR** qualifier causes carriage return characters to be held back and output only if the next character is not a form feed or vertical tab. Carriage return characters are always output on devices that have the carriage return function characteristic set.

/FALLBACK
/NOFALLBACK (default)

Determines whether the printer attempts to translate characters belonging to the DEC Multinational character set into 7-bit equivalent representations. If a character cannot be translated, an underscore (_) character is substituted.

If the **/PASSALL** qualifier is in effect, it has precedence over the **/FALLBACK** qualifier.

/FF (default)
/NOFF

Indicates whether the printer performs a mechanical form feed. Use the **/NOFF** qualifier when the printer does not automatically perform mechanical form feeds. This qualifier allows the driver to convert form feeds into multiple line feeds and to output them.

/LA11

Specifies the printer as an LA11. This qualifier provides information for the **SHOW PRINTER** command, which, in turn, provides the user with information about specific printers. If no printer type is specified, LP11 is assumed.

/LA180

Specifies the printer as an LA180. This qualifier provides information for the **SHOW PRINTER** command, which, in turn, provides the user with information about specific printers. If no printer type is specified, LP11 is assumed.

/LOG
/NOLOG (default)

Determines whether information confirming the printer setting is displayed at the terminal from which the **SET PRINTER** command was entered.

/LOWERCASE
/NOLOWERCASE

Indicates whether the printer prints both uppercase and lowercase letters or only uppercase. When the operator specifies the **/NOLOWERCASE** qualifier, all letters are translated to uppercase.

The **/[NO]LOWERCASE** and **/[NO]UPPERCASE** qualifiers are complementary; that is, the **/LOWERCASE** qualifier is equivalent to the **/NOUPPERCASE** qualifier, and the **/NOLOWERCASE** qualifier is equivalent to the **/UPPERCASE** qualifier.

/LP11 (default)

Specifies the printer as an LP11. This qualifier provides information for the **SHOW PRINTER** command, which, in turn, provides the user with information about specific printers. LP11 is the default printer type.

/PAGE=*lines-per-page*

Establishes the number of lines per page on the currently installed form; the number of lines can range from 1 to 255 and defaults to 64. The printer driver uses this value to determine the number of line feeds that must be entered to simulate a form feed.

/PASSALL
/NOPASSALL (default)

Controls whether the system interprets special characters or passes them as 8-bit binary data.

If you specify the **/PASSALL** qualifier, the printer driver does not expand tab characters to spaces, fill carriage return or line feed characters, or recognize control characters.

/POLLED
/NOPOLLED (default)

Instructs the LRDRIVER to run in polled mode instead of interrupt mode.

In polled mode, the printer is checked on a regular basis to determine if it is ready to receive additional data. The polling timer runs only while there is more data to be sent to the printer.

/PRINTALL
/NOPRINTALL (default)

Controls whether the line printer driver outputs printable 8-bit multinational characters.

/RESET

Resets the USB pipe associated with the USB printer, thereby canceling all print jobs.

/TAB
/NOTAB (default)

Controls how the printer handles TAB characters. The **/NOTAB** qualifier expands all tab characters to spaces and assumes tab stops at 8-character intervals.

Use the **/TAB** qualifier when you do not want the system to convert tabs to spaces, but want the printer to process the tab characters. The OpenVMS system requires that printers expand tabs at 8-character intervals.

/TRUNCATE (default)
/NOTRUNCATE

Controls whether the printer truncates data exceeding the value specified by the **/WIDTH** qualifier. Note that the **/TRUNCATE** and **/WRAP** qualifiers are incompatible.

/UNKNOWN

Specifies the printer as nonstandard. This qualifier provides information for the **SHOW PRINTER** command, which, in turn, provides the user with information about specific printers. If no printer type qualifier is specified, LP11 is assumed.

/UPPERCASE
/NOUPPERCASE

Indicates whether the printer prints both uppercase and lowercase letters or only uppercase ones. When you specify the **/UPPERCASE** qualifier, all letters are translated to uppercase.

The **/[NO]UPPERCASE** and **/[NO]LOWERCASE** qualifiers are complementary; that is, the **/UPPERCASE** qualifier is equivalent to the **/NOLOWERCASE** qualifier, and the **/NOUPPERCASE** qualifier is equivalent to the **/LOWERCASE** qualifier.

/WIDTH=*n*

Establishes the number of characters per output line on currently installed forms. The width, *n*, can range from 0 to 65535 for LP11 controllers, and from 0 to 255 for DMF32 controllers. The default value is 132 characters per line.

/WRAP
/NOWRAP (default)

Controls whether the printer generates a carriage return and a line feed when it reaches the end of a line.

If the **/NOWRAP** qualifier is specified, the printer writes characters out in the last position on the line.

If the **/WRAP** qualifier is specified, the terminal generates a carriage return and a line feed whenever the end of a line is reached.

Note that the **/TRUNCATE** and **/WRAP** qualifiers are incompatible.

Examples

1. `$ SET PRINTER/PAGE=60/WIDTH=80  LPA0:`

   The **SET PRINTER** command in this example establishes the size of an output page as `60` lines and the width of a line as `80` characters for printer `LPA0`.

2. `$ SET PRINTER/LA11  LPB0:`

   The **SET PRINTER** command in this example establishes the line printer `LPB0` as an `LA11` printer.

3. `$ SET PRINTER/LOWERCASE LPA0:`

   The **SET PRINTER** command in this example requests that lowercase printing be enabled on line printer `LPA0`.

SET PROCESS

SET PROCESS — Changes the execution characteristics associated with the specified processor kernel thread for the current terminal session or job. If no process is specified, changes are made to the current process. **Requires GROUP privilege to change other processes in the same group. Requires WORLD privilege to change processes outside your group.**

Format

SET PROCESS [*process-name*]

Parameter

process-name

Requires that you own the process or that you have GROUP privilege and that the process is in your group.

Specifies the name of the process for which the characteristics are to be changed. Process names can be up to 23 alphanumeric characters long in the following format:

`[`*node-name*`::]`*process-name*

- The node name can have as many as 6 alphanumeric characters.

- The colons (:) count for 2 characters.
- The process name can have as many as 15 characters.

A local process name can look like a remote process name; therefore, if you specify ATHENS::SMITH, the system checks for a process named ATHENS::SMITH on the local node before checking node ATHENS for a process named SMITH.

The default process is the current process.

Process names are unique only within a particular UIC group. You cannot specify the process name for a process outside of your group. To change the characteristics of a process outside of your group, you must use the qualifier **/IDENTIFICATION=*pid***. The process name parameter is ignored. If you include neither the process name nor the **/IDENTIFICATION** qualifier, the current process is assumed.

Qualifiers

/AFFINITY (Alpha/IA-64 only)
/NOAFFINITY

Allows bits in the kernel thread affinity mask to be set or cleared individually, in groups, or all at once. This qualifier is mutually exclusive with the **/CAPABILITY** qualifier.

Note

The **SET PROCESS/[NO]AFFINITY** command fails if none of the specified CPUs has the capabilities required by the process.

The **/NOAFFINITY** qualifier clears all affinity bits currently set in the current or permanent affinity masks, based on the setting of the **/PERMANENT** qualifier. Specifying the **/AFFINITY** qualifier has no direct effect, but merely indicates the target of the operations specified by the following secondary qualifiers:

Qualifier	Description
/SET=(*n*[,...])	Sets affinity for currently active CPUs defined by the CPU IDs *n*, where *n* has the range of 0 to 31.
/CLEAR=(*n*[,...])	Clears affinity for currently active CPUs defined by the position values *n*, where *n* has the range of 0 to 31.
/PERMANENT	Performs the operation on the permanent affinity mask as well as the current affinity mask, making the changes valid for the life of the kernel thread. (The default behavior is to affect only the affinity mask for the running image.)

The secondary qualifiers can all be used at once as long as the affinity bits defined in the **/SET** and **/CLEAR** parameters do not overlap.

The privileges required to execute the **SET PROCESS/AFFINITY** command match those required by the $PROCESS_AFFINITY system service. ALTPRI is the base privilege required to make any modifications, and the only privilege required to modify the current owner's kernel thread. Modifications within the same UIC group require GROUP privilege. Modifications to any unrelated kernel thread require WORLD privilege.

As with the other **SET PROCESS** qualifiers, the bit operations occur on the current process if no **/IDENTIFICATION** qualifier or explicit process name parameter is specified. Specifying a process name does not imply that all kernel threads associated with the process are affected; the **SET PROCESS** command affects only the initial kernel thread of a multithreaded process.

/AUTO_UNSHELVE
/NOAUTO_UNSHELVE

Controls whether the process automatically unshelves files.

Note that the **/NOAUTO_UNSHELVE** qualifier does not work across a cluster. It can be issued only for a process on the same node, including as the default case, the process from which the command is issued.

/CAPABILITY
/NOCAPABILITY

Allows bits in the process user capability mask to be set or cleared individually, in groups, or all at once. This qualifier is mutually exclusive with the **/AFFINITY** qualifier.

Note

The **SET PROCESS/[NO]CAPABILITY** command fails if there is no CPU with the required set of capabilities.

The **/NOCAPABILITY** qualifier clears all user capability bits currently set in the current or permanent capability masks, based on the setting of the **/PERMANENT** qualifier. Specifying the **/CAPABILITY** qualifier has no direct effect, but merely indicates the target of the operations specified by the following secondary qualifiers:

Qualifier	Description
/SET=(*n*[,...])	Sets all user capabilities defined by the position values *n*, where *n* has the range of 1 to 16.
/CLEAR=(*n*[,...])	Clears all user capabilities defined by the position values *n*, where *n* has the range of 1 to 16.
/PERMANENT	Performs the operation on the permanent user capability mask as well as the current user capability mask, making the changes valid for the life of the kernel thread. (The default behavior is to affect only the capabilities mask for the running image.)

The secondary qualifiers can all be used at once as long as the user capability bits defined in the **/SET** and **/CLEAR** parameters do not overlap.

The privileges required to execute the **SET PROCESS/CAPABILITY** command match those required by the $PROCESS_CAPABILITIES system service. ALTPRI is the base privilege required to make any modifications, and the only privilege required to modify the current owner's kernel thread. Modifications within the same UIC group require GROUP privilege. Modifications to any unrelated kernel thread require WORLD privilege.

As with the other **SET PROCESS** qualifiers, the bit operations occur on the current process if no **/IDENTIFICATION** qualifier or explicit process name parameter is specified. Specifying a process name does not imply that all kernel threads associated with the process are affected; the **SET PROCESS** command affects only the initial kernel thread of a multithreaded process.

/CASE_LOOKUP=*keyword*

The valid keywords for this qualifier are BLIND and SENSITIVE.

VSI strongly recommends that you use caution when enabling case sensitivity in your processes.

See the *Guide to OpenVMS File Applications* [https://docs.vmssoftware.com/guide-to-openvms-file-applications/] for additional information.

/CLEAR=(*n*[,...])

Clears all user capabilities or CPU affinities defined by the position values *n* specified by the /CAPABILITY or the **/AFFINITY** qualifier.

/DUMP[=NOW] [/ID=*pid*] [*process-name*]
/NODUMP (default)

Causes the contents of the address space to be written to the file named (image-name).DMP in the current default directory (where the image name is the same as the file name) when an image terminates because of an unhandled error. To specify a target process, indicate either the process ID or the process name.

Using the NOW option (available on Alpha and IA-64 systems only) causes the targeted process to dump as soon as possible. This is especially useful for hung processes.

You can then analyze the dump with the **ANALYZE/PROCESS_DUMP** utility, the Debugger, or the System Dump Analyzer (SDA).

/GLOBAL

Modifies the global cell SCH$GL_DEFAULT_PROCESS_CAP. This global cell is used to initialize the user capability mask of processes when they are started.

/IDENTIFICATION=*pid*

Requires GROUP or WORLD privilege for processes other than your own.

Specifies the process identification (PID) value of the kernel thread for which characteristics are to be changed. The target process must be on the same node as the process from which the command is issued. The **/IDENTIFICATION** qualifier overrides the *process-name* parameter.

The PID is assigned by the system when the process is created. When you specify a PID, you can omit the leading zeros.

If you use the **/IDENTIFICATION** qualifier, the *process-name* parameter is ignored.

Note

The **/IDENTIFICATION** qualifier allows the **SET PROCESS** command to affect individual kernel thread PIDs; because each thread is a separate runnable entity, this command treats them as discrete entities in terms of affinities and capabilities.

/KERNEL_THREAD_LIMIT=*n*

Changes the limit of the number of kernel threads that can be created in the process. If the value specified is higher than the SYSGEN parameter MULTITHREAD, an error message is returned.

Specifying the value 0 makes the process use the systemwide limit set by the MULTITHREAD parameter.

Per-process kernel thread limits can be controlled using one of the following DCL commands:

```
$ SET PROCESS/KERNEL_THREAD_LIMIT=n
$ RUN/KERNEL_THREAD_LIMIT=n
$ SPAWN/KERNEL_THREAD_LIMIT=n
```

/NAME=*string*

Changes the name of the current process to a string of 1 to 15 characters.

/PARSE_STYLE=(*keyword*)

Allows a user to indicate how commands are to be parsed.

Setting a particular parse style tells DCL how it should handle command syntax. User programs can also examine the state of this setting if they need to use different parse rules.

The valid keywords for this qualifier are TRADITIONAL and EXTENDED. These keywords are mutually exclusive and cannot be negated.

If the **/PARSE_STYLE** qualifier is not specified, the default is the TRADITIONAL format.

Keyword	Description
TRADITIONAL (default)	Indicates that commands should be examined using the former (prior to Version 7.2) rules for DCL syntax.
EXTENDED (Alpha/IA-64 only)	Indicates that commands should be examined using a syntax that allows ODS-5 file specifications.

The main differences for DCL when EXTENDED parse rules are in effect are:

- Arguments to foreign commands are case preserved. You can get the command string by calling LIB$GET_FOREIGN.

  C/C++ programs that use the argc/argv mechanism will have unquoted arguments in lowercase unless the C Run-Time Library logical DECC$ARGV_PARSE_STYLE is set to ENABLE. When DECC$ARGV_PARSE_STYLE is enabled, case is preserved in command line arguments when the process is set up for extended DCL parsing using **/PARSE_STYLE=EXTENDED**.

- Some characters that were previously treated as token delimiters are no longer delimiters. The pound sign (#), circumflex (^), and question mark (?) fall into this category.

- A circumflex (^) is an escape character, which can be used to indicate that the next character in the command string is to be treated as if it were quoted, thereby losing its syntactic significance.

EXTENDED parsing also modifies DCL's rules for parsing a parameter or qualifier that is defined as a file specification in a command's definition:

- File specifications will not be in uppercase.

- Any number of commas (,) may be placed between directory delimiters ([] and <>).

- Directory file ID's (DIDs) can be included in the file specification.

- Any number of periods (.) or semi-colons (;) may be included in the file specification.

See the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/] for more information.

/PERMANENT

Performs the operation on the permanent affinity or capability mask as well as the current mask, making the changes valid for the life of the kernel thread. (The default behavior is to affect the current mask only for the running image.) The **/PERMANENT** qualifier is only valid in conjunction with the **/CAPABILITY** or the **/AFFINITY** qualifier.

/PRIORITY=*n*

Requires ALTPRI (alter priority) privilege to set the priority higher than the base priority of the specified process.

Changes the priority for the specified kernel thread. If you do not have the ALTPRI privilege, the value you specify is compared to your current base priority, and the lower value is always used.

/PRIVILEGES=(*privilege*[,...])

Requires SETPRV (set privilege) privilege as an authorized privilege to enable any privilege you do not have in your authorized privilege mask.

Enables the following process privileges:

ACNT	ALLSPOOL	ALTPRI	AUDIT
BUGCHK	BYPASS	CMEXEC	CMKRNL
DETACH	DIAGNOSE	DOWNGRADE	EXQUOTA
GROUP	GRPNAM	GRPPRV	IMPORT
LOG_IO	MOUNT	NETMBX	OPER
PFNMAP	PHY_IO	PRMCEB	PRMGBL
PRMMBX	PSWAPM	READALL	SECURITY
SETPRV	SHARE	SHMEM	SYSGBL
SYSLCK	SYSNAM	SYSPRV	TMPMBX
UPGRADE	VOLPRO	WORLD	

Use the **SHOW PROCESS/PRIVILEGES** command to determine what authorized privileges are enabled currently.

/RAD=HOME=*n*

Changes the home resource affinity domain (RAD) of a process.

This command only works on the current process.

RAD is supported on AlphaServer GS series systems and, starting from OpenVMS Version 8.4, support is extended to NUMA-capable IA-64 systems.

/RESOURCE_WAIT
/NORESOURCE_WAIT

Enables resource wait mode so that the process waits for resources to become available. If you specify the **/NORESOURCE_WAIT** qualifier, the process receives an error status code when system dynamic memory is not available or when the process exceeds one of the following resource quotas: direct I/O (DIOLM) limit, buffered I/O (BIOLM) limit, buffered I/O byte (BYTLM) count limit (buffer space), timer queue quota, mailbox buffer quota, or pipe quota.

Caution

Disabling resource waiting should be performed with caution, as doing so can have unexpected effects on constituent sharable images and runtime libraries. See the $SETRWM service in the *VSI OpenVMS System Services Reference Manual: GETUTC–Z* [https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-getutc-z/#JUN_518] for additional information.

/RESUME

Allows a process suspended by a previous **SET PROCESS/SUSPEND** command to resume operation. The **/RESUME** qualifier is equivalent to the **/NOSUSPEND** qualifier.

/SCHEDULING_CLASS= "*class_name*"
/NOSCHEDULING_CLASS

Adds a process to, or removes a process from, a scheduling class. This command does not modify the permanent class scheduling database file. Therefore, the effect of this command can be viewed as temporary placement into, or removal from, a scheduling class.

/SET=(*n*[,...])

Sets all user capabilities or CPU affinities defined by the position values *n* specified by the **/CAPABILITY** or the **/AFFINITY** qualifier.

/SSLOG=(STATE=ON | OFF | UNLOAD [,COUNT=*n*][,FLAGS=[NO]ARG])

Valid on Alpha and IA-64 systems only.

Requires CMEXEC, CMKRNL, or SETPRV privilege to log argument values. The SYSGEN parameter SYSSER_LOGGING must be enabled or the command will fail.)

Enables or disables system service logging, using a log file (named SSLOG.DAT by default) to log data.

Keywords are as follows:

Keyword	Description	
COUNT= *n*	Specifies how many P2-space buffers to log. (Default: 2)	
FLAGS=[NO]ARG	Specifies whether service argument values are to be logged. (Default: ARG) ARG requires CMEXEC, CMKRNL, or SETPRV privilege.	
STATE=*state*	Turns system service logging on or off. Possible states are:	
	ON	Enables system service logging.

Keyword	Description	
	OFF	Disables (turns off) system service logging; logging can still be reenabled.
	UNLOAD	Stops logging and closes the log file, which is named SSLOG.DAT by default.

When enabling SSLOG for a process, you can specify the number of buffers to be used for logging. Buffers are allocated in P2 space and are charged against the process's paging file quota. Each buffer is $65{,}024_{10}$ bytes or $FE00_{16}$ bytes. The buffer space remains allocated and the quota is charged until the process is deleted.

Between the time when SSLOG is first enabled and when the log file is closed, logging can be stopped and resumed.

Before you delete the process, you should stop all logging and close the log file. The log file does not close automatically.

To analyze the log file, use the DCL command **ANALYZE/SSLOG**, which is described in online help and in the relevant section in the *VSI OpenVMS System Analysis Tools Manual* [https://docs.vmssoftware.com/vsi-openvms-system-analysis-tools-manual/#CHAPT14].

/SUSPEND[=SUPERVISOR]
/SUSPEND=KERNEL
/NOSUSPEND

Requires privileges as described in text.

Temporarily stops the process's activities. The process remains suspended until another process resumes or deletes it. Use the qualifiers **/NOSUSPEND** and **/RESUME** to resume a suspended process.

Specify either of the following keywords with the **/SUSPEND** qualifier to produce different results:

Keyword	Description
SUPERVISOR (default)	Specifies that the named process is to be suspended to allow the delivery of asynchronous system traps (ASTs) at EXEC or KERNEL mode. Specifying this keyword is optional.
KERNEL	Specifies that the named process is to be suspended such that no ASTs can be delivered. To specify the KERNEL keyword, you must be in either KERNEL mode or EXEC mode, or have CMKRNL (change mode to kernel) and CMEXEC (change mode to executive) privilege enabled. Note that this was the default behavior of the **SET PROCESS/SUSPEND** command for versions of OpenVMS prior to VMS Version 5.0.

Depending on the operation, the process from which you specify the **/SUSPEND** qualifier requires privileges. You must have GROUP privilege to suspend another process in the same group, unless that process has the same user identification code (UIC). You must have WORLD privilege to suspend any other process in the system.

When you enter the **SET PROCESS/SUSPEND=KERNEL/ID=** command in a cluster environment, the KERNEL keyword is ignored if the target process and the current process reside

on different cluster nodes. As a result, process suspension is handled as if you had specified the SUPERVISOR keyword (the default).

Note that you can specify **SET PROCESS/SUSPEND=KERNEL** to override a previous **SET PROCESS/SUSPEND=SUPERVISOR**. **SET PROCESS/SUSPEND=SUPERVISOR** does not, however, override **SET PROCESS/SUSPEND=KERNEL**.

/SWAPPING (default)
/NOSWAPPING

Requires the PSWAPM (process swap) privilege to disable swapping for your process.

Permits the process to be swapped. By default, a process that is not currently executing can be removed from physical memory so that other processes can execute. If you specify the **/NOSWAPPING** qualifier, the process is not swapped out of the balance set when it is in a wait state.

/SYMLINK=*keyword*

Controls the behavior of all directory wildcard searches used in the RMS$SEARCH service. Note that these directory wildcards are not overridden by command qualifiers or other explicit RMS characteristics. Keyword options are:

Keyword	Description
NOWILDCARD	Indicates that symlinks are disabled during directory wildcard searches.
WILDCARD	Indicates that symlinks are enabled during wildcard searches.
NOELLIPSIS	Indicates that symlinks are matched for all wildcard fields except for ellipsis.
ELLIPSIS	Equivalent to WILDCARD (included for command symmetry).

On the command line, you can set a flag to select the **/SYMLINK** qualifier. It uses the keywords DEFAULT, [NO]WILDCARD,[NO]ELLIPSIS, and [NO]TARGET. In addition, **/SYMLINK** and **/NOSYMLINK** without any keywords are interpreted as **/SYMLINK=NOTARGET** and **/SYMINK=TARGET**, respectively.

/TOKEN=*keyword*

Changes the maximum size of tokens (elements) in a DCL command. (A token is any element in a command line that is bounded by spaces. For example, the command **COPY X.TXT Y.TXT** contains three tokens.) The token size is determined by the setting of bit 1 in the DCL_CTLFLAGS system parameter. By default, the bit is clear, indicating that traditional tokens (255 bytes) are being used. If the bit is set, extended tokens (4000 bytes) are used. The corresponding key words for **/TOKEN** are TRADITIONAL and EXTENDED. You can use **SHOW PROCESS/TOKEN** to determine the current token size setting.

/UNITS[=*keyword*]

Specifies whether the amount of disk space reported by certain utilities is to be displayed in blocks or bytes. Keyword options are:

Keyword	Description
BLOCKS	Displays disk space in blocks.
BYTES	Displays disk space in bytes.

Blocks is the default until **/UNITS** is set to BYTES. If you specify **/UNITS** with no keyword, disk space is reported in blocks.

Displays that are affected by changing the value of **/UNITS** include output from certain forms of the following commands: **COPY**, **DELETE**, **DIRECTORY**, **PURGE**, **SHOW DEVICE**, **SHOW MEMORY**, and **SHOW QUOTA**. Note that input to these commands can be specified only in blocks. The **DIRECTORY**, **SHOW DEVICES**, and **SHOW MEMORY** commands have a qualifier that lets you override the default **SET PROCESS/UNITS** setting for a single command.

Examples

1. `$ SET PROCESS/PRIVILEGE=EXQUOTA`

   The **SET PROCESS** command in this example assigns the current process the privilege of exceeding quotas.

2. `$ SET PROCESS/NORESOURCE_WAIT`

   The **SET PROCESS** command in this example disables resource wait mode for the current process.

3. ```
   $ RUN/PROCESS_NAME=TESTER  CALC
   %RUN-S-PROC_ID, identification of created process is 0005002F
   $ SET PROCESS/PRIORITY=10  TESTER
   ```

   The **RUN** command in this example creates a subprocess and gives it the name `TESTER`. Subsequently, the **SET PROCESS/PRIORITY** command assigns the subprocess a priority of `10`.

4. ```
   $ SHOW PROCESS/SUBPROCESS

   19-APR-2001 15:17:28.41  User: DAVIS Process ID:   31900218
                            Node: OCALA  Process name: "DAVIS"

   Processes in this tree:

   DAVIS *
     DAVIS_1
     DAVIS_2

   $ SET PROCESS/SUSPEND DAVIS_1
   $
   ```

   The **SET PROCESS/SUSPEND** command in this example suspends the process `DAVIS_1` such that ASTs can be delivered to it. Because no key word was specified, the **/SUSPEND=SUPERVISOR** version is assumed.

5. ```
   $ SHOW PROCESS/SUBPROCESS OCALA::TESTA

   19-APR-2001 12:17:24.45   User: TESTA    Process ID:   31400208
                             Node: OCALA    Process name:  "TESTA"

   Processes in this tree:

   TESTA *
     TESTA_1
     TESTA_2
   $ SET PROCESS OCALA::TESTA_2 /SUSPEND=KERNEL
   $
   ```
   ```

The **SET PROCESS/SUSPEND=KERNEL** command in this example suspends the process TESTA_2 on node OCALA such that no ASTs can be delivered to it.

6.
```
$ SET PROCESS CPUSCHED/AFFINITY/SET=1/PERMANENT
$ SET PROCESS CPUSCHED/CAPABILITY/SET=8/PERMANENT
$ SHOW CPU/FULL
COBRA7, a DEC 4000 Model 620
Multiprocessing is ENABLED. Streamlined synchronization image loaded.
Minimum multiprocessing revision levels: CPU = 1

System Page Size = 8192
System Revision Code =
System Serial Number =
Default CPU Capabilities:
        System:         QUORUM RUN
Default Process Capabilities:
        System:         QUORUM RUN

PRIMARY CPU = 00

CPU 00 is in RUN state
Current Process: *** None ***
Serial Number:  AY24870417
Revision:       A200
VAX floating point operations supported.
IEEE floating point operations and data types supported.
Processor is Primary Eligible.
PALCODE: Revision Code = 5.48
         PALcode Compatibility = 0
         Maximum Shared Processors = 2
         Memory Space:  Physical address = 00000000 00000000
                        Length = 0
         Scratch Space: Physical address = 00000000 00000000
                        Length = 0
Capabilities of this CPU:
      System:           PRIMARY QUORUM RUN
      User bitmask:     00000040
Processes which can only execute on this CPU:
      NETACP       PID = 0000008E      Reason: PRIMARY Capability

CPU 01 is in RUN state
Current Process: CPUSCHED        PID = 00000095
Serial Number:  AY24870406
Revision:       A200
VAX floating point operations supported.
IEEE floating point operations and data types supported.
PALCODE: Revision Code = 5.48
         PALcode Compatibility = 0
         Maximum Shared Processors = 2
         Memory Space:  Physical address = 00000000 00000000
                        Length = 0
         Scratch Space: Physical address = 00000000 00000000
                        Length = 0
Capabilities of this CPU:
      System:           QUORUM RUN
      User bitmask:     00000080
Processes which can only execute on this CPU:
```

```
      CPUSCHED   PID = 00000095 Reason: Affinitized to this CPU
                                Reason: User capabilities - 00000080)

$ SET PROCESS/NOAFFINITY/PERMANENT
$ SET PROCESS/NOCAPABILITY/PERMANENT
$ SHOW CPU/FULL
COBRA7, a DEC 4000 Model 620
Multiprocessing is ENABLED. Streamlined synchronization image loaded.
Minimum multiprocessing revision levels: CPU = 1

System Page Size = 8192
System Revision Code =
System Serial Number =
Default CPU Capabilities:
        System:         QUORUM RUN
Default Process Capabilities:
        System:         QUORUM RUN

CPU 00 is in RUN state
Current Process: *** None ***
Serial Number:  AY24870417
Revision:       A200
VAX floating point operations supported.
IEEE floating point operations and data types supported.
Processor is Primary Eligible.
PALCODE: Revision Code = 5.48
         PALcode Compatibility = 0
         Maximum Shared Processors = 2
         Memory Space:  Physical address = 00000000 00000000
                        Length = 0
         Scratch Space: Physical address = 00000000 00000000
                        Length = 0
Capabilities of this CPU:
        System:         PRIMARY QUORUM RUN
        User bitmask:   00000040
Processes which can only execute on this CPU:
        NETACP           PID = 0000008E  Reason: PRIMARY Capability

CPU 01 is in RUN state
Current Process: CPUSCHED        PID = 00000095
Serial Number:  AY24870406
Revision:       A200
VAX floating point operations supported.
IEEE floating point operations and data types supported.
PALCODE: Revision Code = 5.48
         PALcode Compatibility = 0
         Maximum Shared Processors = 2
         Memory Space:  Physical address = 00000000 00000000
                        Length = 0
         Scratch Space: Physical address = 00000000 00000000
                        Length = 0
Capabilities of this CPU:
        System:         QUORUM RUN
        User bitmask:   00000080
Processes which can only execute on this CPU:
        *** None ***
$ SET PROCESS/NOAFFINITY/PERMANENT
$ SET PROCESS/NOCAPABILITY/PERMANENT
```

```
$ SHOW CPU/FULL
System: EUROS, HPE rx2600  (900MHz/1.5MB)  SMP execlet   = 3 : Enabled :
 Streamlined.  Config tree   = Version 6  Primary CPU   = 0  HWRPB
 CPUs    = 2  Page Size     = 8192  Revision Code =  Serial Number
 = US30464615  Default CPU Capabilities:        System: QUORUM RUN
  Default Process Capabilities:        System: QUORUM RUNCPU 0
  State: RUN                CPUDB: 8901C000     Handle: 00005D70
        Owner: 000004C8         Current: 000004C8     Partition 0
 (EUROS)        ChgCnt:        1           State: Present, Primary,
 Reassignable       Process: * None *  Capabilities:        System:
 PRIMARY QUORUM RUN RAD0          User: Usr2  Slot Context: 901EB000
     CPU     -  State..........: RC, PA, PP, CV, PV, PMV, PL
       Type...........: Itanium Major = 31,  Minor = 0
 Speed..........: 900 Mhz                LID............: 00000000
             Variation......: IEEE FP, Primary Eligible
  Serial Number..: 0001a968930a1a25               Revision.......:
              Halt Request...: 0               Software Comp..: 7.0
   PALCODE -  Revision Code..: 1.1                Compatibility..: 0
              Max Shared CPUs: 0  Bindings:     * None *  Fastpath:
     FGB0        PKA0        PEA0        EIA0  Features:     Autostart
 - Enabled.     Fastpath  - Selection enabled as Preferred CPU.CPU 1
   State: RUN                CPUDB: 89163480     Handle: 00005E80
     Owner: 000004C8         Current: 000004C8     Partition 0 (EUROS)
        ChgCnt:        1           State: Present, Reassignable
 Process: TEST11            PID: 20201192  Capabilities:        System:
 QUORUM RUN RAD0  Slot Context: 901EC000     CPU     -  State..........:
 RC, PA, PP, CV, PV, PMV, PL                Type...........: Itanium
 Major = 31,  Minor = 0                Speed..........: 900 Mhz
           LID............: 01000000                Variation......:
 IEEE FP                Serial Number..: 000363b4b0c5c862
     Revision.......:                Halt Request...: 0
   Software Comp..: 7.0     PALCODE -  Revision Code..: 1.1
      Compatibility..: 0               Max Shared CPUs: 0  Bindings:
     TEST11         PID = 20201192  Reason: Affinitized to this CPU
 Fastpath:        FGA0        EWA0        PKB0        BG0  Features:
   Autostart - Enabled.     Fastpath  - Selection enabled as Preferred
 CPU.
 $
```

In this example, CPU `1` has user capability `8` enabled (user mask = 80). User capability `8` is set for the initial kernel thread of process `CPUSCHED`, and CPU affinity is set to `1`. Either of these settings forces the initial kernel thread to run only on CPU `1`, as shown by the **`SHOW PROCESS`** command.

Then, the **`SET PROCESS/NOAFFINITY/NOCAPABILITY`** command clears all CPU affinities and user capability requirements for the initial kernel thread of process (`CPUSCHED`), and the binding to CPU `1` disappears.

7. `$ SET PROCESS/SSLOG=(STATE=ON,COUNT=4)`

   This command turns on system service logging with four P2 space buffers, each having a size of $FE00_{16}$ bytes. If the process has SETPRV, CMKRNL, or CMEXEC privilege, argument values are logged.

8. `$ SET PROCESS/SSLOG=(STATE=UNLOAD)`

   This command stops logging and closes the log file.

SET PROMPT

SET PROMPT — Replaces the default DCL prompt ($) with the specified string.

Format

SET PROMPT [=*string*]

Parameter

string

Specifies the new prompt string. The following rules apply:

- All valid ASCII characters can be used.
- No more than 64 characters are allowed.
- To include spaces or lowercase letters, enclose the string in quotation marks (" "). Otherwise, letters are converted automatically to uppercase; leading and trailing spaces are removed.

If you do not specify the string parameter with the **SET PROMPT** command, the default DCL prompt ($) is restored.

Description

The **SET PROMPT** command customizes prompts for your main process or a subprocess.

When a continued command is read from the terminal or an indirect command is read from a command procedure, DCL places an underscore in front of the prompt string.

Qualifier

/CARRIAGE_CONTROL (default)
/NOCARRIAGE_CONTROL

Inserts carriage return and line feed characters before the prompt string. Type the qualifier after the string parameter.

Example

```
$ SET PROMPT ="What's next?"
What's next? SHOW TIME
 19-APR-2001 14:08:58
```

The **SET PROMPT** command in this example replaces the DCL prompt ($) with the phrase `What's next?`. When you see the prompt on your screen, you can enter any DCL command. This example uses the **SHOW TIME** command.

SET PROTECTION/DEFAULT

SET PROTECTION/DEFAULT — Establishes the default protection for files created by the current process.

Format

SET PROTECTION[=(*code*)]/DEFAULT

Parameter

code

Specifies the category of protection to be assigned by default to all files subsequently created by the current process. To override this protection, you must specify an explicit protection using either the **SET SECURITY/PROTECTION** command or the **/PROTECTION** qualifier on the **CREATE** command. If you do not specify a protection code, the current default remains unchanged.

The protection code is made up of the following components:

- Ownership category – system (S), owner (O), group (G), or world (W). Each category can be abbreviated to its first character.
- Access category – Read (R), write (W), execute (E), or delete (D). The access category is assigned to each ownership category. A null access specification means no access.

Note

Use the **SET SECURITY/PROTECTION** command to reset security characteristics of the directory or file.

Example

```
$ SET PROTECTION=(GROUP:RWED,WORLD:R)/DEFAULT
```

The **SET PROTECTION/DEFAULT** command in this example sets the default protection to grant unlimited access to other users in the same group and read (R) access to all users. The default protections for system and owner are not changed.

SET QUEUE

SET QUEUE — Changes the attributes of the specified queue. **Requires manage (M) access to the queue.**

Format

SET QUEUE *queue-name*[:]

Parameter

***queue-name*[:]**

Specifies the name of an execution queue or a generic queue.

Description

After you have created a queue with the **INITIALIZE/QUEUE** command, use the **SET QUEUE** command to change the queue's status or attributes.

The defaults for the **SET QUEUE** qualifiers depend on the qualifiers of the queue when it was initialized. For example, the default for **/JOB_LIMIT** with **INITIALIZE/QUEUE** is 1. However, if the queue you are altering was initialized with a job limit of 3, and if you do not specify the **/JOB_LIMIT** qualifier with the **SET QUEUE** command, the job limit remains at 3 for that queue.

Qualifiers

/BASE_PRIORITY=*n*

Specifies the base process priority at which jobs are initiated from a batch execution queue. The base priority specifier can be any decimal value from 0 to 15.

You also can specify this qualifier for an output execution queue. In this context the **/BASE_PRIORITY** qualifier establishes the base priority of the symbiont process when the symbiont process is created.

/BLOCK_LIMIT=([*lowlim*,]*uplim*)
/NOBLOCK_LIMIT

Limits the size of print jobs that can be processed on an output execution queue. This qualifier allows you to reserve certain printers for certain size jobs. You must specify at least one of the parameters.

The *lowlim* parameter is a decimal number referring to the minimum number of blocks that are accepted by the queue for a print job. If a print job is submitted that contains fewer blocks than the *lowlim*value, the job remains pending until the block limit for the queue is changed. After the block limit for the queue is decreased sufficiently, the job is processed.

The *uplim* parameter is a decimal number referring to the maximum number of blocks that are accepted by the queue for a print job. If a print job is submitted that exceeds this value, the job remains pending until the block limit for the queue is changed. After the block limit for the queue is increased sufficiently, the job is processed.

If you specify only an upper limit for jobs, you can omit the parentheses. For example, **/BLOCK_LIMIT=1000** means that only jobs with 1000 blocks or less are processed in the queue. To specify only a lower job limit, you must use a null string ("") to indicate the upper limit. For example, **/BLOCK_LIMIT=(500,"")** means any job with 500 or more blocks is processed in the queue. You can specify both a lower and upper limit. For example, **/BLOCK_LIMIT=(200,2000)** means that jobs with less than 200 blocks or more than 2000 blocks are not processed in the queue.

The **/NOBLOCK_LIMIT** qualifier cancels the previous block limit setting for that queue.

/CHARACTERISTICS= (*characteristic*[,...])
/NOCHARACTERISTICS

Specifies one or more characteristics for processing jobs on an execution queue. If a queue does not have all the characteristics that have been specified for a job, the job remains pending. If you specify only one characteristic, you can omit the parentheses. Each time you specify the **/CHARACTERISTICS** qualifier, all previously set characteristics are canceled. Only the characteristics specified with the qualifier are established for the queue.

Queue characteristics are installation specific. The characteristic parameter can be either a value from 0 to 127 or a characteristic name that has been defined by the **DEFINE/CHARACTERISTIC** command.

The **/NOCHARACTERISTICS** qualifier cancels any characteristics settings previously established for that queue.

/CLOSE

Prevents jobs from being entered in the queue through **PRINT** or **SUBMIT** commands or as a result of requeue operations. To allow jobs to be entered, use the **/OPEN** qualifier. Whether a queue accepts or rejects new job entries is independent of the queue's state (such as paused, stopped, or stalled). When a queue is marked closed, jobs executing continue to execute and jobs pending in the queue continue to be candidates for execution.

/CPUDEFAULT=*time*

Defines the default CPU time limit for jobs in a batch execution queue. You can specify time as delta time, 0, INFINITE, or NONE. You can specify up to 497 days of delta time.

If the queue does not have a defined CPUMAXIMUM time limit and the value established in the user authorization file (UAF) has a specified CPU time limit of NONE, either the value 0 or the keyword INFINITE allows unlimited CPU time. If you specify NONE, the CPU time value defaults to the value specified either in the UAF or by the **SUBMIT** command (if included). CPU time values must be greater than or equal to the number specified by the system parameter PQL_MCPULM. The time cannot exceed the CPU time limit set by the **/CPUMAXIMUM** qualifier. For information on specifying delta time, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date. For more information on specifying CPU time limits, see the "CPU Time Limit Specifications and Actions" table for the **INITIALIZE/QUEUE** command in the *VSI OpenVMS DCL Dictionary: A-M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_119].

/CPUMAXIMUM=*time*

Defines the maximum CPU time limit for all jobs in a batch execution queue. You can specify time as delta time, 0, INFINITE, or NONE. You can specify up to 497 days of delta time.

The **/CPUMAXIMUM** qualifier overrides the time limit specified in the user authorization file (UAF) for any user submitting a job to the queue. Either the value 0 or the keyword INFINITE allows unlimited CPU time. If you specify NONE, the CPU time value defaults to the value specified either in the UAF or by the **SUBMIT** command (if included). CPU time values must be greater than or equal to the number specified by the system parameter PQL_MCPULM.

For information on specifying delta time, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date. For more information on specifying CPU time limits, see the "CPU Time Limit Specifications and Actions" table for the **INITIALIZE/QUEUE** command in the *VSI OpenVMS DCL Dictionary: A-M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_119].

/DEFAULT=(*option*[,...])
/NODEFAULT

Establishes defaults for certain options of the **PRINT** command. Defaults are specified by the list of options. If you specify only one option, you can omit the parentheses. After you set an option for the queue with the **/DEFAULT** qualifier, you do not have to specify that option in your **PRINT** commands. If you do specify these options in your **PRINT** command, the values specified with the PRINT command override the values established for the queue with the **/DEFAULT** qualifier. Possible options are as follows:

Option	Description
[NO]BURST[=*keyword*]	Controls whether two file flag pages with a burst bar between them are printed preceding output. If you specify the value ALL (default), these flag pages are printed before each file in the job. If you specify the value ONE, these flag pages are printed once before the first file in the job.
[NO]FEED	Specifies whether a form feed is inserted automatically at the end of a page.
[NO]FLAG[=*keyword*]	Controls whether a file flag page is printed preceding output. If you specify the value ALL (default), a file flag page is printed before each file in the job. If you specify the value ONE, a file flag page is printed once before the first file in the job.
FORM=*type*	Specifies the default form for an output execution queue. If a job is submitted without an explicit form definition, this form is used to process the job. If no form type is explicitly specified with the FORM keyword, the system assigns the form "DEFAULT" to the queue. See also the description of the **/FORM_MOUNTED** qualifier.
[NO]TRAILER[=*keyword*]	Controls whether a file trailer page is printed following output. If you specify the value ALL (default), a trailer page is printed with each file in the job. If you specify the value ONE, a trailer page is printed once with the last file in the job.

When you specify the BURST option for a file, the [NO]FLAG option does not add or subtract a flag page from the two flag pages that are printed preceding the file.

For information on establishing mandatory queue attributes, see the description of the **/SEPARATE** qualifier. For information on specifying default queue attributes, see the relevant section in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#_6017QUEUE_MGR].

/DESCRIPTION=*string*
/NODESCRIPTION

Specifies a string of up to 255 characters used to provide operator-supplied information about the queue.

Enclose strings containing lowercase letters, blanks, or other non alphanumeric characters (including spaces) in quotation marks (" ").

The **/NODESCRIPTION** qualifier removes any descriptive text that may have been associated with the queue.

/DISABLE_SWAPPING
/NODISABLE_SWAPPING

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

/ENABLE_GENERIC
/NOENABLE_GENERIC

Specifies whether files queued to a generic queue that does not specify explicit queue names can be placed in this execution queue for processing.

/FORM_MOUNTED=*type*

Specifies the mounted form for an output execution queue.

If no form type is explicitly specified, the system assigns the form "DEFAULT" to the queue.

If the stock of the mounted form does not match the stock of the default form, as indicated by the **/DEFAULT=FORM** qualifier, all jobs submitted to this queue without an explicit form definition enter a pending state and remain pending until the stock of the queue's mounted form is identical to the stock of the form associated with the job.

If a job is submitted with an explicit form and the stock of the explicit form is not identical to the stock of the mounted form, the job enters a pending state and remains pending until the stock of the mounted form of the queue is identical to the stock of the form associated with the job.

To specify the form type, use either a numeric value or a form name that has been defined by the **DEFINE/FORM** command. Form types are installation-specific. You cannot use the **/FORM_MOUNTED** qualifier with the **/GENERIC** qualifier.

/JOB_LIMIT=*n*

Indicates the number of batch jobs that can be executed concurrently from the queue. Specify a number in the range 1 to 65535.

/NO_INITIAL_FF
/NONO_INITIAL_FF (default)

Allows user to set or modify whether a form feed should be sent to a printer device when a queue starts. To suppress the initial form feed, set the queue with **/NO_INITIAL_FF** qualifier.

Setting the queue with **/NONO_INITIAL_FF** qualifier sends a form feed to the output device before printing begins when the queue restarts.

/OPEN

Allows jobs to be entered in the queue through **PRINT** or **SUBMIT** commands or as the result of requeue operations. To prevent jobs from being entered in the queue, use the **/CLOSE** qualifier. Whether a queue accepts or rejects new job entries is independent of the queue's state (such as paused, stopped, or stalled).

/OWNER_UIC=*uic*

Requires OPER (operator) privilege or control (C) access to the queue.

Enables you to change the user identification code (UIC) of the queue. Specify the UIC by using the standard format as described in the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#UIC_SECTION].

/PROTECTION= (*ownership*[:*access*],...)

Requires OPER (operator) privilege to control access to the queue.

Specifies the protection of the queue. The following rules apply:

- Specify the *ownership* parameter as system (S), owner (O), group (G), or world (W).
- Specify the *access* parameter as read (R), submit (S), manage (M), or delete (D). A null access specification means no access.

If you include only one protection code, you can omit the parentheses.

For more information on specifying protection codes, see the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#PROTECT_CODE_DETAILS]. For more information on controlling queue operations through UIC-based protection, see the relevant section in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#SECURITY].

/RAD=*n*
/NORAD

Specifies the RAD number on which to run batch jobs assigned to the queue. The RAD value is validated as a positive integer between 0 and the value returned by the $GETSYI item code, SYI$_RAD_MAX_RADS.

To eliminate the RAD value for a batch queue, use the **/NORAD** qualifier.

Note that when you change the RAD value on a batch queue, the jobs currently in the batch queue are not dynamically updated with the new RAD value. Any executing jobs will complete processing using the original RAD value. Jobs in the pending, holding, or timed execution states will retain the old RAD value on the job; however, when such a job becomes executable, the job is updated with the new RAD value.

RAD is supported on AlphaServer GS series systems and, starting from OpenVMS Version 8.4, support is extended to NUMA-capable IA-64 systems.

/RECORD_BLOCKING
/NORECORD_BLOCKING

Determines whether the symbiont can concatenate (or block together) output records for transmission to the output device. If you specify the **/NORECORD_BLOCKING** qualifier, the symbiont sends each formatted record in a separate I/O request to the output device. For the standard OpenVMS system print symbiont, record blocking can have a significant performance advantage over single-record mode.

/RETAIN[=*option*]
/NORETAIN

Holds jobs in the queue in a retained status after they have executed. The **/NORETAIN** qualifier enables you to reset the queue to the default. Possible options are as follows:

Option	Description
ALL	Holds all jobs in the queue after execution (default).
ERROR	Holds in the queue only jobs that complete unsuccessfully.

/SCHEDULE=[NO]SIZE

Specifies whether pending jobs in an output queue are scheduled for printing based on the size of the job. When the **/SCHEDULE=SIZE** qualifier is in effect, shorter jobs print before longer ones. When the **/SCHEDULE=NOSIZE** qualifier is in effect, jobs are printed in the order they were submitted, regardless of size.

If you enter this command while there are pending jobs in any queue, its effect on future jobs is unpredictable.

/SEPARATE=(*option*[,...])
/NOSEPARATE

Specifies the mandatory queue attributes or job separation options for an output execution queue. Job separation options cannot be overridden by the **PRINT** command.

The job separation options are as follows:

Option	Description
[NO]BURST	Specifies whether two job flag pages with a burst bar between them are printed at the beginning of each job.
[NO]FLAG	Specifies whether a job flag page is printed at the beginning of each job.
[NO]RESET=(*module*[,...])	Specifies one or more device control library modules that contain the job reset sequence for the queue. The specified modules from the queue's device control library (by default SYS$LIBRARY:SYSDEVCTL) are used to reset the device each time a job reset occurs. The RESET sequence occurs after any file trailer and before any job trailer; therefore, all job separation pages are printed when the device is in its RESET state.
[NO]TRAILER	Specifies whether a job trailer page is printed at the end of each job.

When you specify the **/SEPARATE=BURST** qualifier, the [NO]FLAG separation option does not add or subtract a flag page from the two flag pages that are printed preceding the job.

For information on establishing queue attributes that can be overridden, see the description of the **/DEFAULT** qualifier.

For more information on specifying mandatory queue attributes, see the relevant section in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#CHOOSE_OPTIONS].

/WSDEFAULT=*n*

Defines for a batch job a working set default, the default number of physical pages that the job can use.

The value set by this qualifier overrides the value defined in the user authorization file (UAF) of any user submitting a job to the queue.

Specify the value of *n* as a number of 512-byte pagelets on Alpha. Note that the OpenVMS system rounds up this value to the nearest CPU-specific page so that actual amount of physical memory allowed may be larger than the specified amount on Alpha.

If you specify the value 0 or NONE, the working set default value defaults to the value specified in the UAF or by the **SUBMIT** command (if included).

For more information about the way a working set default affects batch jobs, see *Table 6, "Working Set Default, Extent, and Quota Decision"*.

/WSEXTENT=*n*

Defines for the batch job a working set extent, the maximum amount of physical memory that the job can use. The job uses the maximum amount of physical memory only when the system has

excess free pages. The value set by this qualifier overrides the value defined in the user authorization file (UAF) of any user submitting a job to the queue.

Specify the value of *n* as a number of 512-byte pagelets on Alpha. Note that the OpenVMS system rounds up this value to the nearest CPU-specific page so that actual amount of physical memory allowed may be larger than the specified amount on Alpha.

If you specify the value 0 or NONE, the working set extent value defaults to the value specified in the UAF or by the **SUBMIT** command (if included).

For more information about the way a working set extent affects batch jobs, see *Table 6, "Working Set Default, Extent, and Quota Decision"*.

/WSQUOTA=*n*

Defines for a batch job the working set quota, the amount of physical memory that is available to the job. The value set by this qualifier overrides the value defined in the user authorization file (UAF) of any user submitting a job to the queue.

Specify the value of *n* as a number of 512-byte pagelets on Alpha systems. Note that the OpenVMS system rounds up this value to the nearest CPU-specific page so that actual amount of physical memory allowed may be larger than the specified amount on Alpha. If you specify the value 0 or NONE, the working set quota value defaults to the value specified in the UAF or by the **SUBMIT** command (if included).

A working set default size and a working set quota (maximum size) are included in each user record in the system user authorization file (UAF). You can specify a working set default and a working set quota for both individual jobs and for all jobs in a given queue. The decision table (*Table 6, "Working Set Default, Extent, and Quota Decision"*) shows the action taken for different combinations of specifications that involve working set size and working set quota values.

Table 6. Working Set Default, Extent, and Quota Decision

Value Specified by the SUBMIT Command?	Value Specified for the Queue?	Action Taken
No	No	Use the UAF value.
No	Yes	Use the value for the queue.
Yes	Yes	Use the smaller of the two values.
Yes	No	Compare the specified value with the UAF value; use the smaller.

Examples

1.
```
$ INITIALIZE/QUEUE/DEFAULT=BURST/FORM_MOUNTED=LETTER/START SYS$PRINT
   .
   .
   .
$ STOP/QUEUE/NEXT SYS$PRINT
$ SET QUEUE/DEFAULT=BURST/FORM_MOUNTED=MEMO SYS$PRINT
```

In this example, the queue is initialized with the **INITIALIZE/QUEUE** command. The queue has the following attributes: two file flag pages preceding each file in the job and the mounted form `LETTER`. Later the queue is stopped with the **STOP/QUEUE/NEXT** command so that the current

job finishes processing before the queue stops. The **SET QUEUE** command changes the mounted form to MEMO.

2. `$ SET QUEUE/DEFAULT=FORM=LN01_PORTRAIT LN01_PRINT`

   In this example, the **SET QUEUE** command changes the default form to LN01_PORTRAIT for the LN01_PRINT queue.

3. `$ SET QUEUE/CLOSE SYS$BATCH`

   In this example, the batch queue SYS$BATCH is modified to prevent jobs from being entered in this queue.

4. ```
   $ SET/QUEUE/RAD=0 BATCHQ1
   $ SHOW QUEUE/FULL BATCHQ1
   Batch queue BATCHQ1, idle, on QUEBID::
    /BASE_PRIORITY=4 /JOB_LIMIT=3 /OWNER=[SYSTEM]
    /PROTECTION=(S:M,O:D,G:R,W:S) /RAD=0
   ```

   This example modifies the batch queue to run all assigned jobs on RAD 0 of QUEBID. Any new jobs assigned to the queue will run on RAD 0. Jobs already executing on the queue will continue to completion executing on the previous RAD value.

5. ```
   $ SET/QUEUE/NORAD BATCHQ1
   $ SHOW QUEUE/FULL BATCHQ1
   Batch queue BATCHQ1, idle, on QUEBID::
    /BASE_PRIORITY=4 /JOB_LIMIT=3 /OWNER=[SYSTEM]
    /PROTECTION=(S:M,O:D,G:R,W:S)
   ```

   This example eliminates the RAD value for a batch queue.

SET RESTART_VALUE

SET RESTART_VALUE — Assigns a value to the global symbol BATCH$RESTART. This global symbol defines the location at which a batch job is restarted after its execution has been interrupted. The **SET RESTART_VALUE** command is meaningful only in command procedures.

Format

SET RESTART_VALUE=*string*

Parameter

string

Specifies a string of up to 255 characters, which specifies the label at which the batch job should begin executing when it is restarted.

Description

Use the **SET RESTART_VALUE** command in restartable command procedures. (A restartable command procedure is a command procedure that is submitted to run in batch mode with the **SUBMIT/RESTART** command.) **SET RESTART_VALUE** assigns a value to the global symbol BATCH$RESTART, the location at which the batch job should be restarted after its execution has been interrupted.

When writing a restartable command procedure, begin each possible starting point in the command procedure with a label. After the label, use the **SET RESTART_VALUE** command to assign the symbol BATCH$RESTART to equal the label. If the batch job is interrupted by a system crash and is then restarted, the command procedure can resume execution at the label pointed to by BATCH$RESTART.

The **SET RESTART_VALUE** command is used in conjunction with the reserved global symbol $RESTART. $RESTART is a special symbol that is maintained by the system and cannot be deleted. $RESTART has the value TRUE if the batch job has been restarted; otherwise, $RESTART has the value FALSE.

At the beginning of a restartable command procedure, test the value of the reserved global symbol $RESTART. If $RESTART is true, execute a **GOTO** statement using the symbol BATCH$RESTART as the transfer label. If a **SET RESTART_VALUE** command was not executed before the batch job was interrupted, the symbol BATCH$RESTART has no definition and the batch job should restart from the beginning.

Most of your process environment is not maintained when the system fails. The only symbols maintained across a system failure are $RESTART and BATCH$RESTART; therefore, you should redefine any symbols or process logical names used in your command procedure after each **SET RESTART_VALUE** command.

If a command procedure has **SET RESTART_VALUE** commands in it, but you want the job to restart at the beginning, enter the **SET ENTRY/NOCHECKPOINT** command to delete the global symbol BATCH$RESTART.

Example

```
$ IF $RESTART THEN GOTO 'BATCH$RESTART'
  .
  .
  .
$ FIRSTPART:
$ SET RESTART_VALUE = FIRSTPART
$ RUN PART1
  .
  .
  .
$ SECONDPART:
$ SET RESTART_VALUE = SECONDPART
$ RUN PART2
  .
  .
  .
```

In this example, the first command states that, if `$RESTART` is true, the procedure is to jump to the value contained in `BATCH$RESTART`. (`$RESTART` is true only if the job has been executed before, that is, the job is being rerun after a crash or after having been requeued.)

The first **SET RESTART_VALUE** command assigns the label `FIRSTPART` to be equal to the symbol `BATCH$RESTART`. The next line contains the command to run `PART1.EXE`.

The second **SET RESTART_VALUE** command assigns the label `SECONDPART` to be equal to the symbol `BATCH$RESTART`. The last line shown contains the command to run `PART2.EXE`.

When the job is first submitted using the **SUBMIT/RESTART** command, the value of `$RESTART` is FALSE, so the **IF** expression is ignored. If the job is stopped during the run of `PART1.EXE`, the

value of BATCH$RESTART is FIRSTPART. When the job is restarted, the value of $RESTART is TRUE; therefore, the **IF** expression is processed and transfers control to the FIRSTPART label in the procedure. PART1.EXE is rerun.

If the job is stopped during the run of PART2.EXE, the value of BATCH$RESTART is SECONDPART. When the job is restarted, the value of $RESTART is TRUE. In this instance, the **IF – GOTO** command transfers control to the SECONDPART label in the procedure so that PART2.EXE can be run. PART1.EXE is not rerun.

SET RIGHTS_LIST

SET RIGHTS_LIST — Allows users to modify the process or system rights list. You must specify either the **/DISABLE** or the **/ENABLE** qualifier with the **SET RIGHTS_LIST** command.

Format

SET RIGHTS_LIST ***id-name*****[,...]**

Parameter

id-name**[,...]**

Specifies identifiers to be added to or removed from the process or system rights list. The *id-name* parameter is a string of 1 to 31 alphanumeric characters, underscores (_), and dollar signs ($); each name must contain at least one nonnumeric character.

Description

The **SET RIGHTS_LIST** command modifies identifiers in your current process rights list, the rights list of another process on the system, or the system rights list. Use the following guidelines to determine which privileges are required for each case:

- Adding new identifiers or modifying existing identifiers in your process rights list that do not have the Dynamic attribute requires CMKRNL (change mode to kernel) privilege.
- Modifying the rights list of other processes on the system requires CMKRNL privilege and either GROUP or WORLD privilege.
- Modifying the system rights list requires both CMKRNL and SYSNAM (system logical name) privileges.
- Adding or removing more than ten identifiers using the **/ENABLE** qualifier or the **/DISABLE** qualifier in a single command invocation requires CMKRNL privilege. You must specify either the **/DISABLE** or the **/ENABLE** qualifier with the **SET RIGHTS_LIST** command.

This command can also be used to add attributes to existing identifiers.

Qualifiers

/ATTRIBUTES=(*keyword*[,...])

Specifies attributes to be associated with the identifiers. Attributes may be added to new or existing identifiers. The following are valid keywords:

Keyword	Description
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**.
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 **/ATTRIBUTE=NORESOURCE**.

The default uses the current processes in the rights database. Use the command **SHOW PROCESS/RIGHTS** to see the attributes of the current process.

/DISABLE

Removes the identifiers from the process or system rights list. You cannot use the **/DISABLE** qualifier with the **/ENABLE** qualifier. Note that removing more than ten identifiers in a single command invocation requires CMKRNL privilege.

/ENABLE

Adds the identifiers to the process or system rights list. You cannot use the **/ENABLE** qualifier with the **/DISABLE** qualifier. Note that adding more than ten identifiers in a single command invocation requires CMKRNL privilege.

/IDENTIFICATION=*pid*

Specifies the process identification (PID) value of the process whose rights list is to be modified. The PID is assigned by the system when the process is created. When you specify a PID, you can omit the leading zeros.

If you specify the **/IDENTIFICATION** qualifier, you cannot use the **/PROCESS** qualifier. By default, if neither the **/IDENTIFICATION** nor the **/PROCESS** qualifier is specified, the current process is assumed. You cannot use the **/IDENTIFICATION** qualifier with the **/SYSTEM** qualifier.

/PROCESS[=*process-name*]

Specifies the name of the process whose rights list is to be modified. The process name can contain from 1 to 15 alphanumeric characters.

If you specify the **/PROCESS** qualifier, you cannot use the **/IDENTIFICATION** qualifier. By default, if neither the **/PROCESS** nor the **/IDENTIFICATION** qualifier is specified, the current process is assumed.

You cannot use the **/PROCESS** qualifier with the **/SYSTEM** qualifier.

/SYSTEM

Specifies that the desired operation (addition or removal of an identifier)be performed on the system rights list. You cannot use the **/SYSTEM** qualifier with the **/PROCESS** or the **/IDENTIFICATION** qualifier.

Examples

1. `$ SET RIGHTS_LIST/ENABLE/ATTRIBUTES=RESOURCE MARKETING`

   The **SET RIGHTS_LIST** command in this example adds the `MARKETING` identifier to the process rights list of the current process. Specifying the `RESOURCE` attribute allows holders of the `MARKETING` identifier to charge resources to it.

2. ```
   $ SET RIGHTS_LIST/ENABLE/SYSTEM PHYSICS101
   %SYSTEM-F-NOPRIV, insufficient privilege or object
    protection violation
   $ SET PROCESS/PRIVILEGES=(CMKRNL,SYSNAM)
   $ SET RIGHTS_LIST/ENABLE/SYSTEM PHYSICS101
   ```

   The **SET RIGHTS_LIST** command in this example adds the `PHYSICS101` identifier to the system rights list. You must have both the `CMKRNL` (change mode to kernel) and `SYSNAM` (system logical name) privileges to modify the system rights list.

SET RMS_DEFAULT

SET RMS_DEFAULT — Defines default values for multiblock and multibuffer counts, network transfer sizes, prolog level, extend quantity, and the query lock option used by OpenVMS Record Management Services (RMS) for file operations. If you set the default value for either the multiblock count or the multibuffer count at 0, RMS tries to use the process default value or the system default value, in that order. If these are set at 0, RMS uses a default value of 1. Defaults are set for sequential, relative, or indexed file organizations on a process-only basis, unless a systemwide basis is requested.

Format

SET RMS_DEFAULT

Parameters

None.

Description

Multiblocking and multibuffering of file operations can enhance the speed of I/O operations with RMS. The defaults set with the **SET RMS_DEFAULT** command are applied for all file operations that do not specify explicit multiblock or multibuffer counts.

For more information on multiblock and multibuffer operations, see the *VSI OpenVMS System Services Reference Manual: A–GETUAI* [https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-a-getuai/].

For indexed files, the **SET RMS_DEFAULT** command defines default prolog level options.
   ```

For sequential files, the **SET RMS_DEFAULT** command defines default extensions. If a default extension is not specified in your program, the process or system default is used.

For network operations, the **SET RMS_DEFAULT** command defines network buffer sizes for transfer.

Qualifiers

/BLOCK_COUNT=*count*

Specifies a default multiblock count (0 to 255) for record I/O operations *only*, where *count* is the number of blocks to be allocated for each I/O buffer.

For more information on multiblock count, see the description of the RAB$B_MBC in the *VSI OpenVMS Record Management Services Reference Manual* [https://docs.vmssoftware.com/vsi-openvms-record-management-services-reference-manual/#_100_RABB_MBCFIELD].

/BUFFER_COUNT=*count*

Specifies a default multibuffer count (0 to 255) for local file operations, where *count* is the number of buffers to be allocated. If you use the **/SYSTEM** qualifier to extend the default value systemwide, the maximum default value is 127.

When you use the **/BUFFER_COUNT** qualifier, you can use the **/DISK**, **/INDEXED**, **/MAGTAPE**, **/RELATIVE**, **/SEQUENTIAL**, and **/UNIT_RECORD** qualifiers to specify the types of file for which the default is to be applied. If the **/BUFFER_COUNT** qualifier is specified without any of these qualifiers, the **/SEQUENTIAL** qualifier is assumed. If file type is not specified, the default is applied to sequential files.

For more information on multibuffer count, see the description of the RAB$B_MBF field in the *VSI OpenVMS Record Management Services Reference Manual* [https://docs.vmssoftware.com/vsi-openvms-record-management-services-reference-manual/#_101_RABB_MBFFIELD].

/CONTENTION_POLICY=*keyword* [/SYSTEM]

Allows you to specify fairness under high contention conditions for write-shared files at the process or system level. Note that selecting this option may noticeably increase locking overhead.

This option is disabled by default.

In a mixed cluster environment with high contention for specific buckets, it is possible for accesses to write-shared files on nodes using read-mode global bucket locking to dominate access to a bucket. Nodes without this support might be denied timely access to the bucket.

Additionally, it is possible to observe comparable behavior on all OpenVMS versions when dealing with accesses to write-shared files without global buffers enabled. A similar fairness issue between lock conversions and new lock requests may be observed in which the new lock requests may remain ungranted for an extended period of time. Use the **/CONTENTION_POLICY** qualifier to specify fairness under high contention conditions.

The following table lists valid PROCESS keywords (**/SYSTEM** not specified):

Keyword	Description
NEVER	Never use the higher overhead option to improve fairness for any write-shared files accessed by this process; minimal overhead.

Keyword	Description
SOMETIMES	Use this option for fairer bucket access (but higher overhead) to any write-shared files with global buffers enabled accessed by this process.
ALWAYS	Use this option for fairer bucket access (but higher overhead) to all write-shared files accessed by this process.
SYSTEM_DEFAULT (default)	Use system setting. Note that this keyword is disallowed with **/SYSTEM**.

The following table lists valid SYSTEM keywords (**/SYSTEM** also specified):

Keyword	Description
NEVER (default)	Never use the higher overhead option to improve fairness for any write-shared files accessed on the system; minimal overhead.
SOMETIMES	Use this option for fairer bucket access (but higher overhead) to any write-shared files with global buffers enabled accessed on the system.
ALWAYS	Use this option for fairer bucket access (but higher overhead) to all write-shared files accessed on the system.

/DISK

Applies the specified defaults to disk file operations. Values applied using the **/SEQUENTIAL** qualifier take precedence over values applied using the **/DISK** qualifier.

/EXTEND_QUANTITY=*n*

Specifies the number of blocks *n* to extend a sequential file where *n* can range from 0 to 65535. If you do not specify the **/EXTEND_QUANTITY** qualifier, RMS calculates its own extend value. The **/EXTEND_QUANTITY** qualifier value is used when the program does not specify an extent quantity explicitly.

/INDEXED

Applies the multibuffer default to indexed file operations.

/MAGTAPE

Applies the multibuffer default to magnetic tape operations. Values applied using the **/SEQUENTIAL** qualifier take precedence over values applied using the **/MAGTAPE** qualifier.

/NETWORK_BLOCK_COUNT=*count*

Specifies a default block count (0 to 127) for network access to remote files, where count is the number of blocks to be allocated for each I/O buffer.

For remote file access, the buffer size is negotiated between RMS and the remote system's file access listener (FAL) with the smaller of the two sizes being selected.

Thus, the **/NETWORK_BLOCK_COUNT** value places an upper limit on the network buffer size that is used. It also places an upper limit on the largest record that may be transferred to or from a remote file. The largest record must be less than or equal to 512×network_block_count. The network_block_count is in block units and the record is in byte units.)

If you omit the value or specify a value of 0, RMS uses the systemwide block count value. If this value is also 0, RMS uses a size of one block.

/PROLOG=*n*

Specifies a default prolog level for indexed files where acceptable values for *n* are 0, 2, or 3. If 0 (default) is specified, RMS sets an appropriate prolog level.

/QUERY_LOCK=*keyword* [/SYSTEM]

Enables or disables query locking at the process and system levels.

The following table lists valid PROCESS keywords (**/SYSTEM** not specified):

Keyword	Description
DISABLE	Disable query locking for the duration of any file opens done by images within the process for any read ($GET or $FIND) record operations that have both RAB$V_NLK (no lock) and RAB$V_RRL (read-regardless) set in the RAB$L_ROP field of the RAB (Record Access Block). Disabling query locking directs RMS not to make any calls to the lock manager for a read record operation just to find out whether a record is locked by another stream.
ENABLE	Use RMS's default record locking for images run within this process.
SYSTEM_DEFAULT (default)	Use system setting. Note that this keyword is disallowed with **/SYSTEM**.

The following table lists valid SYSTEM keywords (**/SYSTEM** also specified):

Keyword	Description
DISABLE	Disable query locking for the duration of any file opens done by images on the system for any read ($GET or $FIND) record operations that have both RAB$V_NLK (no lock) and RAB$V_RRL (read-regardless) set in the RAB$L_ROP field of the RAB (Record Access Block). Disabling query locking directs RMS not to make any calls to the lock manager for a read record operation just to find out whether a record is locked by another stream.
ENABLE (default)	Use RMS's default record locking for images run within this process.

There is no system parameter associated with the QUERY_LOCK setting.

/RELATIVE

Applies the multibuffer default to relative file operations.

/SEQUENTIAL (default)

Applies the multibuffer default to sequential file operations. The **/SEQUENTIAL** qualifier overrides values applied using either the **/DISK**, the **/MAGNETIC TAPE**, or the **/UNIT RECORD** qualifier.

The **/SEQUENTIAL** qualifier is the default if you do not specify either the **/RELATIVE** or the **/INDEXED** qualifier.

/SYSTEM

Requires CMKRNL (change mode to kernel) privilege.

Applies specified defaults on a systemwide basis to all file operations.

Systemwide settings are implemented as system parameters, with the exception of **/QUERY_LOCK**. For a systemwide setting to survive a system reboot, it must be written to the system parameter file by using SYSGEN.

/UNIT_RECORD

Applies the multibuffer default to file operations on unit record devices. Values applied using the **/SEQUENTIAL** qualifier take precedence over values applied using the **/UNIT_RECORD** qualifier.

Examples

1. ```
   $ SET RMS_DEFAULT/BLOCK_COUNT=24
   $ SHOW RMS_DEFAULT

           MULTI-  |              MULTIBUFFER COUNTS             | NETWORK
           BLOCK   | Indexed Relative         Sequential         |  BLOCK
           COUNT   |                   Disk Magtape  Unit Record |  COUNT
   Process  24     |    0        0       0     0          0       |    0
   System   16     |    0        0       0     0          0       |    8

            Prolog    Extend Quantity     QUERY_LOCK
   Process    0              0            System
   System     0              0            Enabled

            CONTENTION_POLICY
   Process      System
   System       Never
   ```

   The **SET RMS_DEFAULT** command in this example sets the multiblock count for disk file I/O at 24 for user programs that do not set the multiblock count explicitly. The command applies only to the current process.

2. ```
   $ SET RMS_DEFAULT/BUFFER_COUNT=8/MAGTAPE
   $ SHOW RMS_DEFAULT

           MULTI-  |              MULTIBUFFER COUNTS             | NETWORK
           BLOCK   | Indexed Relative         Sequential         |  BLOCK
           COUNT   |                   Disk Magtape  Unit Record |  COUNT
   Process   0     |    0        0       0     8          0       |    0
   System   16     |    0        0       0     0          0       |    8

            Prolog    Extend Quantity     QUERY_LOCK
   Process    0              0            System
   System     0              0            Enabled

            CONTENTION_POLICY
   Process      System
   System       Never
   ```

   The **SET RMS_DEFAULT** command in this example defines the default multibuffer count for I/O magnetic tape operations at `8`.

3.
```
$ SET RMS_DEFAULT/BUFFER_COUNT=7/NETWORK_BLOCK_COUNT=16/SYSTEM
$ SHOW RMS_DEFAULT

        MULTI- |              MULTIBUFFER COUNTS             | NETWORK
        BLOCK  | Indexed Relative         Sequential         |  BLOCK
        COUNT  |                   Disk Magtape  Unit Record |  COUNT
Process   0    |    0        0      0     0          0      |    0
System   16    |    0        0      7     7          0      |   16

         Prolog    Extend Quantity     QUERY_LOCK
Process    0              0            System
System     0              0            Enabled

         CONTENTION_POLICY
Process      System
System       Never
```

   The **SET RMS_DEFAULT** command in this example defines the systemwide default multibuffer count at 7 for all sequential file operations on disk, magnetic tape, and unit record devices. The command also sets the network block count at 16.

4.
```
$ SET RMS_DEFAULT/EXTEND=50/INDEXED/BUFFER_COUNT=5
$ SHOW RMS_DEFAULT

        MULTI- |              MULTIBUFFER COUNTS             | NETWORK
        BLOCK  | Indexed Relative         Sequential         |  BLOCK
        COUNT  |                   Disk Magtape  Unit Record |  COUNT
Process   0    |    5        0      0     0          0      |    0
System   16    |    0        0      0     0          0      |    8

         Prolog    Extend Quantity     QUERY_LOCK
Process    0             50            System
System     0              0            Enabled

         CONTENTION_POLICY
Process      System
System       Never
```

   The **SET RMS_DEFAULT** command in this example sets the default multibuffer count for I/O operations on indexed files at 5. It also defines the default extend quantity for sequential I/O operations at 50 blocks. These defaults apply only to disk operations for user programs that do not set the multiblock count explicitly. These defaults are limited to the current process.

5.
```
$ SET RMS_DEFAULT/QUERY_LOCK=DISABLE
$ SHOW RMS_DEFAULT

        MULTI- |              MULTIBUFFER COUNTS             | NETWORK
        BLOCK  | Indexed Relative         Sequential         |  BLOCK
        COUNT  |                   Disk Magtape  Unit Record |  COUNT
Process   0    |    0        0      0     0          0      |    0
System   16    |    0        0      0     0          0      |    8

         Prolog    Extend Quantity     QUERY_LOCK
Process    0              0            Disabled
System     0              0            Enabled

         CONTENTION_POLICY
Process      System
```

```
System       Never
```

The **SET RMS_DEFAULT** command in this example requests that query locking be disabled for the duration of any file opens done by images within this process for any read ($GET or $FIND) record operations that have both RAB$V_NLK (no lock) and RAB$V_RRL (read-regardless) set in the RAB$L_ROP field of the RAB (Record Access Block).

SET ROOT

SET ROOT — Defines the POSIX root for the system.

Format

SET ROOT *device-name*:*directory-spec*

Parameter

device-name*:*directory-spec

Defines the root pathname.

Description

In POSIX pathname processing mode, RMS and the C Run-Time Library treat the leading slash (/) of a pathname as referring to the defined root. By default, the root is SYS$SYSDEVICE:[PSX$ROOT]. The root definition does not persist across a reboot. The **SET ROOT** commands requires SYSNAM privilege and has the following qualifier:

Qualifier

/LOG (default)
/NOLOG

Controls whether the **SET ROOT** command displays a success indication after the root definition is set.

Example

```
$ SHOW ROOT
DISK$ALPHASYS:[000000]
$ SET ROOT DISK$ALPHASYS:[PSX$ROOT]
%SET-I_SXROOSET, system POSIX root set to DISK$ALPHASYS:[PSX$ROOT]
$ SHOW ROOT
DISK$ALPHASYS:[PSX$ROOT]
$
```

The **SHOW ROOT** command in this example displays the current root path. The **SET ROOT** command changes these defaults, and the next **SHOW ROOT** command displays the new default root.

SET SECURITY

SET SECURITY — Modifies the security profile of an object.

Format

SET SECURITY ***object-name***

Parameter

object-name

Specifies the name of an object, such as a file or device, whose security profile is to be modified. An object is identified by an object name and a class name. The default class name is FILE.

An object name of the FILE class (explicitly or implicitly specified) can include an asterisk (*) or a percent sign (%) wildcard character, but wildcard characters are not allowed in any class other than FILE. **SET SECURITY** does not operate on remote files and devices, alias directory entries, or directory names in UIC format (for example, [14,5]).

Description

The **SET SECURITY** command modifies the security profile of an object. Such a profile contains the following elements:

- An access control list editor (ACL editor).
- A protection code, which defines access to objects based on the categories of system, owner, group, and world.
- An owner. The system uses the owner element to interpret the protection code.

There are three different ways to use the command:

- You can provide new values explicitly with the qualifiers **/ACL**, **/PROTECTION**, and **/OWNER**. (For extensive ACL work, use **/EDIT** to invoke the ACL editor.)
- You can copy from another object's profile with the **/LIKE** qualifier.
- If the object is of the FILE class, you can reset its profile to the default setting with the **/DEFAULT** qualifier.

To modify a security profile, you need control access to the object. An ACL grants control access explicitly whereas a protection code grants it implicitly to anyone belonging to the owner or system categories. If an object profile is modified while the object is being accessed, the existing access is unaffected.

The following table identifies object classes and the access types they support:

Object Class	**Access Types**
COMMON_EVENT_FLAG_CLUSTER	Associate, Delete
DEVICE	Read, Write, Physical, Logical
FILE (including directory file)	Read, Write, Execute, Delete
GROUP_GLOBAL_SECTION	Read, Write, Execute
ICC_ASSOCIATION[1]	Open, Access
LOGICAL_NAME_TABLE	Read, Write, Create, Delete

Object Class	Access Types
QUEUE	Read, Submit, Manage, Delete
RESOURCE_DOMAIN	Read, Write, Lock
SECURITY_CLASS	Read, Write, Logical I/O, Physical I/O
SYSTEM_GLOBAL_SECTION	Read, Write, Execute
VOLUME	Read, Write, Create, Delete

[1]The class ICC_ASSOCIATION has special semantics: there are both permanent and temporary objects for this class. Permanent objects are created using the command procedure SYS$MANAGER:ICC$CREATE_SECURITY_OBJECT.COM. The **SET SECURITY** command applies to both permanent and temporary ICC_ASSOCIATION security objects.

The *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/] provides a full explanation of protected objects and how to modify them.

Table 7, "SET SECURITY Qualifier Categories" shows the qualifier categories for the **SET SECURITY** command. The explanations for the qualifiers following *Table 7, "SET SECURITY Qualifier Categories"* occur in alphabetical order.

Table 7. SET SECURITY Qualifier Categories

General Qualifiers	ACL-Modifying Qualifiers	Security Class Qualifier	File-Specific Qualifiers	Transfer Qualifiers
/ACL	/AFTER	/PROFILE	/BACKUP	/COPY_ATTRIBUTE
/CLASS	/DELETE		/BEFORE	/LIKE
/LOG	/EDIT		/BY_OWNER	
/OWNER	/REPLACE		/CONFIRM	
/PROTECTION			/CREATED	
			/DEFAULT	
			/EXCLUDE	
			/EXPIRED	
			/MODIFIED	
			/SINCE	
			/STYLE	

Qualifiers

/ACL[=(*ace*[,...])]

Identifies one or more access control list entries (ACEs) to add, replace, or delete. Enclose each ACE in parentheses and separate multiple ACEs by commas (,). The most common type of entry, the Identifier ACE, has the format (IDENTIFIER=*identifier*, ACCESS=*access-type*(+...)). By default, **SET SECURITY** adds an ACE to the top of the ACL. This behavior changes when you include one of the positional qualifiers: **/AFTER**, **/DELETE**, or **/REPLACE**. See the discussion of ACL

ordering in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#ACL_DETAILS].

/AFTER=*ace*

Positions all ACEs specified with the **/ACL** qualifier after the ACE named with the **/AFTER** qualifier.

/BACKUP

Modifies the time value provided with the **/BEFORE** or the **/SINCE** qualifier. The **/BACKUP** qualifier selects files according to the date of their most recent backup (rather than by the creation, expiration, or modification date). By default, **SET SECURITY** selects files according to their creation date.

/BEFORE[=*time*]

Selects only those files dated prior to the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify the **/CREATED** or the **/MODIFIED** qualifier to indicate the time attribute to be used as the basis for selection. The **/CREATED** qualifier is the default.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/BY_OWNER[=*uic*]

Selects files whose owner's UIC matches the UIC specified. The default UIC is that of the current process.

/CLASS=*class-name*

Specifies the class of the object whose profile is to be modified. By default, the command assumes the object class is FILE.

/CONFIRM

Controls whether **SET SECURITY** prompts for verification before performing the operation. Valid responses are YES, NO, TRUE, and FALSE. Answers are not case sensitive and can be abbreviated to one letter. To stop processing the command at any point, type QUIT or press **Ctrl/Z**. To cancel the verification procedure but to proceed with the command, type ALL.

/COPY_ATTRIBUTE=(*keyword*[,...])

Specifies a subset of security elements to transfer from a source object to a target object. Valid keywords include the following:

Keyword	Description
ALL (default)	Copy all security elements
ACL	Copy the access control list
OWNER	Copy the owner
PROTECTION	Copy the protection code

Use the **/COPY_ATTRIBUTE** qualifier with the **/LIKE** qualifier. For example, you can create an ACL for an object and then copy its ACL to new objects.

/CREATED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/CREATED** qualifier selects files according to the date they were created (rather than by the backup, expiration, or modification date). By default, **SET SECURITY** selects files according to their creation date.

/DELETE[=ALL]

Deletes ACEs according to the following rules:

- The expression **/ACL=*aces*/DELETE** deletes the named ACEs.
- The expression **/ACL/DELETE** deletes all unprotected ACEs.
- The expression **/ACL/DELETE=ALL** deletes all ACEs including protected ACEs.
- The expression **/ACL=*aces*/DELETE=ALL** deletes the existing ACL (if any) and create a new ACL with the ACEs specifies on the **/ACL** qualifier.

/DEFAULT

Regenerates the security profile of a file. The default qualifier changes the protection code, the ACL, and the owner elements of a file to what it would be if the file had just been created. The profile is recreated according to the following rules:

- The protection code is propagated from the default protection ACE on the directory (if one exists), or else it is propagated from the process default.
- The ACL is propagated from the parent directory for those ACEs that have the default option.
- The owner is set to the owner of the parent directory.

With subdirectory files, **SET SECURITY** assigns the owner, protection, and ACL elements of the parent directory.

SET SECURITY does not copy any ACE on the source object if the ACE holds the nopropagate attribute nor does it change any ACE on the target object if the ACE holds the protected attribute. To apply new elements to all versions of the file, specify ;* in the object name. See the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#ADJUST_DEFAULTS] for more information on propagation rules.

/EDIT

Invokes the access control list editor (ACL editor) and allows you to modify an ACL interactively. The ACL editor does not allow the asterisk (*) and the percent sign (%) wildcard characters in an object name. You must specify the object whose ACL you are editing.

The **/EDIT** qualifier must be the first qualifier on the command line; other qualifiers can include **/CLASS** and, if the class is SECURITY_CLASS, you can include the **/PROFILE** qualifier. Whenever an object does not belong to the FILE class, you also need to specify **/CLASS**.

See the section on the ACL editor in the *VSI OpenVMS System Management Utilities Reference Manual* [https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#ACL_PART] for more information.

/EXCLUDE=(*filespec*[,...])

Excludes the specified files from the **SET SECURITY** operation. You can include a directory, but not a device, in the file specification. You cannot use relative version numbers to exclude a specific version.

/EXPIRED

Modifies the time specified with the **/BEFORE** or the **/SINCE** qualifier. The **/EXPIRED** qualifier selects files according to their expiration dates rather than by the backup, creation, or modification date. (The expiration date is set with the **SET FILE/EXPIRATION_DATE** command.) By default, files are selected according to their creation date.

/LIKE=(NAME=*source-object-name* [,CLASS=*source-object-class*] [,PROFILE=TEMPLATE=*template-name*])

Identifies the object from which **SET SECURITY** should copy security elements. The **/LIKE** qualifier replaces an object's existing elements with those of the source object. Nopropagate ACEs are not transferred and protected ACEs on the target object are not deleted. Use the **/COPY_ATTRIBUTE** qualifier with the **/LIKE** qualifier to copy an object's elements. See the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/] for information about the special handling of protected and nopropagate ACEs.

The object class of the source object defaults to the class of the target object. When the **/CLASS** qualifier is omitted, the CLASS keyword defaults to FILE.

The PROFILE keyword applies to security class objects. It identifies which template of the security class you want to copy and modify. See the **/PROFILE** qualifier for more information.

/LOG

Controls whether the **SET SECURITY** command displays the name of the object that has been modified by the command. The qualifier is invalid with the **/EDIT** qualifier.

/MODIFIED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/MODIFIED** qualifier selects files according to the dates on which they were last modified, rather than by the backup, creation, or expiration date. By default, files are selected according to their creation date.

/OWNER=*identifier*

Requires GRPPRV (group privilege) to set the owner to another member of the same group. Requires SYSPRV (system privilege) to set the owner to any user identification code (UIC) outside your group.

Modifies the owner element of an object. Specify the user identification code (UIC) or general identifier in the standard format. Modifying the owner element of a file usually requires privileges. See the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#UIC_SECTION] for more information.

/PROFILE=TEMPLATE[=*template-name*]

Identifies which template profile of a security class object you want to modify. All object classes except FILE have at least one template profile. These template profiles define the basis of the profile

of new objects. Use the DCL command **SHOW SECURITY/CLASS=SECURITY_CLASS** to display template names. When no value is given for *template-name*, **SET SECURITY** uses the template named DEFAULT.

Include the **/CLASS=SECURITY_CLASS** qualifier to identify which profile you want to modify.

/PROTECTION=(*ownership*[:*access*][,...])

Cannot be used to change the protection on a file by using DECnet software.

Modifies the protection code of an object. The protection code defines the type of access allowed to users, based on their relationship to the object's owner.

Specify the *ownership* parameter as system (S), owner (O), group (G), or world (W).

Access types are class specific and are shown in the Description section. For *access*, use the first letter of the access name. The Examples section provides you with models of protection codes.

/REPLACE=(*ace*[,...])

Eliminates entries listed with the **/ACL** qualifier and adds entries listed with the **/REPLACE** qualifier. **SET SECURITY** inserts the entries listed with **/REPLACE** in the position of the last deleted ACE.

/SECRECY

Reserved for use by VSI.

/SINCE[=*time*]

Selects only those files dated on or after the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, JOB_LOGIN, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify the **/CREATED** or the **/MODIFIED** qualifier to indicate the time attribute to be used as the basis for selection. The **/CREATED** qualifier is the default.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/STYLE=*keyword*

Specifies the file name format for display purposes.

The valid keywords for this qualifier are CONDENSED and EXPANDED. Descriptions are as follows:

Keyword	Description
CONDENSED (default)	Displays the file name representation of what is generated to fit into a 255-length character string. This file name may contain a DID or FID abbreviation in the file specification.
EXPANDED	Displays the file name representation of what is stored on disk. This file name does not contain any DID or FID abbreviations.

The keywords CONDENSED and EXPANDED are mutually exclusive. This qualifier specifies which file name format is displayed in the output message, along with the confirmation if requested.

File errors are displayed with the CONDENSED file specification unless the EXPANDED keyword is specified.

See the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#FILE_ACCESS] for more information.

/SYMLINK=*keyword*

The valid keywords for this qualifier are [NO]WILDCARD and [NO]ELLIPSIS. Descriptions are as follows:

Keyword	Description
WILDCARD	Indicates that symlinks are enabled during wildcard searches.
NOWILDCARD	Indicates that symlinks are disabled during directory wildcard searches.
ELLIPSIS	Equivalent to WILDCARD (included for command symmetry).
NOELLIPSIS	Indicates that symlinks are matched for all wildcard fields except for ellipsis.

If the file named in the **SET SECURITY** command is a symlink, the command operates on the symlink itself.

Examples

1.
```
(WIDE)$  SHOW SECURITY LNM$GROUP /CLASS=LOGICAL_NAME_TABLE

LNM$GROUP object of class LOGICAL_NAME_TABLE

     Owner: [SYSTEM]
     Protection: (System: RWCD, Owner: R, Group: R, World: R)
     Access Control List:
          (IDENTIFIER=[USER,VARANESE],ACCESS=CONTROL)

$  SET SECURITY LNM$GROUP /CLASS=LOGICAL_NAME_TABLE -
_$       /ACL=((IDENTIFIER=CHEKOV,ACCESS=CONTROL), -
_$             (IDENTIFIER=WU,ACCESS=READ+WRITE)) -
_$        /DELETE=ALL -
_$        /PROTECTION=(S:RWCD, O:RWCD, G:R, W:R)

$  SHOW SECURITY LNM$GROUP /CLASS=LOGICAL_NAME_TABLE

LNM$GROUP object of class LOGICAL_NAME_TABLE

     Owner: [SYSTEM]
     Protection: (System: RWCD, Owner: RWCD, Group: R, World: R)
     Access Control List:
          (IDENTIFIER=[USER,CHEKOV],ACCESS=CONTROL)
          (IDENTIFIER=[USER,WU],ACCESS=READ+WRITE)
```

This example shows how to make a straightforward change to the security elements of an object. The first **SHOW SECURITY** command displays the current settings of the LNM$GROUP logical name

table. The **SET SECURITY** command resets the ACL to allow control access for user Chekov, and to allow read and write access for user Wu. Note that without the **/DELETE=ALL** qualifier, these ACEs would have been added to the existing ACL rather than superseding it. The protection is also changed to allow read, write, create, and delete access for the owner. The last command displays the results of the changes.

2.
```
$  SHOW SECURITY LNM$GROUP /CLASS=LOGICAL_NAME_TABLE

LNM$GROUP object of class LOGICAL_NAME_TABLE

     Owner: [SYSTEM]
     Protection: (System: RWCD, Owner: R, Group: R, World: R)
     Access Control List:
          (IDENTIFIER=[USER,FERNANDEZ],ACCESS=CONTROL)

$  SHOW SECURITY LNM$JOB /CLASS=LOGICAL_NAME_TABLE

LNM$JOB object of class LOGICAL_NAME_TABLE

     Owner: [USER,WEISS]
     Protection: (System: RWCD, Owner: RWCD, Group, World)
     Access Control List:  <empty>


$  SET SECURITY LNM$JOB /CLASS=LOGICAL_NAME_TABLE -
_$      /LIKE=(NAME=LNM$GROUP, CLASS=LOGICAL_NAME_TABLE) -
_$      /COPY_ATTRIBUTES=PROTECTION
$  SET SECURITY LNM$JOB /CLASS=LOGICAL_NAME_TABLE -
_$      /ACL=(IDENTIFIER=FERNANDEZ, ACCESS=READ)

$  SHOW SECURITY LNM$JOB /CLASS=LOGICAL_NAME_TABLE

LNM$JOB object of class LOGICAL_NAME_TABLE

     Owner: [USER,WEISS]
     Protection: (System: RWCD, Owner: R, Group: R, World: R)
     Access Control List:
          (IDENTIFIER=[USER,FERNANDEZ],ACCESS=READ)
```

This example shows how to copy security access information from one object to another and, at the same time, set some elements explicitly. The first **SHOW SECURITY** commands display the current settings for the LNM$GROUP and LNM$JOB logical name tables. The **SET SECURITY** command copies the protection code from the LNM$GROUP logical name table to the LNM$JOB logical name table and adds an ACE to allow read access to another user. The final **SHOW SECURITY** command shows the effect of the changes.

3.
```
$  SHOW SECURITY SECURITY_CLASS /CLASS=SECURITY_CLASS

SECURITY_CLASS object of class SECURITY_CLASS

     Owner: [SYSTEM]
     Protection: (System: RWED, Owner: RWED, Group: R, World: R)
     Access Control List:  <empty>

  Template: DEFAULT

     Owner: [SYSTEM]
```

```
     Protection: (System: RWED, Owner: RWED, Group, World: RE)
     Access Control List:  <empty>

$  SET SECURITY SECURITY_CLASS /CLASS=SECURITY_CLASS -
_$      /PROFILE=TEMPLATE=DEFAULT -
_$      /PROTECTION=(S:RWE, O:RWE, G:RE)

$  SHOW SECURITY SECURITY_CLASS /CLASS=SECURITY_CLASS

SECURITY_CLASS object of class SECURITY_CLASS
     Owner: [SYSTEM]
     Protection: (System: RWED, Owner: RWED, Group: R, World: R)
     Access Control List:  <empty>

  Template: DEFAULT

     Owner: [SYSTEM]
     Protection: (System: RWE, Owner: RWE, Group: RE, World: RE)
     Access Control List:  <empty>
```

This example demonstrates how to change the security elements for the template of a security class object. The first command shows the current settings for the `SECURITY_CLASS` object. The second command changes the `DEFAULT` template of the `SECURITY_CLASS` object such that the protection is (`S:RWE`, `O:RWE`, `G:RE`). The change is shown in the display of the last command. The world protection of `RE` remains unchanged.

4.
```
$  DIRECTORY/SECURITY

Directory DKA200:[DATA]

FILE001.DAT;1        [SYSTEM]                          (RWED,RWED,RE,)

Total of 1 file.

$  SET SECURITY/CLASS=FILE/PROTECTION=(WORLD:RE)/LOG FILE001.DAT
%SET-I-MODIFIED, DKA200:[DATA]FILE001.DAT;1 modified

$  DIRECTORY/SECURITY

Directory DKA200:[DATA]

FILE001.DAT;1        [SYSTEM]                        (RWED,RWED,RE,RE)

Total of 1 file.
$
```

This example shows how to set UIC-based protection codes on an object. The first **`DIRECTORY`** command displays the current security settings on the file `FILE001.DAT`. The **`SET SECURITY`** command changes the protection codes on the file to allow read and execute access for all users. The last command displays the results of the change.

SET SERVER ACME_SERVER

SET SERVER ACME_SERVER — Controls the ACME server. **Requires SYSPRV privilege.**

Format

SET SERVER ACME_SERVER

Description

The **SET SERVER ACME_SERVER** command provides a system manager with the ability to manage the Authentication and Credential Management Extension (ACME) server process, ACME_SERVER. Specifying this command allows you to start, stop, and restart the ACME server.

The ACME_SERVER process is a detached, kernel-threaded server process that services user authentication requests. These requests are received from client processes that use the SYS$ACM system service. One or more ACME agents assist in processing these requests. An ACME agent is a shareable image that is dynamically activated within the ACME_SERVER process.

The ACME_SERVER process is created automatically at system startup with the VMS ACME agent configured and enabled.

By default, the ACME_SERVER process runs under the SYSTEM account profile ([1,4]) with the following privileges: IMPERSONATE, SYSNAM, PSWAPM, SHARE, TMPMBX, EXQUOTA, AUDIT, CMKRNL, WORLD, and SETPRV.

The ACME_SERVER process is created with the following quotas:

```
ASTLM     = 200
BIOLM     = 200
BYTLM     = 100000
DIOLM     = 200
ENQLM     = 1000
FILLM     = 64
JTQUOTA   = 4096

PGFLQUOTA = 200000 (Alpha)
PGFLQUOTA = 300000 (Integrity servers

PRCLM     = 4
TQELM     = 64
WSQUOTA   = 2048
WSDEFAULT = 1024
WSEXTENT  = 4096
```

Default UIC, privileges, and quotas can be overridden when the server is started.

In the event of an abnormal process termination, the server will automatically restart and run the SYS$MANAGER:ACME$START.COM command procedure. The executive-mode logical name ACME$START is used to locate this file.

SYS$MANAGER:ACME$START.COM is also run as a result of one of the following commands:

- **SET SERVER ACME/START=AUTO** command is issued.
- **SET SERVER ACME/RESTART** command is issued.

The SYS$MANAGER:ACME$START.LOG file contains any information produced during restart in the event of problems.

SYS$MANAGER:ACME$START.COM file can be modified by the user to define which agents are configured during a restart as well as any other server configuration options. It is not replaced during system upgrades. The SYS$MANAGER:ACME$START.TEMPLATE file contains the VSI-supplied version of the file.

The ACME_SERVER process directs SYS$ERROR output to the SYS$MANAGER:ACME_SERVER_ERROR.LOG file. This file exists only if unexpected errors are encountered during operation.

Log file entries generated by the server and ACME agents are written by default to the SYS$MANAGER:ACME$SERVER.LOG file.

The **SET SERVER ACME** commands are used to perform these operations:

- Start and stop the ACME_SERVER process.
- Configure (dynamically load) ACME agents.
- Enable and disable ACME agents (after they are configured).
- Suspend and resume requests processing. (ACME agents reenter "initial" state for system management reasons, for example, system backup operations.)
- Control logging and tracing.

The **SHOW SERVER ACME** commands are used to display server and agent information.

The following sequence of commands is used to create the server and enable request processing:

1. **SET SERVER ACME/START**

   This command creates the server process.

2. **SET SERVER ACME/CONFIGURE**

   This command configures one or more ACME agents.

3. **SET SERVER ACME/ENABLE**

   This command enables ACME agent request processing.

For additional information, see the **SHOW SERVER ACME_SERVER** command and see the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#CONFIGURE_ACME_SERVER_AND_AGENTS].

For information about the SYS$ACM system service, see the relevant section in the *VSI OpenVMS System Services Reference Manual: A–GETUAI* [https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-a-getuai/#_4527acm].

Qualifiers

/ABORT

Forces the ACME_SERVER process to terminate without graceful shutdown of ACME agents. Pending requests are cancelled.

This command can be used if a malfunctioning ACME agent prevents a graceful shutdown.

/CANCEL

Cancels pending dialogue requests. Pending dialogue requests are outstanding requests to SYS$ACM callers to supply dialogue response data. Active requests being serviced by the ACME_SERVER process are allowed to complete normally.

Can be used only if the **/EXIT** or **/DISABLE** qualifier is also specified.

/CLUSTER

Issues the **SET** command to each ACME server in the cluster.

/CONFIGURE=(NAME=*name*[,CREDENTIALS=*credentials*][,FACILITY=*facility*][,FILE=*file*][,THREAD_MAX=*n*][,...])

Dynamically loads an ACME agent. An ACME agent is a shareable image that conforms to the ACME agent plug-in interface specification.

By default, the ACME server looks for an ACME agent file named *prefix*$*name*_ACMESHR.EXE, where the *prefix* is VMS by default and *name* is the case-insensitive string specified by the NAME keyword. The prefix can be overridden by the FACILITY keyword. Otherwise, a complete file specification can be specified with the FILE keyword.

You must specify the CREDENTIALS keyword if the ACME agent is a domain of interpretation (DOI) agent capable of authenticating users and issuing credentials. The CREDENTIALS keyword specifies the name of the persona extension associated with the ACME agent (see the $PERSONA_EXTENSION_LOOKUP system service in the *VSI OpenVMS System Services Reference Manual: GETUTC–Z* [https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-getutc-z/#_4527pers_ext_lookup]). The credentials name must match the name registered by the executive loadable image that implements the persona extension. For credentials other than "VMS", see the documentation provided for the ACME agent and the specific persona extension.

A maximum of eight ACME agents can be configured.

THREAD_MAX Keyword: Usage Considerations

An ACME server worker thread is the authentication request processing thread of execution. The number of worker threads determines the number of authentication requests the ACME server can process simultaneously.

The ACME server assigns a request slot to each in-progress authentication request. The total number of request slots sets a limit on the maximum number of requests the ACME server can have outstanding.

The ACME server limits the number of unprivileged authentication requests it can accept for processing to half the total number of request slots.

If a process makes a call to $ACM and no request slots are available, the process remains in resource wait state until a request slot becomes free.

The number of ACME server worker threads can range from 1 to 32. The default of 4 satisfies most operating environments in which authentication processing time is largely spent waiting for I/O. You may, however, be able to improve throughput by increasing the number of worker threads if, for example, you have a user-written ACME agent that requires more CPU time than most.

You can set the maximum number of ACME server worker threads before you start or restart the server by defining the inner-mode logical name ACME$THREAD_MAXIMUM in the systemwide logical name table. For example:

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE ACME$THREAD_MAXIMUM 8
```

You can also use the **/CONFIGURE** qualifier to set the maximum number of server worker threads before you enable the ACME server. For example:

```
$ SET SERVER ACME /CONFIGURE=THREAD_MAX=6
```

The **/CONFIGURE** qualifier overrides the value of the ACME$THREAD_MAXIMUM logical, if it is defined.

Because most authentication processing is I/O bound, defining more threads than about one quarter the number of request slots does not gain more throughput.

The ACME server calculates a default number of request slots from the value of MAXPROCESSCNT and the number of available CPUs. You can override the calculated value by defining the inner-mode logical name ACME$REQUEST_MAXIMUM in the systemwide name table before you start or restart the ACME server. For example:

```
$ DEFINE /SYSTEM /EXECUTIVE_MODE ACME$REQUEST_MAXIMUM 64$ SET SERVER
 ACME /RESTART
```

/DISABLE

Disables (shuts down) all ACME agents. Additional ACME agents can be configured.

/ENABLE[=NAME=(*name*[,...])]

Enables one or more ACME agents in the specified order. Ordering of ACME agents is significant, as the ACME_SERVER processes user authentication requests in the order specified. The *name* variable is a case-insensitive string argument.

If you do not specify the NAME keyword, the same list of ACME agents will be enabled that were enabled by a previous **/ENABLE** command. If no previous **/ENABLE** command was issued, all configured ACME agents are enabled.

/EXIT

Requests a graceful shutdown of the ACME agents. The ACME_SERVER process will be shut down after pending requests are completed.

/LOG
/NOLOG (default)

Creates and opens a new log file. The log file is named SYS$MANAGER:ACME$SERVER.LOG. It may be redirected by defining the system logical name ACME$SERVER.

/NODE=(*node-name*[,...])

Issues the **SET** command to the ACME server on the specified nodes in the order they are entered. The node names must be located within the current cluster.

/RESTART

Requests a graceful shutdown of the ACME agents. The ACME_SERVER process will be shut down after pending requests are completed.

The ACME_SERVER is restarted and the SYS$MANAGER:ACME$START.COM startup procedure runs.

/RESUME

Resumes normal operation following use of the **/SUSPEND** qualifier.

/START[=AUTO]

/START cannot be used with the **/ABORT**, **/EXIT**, or **/RESTART** qualifiers.

Starts the server process.

The optional keyword, AUTO, causes the server to startup and configure itself using the SYS$MANAGER:ACME$START.COM procedure. By default, the server starts with only the OpenVMS ACME agent configured. The following options are available:

To change the default privileges:

```
/PRIVILEGES=(privilege[,...])
```

To change the default UIC:

```
/UIC=uic
```

To change the default quotas:

```
/AST_LIMIT=n
/BUFFER_LIMIT=n
/ENQUEUE_LIMIT=n
/EXTENT=n
/FILE_LIMIT=n
/IO_BUFFERED=n
/IO_DIRECT=n
/JOB_TABLE_QUOTA=n
/MAXIMUM_WORKING_SET=n
/PAGE_FILE=n
/QUEUE_LIMIT=n
/SUBPROCESS_LIMIT=n
/WORKING_SET=n
```

You can use the following alternate syntax to specify a list of quotas and their values:

```
/QUOTAS=(quota=n[,...])
```

quota is a PQL-style quota name (for example: ASTLM, BIOLM).

See the $CREPRC system service in the *VSI OpenVMS System Services Reference Manual: A–GETUAI* [https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-a-getuai/#JUN_147] for a list of PQL-style quota names and descriptions.

To change the image dump setting:

```
/DUMP (default)
/NODUMP
```

Forces a process dump in the event of an abnormal process termination. The dump is written to SYS$MANAGER: using the image file name and .DMP extension.

/SUSPEND

Suspends operation after all pending requests are completed.

ACME agents are returned to their initial state (open files are closed, for example). This permits system management functions, such as system backups, to be performed without file-open conflicts.

See also the **/RESUME** qualifier.

/TRACE= *n*
/NOTRACE

Enables trace mode in the ACME$SERVER log file. Opens the log file if not already opened.

n is a bit-mask with the following assigned bits:

```
0   agent             Enable agent tracing
1   general           General (non-specific) tracing
2   vm                VM operations
3   ast               AST processing
4   wqe               WQE parameter values
5   report            Agent status/attributes operations
6   message           Messaging operations
7   dialogue          Dialogue operations
8   resource          Agent resource operations
9   callout           Agent callout routine
10  callout_status    Agent callout return status
```

If you specify **/NOTRACE**, trace mode is disabled and the log file remains open.

/WAIT
/NOWAIT (default)

Returns to caller once the server returns a status other than ACME$_BUSY.

Can only be used with **/EXIT**, **/DISABLE**, and **/SUSPEND** qualifiers.

Examples

1. `$ SET SERVER ACME_SERVER/START/LOG`

   This command starts the ACME server and creates and opens a new log file.

2. `$ SET SERVER ACME_SERVER/DISABLE/WAIT/CANCEL`

   This command disables all configured ACME agents, cancelling all requests in progress. Control does not return to the DCL command prompt until the ACME server has completed the cancellations.

3. `$ SET SERVER ACME_SERVER/CONFIGURE=(NAME=VMS,CREDENTIALS=VMS)`

   This command loads the VMS ACME agent and identifies its credentials as `VMS`.

SET SERVER REGISTRY_SERVER (Alpha/IA-64 only)

SET SERVER REGISTRY_SERVER — Controls the Registry server. **Requires SYSPRV privilege.**

Format

SET SERVER REGISTRY_SERVER

Description

The **SET SERVER REGISTRY_SERVER** command provides a system manager with the ability to start, stop, and restart the Registry server. The Registry server maintains information stored in the Registry database.

The Registry database is used by COM, Advanced Server for OpenVMS, and other applications.

For more information about the Registry database and the $REGISTRY system service, see the relevant section in the *VSI OpenVMS System Services Reference Manual* [https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-getutc-z/#_4527REGISTRY]. See also the **SHOW SERVER REGISTRY_SERVER** command.

Qualifiers

/ABORT

Aborts the Registry server on the specified node or nodes in the cluster.

Cannot be used with the **/EXIT**, **/RESTART**, or **/START** qualifiers.

/CLUSTER

Issues the **SET** command to each Registry server in the cluster, setting the Registry master server last.

Cannot be used with the **/MASTER** or **/NODE** qualifiers.

/EXIT

Stops the Registry server on the specified node or nodes in the cluster.

Cannot be used with the **/ABORT**, **/RESTART**, or **/START** qualifiers.

/LOG
/NOLOG (default)

Closes the current Registry server log file and creates a new file.

/MASTER

Requires SYSLCK privilege.

Issues the command to the Registry master server only.

Cannot be used with the **/CLUSTER**, **/NODE**, or **/START** qualifiers.

/NODE=(*node-name*[,...])

Issues the **SET** command to the Registry servers on the specified nodes in the order they are entered. The node names must be within the current cluster.

Cannot be used with the **/CLUSTER** or **/MASTER** qualifiers.

/RESTART

Restarts the Registry server on the specified node or nodes in the cluster.

Cannot be used with the **/ABORT**, **/EXIT**, or **/START** qualifiers.

/START

/START cannot be used with the **/ABORT**, **/EXIT**, or **/RESTART** qualifiers.

Starts the server process. The following options are available:

To change the default privileges:

```
/PRIVILEGES=(privilege[,...])
```

To change the default UIC:

```
/UIC=uic
```

To change the default quotas:

```
/AST_LIMIT=n
/BUFFER_LIMIT=n
/ENQUEUE_LIMIT=n
/EXTENT=n
/FILE_LIMIT=n
/IO_BUFFERED=n
/IO_DIRECT=n
/JOB_TABLE_QUOTA=n
/MAXIMUM_WORKING_SET=n
/PAGE_FILE=n
/QUEUE_LIMIT=n
/SUBPROCESS_LIMIT=n
/WORKING_SET=n
```

You can use the following alternate syntax to specify a list of quotas and their values:

```
/QUOTAS=(quota=n[,...])
```

The *quota* value is a PQL-style quota name (for example: ASTLM, BIOLM).

See the $CREPRC system service in the *VSI OpenVMS System Services Reference Manual: A–GETUAI* [https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-a-getuai/#JUN_147] for a list of PQL-style quota names and descriptions.

To change the image dump setting:

```
/DUMP
/NODUMP (default)
```

Forces a process dump in the event of an abnormal process termination. The dump is written to SYS$MANAGER: using the image file name and .DMP extension.

Examples

1. `$ SET SERVER REGISTRY_SERVER/RESTART/MASTER`

   This command restarts the detached Registry server on the master server.

2. `$ SET SERVER REGISTRY_SERVER/LOG/CLUSTER`

   This command closes the current log files and opens new files on all systems across the cluster.

3. `$ SET SERVER REGISTRY_SERVER/EXIT/NODE=(KAKADU,CAIRNS)`

   This command stops the detached Registry server process on nodes `KAKADU` and `CAIRNS`.

SET SERVER SECURITY_SERVER

SET SERVER SECURITY_SERVER — Controls the Security server. **Requires SYSPRV privilege.**

Format

SET SERVER SECURITY_SERVER

Description

The **SET SERVER SECURITY_SERVER** command provides a system manager with the ability to start, stop, and restart the security server. The security server maintains information stored in the system intrusion and proxy databases.

The system intrusion database is used by **LOGINOUT**, DECnet-Plus, DECwindows, **SHOW INTRUSION**, **DELETE INTRUSION**, and other applications. For more information about the system intrusion database and $DELETE_INTRUSION, $SCAN_INTRUSION, and $SHOW_INTRUSION system services, see the *VSI OpenVMS System Services Reference Manual: A–GETUAI* [https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-a-getuai/] and *GETUTC-Z* [https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-getutc-z/]. For further information, see the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/].

The system proxy database is used by AUTHORIZE, DECnet-Plus, DFS, and other applications to access information stored in the network proxy database. Additional information can be found in the *VSI OpenVMS System Management Utilities Reference Manual, Volume I: A–L* [https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/] and *Volume 2: M-Z* [https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-ii-m-z/]. See also the $ADD_PROXY, $DELETE_PROXY, $DISPLAY_PROXY, $VERIFY_PROXY system services in the *VSI OpenVMS System Services Reference Manual: A–GETUAI* [https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-a-getuai/] and *GETUTC-Z* [https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-getutc-z/].

Qualifiers

/EXIT

Stops the detached security server process.

/EXIT cannot be used with any other qualifier.

/RESTART

Restarts the detached security server process.

/RESTART cannot be used with any other qualifier.

/START

Starts the detached security server process.

/START cannot be used with any other qualifier.

Examples

1. `$ SET SERVER SECURITY_SERVER/EXIT`

   This command stops the detached security server process.

2. `$ SET SERVER SECURITY_SERVER/RESTART`

   This command restarts the detached security server process.

SET SHADOW

SET SHADOW — Changes the characteristics of shadow sets created using Volume Shadowing for OpenVMS. For more information about volume shadowing, see the *VSI OpenVMS Volume Shadowing Guide* [https://docs.vmssoftware.com/vsi-openvms-volume-shadowing-guide/]. **Requires SYSPRV privilege.**

Format

SET SHADOW [*device-name*[:]]

Parameter

***device-name*[:]**

Specifies the name of a shadow set (DSA*n*:) or shadow set member (*ddcu*:), depending on the command qualifiers. Refer to parameter formats specified in the descriptions of specific qualifiers.

For most qualifiers that take a shadow set (DSA*n*:) as a parameter the **/ALL** qualifier can be used in place of the shadow set name to indicate that the requested action applies to all shadow sets on the system. Exceptions are noted in the description of the **/ALL** qualifier.

Description

The **SET SHADOW** command and its qualifiers are designed to give the user greater control over the behavior of shadow sets. These commands act upon the shadow set (also known as the virtual unit) or, in some cases, on the individual shadow set member that you specify.

Unless otherwise noted, the qualifier settings typically remain in effect until they are changed using either the **SET SHADOW** or **SET DEVICE** command or until the shadow set or shadow set member is dismounted. If the shadow set is remounted or a shadow set member is returned to the shadow set from which it was dismounted, the settings must be specified again.

Some **SET SHADOW** qualifiers also can be specified using the **SET DEVICE** command, where they have the same effect. However, if your focus is on changing characteristics of shadow sets, the **SET SHADOW** command provides more options.

Note

The **/DELETE**, **/DISABLE**, **/ENABLE**, **/NAME**, and **/POLICY** qualifiers are used exclusively to manage host-based minimerge (HBMM) operations and do not apply to other operations.

If you specify any other (non-HBMM) qualifiers in a command that includes HBMM qualifiers, the command will fail.

Qualifiers

/ABORT_VIRTUAL_UNIT DSA*n*:
/ABORT_VIRTUAL_UNIT /ALL

Aborts mount verification on the specified shadow set or on all shadow sets in mount verification on the system.

Use this qualifier when you know that the unit cannot be recovered. When you use this qualifier, the shadow set must be in mount verification. The shadow set aborts mount verification immediately on the system from which the command was issued. If the shadow set is not in mount verification, the command returns the following error:

```
%SYSTEM-E-UNSUPPORTED, unsupported operation or function.
```

After the command completes, the shadow set still must be dismounted. Use the following command to dismount the shadow set:

```
$ DISMOUNT/ABORT/OVERRIDE=CHECKS DSAn:
```

/ALL

Causes the command to operate on all shadow sets that are mounted on the system from which the command is issued.

The **/ALL** qualifier can be used instead of DSA*n*: in most commands that take a shadow set device specification as a parameter. Exceptions are the **/DEMAND_MERGE**, **/DELETE**, and **/EVALUATE=RESOURCES** qualifiers. **/ALL** also cannot be specified with the **/POLICY** qualifier or any qualifier that operates only on individual shadow set members (such as **/MEMBER_TIMEOUT** and **/FORCE_REMOVAL**).

/CONFIRM
/NOCONFIRM (default)

Specifies whether a query is made before each merge operation to confirm that the operation should be performed on the designated shadow set.

This qualifier can be used only in conjunction with the **/DEMAND_MERGE** qualifier.

The following responses are valid in response to the query:

- Affirmative: YES, TRUE, or 1.
- Negative: NO, FALSE, 0 (zero), or pressing the **Return** key.
- End the process: QUIT or **Ctrl/Z**.
- When you enter ALL, the command continues to process, but no further prompts are given.

You can enter word responses in uppercase or lowercase letters, and words can be abbreviated to one or more letters. If you enter an illegal response, DCL redisplays the prompt.

/COPY_SOURCE *ddcu*:
/COPY_SOURCE DSA*n*:
/COPY_SOURCE /ALL

Specifies which source member of a shadow set to use as the source for read data during full copy operations when a new member is added to a shadow set that already contains two or more full members. This qualifier affects only those copy operations that do not use disk copy data (DCD) commands. The source specified by this qualifier persists until the shadow set is dismounted.

Some storage controllers, such as the HSG80, have a read-ahead cache, which significantly improves a device's read performance. Copy operations normally alternate reads between the source members, which effectively nullifies the benefits of the read-ahead cache. This qualifier lets you force all reads from a single, specified source member for the duration of a copy operation.

In addition to improving copy performance, **/COPY_SOURCE** can be used to prevent read operations from a specific shadow set member that is considered unreliable. By specifying only the healthy shadow set member (or members), the copy operations can continue to completion. The unreliable shadow set member can be removed once the copy operation completes successfully.

If a shadow set (DSA*n*:) is specified, all reads for full copy operations are performed from the device that is the current "master" member, regardless of the physical location of that device.

If a shadow set member (*ddcu*:) is specified, that member is used as the read source for all copy operations. This setting allows you to choose any source member. For example, you can choose a source member that is at the same site as the member being added, rather than using a master member that is not at the same site.

If **/ALL** is specified, all reads for full copy operations on all currently mounted virtual units are performed from the master member.

/DELETE DSA*n*:
/DELETE /NAME

Used in conjunction with **/POLICY=HBMM**, **/DELETE** removes a host-based minimerge (HBMM) policy from a specified shadow set, or deletes an HBMM named policy from the entire cluster. For example, the following command removes the policy that is currently associated with shadow set DSA1:

```
$ SET SHADOW /DELETE DSA1 /POLICY=HBMM
```

In contrast, the following command removes COMPANY_POLICY from the cluster:

```
$ SET SHADOW /DELETE /NAME=COMPANY_POLICY /POLICY=HBMM
```

You cannot delete the NODEFAULT policy.

You cannot specify **/ALL** with **/DELETE**.

/DEMAND_MERGE DSA*n*:

Initiates a merge operation on the specified shadow set. This qualifier is useful if the shadow set was created with the **INITIALIZE/SHADOW** command without the use of the **/ERASE** qualifier. For more information about using **/DEMAND_MERGE**, see the relevant section in the *VSI OpenVMS Volume Shadowing Guide* [https://docs.vmssoftware.com/vsi-openvms-volume-shadowing-guide/#demand_merge_start_merge_oper_sect].

You cannot specify **/ALL** with **/DEMAND_MERGE**.

An OPCOM message is displayed for each shadow set indicating that a demand merge has been invoked and recording the process ID (PID) of the process that executed the command. For example:

```
%%%%%%%%%%  OPCOM   9-MAR-2004 10:35:23.24  %%%%%%%%%%
Message from user SYSTEM on NODE1
Demand Merge requested for _DSA721:, PID: 2760009A
```

/DISABLE=HBMM DSA*n*:
/DISABLE=HBMM /ALL
/DISABLE=SPLIT_READ_LBNS

Disables host-based minimerge (HBMM) on the specified shadow set or clusterwide on all shadow sets.

/DISABLE=SPLIT_READ_LBNS disables the split behavior of logical block numbers and as a result the reads are alternated between the source shadow set members having the same READ_COST and device queue length.

/ENABLE=HBMM
/ENABLE=SPLIT_READ_LBNS

Enables host-based minimerge (HBMM) on the specified shadow set or across the entire cluster if an applicable HBMM policy exists.

The **/ENABLE=SPLIT_READ_LBNS** qualifier logically divides the shadow set members having the same read cost into equal groups of Logical Block Numbers (LBNs).When a virtual unit performs a read, it does so by reading from the corresponding LBN group. This results in the maximum usage of the controller read-ahead cache.

/EVALUATE=RESOURCES

Forces the system to evaluate whether it should act on most shadow copy and merge operations currently being managed on the system. It cancels most operations and then, based on the value of system parameter SHADOW_MAX_COPY and the copy/merge priority of each shadow set, it evaluates the order in which the pending copies and merges should be restarted.

RESOURCES is the only supported value for **/EVALUATE** and must be included.

/EVALUATE does not apply to MSCP-based minimerge operations. MSCP-based minimerge operations are not subject to cancellation and restart by **/EVALUATE**.

This command is intended to be used after changing the value of the dynamic system parameter SHADOW_MAX_COPY or after issuing a **SET SHADOW /PRIORITY=*n*** command for a shadow set. After a suitable delay, all available SHADOW_MAX_COPY slots on the system are allocated using the priority list.

/FORCE_REMOVAL *ddcu*:
/NOFORCE_REMOVAL *ddcu*:

Expels the specified shadow set member from the shadow set. The specified device must be a member of a shadow set that is mounted on the system where the command is issued. You cannot specify **/ALL** with **/FORCE_REMOVAL**.

If connectivity to a device has been lost and the shadow set is in mount verification, this qualifier causes the member to be expelled from the shadow set immediately.

If the shadow set is not currently in mount verification, no immediate action is taken. If connectivity to a device has been lost but the shadow set is not in mount verification, this qualifier lets you flag the member to be expelled from the shadow set as soon as the shadow set enters mount verification. If no action has been taken on the specified member and you wish to clear the flag, use the **/NOFORCE_REMOVAL** qualifier.

If the shadow set is dismounted before the member is expelled, the FORCE_REMOVAL request expires.

/LOG

Instructs the volume shadowing software to display a brief message confirming that the **SET SHADOW** command completed. If the **/OUTPUT** qualifier is also specified, this information is written to the output file.

/MEMBER_TIMEOUT=*n ddcu*:

Specifies the timeout value to be used for a shadow set member. The specified device must be a member of a shadow set that is mounted on the system where the command is issued.

The value supplied by this qualifier overrides the system parameter SHADOW_MBR_TMO for this specific device. Each member of a shadow set can be assigned a different MEMBER_TIMEOUT value.

The valid range for *n* is 1 through 16 777 215 seconds.

The timeout value set by **/MEMBER_TIMEOUT** does not persist after the shadow set is dismounted.

/MVTIMEOUT=*n* DSA*n*:
/MVTIMEOUT=*n* /ALL

Specifies the mount verification timeout value to be used for all shadow sets on the cluster or for the shadow set specified by its virtual unit name (DSA*n*:). The specified shadow set must be mounted on the system where the command is issued.

The value supplied by this qualifier overrides the value specified by the system parameter MVTIMEOUT for this specific shadow set.

Note

You cannot change the value of MVTIMEOUT for a system disk. Any attempt to do so results in an error.

The valid range for *n* is 1 through 16 777 215 seconds.

The timeout value set by **/MVTIMEOUT** does not persist after the shadow set is dismounted.

/NAME=*policy-name*

Used with **/POLICY=HBMM** to define a named host-based minimerge (HBMM) policy or used with **/DELETE** to delete a policy. The policy is defined clusterwide. See detailed descriptions under the **/DELETE** and **/POLICY** qualifiers.

Policy names are case insensitive and must consist of from 1 to 64 characters. Only letters, numbers, the dollar sign ($), and the underscore (_) are allowed.

If you create a default policy, you must assign it the name DEFAULT.

For details about creating and using policy names, see the relevant section in the *VSI OpenVMS Volume Shadowing Guide* [https://docs.vmssoftware.com/vsi-openvms-volume-shadowing-guide/#hbmm_policies_sect].

/OUTPUT=*file-name*

Outputs any messages to the specified file.

/POLICY=HBMM[=*policy-name*]
/POLICY=HBMM[=*policy-specification*]

Creates or deletes a policy for host-based minimerge (HBMM).

HBMM is the only supported value for the **/POLICY** qualifier, and it must be included. You can optionally specify a named policy, including DEFAULT, or you can specify NODEFAULT to indicate that the shadow set to which it is applied is not to use HBMM, including any DEFAULT policy. For details about specifying policies and using the DEFAULT and NODEFAULT policy names, see the relevant section in the *VSI OpenVMS Volume Shadowing Guide* [https://docs.vmssoftware.com/vsi-openvms-volume-shadowing-guide/#hbmm_policies_sect] manual.

When **/POLICY** is specified with **/DELETE**, it removes either a specified HBMM named policy or the HBMM policy for a specific shadow set. You cannot delete the NODEFAULT policy.

When **/POLICY** is specified with **/NAME**, it defines a clusterwide named policy. When no qualifiers other than **/NAME** or **/DELETE** are specified, **/POLICY** defines a policy for a specific shadow set.

Deleting bitmaps with the **DELETE/BITMAP** command causes a bitmap to be deleted. However, the shadowing software recognizes this condition and starts a new bitmap immediately. To disable HBMM bitmaps, you have to use the command **SET SHADOW/DISABLE=HBMM**.

When defining a policy, you use five keywords (MASTER_LIST, COUNT, RESET_THRESHOLD, MULTIUSE, and DISMOUNT) to control the placement and management of HBMM bitmaps. An HBMM policy specification consists of a list of these keywords enclosed within parentheses. Only the MASTER_LIST keyword is required. If COUNT and RESET_THRESHOLD are omitted, default values are applied.

The MULTIUSE and DISMOUNT keywords specify the number of bitmaps to be converted to multiuse bitmaps during the automatic and manual removal of members respectively. If MULTIUSE is omitted, then automatic minicopy on volume processing is not enabled. As a result, no HBMM bitmap is converted to multiuse bitmap. If DISMOUNT is omitted, only a maximum of 6 HBMM bitmaps can be used as multiuse bitmaps.

- **MASTER_LIST=*list***

  The MASTER_LIST keyword is used to identify a set of systems as candidates for a master bitmap. The *list* value can be a single system name; a parenthesized, comma-separated list of system names; or the wildcard character, as shown in the following examples:

  ```
  MASTER_LIST=NODE1MASTER_LIST=(NODE1,NODE2,NODE3)MASTER_LIST=*
  ```

  When the system list consists of a single system name or the wildcard character, parentheses are optional.

  An HBMM policy must include at least one MASTER_LIST. Multiple master lists are optional. If a policy has multiple master lists, the entire policy must be enclosed with parentheses, and each constituent master list must be separated by a comma as shown in the following example:

```
(MASTER_LIST=(NODE1,NODE2), MASTER_LIST=(NODE3,NODE4))
```

There is no significance to the position of a system name in a master list.

- **COUNT=*n***

  The COUNT keyword specifies how many systems in the master list can have master bitmaps. Therefore, the COUNT keyword and its associated MASTER_LIST must be enclosed within a single parenthetical statement.

  The COUNT value specifies the number of systems on which you want master bitmaps. It does not necessarily mean that the first *n* systems in the list will be chosen.

  When the COUNT keyword is omitted, the default value is 6 or the number of systems in the master list, whichever is smaller.

  You cannot specify more than one COUNT keyword per master list.

  Examples:

```
(MASTER_LIST=(NODE1,NODE2,NODE3), COUNT=2)

(MASTER_LIST=(NODE1,NODE2,NODE3),COUNT=2),
(COUNT=2,MASTER_LIST=(NODE4,NODE5,NODE6))
```

- **RESET_THRESHOLD=*n***

  The RESET_THRESHOLD keyword specifies the number of blocks that can be set before the bitmap is eligible to be cleared. Each set bit in a master bitmap corresponds to a set of blocks to be merged, so this value can affect the merge time.

  Bitmaps are eligible to be cleared when the RESET_THRESHOLD is exceeded. However, the reset is not guaranteed to occur immediately when the threshold is crossed. For more information about choosing a value for this attribute, see the relevant section in the *VSI OpenVMS Volume Shadowing Guide* [https://docs.vmssoftware.com/vsi-openvms-volume-shadowing-guide/#reset_treshold_value_sect].

  The reset threshold is associated with a specific HBMM policy, so the RESET_THRESHOLD keyword can be defined only once in a policy specification. Because its scope is the entire policy, the RESET_THRESHOLD keyword cannot be specified inside a constituent master list when the policy uses multiple master lists.

  When the RESET_THRESHOLD keyword is omitted, the value of 1,000,000 is used by default. See the following example:

```
(MASTER_LIST=*, COUNT=4, RESET_THRESHOLD=1000000)
```

  Example:

  The command in the following example defines the HBMM named policy `POLICY_2`, which has two master lists. Having multiple master lists can be useful in a multiple-site OpenVMS Cluster configuration because a policy can be defined to ensure that at least one surviving system has an HBMM bitmap in the event of an outage at one or more sites.

```
$ SET SHADOW /POLICY=HBMM=( -
_$        (MASTER_LIST=(NODE1,NODE2,NODE3), COUNT=2), -
```

```
_$        (MASTER_LIST=(NODE4,NODE5,NODE6), COUNT=2), -
_$                RESET_THRESHOLD=150000) -
_$                /NAME=POLICY_2
```

In a policy with multiple master lists, a given system name can appear in only one master list.

A shadow set need not be mounted to have an HBMM policy defined for it.

See the **SET SHADOW** Examples section for several more **/POLICY** examples.

- **MULTIUSE=*n***

  The MULTIUSE keyword enables automatic minicopy on volume processing. *n* specifies the number of existing HBMM master bitmaps to be converted to multiuse bitmaps in the event that a shadow set member is removed from the shadow set by the shadowing driver.

  During a loss of connectivity to a site or controller, shadowing may remove a member from the shadow set. When the member is added back to the shadow set, a full shadow copy occurs.

  By converting a few of the HBMM bitmaps to multiuse, all writes that are performed to the shadow set are recorded. Thus, when the member is added back to the shadow set, the multiuse bitmap can be used for a minicopy operation. This is much faster than a full copy operation.

  The value of *n* cannot exceed the implied or explicit value of COUNT. If MULTIUSE is not specified, then bitmaps are not converted to multiuse and a full copy operation is required. Fatal drive errors that remove a shadow set member do not cause a multiuse conversion as the drive has to be replaced and therefore requires a full copy operation.

- **DISMOUNT=*n***

  The DISMOUNT keyword allows all the 12 write bitmaps to be used by Shadowing as multiuse bitmaps, thereby reducing the single point of failure of single minicopy master bitmaps. *n* specifies the number of HBMM bitmaps to be converted to multiuse bitmaps every time a member is dismounted from a shadow set with the following command:

  ```
  DISMOUNT/POLICY=MINICOPY
  ```

/PRIORITY=*n* DSA*n*:

Overrides the current default priority setting. Priorities range from 0 (lowest) to 10 000 (highest). The default priority is 5000. A shadow set with a priority of 0 is never considered for a merge or a copy on the system.

When a recovery operation (either a merge or a copy) is needed on multiple shadow sets, the shadow sets are recovered in priority order from highest to lowest. The priority setting is system specific; any change in priority made on a single system does not propagate to the entire cluster and does not persist across a system reboot.

Once this qualifier has been applied to a virtual unit that is mounted, the setting persists across any subsequent **DISMOUNT** and **MOUNT** commands.

/READ_COST=*n* *ddcu*:
/READ_COST=*n* DSA*n*:
/READ_COST=*n* $*n*$DGA*n*:

Enables you to modify the default cost assigned to each shadow set member (*ddcu*:). By modifying the assignments, you can bias the reads in favor of one member of a two-member shadow set or, in

the case of shadow sets with more than two members, in favor of one or more members of the set over the remaining members. The device specified must be a shadow set or member of a shadow set that is mounted on the system where the command is issued.

The valid range for the specified cost is 1 through 65,535 units.

The shadowing driver assigns default READ_COST values to shadow set members when each member is initially mounted. The default value depends on the device type and its configuration relative to the system mounting it. The following list of device types is ordered by the default READ_COST assignments, from the lowest cost to the highest cost:

- DECram device
- Directly connected device in the same physical location
- Directly connected device in a remote location
- DECram served device
- Default value for other served devices

The value supplied by the **/READ_COST** qualifier overrides the default assignment. The shadowing driver adds the value of the current queue depth of the shadow set member to the READ_COST value and then reads from the member with the lowest value.

Different systems in the cluster can assign different costs to each shadow set member.

When this qualifier specifies a shadow set (DSA*n*:) instead of a shadow set member, the **/READ_COST** qualifier switches the read cost setting for *all* shadow set members back to the default read cost settings established automatically by the shadowing software. The specified shadow set must be mounted on the system where the command is issued.

You cannot specify a value for **/READ_COST** when you specify **/ALL**.

If the **/SITE** command qualifier has been specified, the shadowing driver takes site values into account when it assigns default READ_COST values. In order for the shadowing software to determine whether a device is in the category of "directly connected device in a remote location", the **/SITE** command qualifier must have been applied to both the shadow set *and* the shadow set member.

Reads requested for a shadow set from a system at site 1 are performed from a shadow set member that is also at site 1. Reads requested for the same shadow set from site 2 can read from the member located at site 2.

/RECOVERY_OPTIONS= DELAY_PER_SERVED_MEMBER=*n*

Allows the system manager to adjust the rating assigned to a system based on a delay assessed for each MSCP served shadow set member on that system. The value specified by this qualifier overrides the value established by the SHADOW_PSM_RDLY system parameter. The default delay for each MSCP served member is 30 seconds and the valid range for the specified delay is 0 through 65,535 seconds.

When a copy or merge operation is needed on a shadow set that is mounted on multiple systems, OpenVMS Volume Shadowing attempts to perform this work on a system that has a local connection to all of the shadow set members. Systems are rated with a penalty (delay time) assessed for each shadow set member that is MSCP served to the system. No delay is added for local members, so a system with all locally accessible shadow set members is likely to perform the work before a system where one or more members is served.

If the **/ALL** qualifier is also included, the specified delay is applied to all currently mounted shadow sets.

See the relevant section in the *VSI OpenVMS Volume Shadowing Guide* [https://docs.vmssoftware.com/vsi-openvms-volume-shadowing-guide/#manage_copy_merge_sect] for more information.

/RESET_COUNTERS

Resets the shadowing specific counters that are maintained for each shadow set. These counters can be displayed using the **SHOW SHADOW** command.

The counters that are reset to 0 are the following:

HBMM Reset Count
Copy Hotblocks
Copy Collisions
SCP Merge Repair Cnt
APP Merge Repair Cnt

The HBMM Reset Count is a running total of the number of times that the HBMM Reset Threshold was exceeded and therefore, the number of times the bitmaps were reset. The threshold is set by the RESET_THRESHOLD keyword in the **/POLICY** qualifier. For more information, see the relevant section in the *VSI OpenVMS Volume Shadowing Guide* [https://docs.vmssoftware.com/vsi-openvms-volume-shadowing-guide/#manage_copy_merge_sect].

/SITE=*n ddcu*:
/SITE=*n* DSA*n*:

Indicates to the shadowing driver the site location of the specified shadow set (DSA*n*:) or shadow set member (*ddcu*:).

The SHADOW_SITE_ID system parameter defines the default site location of the shadow set. You can override the default location of the shadow set with the **/SITE** qualifier.

The valid range for the site location, represented by *n*, is 1 through 255.

If **/ALL** is specified, all virtual units are assigned the new value. The shadow set's member site values remain unchanged.

After you apply this qualifier, the setting remains in effect until you change it using a **SET SHADOW/SITE** command.

This qualifier can improve read performance because the member that is physically local to the system will be the preferred disk from which to read, provided that you specify the **/SITE** qualifier for each shadow set member and for the shadow set. In a Fibre Channel configuration, shadow set members at different sites are directly attached to the system. For the Volume Shadowing and OpenVMS Cluster software, there is no distinction between local and remote in multiple-site Fibre Channel configurations.

/STALL=WRITES[=*nnn*]
/NOSTALL=WRITES[=*nnn*]

Using the **/STALL=WRITES[=*nnn*]** qualifier, you can stall the write operations for *nnn* seconds. *nnn* is the number of seconds for which you can stop write operations. If no value is

specified for *nnn*, the lock is released after SHADOW_MBR_TMO seconds. The default is SHADOW_MBR_TMO.

/NOSTALL=WRITES[=*nnn*] releases the lock after *nnn* seconds so that write operations can continue on the shadow set.

Examples

1.
```
$ ! Command sequence used on system 1 at site 1:
$ MOUNT/SYSTEM DSA0:/SHADOW=($1$DGA0:,$1$DGA1:) TEST
$ SET SHADOW/SITE=1  DSA0:
$ !
$ ! Command sequence used on system 2 at site 2:
$ MOUNT/SYSTEM DSA0:/SHADOW=($1$DGA0:,$1$DGA1:) TEST
$ SET SHADOW/SITE=2  DSA0:
$ !
$ ! Commands used on both system 1 and system 2:
$ SET SHADOW/SITE=1  $1$DGA0:
$ SET SHADOW/SITE=2  $1$DGA1:
```

   In this example, the system at site `1` will perform reads from `$1$DGA0`, which is located at that site, while the system at site `2` will perform reads from `$1$DGA1`, which is located at site `2`.

2.
```
$ SET SHADOW DSA1: /POLICY=HBMM=  -
_$ (MASTER_LIST=(NODE1,NODE2,NODE3), COUNT=2, -
_$      RESET_THRESHOLD=75000)
```

   This command defines an HBMM policy for shadow set `DSA1:`. The policy specifies that no more than two systems of the three systems listed can have an HBMM master bitmap. Once `75000` blocks have been modified on `DSA1:`, any existing HBMM bitmaps are zeroed.

3.
```
$ SET SHADOW /POLICY=HBMM=(MASTER_LIST= -
_$ (NODE1,NODE2,NODE3), RESET_THRESHOLD=75000, COUNT=2) -
_$ /NAME=HONESTY_IS_BEST
```

   This command defines the named policy `HONESTY_IS_BEST`. This policy is available to all cluster members, but is not applied to any specific shadow set.

4.
```
$ SET SHADOW DSA1: /POLICY=HBMM=HONESTY_IS_BEST
```

   This command associates the named policy `HONESTY_IS_BEST` with shadow set `DSA1`.

5.
```
$ SET SHADOW /POLICY=HBMM=( -
_$       (MASTER_LIST=(NODE1,NODE2,NODE3), COUNT=2), -
_$       (MASTER_LIST=(NODE4,NODE5,NODE6), COUNT=2), -
_$               RESET_THRESHOLD=150000) -
_$               /NAME=POLICY_2
```

   This command defines the HBMM named policy `POLICY_2`, which has two master lists.

6.
```
$ SET SHADOW DSA2:/POLICY=HBMM=POLICY_2
```

   This command associates the named policy `POLICY_2` with shadow set `DSA2`.

7.
```
$ SET SHADOW /POLICY=HBMM /DELETE /NAME=POLICY_2
```

   This command deletes the named policy `POLICY_2`. Note that any shadow set that had acquired the attributes of `POLICY_2` by execution of a

SET SHADOW DSA*n*: /POLICY=HBMM=POLICY_2 command is *not* affected. Changes made to a named policy are *not* retroactive to a mounted shadow set that was assigned the characteristics of a previous version of that named policy.

8. `$ SET SHADOW DSA1: /POLICY=HBMM /DELETE`

   This command deletes the existing HBMM policy from `DSA1:`. If this HBMM policy is enabled on `DSA1:`, you must first execute the following command to disable the policy before you can delete it:

   ```
   $ SET SHADOW DSA1: /DISABLE=HBMM
   ```

9. ```
   $ SET SHADOW /DEMAND_MERGE DSA721:/CONFIRM
   Allow merge of _DSA721:? [N]:Y
   $
   ```

   In this example, a merge operation is initiated on `DSA721:`.

10. `$ SET SHADOW DSA1/POLICY=HBMM=(MASTER=*,COUNT=12,MULTIUSE=12,DISMOUNT=1)`

    In this example, a policy is set in which all `12` bitmaps can be used as multiuse bitmaps. When you execute the command **DISMOUNT/POLICY=MINICOPY**, `1` minimerge bitmap is converted to multiuse bitmap. You can use this multiuse bitmap with the **MINICOPY** command to add the dismounted member back to the shadow set. In other words, it specifies that `12` bitmaps can be used during the automatic and `1` bitmap during the manual removal of the shadow set member.

11. `$ SET SHADOW DSA42 /STALL=WRITES`

    In this example, the writes are stalled to the shadow set for a period of `SHADOW_MBR_TMO` seconds.

12. `$ SET SHADOW DSA42 /STALL=WRITES=60`

    In this example, the writes are stalled to the shadow set for a period of `60` seconds.

13. `$ SET SHADOW DSA42 /NOSTALL=WRITES=30`

    In this example, the writes are allowed to the shadow set after a period of `30` seconds.

SET SYMBOL

SET SYMBOL — Controls access to local and global symbols in command procedures.

Format

SET SYMBOL

Description

The **SET SYMBOL** command controls access to local and global symbols in command procedures by treating symbols as undefined. Because all global and local symbols defined in an outer procedure level are accessible to inner procedure levels, it is often necessary to mask these symbols without deleting them.

The **SET SYMBOL** command also controls whether DCL will attempt to translate the verb string (the first token on the command line) as a symbol before processing the line. The default behavior is that the
   ```

translation is attempted. The advantage to changing this behavior is that a command procedure will not be affected by outer procedure level environments when invoking a command.

The symbol scoping context is different for local and global symbols. Local symbols are procedure level dependent. Local symbols defined in an outer subroutine level can be read at any inner subroutine level, but they cannot be written to. If you assign a value to a symbol that is local to an outer subroutine level, a new symbol is created at the current subroutine level; however, the symbol in the outer procedure level is not modified.

This means that the **SET SYMBOL/SCOPE=NOLOCAL** command causes all local symbols defined at an outer procedure level to be inaccessible to the current procedure level and any inner levels. For example, if **SET SYMBOL/SCOPE=NOLOCAL** was specified at procedure levels 2 and 4, procedure level 2 can read and write to only level 2 local symbols. Level 3 can read (but not write to) level 2 local symbols and can read and write to level 3 local symbols. Level 4 can read and write to only level 4 local symbols.

Global symbols are procedure-level independent. The current global symbol scoping context is applied subsequently to all procedure levels. Specifying the **/SCOPE=NOGLOBAL** qualifier causes all global symbols to become inaccessible for all subsequent commands until either the **/SCOPE=GLOBAL** qualifier is specified or the procedure exits to a previous level at which global symbols were accessible. In addition, specifying the **/SCOPE=NOGLOBAL** qualifier prevents you from creating any new global symbols until the **/SCOPE=GLOBAL** qualifier is specified.

When you exit a procedure level to return to a previous procedure, the symbol scoping context from the previous level is restored for both local and global symbols.

To display the current symbol scoping state, use the lexical function F$ENVIRONMENT ("SYMBOL_SCOPE").

Qualifiers

/ALL (default)

Specifies that the values of the **/SCOPE** qualifier pertain both to the translation of the first token on a command line and to general symbol substitution.

The **/ALL** qualifier is incompatible with the **/GENERAL** or the **/VERB** qualifier.

/GENERAL

Specifies that the values of the **/SCOPE** qualifier pertain to the translation of all symbols except the first token on a command line.

The **/GENERAL** qualifier is incompatible with the **/ALL** or the **/VERB** qualifier.

/SCOPE=(*keyword*,...)

Controls access to local and global symbols. Lets you treat symbols as being undefined. Possible keywords are as follows:

Keyword	Description
NOLOCAL	Causes all local symbols defined in outer procedure levels to be treated as being undefined by the current procedure and by all inner procedure levels.
LOCAL	Removes any symbol translation limit set by the current procedure level.

Keyword	Description
NOGLOBAL	Causes all global symbols to be inaccessible to the current procedure level and to all inner procedure levels unless otherwise changed.
GLOBAL	Restores access to all global symbols.

/VERB

Specifies that the values of the **/SCOPE** qualifier pertain to the translation of the first token on a command line as a symbol before processing only. It does not affect general symbol substitution.

Note

Caution must be used if the **SET SYMBOL/VERB/SCOPE** command is used more than once in a command procedure. Because DCL uses the translation behavior when looking for a label or subroutine, execution may be different running in one mode than in another. VSI recommends that the **SET SYMBOL/VERB/SCOPE** command be used once as part of the command procedure setup and left in that mode for the duration of the procedure.

The **/VERB** qualifier is incompatible with the **/ALL** or the **/GENERAL** qualifier.

Examples

1.
```
$ SET SYMBOL/SCOPE=NOLOCAL
```

   In this example, all local symbols defined in outer procedure levels are now undefined for the current procedure level and all inner procedure levels.

2.
```
$ SET SYMBOL/SCOPE=NOGLOBAL
```

   In this example, all global symbols are now inaccessible to the current procedure level and all inner procedure levels unless otherwise changed.

3.
```
$ NOW :== SHOW TIME
$ !
$ NOW
   3-NOV-2001 11:48:58
$ !
$ SET SYMBOL /VERB /SCOPE=NOGLOBAL
$ NOW
%DCL-W-IVVERB, unrecognized command verb-check validity and spelling
\NOW\
$ !
$ SHOW SYMBOL NOW
  NOW == "SHOW TIME"
```

   This example demonstrates the use of the **/VERB** qualifier.

   The symbol NOW is assigned to the **SHOW TIME** command. The next line shows the default behavior, where DCL attempts to translate the first string on the command line (NOW). Because NOW translates to the **SHOW TIME** command, this is used instead of NOW.

   The **SET SYMBOL** command on the next line changes the behavior so that DCL does not attempt a translation. When NOW is subsequently entered, DCL uses the string NOW as the command verb and cannot find it in the command table. This results in the error message.

Notice that the scoping of the verb translation has no effect on general symbol translations, as demonstrated by the **SHOW SYMBOL** command in the example.

SET TERMINAL

SET TERMINAL — Sets the characteristics of a terminal. Entering a qualifier changes a characteristic; omitting a qualifier leaves the characteristic unchanged.

Format

SET TERMINAL [*device-name*[:]]

Parameter

***device-name*[:]**

Specifies the device name of the terminal. The default is SYS$COMMAND if that device is a terminal. If the device is not a terminal, an error message is displayed.

Description

The **SET TERMINAL** command modifies specific terminal characteristics for a particular application or overrides system default characteristics. (These defaults are defined at each installation, based on the most common type of terminal in use.) The default characteristics for terminals are listed in *Figure 2, "Default Characteristics for Terminals"*. The table extends across two pages to document a range of terminals.

The terminal characteristics, local or remote, are determined automatically by the terminal driver for terminals that have the modem characteristic enabled. These characteristics are not affected by the **SET TERMINAL** command. For example, when you successfully dial in to an OpenVMS system processor, you establish your terminal as remote. When you hang up, the terminal characteristic is set back to local.

The set of terminals supported by the OpenVMS system includes a set of VT100 family terminals that support special DIGITAL ANSI characteristics and escape sequences. For a description of these special characteristics and escape sequences, see the *VSI OpenVMS I/O User's Reference Manual* [https://docs.vmssoftware.com/vsi-openvms-io-user-s-reference-manual/].

Figure 2. Default Characteristics for Terminals

Name	Unknown	Foreign	LA12	LA34 LA38 LA100 LQP02	LA36	LA120	LA210 LN03 LN01K	VT101	VT102
ADVANCED_VIDEO	No	*	No	No	No	No	No	No	Yes
ALTYPEAHD	No	*	*	*	*	*	*	*	*
ANSI_CRT	No	*	No	No	No	No	No	Yes	Yes
APPLICATION_KEYPAD /NUMERIC_KEYPAD	*	*	*	*	*	*	*	*	*
AUTOBAUD	*	*	*	*	*	*	*	*	*
BLOCK_MODE	No	*	No	No	No	No	No	No	No
BRDCSTMBX	*	*	*	*	*	*	*	*	*
BROADCAST	*	*	*	*	*	*	*	*	*
CRFILL	0	0	0	0	0	0	0	0	0
DEC_CRT	No	*	No	No	No	No	No	Yes	Yes
DEC_CRT2	No	*	No	No	No	No	No	No	No
DEC_CRT3	No	*	No	No	No	No	No	No	No
DEC_CRT4	No	*	No	No	No	No	No	No	No
DEC_CRT5	No	No	No	No	No	No	No	No	No
DIALUP	*	*	*	*	*	*	*	*	*
DISCONNECT	*	*	*	*	*	*	*	*	*
DMA	*	*	*	*	*	*	*	*	*
ECHO	*	*	*	*	*	*	*	*	*
EDIT_MODE	No	*	No	No	No	No	No	No	Yes
EIGHT_BIT	*	*	No	No	No	No	Yes	No	No
ESCAPE	*	*	*	*	*	*	*	*	*
FALLBACK	*	*	*	*	*	*	*	*	*
FORM	*	*	Yes	Yes	No	Yes	Yes	No	No
FRAME	*	*	*	*	*	*	*	*	*
FULLDUP/HALFDUP	*	*	*	*	*	*	*	*	*
HANGUP	*	*	*	*	*	*	*	*	*
HARDCOPY/SCOPE	*	*	Hard	Hard	Hard	Hard	Hard	Scope	Scope
HOSTSYNC	*	*	*	*	*	*	*	*	*
INSERT/OVERSTRIKE	*	*	*	*	*	*	*	*	*
LFFILL	0	0	0	0	0	0	0	0	0
LINE_EDITING	No	*	No	No	No	No	No	*	*
LOCAL_ECHO	*	*	*	*	*	*	*	*	*
LOWERCASE/UPPERCASE	*	*	Low	Low	Low	Low	Low	Low	Low
MODEM	*	*	*	*	*	*	*	*	*
PAGE	*	*	66	66	66	66	66	24	24
PARITY	*	*	*	*	*	*	*	*	*
PRINTER_PORT	No	*	No	No	No	No	No	No	Yes
READSYNC	*	*	*	*	*		*	*	*
REGIS	No	*	No	No	No	No	No	No	No
REMOTE	*	*	*	*	*	*	*	*	*
SECURE_SERVER	*	*	*	*	*	*	*	*	*
SET_SPEED	*	*	*	*	*	*	*	*	*
SIXEL_GRAPHICS	No	*	No	No	No	No	Yes	No	No
SOFT_CHARACTERS	No	*	No	No	No	No	No	No	No
SPEED	*	*	*	*	*	*	*	*	*
SYSPASSWORD	*	*	*	*	*	*	*	*	*
TAB	*	*	Yes	No	No	Yes	No	Yes	Yes
TTSYNC	*	*	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TYPE_AHEAD	*	*	*	*	*	*	*	*	*
WIDTH	*	*	80	132	132	132	80	80	80
WRAP	*	*	Yes	Yes	Yes	Yes	Yes	Yes	Yes

* Indicates that the current setting is not affected by terminal type.
@ Optional terminal feature.

ZK-5170a-AI

Name	VT100 VT105	VT125	VT131 VT132	VT173	VT200	VT300	VT400	VT500
ADVANCED_VIDEO	@	@	Yes	Yes	Yes	Yes	Yes	Yes
ALTYPEAHD	*	*	*	*	*	*	*	*
ANSI_CRT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
APPLICATION_KEYPAD /NUMERIC_KEYPAD	*	*	*	*	*	*	*	*
AUTOBAUD	*	*	*	*	*	*	*	*
BLOCK_MODE	No	No	Yes	Yes	No	No	No	No
BRDCSTMBX	*	*	*	*	*	*	*	*
BROADCAST	*	*	*	*	*	*	*	*
CRFILL	0	0	0	0	0	0	0	0
DEC_CRT	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
DEC_CRT2	No	No	No	No	Yes	Yes	Yes	Yes
DEC_CRT3	No	No	No	No	No	Yes	Yes	Yes
DEC_CRT4	No	No	No	No	No	No	Yes	Yes
DEC_CRT5	No	No	No	No	No	No	No	Yes
DIALUP	*	*	*	*	*	*	*	*
DISCONNECT	*	*	*	*	*	*	*	*
DMA	*	*	*	*	*	*	*	*
ECHO	*	*	*	*	*	*	*	*
EDIT_MODE	@	@	@	Yes	Yes	Yes	Yes	Yes
EIGHT_BIT	No	No	No	No	Yes	Yes	Yes	Yes
ESCAPE	*	*	*	*	*	*	*	*
FALLBACK	*	*	*	*	No	No	No	No
FORM	No	No	No	No	No	No	No	No
FRAME	*	*	*	*	*	*	*	*
FULLDUP/HALFDUP	*	*	*	*	*	*	*	*
HANGUP	*	*	*	*	*	*	*	*
HARDCOPY/SCOPE	Scope	Scope	Scope	Scope	Scope	Scope	Scope	Scope
HOSTSYNC	*	*	Yes	Yes	*	*	*	*
INSERT/OVERSTRIKE	*	*	*	*	*	*	*	*
LFFILL	0	0	0	0	0	0	0	0
LINE_EDITING	*	*	*	No	*	*	*	*
LOCAL_ECHO	*	*	*	*	*	*	*	*
LOWERCASE/UPPERCASE	Low	Low	Low	Low	Low	Low	Low	Low
MODEM	*	*	*	*	*	Yes	Yes	Yes
PAGE	24	24	24	16	24	24	24	24
PARITY	*	*	*	*	*	*	*	*
PRINTER_PORT	@	Yes	No	No	@	Yes	Yes	Yes
READSYNC	*	*	*	*	*	*	*	*
REGIS	No	Yes	No	No	@	@	@	@
REMOTE	*	*	*	*	*	*	*	*
SECURE_SERVER	*	*	*	*	*	*	*	*
SET_SPEED	*	*	*	*	*	*	*	*
SIXEL_GRAPHICS	No	Yes	No	No	@	@	@	@
SOFT_CHARACTERS	No	No	No	No	@	@	@	@
SPEED	*	*	*	*	*	*	*	*
SYSPASSWORD	*	*	*	*	*	*	*	*
TAB	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TTSYNC	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TYPE_AHEAD	*	*	*	*	*	*	*	*
WIDTH	80	80	80	80	80	80	80	80
WRAP	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

* Indicates that the current setting is not affected by terminal type.
@ Optional terminal feature.

ZK-5171a-AI

Qualifiers

/ADVANCED_VIDEO
/NOADVANCED_VIDEO

Controls whether the terminal has advanced video attributes and is capable of 132-column video. If the terminal width is set to 132 columns and you specify the **/ADVANCED_VIDEO** qualifier,

the terminal page limit is set to 24 lines. If you specify the **/NOADVANCED_VIDEO** qualifier, the terminal page limit is set to 14 lines.

/ALTYPEAHD

Causes the terminal driver to create a permanent, alternate type-ahead buffer. The system parameter TTY_ALTYPEAHD determines the size of the type-ahead buffer.

To enable **/ALTYPEAHD**, you must also set the qualifier **/TYPE_AHEAD**.

You should specify **SET TERMINAL/PERMANENT/ALTYPEAHD** in SYS$STARTUP:SYSTARTUP_VMS.COM for those communication lines that require this capability.

To use this feature interactively, specify **SET TERMINAL/PERMANENT/ALTYPEAHD**. This specification is effective at your next login.

/ANSI_CRT (default)
/NOANSI_CRT

Controls whether the terminal conforms to ANSI CRT programming standards. Because ANSI standards are a proper subset of the DEC_CRT characteristics, the default for all VT100 family terminals is **/ANSI_CRT**.

/APPLICATION_KEYPAD

Specifies that the keypad is to be set to application keypad mode, which allows you to enter DCL commands defined with the **DEFINE/KEY** command. By default, the terminal is set to numeric keypad mode.

/AUTOBAUD
/NOAUTOBAUD

Controls whether the terminal baud rate is set when you log in and sets the default terminal speed to 9600. You must press Return two or more times at intervals of at least 1 second for the baud rate to be determined correctly. If you press a key other than Return, the **/AUTOBAUD** qualifier may detect the wrong baud rate. If this happens, wait for the login procedure to time out before continuing. The **/AUTOBAUD** qualifier must be used with the **/PERMANENT** qualifier.

The valid baud rates are as follows:

```
 50       150       1800       4800       38400
 75       300       2000       7200       57600
110       600       2400       9600       76800
134      1200       3600      19200      115200
```

/BACKSPACE=*keyword*

Controls how the system responds to the backspace key (**Ctrl/H**) in line editing mode. There are two possible keywords:

- BACKSPACE (default) – The terminal driver returns the user to the beginning of the line. (This is the traditional way OpenVMS has always worked.)
- DELETE – The terminal driver interprets the backspace key as a delete character instruction.

  Note the following exceptions:

- If the terminal is set in PASSALL or PASTHRU mode, the backspace key is not interpreted as a delete character instruction.
- If the user issues an IO$_READVBLK with IO$M_NOFILTR or IO$_READPBLK, the backspace key is not interpreted as a delete character instruction.

You can use SYSGEN to make **/BACKSPACE=DELETE** the default for all terminals by setting the system parameter TTY_DEFCHAR3 to 16.

If the default is set to DELETE, the user can still go to the start of a line by pressing **F12** or by entering the following sequence: **Ctrl/V Ctrl/H Ctrl/H**.

If you use **SET HOST**, both the local node and the remote node must be capable of responding to your definition of the BACKSPACE key.

/BLOCK_MODE
/NOBLOCK_MODE

Controls whether block mode transmission, local editing, and field protection are performed.

/BRDCSTMBX
/NOBRDCSTMBX

Controls whether broadcast messages are sent to an associated mailbox if one exists.

/BROADCAST (default)
/NOBROADCAST

Controls whether reception of broadcast messages (such as those issued by MAIL and REPLY) is enabled. Specify the **/NOBROADCAST** qualifier when you are using a terminal as a noninteractive device or when you do not want special output to be interrupted by messages. Use the **SET BROADCAST** command to exclude certain types of messages from being broadcast, rather than eliminating all messages.

/COLOR

Sets the ANSI_COLOR terminal characteristic and identifies the terminal as capable of supporting the ANSI color escape sequences.

/COMMSYNC
/NOCOMMSYNC (default)

Allows connection of asynchronous printers and other devices to terminal ports, using standard modem control signals as flow control. Transmission to the device stops if either data set ready (DSR) or clear to send (CTS) EIA modem control signals are dropped. Transmission resumes when both signals are present.

The **/COMMSYNC** qualifier and the **/MODEM** qualifier are mutually exclusive.

The COMMSYNC feature has the following limitations:

- Cannot be used on LAT ports
- Can only be used on ports with full modem control
- Should not be used in conjunction with Xon/Xoff flow control (the port may hang)

Caution

The **/COMMSYNC** qualifier should never be set on a line with a modem that is intended for interactive use. The qualifier disables the modem terminal characteristic that disconnects a user process from the terminal line in case of a modem phone line failure. With the **/COMMSYNC** qualifier enabled, the next call on the terminal line could be attached to the previous user's process. Security administrators should be aware that the characteristic should not be used on interactive terminal ports. In addition, the **/COMMSYNC** qualifier is not supported on a port connected to a LAT line.

/CRFILL[=*fill-count*]

Generates the specified number of null characters after each carriage return before transmitting the next meaningful character (to ensure that the terminal is ready for reception). The value must be an integer in the range 0 to 9. The default is the **/CRFILL=0** qualifier.

/DEC_CRT[= (*value1*,*value2*,*value3*)]
/NODEC_CRT[= (*value1*,*value2*,*value3*)]

Controls whether the terminal conforms to DIGITAL VT100-,VT200-,VT300-,VT400-, or VT500-family standards and supports the minimum standards, including the additional DIGITAL escape sequences.

You can specify one of the following values:

Value	Description
1 (default)	Requests that the DEC_CRT terminal characteristic be set.
2	Requests that the DEC_CRT2 terminal characteristic be set.
3	Requests that the DEC_CRT3 terminal characteristic be set. A level 3 terminal supports the following additional features: ● A status line (line 25, at the bottom of the screen) ● The ISO Latin1 character set ● Terminal state interrogation (describes what state your terminal is in)
4	Requests that the DEC_CRT4 terminal characteristic be set. A level 4 terminal supports the following additional features: ● Extended keyboard ● Key position mode ● Secure reset ● Novice mode ● Selective erase ● On-line transaction processing (OLTP) features: • Page memory

Value	Description
	• Rectangular editing • Text macros • Data integrity reports

Note that DEC_CRT2, DEC_CRT3, and DEC_CRT4 are supersets of DEC_CRT. Clearing DEC_CRT causes DEC_CRT2, DEC_CRT3, and DEC_CRT4 to be cleared. Similarly, setting DEC_CRT4 causes all subsets of DEC_CRT4 (including ANSI_CRT) to be set.

/DEVICE_TYPE=*terminal-type*

Informs the system of the terminal type and sets characteristics according to the device type specified. You can specify any of the following terminal types:

UNKNOWN	LA100	PRO_SERIES	VT102	VT200
FT1--FT8	LA120	VT05	VT105	VT300
LA12	LA210	VT52	VT125	VT400
LA34	LN01K	VT55	VT131	VT500
LA36	LN03	VT100	VT132	
LA38	LQP02	VT101	VT173	

The default characteristics for the VT100-, VT102-, and VT125-series terminals are as follows:

/ADVANCEDVIDEO	**/NOALTYPEAHD**[1]	**/ANSI_CRT**
/NOAUTOBAUD	**/NOBLOCK_MODE**	**/NOBRDCSTMBX**
/BROADCAST	**/CRFILL=0**	**/ECHO**
/NOEIGHT_BIT	**/NOESCAPE**	**/NOFORM**
/FULLDUP	**/NOHOSTSYNC**	**/LFFILL=0**
/LOWERCASE	**/NODMA**	**/PAGE=24**
/NOPARITY	**/NOPASTHRU**	**/NOREADSYN**
/SPEED=9600	**/TAB**	**/TTSYNC**
/TYPE_AHEAD	**/WIDTH=80**	**/WRAP**

[1]This is the default characteristic set by the system and is not a valid qualifier for your use.

The terminal types and characteristics that can be set are listed in *Figure 2, "Default Characteristics for Terminals"*.

/DIALUP
/NODIALUP (default)

Controls whether the terminal is a dialup terminal.

/DISCONNECT
/NODISCONNECT (default)

Controls whether the process connected to this terminal is disconnected if the line detects a hangup. The **/DISCONNECT** qualifier is valid only when the **/PERMANENT** qualifier is specified.

/DISMISS
/NODISMISS (default)

Controls whether the terminal driver ignores data that causes a parity error (instead of terminating the currently outstanding I/O with an error status).

/DMA
/NODMA

Controls whether direct memory access (DMA) mode is used on a controller that supports this feature.

/ECHO (default)
/NOECHO

Controls whether the terminal displays the input it receives. With the **/NOECHO** qualifier, the terminal displays only system or user application output, or both.

/EDIT_MODE
/NOEDIT_MODE

Controls whether the terminal can perform ANSI-defined advanced editing functions.

/EIGHT_BIT
/NOEIGHT_BIT

Controls whether the terminal uses the 8-bit ASCII protocol rather than the 7-bit ASCII protocol. You can use the Terminal Fallback Facility (TFF) to set the 8-bit characteristic on terminals. If the terminal you specify has the TFF enabled, the /EIGHT_BIT qualifier has no effect. For more information on terminal fallback, see the *OpenVMS Terminal Fallback Utility Manual*.

/ESCAPE
/NOESCAPE (default)

Controls whether escape sequences are validated.

/FALLBACK
/NOFALLBACK

Controls whether the 8-bit DEC Multinational character set characters are displayed on the terminal in their 7-bit representation. The default depends on the /EIGHTBIT setting of the terminal. If the OpenVMS Terminal Fallback Facility (TFF) is enabled, it activates the default character conversion tables for the named terminal. For more information, see the *OpenVMS Terminal Fallback Utility Manual*. If TFF is not enabled on your system, the **/FALLBACK** qualifier has no effect and no error message is displayed.

/FORM
/NOFORM

Controls whether a form feed is transmitted rather than translated into multiple line feeds.

/FRAME=*n*

Specifies the number of data bits that the terminal driver expects for every character that is input or output. The value of *n* can be from 5 to 8. The default value depends on the settings for the terminal established by the **/PARITY** and **/EIGHTBIT** qualifiers.

/FULLDUP (default)
/NOFULLDUP

Controls whether the terminal operates in full-duplex mode. The **/FULLDUP** qualifier is equivalent to the **/NOHALFDUP** qualifier.

/HALFDUP
/NOHALFDUP (default)

Controls whether the terminal operates in half-duplex mode. The **/HALFDUP** qualifier is equivalent to the **/NOFULLDUP** qualifier.

/HANGUP
/NOHANGUP (default)

May require LOG_IO (logical I/O) or PHY_IO (physical I/O) privilege depending on system generation parameter settings.

Controls whether the terminal modem is hung up when you log out.

/HARDCOPY
/NOHARDCOPY

Controls whether the device is established as a hardcopy terminal and outputs a backslash (\) when the Delete key is pressed. The **/HARDCOPY** qualifier is equivalent to the **/NOSCOPE** qualifier.

/HOSTSYNC
/NOHOSTSYNC (default)

Controls whether system transmission from the terminal is stopped (by generating a **Ctrl/S**) when the input buffer is full and resumed (by generating a **Ctrl/Q**) when the input buffer is empty.

/INQUIRE

Sets the device type when the **/INQUIRE** qualifier is specified and the DEC_CRT characteristic is set. The **SET TERMINAL** command reads the current screen size from the terminal and sets the corresponding page length and page width values appropriately. The default device type is UNKNOWN. Works only on DIGITAL terminals, and not on LA36 or VT05 terminals. Some VT100 family terminals, including the VT101 and VT105, return a VT100 type response. LA38 terminals respond as LA43 terminals.

Note

The **SET TERMINAL/INQUIRE** command works correctly on DIGITAL supplied VT100 and later terminals. Some personal computer terminal emulators may not work correctly, because they do not correctly emulate all VT100 escape sequences. VSI recommends that users who experience problems with these terminal emulators contact the terminal emulator supplier.

You can include the **SET TERMINAL/INQUIRE** command in your LOGIN.COM file to detect the terminal type automatically.

If you specify **/INQUIRE=OLD**, OpenVMS sets the terminal window to 24 lines by 80 columns and ignores the real terminal size. (This is the behavior of the **SET TERMINAL/INQUIRE** command prior to OpenVMS Version 6.2.)

Caution

This qualifier clears the type-ahead buffer. If the response sequence is unrecognized, no action message or error message is displayed. The **/INQUIRE** qualifier should be used only on DIGITAL terminals; however, the LA36 and VT05 terminals do not support this feature.

/INSERT

Sets the terminal to insert mode. This feature allows you to insert characters when editing command lines. The default mode is overstrike, which allows you to type over the current character when editing a command line. Press **Ctrl/A** to switch from one mode to the other.

/LFFILL[=*fill-count*]

Transmits to the terminal the specified number of null characters after each line feed before transmitting the next meaningful character (to ensure that the terminal is ready for reception). The value must be an integer in the range 0 to 9. The default is installation dependent. See *Figure 2, "Default Characteristics for Terminals"* for a list of default terminal characteristics.

/LINE_EDITING
/NOLINE_EDITING

Controls whether advanced line-editing features are enabled for editing command lines: pressing **Return** and pressing **Ctrl/Z** are recognized as line terminators, as are escape sequences.

/LOCAL_ECHO
/NOLOCAL_ECHO (default)

Controls whether the terminal echoes characters locally (rather than the host echoing them) for command level terminal functions. (Do not use the **/LOCAL_ECHO** qualifier with utilities that require control over echoing, such as line editing or EDT's screen mode.)

Caution

When logging in to terminals with the LOCAL_ECHO characteristic, the OpenVMS system has no control over the echoing of passwords.

/LOWERCASE
/NOLOWERCASE

Controls whether lowercase characters are passed to the terminal. The **/NOLOWERCASE** qualifier translates all input to uppercase. The **/LOWERCASE** qualifier is equivalent to the **/NOUPPERCASE** qualifier.

/MANUAL

Indicates manual switching of terminal lines to dynamic asynchronous DDCMP lines when your local terminal emulator does not support automatic switching. The **/MANUAL** qualifier should be specified with the **/PROTOCOL=DDCMP** and **/SWITCH=DECNET** qualifiers.

/MODEM
/NOMODEM

Specifies whether the terminal is connected to a modem or a cable that supplies standard EIA modem control signals. If your terminal has the MODEM characteristic, entering

SET TERMINAL/NOMODEM automatically logs you out. The **/MODEM** qualifier and the **/COMMSYNC** qualifier are mutually exclusive.

/NUMERIC_KEYPAD (default)

Specifies whether the keys of the numeric keypad are used to type numbers and punctuation marks (**/NUMERIC_KEYPAD**) or to enter DCL commands defined with the **DEFINE/KEY** command (**/APPLICATION_KEYPAD**).

/OVERSTRIKE (default)

Sets the terminal to overstrike mode. This feature allows you to type over the current character when you are editing a command line. Set your terminal to insert mode if you want to insert characters when editing command lines. Press **Ctrl/A** to switch from one mode to the other.

/PAGE[=*lines-per-page*]

Specifies the number of print lines between perforations for hardcopy terminals. (When the terminal reads a form feed, it advances the paper to the next perforation.) The value of the *lines-per-page* parameter can be from 0 to 255 and defaults to 0 (which treats a form feed as a line feed).

/PARITY[=*option*]
/NOPARITY (default)

Controls whether the terminal passes data with only odd or even parity, where *option* equals ODD or EVEN. If you specify the **/PARITY** qualifier without an option, the value defaults to EVEN.

/PASTHRU
/NOPASTHRU (default)

Controls whether the terminal passes all data (including tabs, carriage returns, line feeds, and control characters) to an application program as binary data. The setting of **/TTSYNC** is allowed.

Make sure that you spell both these qualifiers exactly as they appear in the text.

/PERMANENT

Requires LOG_IO (logical I/O) or PHY_IO (physical I/O) privilege.

Sets characteristics on a permanent basis, that is, over terminal sessions; however, the characteristics revert to their initial values if the system is halted and restarted. Use in a system startup file to establish characteristics for all terminals on the system.

/PRINTER_PORT
/NOPRINTER_PORT

Specifies whether the terminal has a printer port (an attribute not set by the **SET TERMINAL/INQUIRE** command). The default is installation dependent. See *Figure 2, "Default Characteristics for Terminals"* for a list of default terminal characteristics.

/PROTOCOL=DDCMP
/PROTOCOL=NONE (default)

On VAX, controls whether the terminal port specified is changed into an asynchronous DDCMP line. The **/PROTOCOL=NONE** qualifier changes an asynchronous DDCMP line back into a terminal line. Note that **/PROTOCOL=DDCMP** is a permanent characteristic; therefore, the **/PERMANENT** qualifier is not required.

/READSYNC
/NOREADSYNC (default)

Controls whether the terminal uses the **Ctrl/S** and **Ctrl/Q** functions to synchronize data transmitted from the terminal.

Caution

`SET TERMINAL/READSYNC` should not be used on LAT terminal lines. Setting this characteristic may cause unexpected results.

The default is the **`/NOREADSYNC`** qualifier; the system does not use the **Ctrl/S** and **Ctrl/Q** functions to control reads to the terminal. The **`/READSYNC`** qualifier is useful for certain classes of terminals that demand synchronization or for special-purpose terminal lines where data synchronization is appropriate.

/REGIS
/NOREGIS

Specifies whether the terminal understands ReGIS graphic commands.

/SCOPE
/NOSCOPE

Controls whether the device is established as a video terminal. The **`/SCOPE`** qualifier is equivalent to the **`/NOHARDCOPY`** qualifier.

/SECURE_SERVER
/NOSECURE_SERVER (default)

Requires either LOG_IO (logical I/O) or PHY_IO (physical I/O) privilege.

Controls whether the Break key on the terminal logs out the current process (except on a virtual terminal). With the **`/SECURE_SERVER`** qualifier in effect, pressing the Break key when there is no current process initiates the login sequence. With the **`/NOSECURE_SERVER`** qualifier in effect, the break is ignored.

On terminals with the AUTOBAUD and SECURE_SERVER characteristics, pressing the Break key disconnects the current process, but is not required to start a new login sequence. However, when the NOAUTOBAUD characteristic is set, the SECURE_SERVER characteristic requires a break to initiate a new login sequence.

/SET_SPEED
/NOSET_SPEED

Requires either LOG_IO (logical I/O) or PHY_IO (physical I/O) privilege.

Controls whether the **`/SPEED`** qualifier can be used to change the terminal speed.

/SIXEL_GRAPHICS
/NOSIXEL_GRAPHICS

Specifies whether the terminal is capable of displaying graphics using the sixel graphics protocol. The default is device dependent. For a list of default terminal characteristics, see *Figure 2, "Default Characteristics for Terminals"*.

/SOFT_CHARACTERS
/NOSOFT_CHARACTERS

Specifies whether the terminal is capable of loading a user-defined character set. The default is device dependent. See *Figure 2, "Default Characteristics for Terminals"* for a list of default terminal characteristics.

/SPEED=(*input-rate*,*output-rate*)

Sets the baud rate at which the terminal receives and transmits data. If the input and output rates are the same, specify **/SPEED=*rate***.

Not all terminals support different input and output baud rates. For specific information on baud rates for your terminal, consult the manual for that terminal.

The default transmission rates are installation dependent.

The valid values for input and output baud rates are as follows:

```
 50       150       1800       4800       38400
 75       300       2000       7200       57600
110       600       2400       9600       76800
134      1200       3600      19200      115200
```

/SWITCH=DECNET

On VAX, causes the terminal lines at each node to be switched to dynamic asynchronous DDCMP lines, when specified with the **/PROTOCOL=DDCMP** qualifier. Note that **/SWITCH=DECNET** is a permanent characteristic; therefore, the **/PERMANENT** qualifier is not required.

/SYSPASSWORD
/NOSYSPASSWORD (default)

Requires LOG_IO (logical I/O) privilege.

Determines whether the terminal requires that a system password be entered before the Username: prompt.

/TAB
/NOTAB

Controls whether tab characters are converted to multiple blanks. The **/NOTAB** qualifier expands all tab characters to blanks and assumes tab stops at 8-character intervals. The default is device dependent. For a list of default terminal characteristics, see *Figure 2, "Default Characteristics for Terminals"*.

/TTSYNC (default)
/NOTTSYNC

Controls whether transmitting to the terminal is stopped when **Ctrl/S** is pressed and resumes transmission when **Ctrl/Q** is pressed.

/TYPE_AHEAD (default)
/NOTYPE_AHEAD

Controls whether the terminal accepts unsolicited input to the limit of the type-ahead buffer.

When you specify the **/NOTYPE_AHEAD** qualifier, the terminal accepts input only when a program or the system issues a read to the terminal, such as for user input at the DCL prompt ($). When you specify the **/TYPE_AHEAD** qualifier, the amount of data that can be accepted is governed by the size of the type-ahead buffer. That size is determined by system generation parameters.

/UNKNOWN

Specifies a terminal type that is unknown to the system, which then uses the default terminal characteristics for unknown terminals. For a summary of the settings, see *Figure 2, "Default Characteristics for Terminals"*.

/UPPERCASE
/NOUPPERCASE

Controls whether lowercase characters are translated to uppercase. The **/UPPERCASE** qualifier is equivalent to the **/NOLOWERCASE** qualifier.

/WIDTH=*characters-per-line*

Specifies the maximum characters per line. This value must be an integer in the range 1 to 511. With the **/WRAP** qualifier, the terminal generates a carriage return and line feed when the width specification is reached.

If the specified width on an ANSI terminal is 132, the screen is set to 132-character mode. If the terminal does not have advanced video option (AVO), the page length limit is set to 14 lines.

/WRAP (default)
/NOWRAP

Controls whether a carriage return and line feed are generated when the value of the **/WIDTH** qualifier is reached.

Examples

1.
```
$ SET TERMINAL/DEVICE=VT102
```

In this example, the **SET TERMINAL** command establishes the current terminal as a VT102 terminal and sets the default characteristics for that terminal type.

2.
```
$ SET TERMINAL/WIDTH=132/PAGE=60/NOBROADCAST
$ TYPE MEMO.DOC
     .
     .
     .
$ SET TERMINAL/DEVICE=LA36
```

In this example, the first **SET TERMINAL** command indicates that the width of terminal lines is `132` characters and that the size of each page is `60` lines. The **/NOBROADCAST** qualifier disables the reception of broadcast messages while the terminal is printing the file `MEMO.DOC`. The next **SET TERMINAL** command restores the terminal to its default state.

SET TIME

SET TIME — Resets the system clock, which is used as a timer to record intervals between various internal events and as a source clock for displaying the time of day. **Requires both OPER (operator) and LOG_IO (logical I/O) privileges.**

Format

SET TIME[=*time*]

Parameter

time

Specifies a date in the format *day-month-year*, or a time in the format *hour*:*minute*:*second*.*hundredth*, or both. These fields accept the following values:

Field	Type	Values
Day	Integer	1 to 31
Month	String	JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, or DEC
Year	Integer	1858 to 9999
Hour	Integer	0 to 23
Minute	Integer	0 to 59
Second	Integer	0 to 59
Hundredth	Integer	0 to 99

The hyphens (-), colons (:), and period (.) are required delimiters. Separate the date and time, when both are specified, with a colon. The syntax is sometimes specified as follows:

```
[dd-mmm-yyyy[:]] [hh:mm:ss.cc]
```

Alpha systems also allow a 2-digit year format:

```
[dd-mmm-yy[:]] [hh:mm:ss.cc]
```

The century applied to the 2-digit year format is determined by the corresponding year in the supported Alpha date range (1957-2056). A 2-digit year in the range 57-99 is assumed to mean the years 1957-1999; a 2-digit year in the range 00-56 is assumed to mean the years 2000-2056. (To specify years outside the 1957-2056 range, you must enter a 4-digit year.)

If the explicit time value is not specified, the interval system clock is automatically reset according to the time-of-year clock.

For further information about the time-of-year clock, see the hardware documentation for your computer.

Note

Alpha systems maintain system time during power failures and system down time. When a system is booted, if the time is known to be earlier than the time value of the last time modification, or greater than five years in the future, you are prompted to enter the time at the console prompt.

Description

Resets the system clock, which is used as a timer to record intervals between various internal events and as a source clock for displaying the time of day.

Note

The **SET TIME** command will not work if there is a time service available on your system or OpenVMS Cluster.

Qualifier

/CLUSTER

Requires OPER and LOG_IO privilege, and in an OpenVMS Cluster environment, SYSLCK privilege.

Sets the time on all nodes in a cluster to the same system time. The format of the **SET TIME /CLUSTER** command is the following:

SET TIME[=*time*] [/CLUSTER]

Examples

1.
```
$ SET TIME=14-DEC-2001:19:31:0.0
```

   The **SET TIME** command in this example sets the date and time to December 14, 2001, 7:31 P.M.

2.
```
$ SET TIME
$ SHOW TIME
14-DEC-2001 03:21:27.53
```

   The **SET TIME** command in this example sets the system time according to the time-of-year clock. The **SHOW TIME** command requests a display of the current time.

3.
```
$ SET TIME=14-DEC-2001:15:31:0.0/CLUSTER
```

   This example sets the date and time to December 14, 2001, 3:31 P.M. on all nodes in the cluster where the command was entered.

SET VERIFY

SET VERIFY — Controls whether command lines and data lines in command procedures are displayed at the terminal or are printed in a batch job log. The information displayed by the **SET VERIFY** command can help you in debugging command procedures.

Format

SET VERIFY [=([NO]PROCEDURE, [NO]IMAGE)]

SET NOVERIFY

Parameter

([NO]PROCEDURE, [NO]IMAGE)

Specifies one or both types of verification. Procedure verification causes each DCL command line in a command procedure to be written to the output device. Image verification causes data lines (input data that is included as part of the SYS$INPUT input stream) to be written to the output device.

By default, both types of verification are set or cleared with **SET VERIFY** and **SET NOVERIFY**. If you specify only one keyword, the other is not affected. If you specify only one keyword, omit the parentheses.

Description

By default, the **SET VERIFY** and **SET NOVERIFY** commands set or clear both types of verification. The default setting for command procedures executed interactively is **SET NOVERIFY**. System responses and error messages are, however, always displayed. The default for batch jobs is **SET VERIFY**.

If you use the **SET VERIFY** command to override the default setting, the system displays each command and data line in the command procedure as it reads it. When verification is in effect, the command interpreter displays each command line after it has completed initial scanning and before the command is parsed and executed. You see the results of symbol substitution performed during scanning, but not the results of symbol substitution performed during parsing and evaluation.

When you change the verification setting, the new setting remains in effect for all command procedures that you subsequently execute.

A file must have read (R) access to allow the **SET VERIFY** command to set verification.

Examples

1.
```
$ SET VERIFY = PROCEDURE
```

   In this example, procedure verification is turned on. If image verification was on, it remains on; if image verification was off, it remains off.

2.
```
$ SET VERIFY
$ INDEX == "$INDEX.EXE"
$ CONTENTS == "$CONTENTS.EXE"
$ TABLE == "$TABLE.EXE"
$ SET NOVERIFY
$ EXIT
```

   Procedure and image verification are turned on at the beginning of the command procedure so that the system displays all the command and data lines in the procedure as it reads them. At the end of the procedure, the **SET NOVERIFY** command restores the system default (no procedure or image verification).

3.
```
$ PROC_VER = F$ENVIRONMENT("VERIFY_PROCEDURE")
$ IMAGE_VER = F$ENVIRONMENT("VERIFY_IMAGE")
$ HOLD_PREFIX = F$ENVIRONMENT("VERIFY_PREFIX")
$ SET PREFIX "(!%T)"
$ SET VERIFY
    .
    .
    .
$ TEMP = F$VERIFY(PROC_VER, IMAGE_VER)
$ SET PREFIX "''HOLD_PREFIX'"
```

   This command procedure uses the F$ENVIRONMENT lexical function to save the current procedure and image verification setting, as well as the current verification prefix string. The **SET PREFIX** command sets the verification prefix to be used in the current command procedure. It uses an FAO control string to produce the time each command is read by the command interpreter (DCL),

surrounded by parentheses. Then the **SET VERIFY** command turns on both procedure and image verification. Subsequently, the F$VERIFY lexical function is used to restore the original verification settings. The **SET PREFIX** command returns the verification prefix to its previous setting. Note how the symbol HOLD_PREFIX is used in the **SET PREFIX** command. This preserves casing and special characters in the stored string.

4.
```
$ SET VERIFY
$ @TEST
$ RUN AVERAGE
1
2
3
$ EXIT
```

In this example, the **SET VERIFY** command turns procedure and image verification on. When the command procedure TEST.COM is executed interactively, the command lines and the data lines for the program AVERAGE are displayed on the terminal. The data lines were entered in the command procedure on lines that did not begin with the DCL prompt.

5.
```
$ SET VERIFY
$ COUNT = 1
$ IF P'COUNT' .NES. "" THEN GOTO &P'COUNT'
    .
    .
    .
$ EXIT
```

When this command procedure is executed interactively, the **SET VERIFY** command causes the command and data lines to be displayed. Symbols that are substituted during the first phase of symbol substitution (such as 'COUNT') are displayed by the **SET VERIFY** command, but other symbols are not. The following lines are displayed when this procedure is executed interactively:

```
$ COUNT = 1
$ IF P1 .NES. "" THEN GOTO &P1
    .
    .
    .
```

Although these values are not displayed, the value for P1 is substituted during the third phase of symbol substitution, and the value for &P1 is substituted during the second phase.

SET VOLUME

SET VOLUME — Changes the characteristics of one or more mounted Files-11 volumes. The **SET VOLUME** command affects only the node on which the command is issued, and will not affect any other nodes, even when a volume is mounted on multiple nodes in an OpenVMS Cluster configuration. **If you are not the owner of the volume, requires control access to the volume.**

Format

SET VOLUME *device-name*[:][,...]

Parameter

***device-name*[:][,...]**

Specifies the name of one or more mounted Files-11 volumes.

Qualifiers

/ACCESSED[=*n*]

Requires OPER (operator) privilege.

Specifies the number of directories to be maintained in system space for ready access. You can specify a value *n* in the range of 0 to 255. If you specify the qualifier **/ACCESSED** and omit the number of directories, a default value of 3 is used. If you specify a value greater than the current value, the new value is effective immediately; otherwise, the new value is not effective until the next time the volume is mounted.

/CACHE=(*option*[,...])

Enables or disables XFC caching for the volume. If CLEAR_DATA option is specified, then contents of the volume already in cache are cleared. (If you do not specify the **/CACHE** qualifier, XFC caching attributes of the volume as well as the contents of the volume already in cache are not affected.) Keywords are as follows:

Keyword	Description
DATA	Enables XFC caching for the volume.
[NO]DATA	Disables XFC caching for the volume.
CLEAR_DATA	Clears contents of the volume present in the XFC cache (if any).

/DATA_CHECK[=(*option*[,...])]

Defines a default for data check operations following all read and write operations to the specified volume. (If you do not specify the **/DATA_CHECK** qualifier, no checks are made.) Keywords are as follows:

Keyword	Description
[NO]READ	Performs checks following all read operations.
[NO]WRITE	Performs checks following all write operations (default).

/ERASE_ON_DELETE
/NOERASE_ON_DELETE (default)

Determines whether the space occupied by a file is overwritten with a system-specified pattern when a file on the volume is deleted.

/EXTENSION[=*n*]

Specifies the number of blocks to be used as a default extension size for all files on the volume. You can specify a value *n* in the range of 0 to 65,535. If you specify the **/EXTENSION** qualifier without specifying a value, a default value of 0 (the OpenVMS RMS default) is used.

For example, during an update operation, the extension default is used when a file increases to a size greater than its initial default allocation.

/FILE_PROTECTION= (*ownership*[:*access*][,...])

Sets the default protection to be applied to all files on the specified disk volume. Specify the *ownership* parameter as system (S), owner (O), group (G), or world (W) and the *access* parameter as read (R), write (W), execute (E), or delete (D). A null access specification means no access.

Note

This attribute is not used while the volume is in use on an OpenVMS system, but the attribute is provided to control the process use of the volume on RSX-11M systems. The OpenVMS system always uses the default file protection; the protection can be changed with the DCL command **SET PROTECTION/DEFAULT**.

/HIGHWATER_MARKING
/NOHIGHWATER_MARKING

Determines whether the file high water mark (FHM) volume attribute is set. The FHM attribute guarantees that a user cannot read data that was not written by the user. Applies to Files-11 On-Disk Structure Level 2 (ODS-2) and 5 (ODS-5) volumes only.

/LABEL=*volume-label*

Specifies a 1- to 12-character ANSI name to be encoded on the volume. The specified label remains in effect until it is changed explicitly; dismounting the volume does not affect the label. VSI strongly recommends that a volume label should consist of only alphanumeric characters, dollar signs ($), underscores (_), and hyphens (-).

Note

Changing the volume label does not change other structures that used the original volume label. For example, the DISK$*label* logical volume name is not changed nor is the device-lock name that is kept internally by OpenVMS. As a result, if you attempt to access another disk that has the same volume label as the original volume of this device, you may get error messages such as the following: "%MOUNT-F-VOLALRMNT, another volume of same label already mounted".

VSI recommends that, if you change a disk volume label, you also dismount and remount the disk on all nodes in the cluster so that the names and locks are consistent.

If you change the volume label on a nonsystem disk and the **PRODUCT INSTALL** command has been used to install software products on that disk, you must update the software product database to reflect this change. To accomplish this, first dismount and remount the disk. Then use the **PRODUCT REGISTER VOLUME** command to replace all occurrences of the old DISK$*label* with the new logical volume name that was created by the **MOUNT** command.

Note

You do not need to take this action when you change the label of the system disk. The POLYCENTER Software Installation Utility continues to use the old logical volume name until the system is rebooted. After reboot (when the system disk is remounted), the utility automatically identifies the system disk using its new logical volume name.

/LIMIT[=*n*]

Specifies that the volume is to be enabled for volume expansion. *n* specifies the upper limit of the expansion (in blocks). If no value is specified, the maximum expansion potential is set up.

The maximum expansion size depends on the cluster size defined for the volume using the **INITIALIZE/CLUSTER_SIZE** command. If the cluster size is 8 or more, the current maximum expansion supported on OpenVMS (2TB) is set up. If the cluster size is less than 8, the expansion limit is set to 65535*4096*Cluster-size because the maximum size of the bitmap is 65535 blocks. For more information about volume expansion, see the relevant section in the *VSI OpenVMS Volume Shadowing Guide* [https://docs.vmssoftware.com/vsi-openvms-volume-shadowing-guide/#dynamic_vol_extention_sec].

This command must be executed while the disk is mounted privately.

If you add additional physical storage in the future, you can use the **/SIZE** qualifier to increase the volume size.

/LOG
/NOLOG (default)

Determines whether the volume specification of each volume is displayed after the modification.

/MOUNT_VERIFICATION
/NOMOUNT_VERIFICATION

Determines whether mount verification is enabled. Mount verification prevents interruption to user input/output operations and notifies the operator of problems with the disk.

/OWNER_UIC[=*uic*]

Sets the owner user identification code (UIC) of the volume to the specified UIC. The default UIC is that of the current process. Brackets ([]) are required around the UIC. Specify the UIC by using standard UIC format as described in the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#UIC_SECTION].

/PROTECTION=(*ownership*[:*access*][,...])

Specifies the protection to be applied to the volume. The following rules apply:

- Specify the *ownership* parameter as system (S), owner (O), group (G), or world (W).
- Specify the *access* parameter as read (R), write (W), create (C), or delete (D).

The default protection is all types of access by all categories of user.

/REBUILD[=FORCE]

Recovers caching limits for a volume that was dismounted improperly. If a disk volume was dismounted improperly (such as during a system failure), and was then remounted with the **MOUNT/NOREBUILD** command, you can use **SET VOLUME/REBUILD** to recover the caching that was in effect at the time of the dismount. The FORCE option forces the disk to be rebuilt unconditionally, thus updating the free block count in the disk volume's lock value block.

/RETENTION=(*min*[,*max*])

Specifies the minimum and maximum retention times to be used by the file system to determine the expiration date for files on the volume. When a file is created, its expiration date is set to the current time plus the maximum time. Each time the file is accessed, the current time is added to the minimum time. If the sum is greater than the expiration date, a new expiration date is computed.

If you omit the maximum value, a default value that is the smaller of (2 x min) or (min + 7) days is used. For example, **/RETENTION=3-** is the same as **/RETENTION=(3-,6-)**, while **/RETENTION=10-** is the same as **/RETENTION=(10-,17-)**.

The command **SET VOLUME/RETENTION=0** disables retention times on the volume.

/SIZE[=*nnn*]

Specifies the amount of disk space (in blocks) that is usable for the file system. This value must be greater than the current logical volume size and less than or equal to the physical size of the disk or the expansion limit (specified by **/LIMIT**), whichever is smaller.

If the value is omitted, the usable space is increased to the physical size of the disk or the expansion limit, whichever is smaller.

/STRUCTURE_LEVEL=*n*

Sets the structure level of the volume. Use the value 5 to set the volume to ODS-5.

When the **/STRUCTURE_LEVEL** qualifier is used with other qualifiers, the volume structure level is set prior to the other qualifiers being performed.

You cannot use the **SET VOLUME** command to change a volume from ODS-5 to ODS-2. To reset a volume to ODS-2, see the instructions in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#_5TO2].

/SUBSYSTEM
/NOSUBSYSTEM

Enables the processing of subsystem ACEs. Requires the SECURITY privilege. By default, the disk from which you boot has protected subsystems enabled but other disks do not. For further details on subsystems, see the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#PROTECT_SUBSYSTEM].

/UNLOAD (default)
/NOUNLOAD

Specifies whether the volume is unloaded (spun down) when the DCL command **DISMOUNT** is entered.

/USER_NAME[=*user-name*]

Specifies a user name of up to 12 alphanumeric characters to be recorded on the volume. The default name is the current process user name.

/VOLUME_CHARACTERISTICS= ([[NO]HARDLINKS,] [[NO]ACCESS_DATES[=*delta-time*]])

Enables or disables hard links and automatic updates of access dates on ODS-5 volumes.

To limit the performance impact if a file is accessed frequently, you can suppress update of the access time if the change is small. A delta time is used to determine when a new access time is significant. The default value for *delta-time* is 1 second, chosen to comply with the "seconds since EPOCH" time interface required by POSIX st_atime. A site can choose a larger delta time to reduce overhead if 1-second granularity is not required.

To disable access date support on a volume, use the **SET VOLUME/VOLUME_CHARACTERISTICS=NOACCESS_DATES** command. This command

affects only the node on which the command is issued. Other nodes are not affected by the change until the next time the volume is mounted.

For additional information about ODS-5 volumes, see the *Guide to OpenVMS File Applications* [https://docs.vmssoftware.com/guide-to-openvms-file-applications/].

/WINDOWS[=*n*]

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

The value of *n* can be from 7 to 80; the default value is 7.

/WRITETHROUGH
/NOWRITETHROUGH

This qualifier only affects applications such as PATHWORKS that can request deferred writes to file headers. You use it to control whether these applications can use the deferred write feature.

Use **/NOWRITETHROUGH** to allow these applications to use the deferred write feature. When one of these applications explicitly requests a deferred write when updating a file header, control returns to the application when the I/O request has been queued; the application does not have to wait until the data is on disk.

Note that although the **SHOW DEVICES/FULL** command shows the volume status as *write-back caching enabled*, the extended file cache (XFC) will still be in write-through mode.

Use **/WRITETHROUGH** to disable the deferred write feature, so that no applications can use it. The **SHOW DEVICES/FULL** command shows the volume status as *write-through caching enabled*.

The deferred write feature is not available on Files-11 ODS-1 volumes.

Examples

1. `$ SET VOLUME/DATA_CHECK=(READ,WRITE) DKA5`

   The **SET VOLUME** command in this example requests that data checks be performed following all read and write operations to `DKA5`.

2. `$ SET VOLUME/FILE_PROTECTION=(S:RWED,O:RWED,G:RE,W:RE) DKA5`

   The **SET VOLUME** command in this example sets the default protection to be applied to all files created on volume `DKA5`. System (`S`) and owner (`O`) are granted all types of access; group (`G`) and world (`W`) are permitted only to read and execute files on `DKA5`.

3. `$ SET VOLUME/LABEL=LICENSES DKA5`

   The **SET VOLUME** command in this example encodes the label `LICENSES` on the volume `DKA5`. Note that if characters in labels are entered in lowercase, they are changed to uppercase by the **/LABEL** qualifier.

4. `$ SET VOLUME/ACCESSED=25/USER_NAME=MANAGER/LOG DKA0:`

   The **SET VOLUME** command in this example specifies that `25` directories are to be maintained in system space for ready access for the volume `DKA0`. The command also assigns the user name `MANAGER` to the volume and displays the volume specification after the volume is modified.

5. `$ SET VOLUME/REBUILD/LOG NODE$DKA2:`

```
%SET-I-MODIFIED, _NODE$DKA2: modified
```

The **SET VOLUME** command in this example causes a rebuild operation to begin on the volume that is mounted on NODE$DKA2. The **/LOG** qualifier directs the **SET VOLUME** command to display a notification message.

6.
```
$ DISMOUNT/CLUSTER/NOUNLOAD $252$DUA716:
$ MOUNT $252$DUA716: TST716
$ SET VOLUME/LIMIT $252$DUA716:
$ DISMOUNT/NOUNLOAD $252$DUA716:
$ MOUNT/CLUSTER $252$DUA716: TST716
```

This example shows how to increase the expansion limit of a volume mounted in a cluster.

7.
```
$ SET VOLUME SYS$DISK/VOLUME_CHARACTERISTICS=NOHARDLINKS
```

This command disables hard links on SYS$DISK.

8.
```
$ SET VOLUME $DKA100/CACHE=CLEAR_DATA
```

This command clears the contents of the volume $DKA100 present in the XFC cache (if any). The caching mode of the volume $DKA100 is not altered.

9.
```
$ SET VOLUME $DKA100/CACHE=DATA
```

This command enables XFC caching for the volume $DKA100. The contents of the volume $DKA100 already present in the XFC cache is not affected in anyway.

10.
```
$ SET VOLUME $DKA100/CACHE=(DATA,CLEAR_DATA)
```

This command enables XFC caching for the volume $DKA100 and clears the contents of the volume $DKA100 present in the XFC cache (if any).

SET WORKING_SET

SET WORKING_SET — Redefines the default working set size for the process, or sets an upper limit to which the working set size can be changed by an image that the process executes. Working set limits cannot be set to exceed those defined in the user authorization file (UAF). Specify the value of *n* as a number of 512-byte pagelets on Alpha. Note that the OpenVMS system rounds up this value to the nearest CPU-specific page so that the actual amount of physical memory allowed may be larger than the specified amount on Alpha.

Format

SET WORKING_SET

Parameters

None.

Description

The **SET WORKING_SET** command enables the user to change the working set size within the authorized limits. A process's working set is the collection of physical pages to which an executing image can refer. Each user is assigned a default working set size to be associated with the process created

during login. The maximum size to which any process can increase its working set is defined in the user authorization file (UAF).

Qualifiers

/ADJUST (default)
/NOADJUST

Controls whether the system can automatically adjust the size of the process working set.

/EXTENT=*n*

Specifies the maximum number of physical pages that can be resident in the working set during image execution.

The extent value must be greater than the minimum working set defined at system generation, and it must be less than or equal to the authorized extent defined in the UAF.

If you specify a value greater than the authorized extent, the command sets the working set limit at the maximum authorized value.

/LIMIT=*n*

Specifies the size to which the working set is to be reduced at image exit.

If you specify a value greater than the current quota, the quota value is also increased.

/LOG
/NOLOG (default)

Determines whether confirmation of the **SET WORKING_SET** command is displayed.

/QUOTA=*n*

Specifies the maximum number of physical pages that any image executing in the process context can request. An image can set the working set size for the process by calling the $ADJWSL (Adjust Working Set Limit) system service.

If you specify a quota value that is greater than the authorized quota, the working set quota is set to the authorized quota value.

Examples

1.
```
$ SHOW WORKING_SET
 Working Set      /Limit=  150  /Quota=  700            /Extent= 700
 Adjustment enabled    Authorized Quota=  700  Authorized Extent= 700
$ SET WORKING_SET/QUOTA=1000
%SET-I-NEWLIMS, new working set: Limit = 150 Quota = 700 Extent = 700
```

   The **SHOW WORKING_SET** command in this example displays the current limit, quota, and extent, as well as the authorized quota and authorized extent. The **SET WORKING_SET** command attempts to set a quota limiting the maximum number of pages any image can request that is greater than the authorized quota. Note from the response that the quota was not increased.

2.
```
$ SHOW WORKING_SET
 Working Set      /Limit=  150  /Quota=  350            /Extent= 350
 Adjustment enabled    Authorized Quota=  350  Authorized Extent= 350
$ SET WORKING_SET/LIMIT=100
```

```
%SET-I-NEWLIMS, new working set: Limit = 100 Quota = 350 Extent = 350
$ SHOW WORKING_SET
 Working Set      /Limit=  100  /Quota=  350            /Extent= 350
 Adjustment enabled    Authorized Quota=  350  Authorized Extent= 350
```

The **SET WORKING_SET** command in this example sets the working set size for any image in the process to `100`.

SHOW

SHOW — Displays information about the current status of a process, the system, or devices in the system.

Format

SHOW ***option***

Description

The **SHOW** command options are described individually in this manual. *Table 8, "SHOW Command Options"* lists all the **SHOW** command options.

Table 8. SHOW Command Options

Option	Displays
ACCOUNTING	Resources the current accounting file is tracking
AUDIT	Security auditing characteristics in effect on the system
BROADCAST	Message classes for which broadcast is enabled
CLUSTER	OpenVMS cluster activity and performance
CPU	Current state of the attached processor
DEFAULT	Current default device and directory
DEVICES	Status of devices in the system
DEVICES/SERVED	Status of devices served by the MSCP server on an OpenVMS cluster system
DISPLAY	Node where the output from a DECwindows application running on the current node will be displayed
DUMP_OPTIONS (x86-64 Only)	Currently defined list of alternate devices where system crash dumps can be written
ENTROPY (x86-64 Only)	Information about derived entropy sources and the status of hardware entropy sources
ENTRY	Information about a user's batch and print jobs or about specific job entries
ERROR	Error count for the CPU, memory, and physical devices
FASTPATH (Alpha/IA-64 only)	Displays fast path port assignment and usage
IMAGE	Displays information about an ELF format image file

Option	Displays
INTRUSION	Contents of the intrusion database
KEY	Key definitions created by the **DEFINE/KEY** command
LICENSE	Software product licenses active on the current node as well as list of the names attached to a license (known as the RESERVE list)
LOGICAL	Current logical name assignments
MEMORY	Availability and usage of memory resources
NETWORK	Information about network services registered on a node
PRINTER	Printer characteristics
PROCESS	Attributes of the current process, including privileges, resource quotas, memory usage, priority, and accounting information
PROTECTION	Current default protection applied to files
QUEUE	Names and types of queues that are available on the system as well as any current jobs belonging to your process
QUEUE/CHARACTERISTICS	Characteristic names and numbers that have been defined for system queues
QUEUE/FORM	Form names and numbers that have been defined for system queues
QUEUE/MANAGERS	Information about the queue manager(s) on the system or OpenVMS cluster
QUOTA	Current disk quota authorized for and used by a specific user on a specific disk
RMS_DEFAULT	Current default multiblock and multibuffer counts used by OpenVMS Record Management Services (RMS) for file operations
SECURITY	Name, class, and profile of a security object
SERVER ACME_SERVER	Information about the ACME server
SERVER REGISTRY_SERVER (Alpha/IA-64 only)	Information about the Registry server
SHADOW	Status of a specified shadow set on the system.
STATUS	Status of the current job, including accumulated CPU time, open file count, and count of I/O operations
SYMBOL	Current symbol definitions
SYSTEM	List of all processes in the system
TERMINAL	Device characteristics of a terminal
[DAY]TIME	Current date and time
TRANSLATION	Current logical name assignment
USERS	Information about users currently on the system
WORKING_SET	Current working set size limit and quota

SHOW ACCOUNTING

SHOW ACCOUNTING — Shows what resources the current accounting file is tracking.

Format

SHOW ACCOUNTING

Parameters

None.

Description

Each node on your system has its own current accounting file. You can see what resources this file is tracking using the **SHOW ACCOUNTING** command.

There are two occasions when the resources used by a process are not tracked, despite the output of the **SHOW ACCOUNTING** command:

- When you use the **RUN** (Process) command with the **/NOACCOUNTING** qualifier.
- When you use the $CREPRC system service with the PRC$M_NOACNT status flag.

Similarly, there is one occasion when the resources used by an image are always tracked, despite the output of the **SHOW ACCOUNTING** command:

- When you install an image using the **/ACCOUNTING** qualifier of the Install utility.

The **SHOW ACCOUNTING** command produces a screen display (see the Example). The keywords in the display (in uppercase) fall into two categories:

- Keywords that show which types of resource are being tracked:

Keyword	Type of Resource
IMAGE	Resources used by an image
LOGIN_FAILURE	Resources used by an unsuccessful attempt to log in
MESSAGE	(Unformatted resource record written to the accounting file by a call to the $SNDJBC system service)
PRINT	Resources used by a print job
PROCESS	Resources used by a process

- Keywords that show which types of process are being tracked. When the resources for processes or images are tracked, these keywords show the process type:

Keyword	Type of Process
BATCH	Batch process
DETACHED	Detached process
INTERACTIVE	Interactive process

Keyword	Type of Process
NETWORK	Network process
SUBPROCESS	Subprocess (the parent process can be a batch, detached, interactive, or network process)

Qualifier

/OUTPUT[=*filespec*]

Requires read (R) and write (W) access to the directory in which the specified file is created.

Controls the output destination of the command. By default, the output is the current SYS$OUTPUT device. To send the output to a file, use the **/OUTPUT** qualifier followed by the file specification.

If the file specification does not include the file name or file type, the defaults SHOW and .LIS are used respectively.

Example

```
$ SHOW ACCOUNTING
Accounting is currently enabled to log the following activities:

      PROCESS        any process termination
      IMAGE          image execution
      INTERACTIVE    interactive job termination
      LOGIN_FAILURE  login failures
      NETWORK        network job termination
      PRINT          all print jobs
```

The keywords in this example show that the local node is tracking the resources used by each:

- Interactive and network process
- Image running in an interactive or network process
- Login failure
- Print job

SHOW AUDIT

SHOW AUDIT — Displays the security auditing characteristics in effect on the system. **Requires the SECURITY privilege.**

Format

SHOW AUDIT

Parameters

None.

Description

The **SHOW AUDIT** command displays the current state of security auditing for the system. The display can include the state of the audit journal, the characteristics of the audit server, and the events for which auditing is enabled. If no auditing has been enabled, the display reports that security alarms and audits are currently disabled.

SHOW AUDIT and **SET AUDIT** provide the principal management interface to the security auditing system.

Qualifiers

/ALL

Displays all available auditing information including the following:

- Location of the system security audit log file
- Security events enabled for auditing
- Location of the security archive file
- Audit server characteristics, such as the action taken if the audit server runs out of memory.

/ALARM

Displays the categories of events that are currently enabled; these events will generate messages on any operator's terminal accepting security class messages.

/ARCHIVE

Displays the name and location of the security archive file (if archiving is enabled).

/AUDIT

Displays the categories of events that are currently enabled to write messages to the system security audit log file.

/EXACT

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify a search string that must match the search string exactly and must be enclosed with quotation marks (" ").

If you specify the **/EXACT** qualifier without the **/SEARCH** qualifier, exact search mode is enabled when you set the search string with the Find (E1) key.

/HIGHLIGHT[=*keyword*]

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify the type of highlighting you want when a search string is found. When a string is found, the entire line is highlighted. You can use the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting.

/JOURNAL

Displays characteristics of the system audit journal.

/OUTPUT[=*filespec*]

Controls where the output of the command is sent. If you do not enter the **/OUTPUT** qualifier or if you enter it without a file specification, the output is sent to the default output stream or device for the current process, which is identified by the logical name SYS$OUTPUT.

If you enter the **/OUTPUT** qualifier with a partial file specification (for example, only a directory name), **SET AUDIT** assigns the file name **SHOW** with the default file type of .LIS. The file specification cannot include the asterisk (*) and the percent sign (%) wildcard characters.

/PAGE[=*keyword*]
/NOPAGE (default)

Controls the display of information on the screen.

You can use the following keywords with the **/PAGE** qualifier:

Keyword	**Description**
CLEAR_SCREEN	Clears the screen before each page is displayed.
SCROLL	Displays information one line at a time.
SAVE[= *n*]	Enables screen navigation of information, where *n* is the number of pages to store.

The **/PAGE=SAVE** qualifier allows you to navigate through screens of information. The **/PAGE=SAVE** qualifier stores up to 5 screens of up to 255 columns of information. When you use the **/PAGE=SAVE** qualifier, you can use the following keys to navigate through the information:

Key Sequence	**Description**
Up arrow key, **Ctrl/B**	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.
Right arrow key	Scroll right one column.
Find (**E1**)	Specify a string to find when the information is displayed.
Insert Here (**E2**)	Scroll right one half screen.
Remove (**E3**)	Scroll left one half screen.
Select (**E4**)	Toggle 80/132 column mode.
Prev Screen (**E5**)	Get the previous page of information.
Next Screen (**E6**), **Return**, **Enter**, **Space**	Get the next page of information.
F10, **Ctrl/Z**	Exit. (Some utilities define these differently.)
Help (**F15**)	Display utility help text.
Do (**F16**)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.

The **/PAGE** qualifier is not compatible with the **/OUTPUT** qualifier.

/SEARCH="*string*"

Use with the **/PAGE=SAVE** qualifier to specify a string that you want to find in the information being displayed. Quotation marks are required for the **/SEARCH** qualifier, if you include spaces in the text string.

You can also dynamically change the search string by pressing the Find key (E1) while the information is being displayed. Quotation marks are not required for a dynamic search.

/SERVER

Displays audit server characteristics.

/WRAP
/NOWRAP (default)

Use with the **/PAGE=SAVE** qualifier to limit the number of columns to the width of the screen and to 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 and can be seen when you use the scrolling (left and right) features provided by the **/PAGE=SAVE** qualifier.

Example

```
$  SHOW AUDIT/ALL
List of audit journals:
 Journal name:           SECURITY
 Journal owner:          (system audit journal)
 Destination:            SYS$COMMON:[SYSMGR]SECURITY.AUDIT$JOURNAL
 Monitoring:             enabled
   Warning thresholds,   Block count:    100   Duration:    2 00:00:00.0
   Action thresholds,    Block count:     25   Duration:    0 00:30:00.0

Security auditing server characteristics:
 Database version:       4.4
 Backlog (total):        100, 200, 300
 Backlog (process):      5, 2
 Server processing intervals:
   Archive flush:        0 00:01:00:00
   Journal flush:        0 00:05:00:00
   Resource scan:        0 00:05:00:00
 Final resource action:  purge oldest audit events

Security archiving information:
 Archiving events:       none
 Archive destination:

System security alarms currently enabled for:
 ACL
 Authorization
 INSTALL
 Time
 Audit:      illformed
 Breakin:    dialup,local,remote,network,detached
```

```
 Login:      batch,dialup,local,remote,network,subprocess,detached
 Logfailure: batch,dialup,local,remote,network,subprocess,detached,server

System security audits currently enabled for:
 ACL
 Mount
 Authorization
 INSTALL
 Time
 Audit:      illformed
 Breakin:    dialup,local,remote,network,detached
 Login:      batch,dialup,local,remote,network,subprocess,detached,server
 Logfailure: batch,dialup,local,remote,network,subprocess,detached,server
 Logout:     batch,dialup,local,remote,network,subprocess,detached,server
 FILE access:
   Failure:  read,write,execute,delete,control
```

The **SHOW AUDIT** command in this example displays the auditing settings after a system installation. See the **SET AUDIT/ENABLE** command for descriptions of the individual audit items.

SHOW BROADCAST

SHOW BROADCAST — Displays the message classes that are currently affected by the **SET BROADCAST** command.

Format

SHOW BROADCAST

Parameters

None.

Description

The **SHOW BROADCAST** command tells which classes of messages are being screened from your terminal by the **SET BROADCAST** command. If some messages are enabled by the **SET BROADCAST** command, but these messages will not be displayed because your terminal is set NO BROADCAST, an informational message is displayed.

Qualifier

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter the **/OUTPUT** qualifier without a file specification, the output is sent to the current process default output stream or device, identified by the logical name SYS$OUTPUT.

If you enter the **/OUTPUT** qualifier with a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS is the default file type. The asterisk (*) and the percent sign (%) wildcard characters are not allowed.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

Examples

1.
```
$ SET BROADCAST=ALL
$ SET TERMINAL/NOBROADCAST
$ SHOW BROADCAST
Broadcasts are enabled for all classes
Terminal is set "No Broadcast", no broadcast messages
will be displayed.
```

   The display in this example shows that all message classes are enabled for broadcast, but that these messages will not be received because the terminal is set `NO BROADCAST`.

2.
```
$ SHOW BROADCAST
Broadcasts are currently disabled for:
    MAIL
```

   The **SHOW BROADCAST** display in this example indicates that **SET BROADCAST=NOMAIL** is in effect.

3.
```
$ SET BROADCAST=(NOALL,USER1)
$ SET TERMINAL/NOBROADCAST
$ SHOW BROADCAST
Broadcasts are currently disabled for:
    GENERAL
    PHONE
    MAIL
    QUEUE
    SHUTDOWN
    URGENT
    DCL
    OPCOM
    USER2
    USER3
    USER4
    USER5
    USER6
    USER7
    USER8
    USER9
    USER10
    USER11
    USER12
    USER13
    USER14
    USER15
    USER16
Terminal is set "No Broadcast", no broadcast messages
will be displayed.
```

   This example shows the display you see when messages from `USER1` are enabled, but will not be received, because the terminal is set `NO BROADCAST`.

4.
```
$ SET BROADCAST=NONE
$ SHOW BROADCAST
Broadcasts are disabled for all classes
```

This example shows the display you see when **SET BROADCAST=NONE** is in effect.

SHOW CLUSTER

SHOW CLUSTER — Invokes the Show Cluster utility to monitor and display cluster activity and performance.

Format

SHOW CLUSTER

Description

For more information about the Show Cluster utility, see the relevant section in the *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/#SHCL_PART] or online help.

SHOW CPU

SHOW CPU — Displays the current state of the processors in an OpenVMS system.

Format

SHOW CPU [*cpu-id*[,...]]

Parameter

***cpu-id*[,...]**

Specifies a decimal value representing the identity of a processor in a OpenVMS multiprocessing system. On an Alpha 7000 system, the CPU ID is the backplane slot number of the processor.

Description

The **SHOW CPU** command displays information about the status, characteristics, and capabilities of the processors active in and available to an OpenVMS multiprocessing or single-CPU system.

You identify the processors to be displayed by using either the **/ACTIVE** qualifier, the **/ALL** qualifier, a CPU ID, or list of CPU IDs. If you specify none of these, the **SHOW CPU** command uses the **/ALL** qualifier by default.

You identify the type of information to be displayed by using the **/BRIEF**, **/FULL**, and **/SUMMARY** qualifiers. If you specify none of these qualifiers, the **SHOW CPU** command assumes the **/BRIEF** qualifier by default; however, if you likewise do not identify a processor or processors as the object of a command, the **SHOW CPU** command assumes a default of **SHOW/ALL/SUMMARY**.

The **SHOW CPU/FULL** command lists the current process on each configured processor without stopping other activity on the system. The current process may change while the data is displayed. As a result, there may be apparent inconsistencies in the display. For example, a process may be listed as the current process on more than one CPU.

Qualifies

/ACTIVE_SET

Selects as the subject of the display only those processors that are members of the system's active set.

/ALL

Selects all configured processors, active and inactive, as the subject of the display.

/BRIEF

Produces information from the summary display and also lists the current CPU state and current process (if any) for each processor in the configuration.

/CONFIGURE_SET (Alpha/IA-64 only)

Selects as the subject of the display only those processors that are members of the system's configure set- those that are actively owned and controlled by the current instance.

/EXACT

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify a search string that must match the search string exactly and must be enclosed with quotation marks (" ").

If you specify the **/EXACT** qualifier without the **/SEARCH** qualifier, exact search mode is enabled when you set the search string with the Find (**E1**) key.

/FULL

Produces information from the summary display. The **/FULL** qualifier also lists the current CPU state, the current process (if any), the revision levels, and the capabilities for each configured processor. It indicates which processes can execute only on certain processors in the configuration. In addition, if one or more uniprocessing drivers are present in the system, the **/FULL** qualifier lists them by name.

The **SHOW CPU/FULL** command lists the current process on each configured processor without stopping other activity on the system. The current process may change while the data is displayed. As a result, there may be apparent inconsistencies in the display. For example, a process may be listed as the current process on more than one CPU.

This qualifier also displays whether the CPU is enabled or disabled for use as a Fast Path preferred CPU, then lists the Fast Path ports assigned to it, and shows whether or not the ports are user-assigned.

For more information about using Fast Path features to improve I/O performance, see the relevant section in the *VSI OpenVMS I/O User's Reference Manual* [https://docs.vmssoftware.com/vsi-openvms-io-user-s-reference-manual/#d0e29396].

/HIGHLIGHT[=*keyword*]

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify the type of highlighting you want when a search string is found. When a string is found, the entire line is highlighted. You can use the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting.

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter the **/OUTPUT** qualifier without a file specification, the output is sent to the current process default output stream or device, identified by the logical name SYS$OUTPUT.

If you enter the **/OUTPUT** qualifier with a partial file specification(for example, specifying only a directory), SHOW is the default file name and .LIS the default file type. If you enter a file specification, it may not include the asterisk (*) and the percent sign (%) wildcard characters.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

/PAGE[=*keyword*]
/NOPAGE (default)

Controls the display of device information on the screen.

You can use the following keywords with the **/PAGE** qualifier:

Keyword	**Description**
CLEAR_SCREEN	Clears the screen before each page is displayed.
SCROLL	Displays information one line at a time.
SAVE[= *n*]	Enables screen navigation of information, where *n* is the number of pages to store.

The **/PAGE=SAVE** qualifier allows you to navigate through screens of information. The **/PAGE=SAVE** qualifier stores up to 5 screens of up to 255 columns of information. When you use the **/PAGE=SAVE** qualifier, you can use the following keys to navigate through the information:

Key Sequence	**Description**
Up arrow key, **Ctrl/B**	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.
Right arrow key	Scroll right one column.
Find (**E1**)	Specify a string to find when the information is displayed.
Insert Here (**E2**)	Scroll right one half screen.
Remove (**E3**)	Scroll left one half screen.
Select (**E4**)	Toggle 80/132 column mode.
Prev Screen (**E5**)	Get the previous page of information.
Next Screen (**E6**), **Return**, **Enter**, **Space**	Get the next page of information.
F10, **Ctrl/Z**	Exit. (Some utilities define these differently.)
Help (**F15**)	Display utility help text.
Do (**F16**)	Toggle the display to oldest/newest page.

Key Sequence	Description
Ctrl/W	Refresh the display.

The **/PAGE** qualifier is not compatible with the **/OUTPUT** qualifier.

/POTENTIAL_SET (Alpha/IA-64 only)

Selects as the subject of the display only those processors that are members of the system's potential set - those CPUs in the hard partition that meet the current instance's requirements to join its active set.

Inclusion in this set does not imply that the CPU is (or ever will be)owned by the current instance. The potential set only describes those physically existing CPUs that currently meet the instance-specific hardware and software compatibility constraints, should they ever become available.

/SEARCH="*string*"

Use with the **/PAGE=SAVE** qualifier to specify a string that you want to find in the information being displayed. Quotation marks are required for the **/SEARCH** qualifier, if you include spaces in the text string.

You can also dynamically change the search string by pressing the Find key (**E1**) while the information is being displayed. Quotation marks are not required for a dynamic search.

/STANDBY_SET (Alpha/IA-64 only)

Selects as the subject of the display only those processors that are members of the system's standby set – those CPUs in the hard partition that are not currently owned by soft partitions.

Supported only on AlphaServer systems that support partitioning.

/SUMMARY

Produces a display listing the processors in the OpenVMS multiprocessing or single-CPU system, indicating which is the primary processor, which processors are configured, and which processors are active.

The **/SUMMARY** qualifier also indicates the minimum revision levels required for processors in the system, which OpenVMS synchronization image has been loaded into the operating system, and whether multiprocessing is enabled. If the presence of one or more uniprocessing drivers in the system prohibits the enabling of multiprocessing, the **SHOW CPU** command displays a warning message.

/SYSTEM (Alpha/IA-64 only)

Displays platform-specific hardware information relating to the current instance.

/WRAP
/NOWRAP (default)

Use with the **/PAGE=SAVE** qualifier to limit the number of columns to the width of the screen and to 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 and can be seen when you use the scrolling (left and right) features provided by the **/PAGE=SAVE** qualifier.

Examples

1.
```
$ SHOW CPU

Alpha:
System: LOWFAT, COMPAQ AlphaServer DS20E 666 MHz
Integrity servers:
System: EUROS, HPE rx2600  (900MHz/1.5MB)

CPU ownership sets:
   Active               0,1
   Configure            0,1

CPU state sets:
   Potential            0,1
   Autostart            0,1
   Powered Down         None
   Not Present          None
   Hard Excluded        None
   Failover             None
```

   The **SHOW CPU** command in this example produces a configuration summary of all configured processors in the Alpha (LOWFAT) and IA-64 server (EUROS) systems. The primary processor is CPU 01, and all configured processors are active.

2.
```
$ SHOW CPU/BRIEF
Alpha:
System: LOWFAT, COMPAQ AlphaServer DS20E 666 MHz
CPU 0    State: RUN               CPUDB: 81C1A000     Handle: * None *
       Process: DAVIS                  PID: 20400EB2

CPU 1    State: RUN               CPUDB: 81D03680     Handle: * None *

Integrity servers:
System: EUROS, HPE rx2600  (900MHz/1.5MB)
CPU 0    State: RUN               CPUDB: 8901C000     Handle: 00005D70
         Owner: 000004C8         Current: 000004C8     Partition 0 (EUROS)

CPU 1    State: RUN               CPUDB: 89163480     Handle: 00005E80
         Owner: 000004C8         Current: 000004C8     Partition 0 (EUROS)
       Process: DAVIS            PID: 20201192
```

   The **SHOW CPU/BRIEF** command in this example produces a configuration summary of the Alpha (LOWFAT) and IA-64 server (EUROS) systems. It also indicates that its two processors are in the RUN state. Only CPU 00 has a current process.

3.
```
$ SHOW CPU/FULL

System: WFGLXD, AlphaServer GS320 6/731

  SMP execlet   = 2 : Enabled : Full checking.
  Config tree   = Version 6
  Primary CPU   = 0
  HWRPB CPUs    = 32
  Page Size     = 8192
  Revision Code =
  Serial Number = BUDATEST
  Default CPU Capabilities:
        System: QUORUM RUN
  Default Process Capabilities:
        System: QUORUM RUN

CPU 0    State: RUN               CPUDB: 81416000     Handle: 00003200
```

```
          Owner: 000006C0         Current: 000006C0     Partition 0 (WFGLXA)
         ChgCnt:        1           State: Present, Primary, Hotswap
        Process: SHADOW_SERVER        PID: 00000434
    Capabilities:
         System: PRIMARY QUORUM RUN RAD0
    Slot Context: 8F5BA180
       CPU     -  State..........: RC, PA, PP, CV, PV, PMV, PL
                  Type...........: EV67, Pass 2.3
                  Speed..........: 730 Mhz
                  Variation......: VAX FP, IEEE FP, Primary Eligible
                  Serial Number..: SM00700054
                  Revision.......:
                  Halt Request...: 0
                  Software Comp..: 8.4
       PALCODE -  Revision Code..: 1.96-01
                  Compatibility..: 6
                  Max Shared CPUs: 32
                  Memory  Space..: Physical = 00000000.00000000  Length = 0
                  Scratch Space..: Physical = 00000000.00000000  Length = 0
    Bindings:
       NETACP           PID = 0000041A  Reason: PRIMARY Capability
    Fastpath:      * None *
    Features:
       Autostart - Enabled.
       Fastpath  - Selection enabled as Preferred CPU.

  CPU 1    State: RUN               CPUDB: 81456880     Handle: 00003500
          Owner: 000006C0         Current: 000006C0     Partition 0 (WFGLXA)
         ChgCnt:        1           State: Present, Hotswap, Reassignable
        Process: SYSTEM               PID: 00000428
    Capabilities:
         System: QUORUM RUN RAD0
    Slot Context: 8F5BA400
       CPU     -  State..........: RC, PA, PP, CV, PV, PMV, PL
                  Type...........: EV67, Pass 2.3
                  Speed..........: 730 Mhz
                  Variation......: VAX FP, IEEE FP, Primary Eligible
                  Serial Number..: NI94370773
                  Revision.......:
                  Halt Request...: 0
                  Software Comp..: 8.4
       PALCODE -  Revision Code..: 1.96-01
                  Compatibility..: 6
                  Max Shared CPUs: 32
                  Memory  Space..: Physical = 00000000.00000000  Length = 0
                  Scratch Space..: Physical = 00000000.00000000  Length = 0
    Bindings:      * None *
    Fastpath:      * None *
    Features:
       Autostart - Enabled.
       Fastpath  - Selection enabled as Preferred CPU.
```

The **SHOW CPU/FULL** command in this example produces a configuration summary of the AlphaServer GS320 system, WFGLXD. All are in the RUN state and all have a current process.

Note that NETACP can only run on CPU 0, because only CPU 0 has primary capability. Note also that CPUs 0 and 1 have Fast path ports.

4. `$ SHOW CPU 2/BRIEF`

```
System: WFGLXA, AlphaServer GS320 6/731

CPU 2    State: RUN               CPUDB: 81457680     Handle: 00003800
         Owner: 000006C0         Current: 000006C0     Partition 0 (WFGLXA)
       Process: SHADOW_SERVER        PID: 00000429
```

The **SHOW CPU/BRIEF** command in this example produces a brief configuration summary of CPU 2. CPU 2 is currently running under instance WFGLXA in partition 0.

5.
```
$ SHOW CPU 11

System: WFGLXA, AlphaServer GS320 6/731

CPU 11   State: NOT_CONFIGURED     CPUDB: 81470E00     Handle: 000125C0
         Owner: 00000440         Current: 00000440     Hard Partition 0
```

This **SHOW CPU** command produces a configuration summary of CPU 11 on instance WFGLXA. CPU 11 is not configured in this instance and is currently assigned to hard partition 0.

6.
```
$ SHOW CPU 8/FULL

System: WFGLXA, AlphaServer GS320 6/731

CPU 8    State: POWERED_DOWN     CPUDB: 81459280    Handle: 00011CC0
         Owner: 000006C0       Current: 000006C0    Partition 0 (WFGLXA)
        ChgCnt:        3         State: Present, In-Console, Powered Down,
                                        Hotswap, Reassignable
      Process: * None *
  Capabilities:
        System: QUORUM RAD0
  Slot Context: 8F5BB580
     CPU     -  State..........: BIP, CV, PV, PMV, PL
  Bindings:     * None *
  Fastpath:     * None *
  Features:
     Autostart - Enabled.
     Fastpath  - Selection enabled as Preferred CPU.
```

This **SHOW CPU** command produces a summary of CPU 8. CPU 8 is currently powered down, but a CPU module is present in the backplane slot.

7.
```
$ SHOW CPU

System: WFGLXA, AlphaServer GS320 6/731

CPU ownership sets:
   Active               0-3,9,10
   Configure            0-3,8-10
   Standby              11
     Hard Partition     11

CPU state sets:
   Potential            0-7,9-15
  Autostart            0-31
   Powered Down         8
   Not Present          16-31
   Failover             None
```

The **SHOW CPU** command shows the configuration characteristics of the instance.

8.
```
$ SHOW CPU/SYSTEM

System: WFGLXA, AlphaServer GS320 6/731

  SMP execlet   = 2 : Enabled : Full checking.
  Config tree   = Version 6
  Primary CPU   = 0
  HWRPB CPUs    = 32
  Page Size     = 8192
```

```
  Revision Code =
  Serial Number = BUDATEST
  Default CPU Capabilities:
        System: QUORUM RUN
  Default Process Capabilities:
        System: QUORUM RUN

CPU ownership sets:
   Active               0-3,9,10
   Configure            0-3,8-10
   Standby              11
     Hard Partition     11

CPU state sets:
   Potential            0-7,9-15
   Autostart            0-31
   Powered Down         8
   Not Present          16-31
   Failover             None
```

This **SHOW CPU/SYSTEM** command shows the platform-specific hardware information relating to instance WFGXLA.

9.
```
$ SET PROCESS /AFFINITY /SET=3 SYSTEM/PERMANENT
$ SHOW CPU 3/FULL

System: WFGLXD, AlphaServer GS320 6/731

CPU 3    State: RUN                CPUDB: 81416000     Handle: 00003200
         Owner: 000006C0         Current: 000006C0     Partition 0 (WFGLXA)
        ChgCnt:        1           State: Present, Primary, Hotswap
       Process: SHADOW_SERVER        PID: 00000434
  Capabilities:
        System: PRIMARY QUORUM RUN RAD0
  Slot Context: 8F5BA180
     CPU     -  State..........: RC, PA, PP, CV, PV, PMV, PL
                Type..........: EV67, Pass 2.3
                Speed.........: 730 Mhz
                Variation.....: VAX FP, IEEE FP, Primary Eligible
                Serial Number..: SM00700054
                Revision.......:
                Halt Request...: 0
                Software Comp..: 8.4
     PALCODE -  Revision Code..: 1.96-01
                Compatibility..: 6
                Max Shared CPUs: 32
                Memory  Space..: Physical = 00000000.00000000  Length = 0
                Scratch Space..: Physical = 00000000.00000000  Length = 0
  Bindings:
     SYSTEM           PID = 00000434  Reason: Affinitized to this CPU
  Fastpath:
        PNA0
  Features:
     Autostart - Enabled.
     Fastpath  - Selection enabled as Preferred CPU.
```

In this example, the SYSTEM process has its affinity set permanently to CPU 3. The **SHOW CPU** command indicates this is the BINDINGS list.

The port PNA0 was assigned to CPU 3 by a user when CPU 3 was enabled as a Fast Path preferred CPU. Subsequent to that operation, CPU 3's bit was cleared in the IO_PREFER_CPUS system parameter. This disabled CPU 3 for use as a Fast Path preferred CPU. Normally this is a

transient condition, as changing IO_PREFER_CPUS results in the FASTPATH_SERVER process redistributing the Fast Path ports across the set of usable CPUs.

However, if there is no FASTPATH_SERVER process, `PNA0` remains assigned to this CPU until any of the following actions occur:

- The user tries to move the port to another CPU by using the **`SET DEVICE/PREFERRED_CPU`** command, $QIO, or the $IO_FASTPATH system service.
- The CPU is stopped.

10.
```
$ SHOW CPU/FULL
System: LOWFAT, COMPAQ AlphaServer DS20E 666 MHz
SMP execlet   = 3 : Enabled : Streamlined.
Config tree   = None
  Primary CPU   = 0
  HWRPB CPUs    = 2
  Page Size     = 8192
  Revision Code =
  Serial Number = 4033DZ2Z1023
  Default CPU Capabilities:
        System: QUORUM RUN
  Default Process Capabilities:
        System: QUORUM RUN

CPU 0    State: RUN               CPUDB: 81C1A000     Handle: * None *
       Process: DAVIS             PID: 20400EB2
  Capabilities:
        System: PRIMARY QUORUM RUN RAD0
          User: Usr1 Usr2
  Slot Context: 857F6180
     CPU     -  State..........: RC, PA, PP, CV, PV, PMV, PL
                Type...........: EV67, Pass 2.2.3 & Pass 2.2.5
                Speed..........: 667 Mhz
                Variation......: VAX FP, IEEE FP, Primary Eligible
                Serial Number..:
                Revision.......:
                Halt Request...: 0
                Software Comp..: 8.4
     PALCODE -  Revision Code..: 1.98-01
                Compatibility..: 79
                Max Shared CPUs: 2
                Memory  Space..: Physical = 00000000.00000000  Length = 0
                Scratch Space..: Physical = 00000000.00000000  Length = 0
  Bindings:     * None *
  Fastpath:
        PKC0
        FGA0
        PEA0
        EWA0
  Features:
     Autostart - Enabled.
     Fastpath  - Selection enabled as Preferred CPU.

CPU 1    State: RUN               CPUDB: 81D03680     Handle: * None *
       Process: * None *
  Capabilities:
        System: QUORUM RUN RAD0
  Slot Context: 857F6400
     CPU     -  State..........: RC, PA, PP, CV, PV, PMV, PL
                Type...........: EV67, Pass 2.2.3 & Pass 2.2.5
                Speed..........: 667 Mhz
                Variation......: VAX FP, IEEE FP, Primary Eligible
                Serial Number..:
```

```
                Revision.......:
                Halt Request...: 0
                Software Comp..: 8.4
     PALCODE -  Revision Code..: 1.98-01
                Compatibility..: 79
                Max Shared CPUs: 2
                Memory  Space..: Physical = 00000000.00000000  Length = 0
                Scratch Space..: Physical = 00000000.00000000  Length = 0
  Bindings:     * None *
  Fastpath:
        PKB0
        PKA0
        FWA0
        BG0
  Features:
     Autostart - Enabled.
     Fastpath  - Selection enabled as Preferred CPU.
 $
```

The **SHOW CPU/FULL** command in this example produces a configuration summary of the Alpha DS20E 666 MHz system LOWFAT, indicating that only CPU 0, the primary CPU, is active and in the RUN state. It also shows that there is a uniprocessing driver loaded in the system, thus preventing the system from being enabled as a multiprocessor.

SHOW DEFAULT

SHOW DEFAULT — Displays the current default device and directory.

Format

SHOW DEFAULT

Description

The **SHOW DEFAULT** command displays the current device and directory names, along with any equivalence strings.

The default disk and directory are established in the user authorization file (UAF). You can change these defaults during a terminal session or in a batch job by using the **SET DEFAULT** command, or by reassigning the logical name SYS$DISK.

Examples

1.
```
$ SHOW DEFAULT
DISK1:[ALAMO]
$ SET DEFAULT DISK5:[SKONETSKI.SOURCES]
$ SHOW DEFAULT
DISK5:[SKONETSKI.SOURCES]
```

   The **SHOW DEFAULT** command in this example displays the current default device and directory names. The **SET DEFAULT** command changes these defaults, and the next **SHOW DEFAULT** command displays the new default device and directory.

2.
```
$  SET DEFAULT NOSUCH:[NOWAY]
$  SHOW DEFAULT
 NOSUCH:[NOWAY]
 %DCL-I-INVDEF, NOSUCH:[NOWAY] does not exist
```

In this example, the default has been set to a nonexistent device and directory. An error message is displayed.

3.
```
$  DEFINE/TRANSLATION_ATTRIBUTES=CONCEALED XYZ WORK:[INVOICES.]
$  SET DEFAULT XYZ:[HESHELMAN]
$  SHOW DEFAULT
XYZ:[HESHELMAN]
```

In this example, a logical name, XYZ, is defined to represent WORK:[INVOICES]. The **/TRANSLATION_ATTRIBUTES=CONCEALED** qualifier tells the system not to display the translation of XYZ in file specifications. Thus, the **SHOW DEFAULT** command displays the logical name XYZ and not its translation.

4.
```
$  SET DEFAULT WRK:[SCHENKENBERGER]
$  SHOW DEFAULT
 WRK:[SCHENKENBERGER]
$  DEFINE KUDOS WRK:[SCHENKENBERGER.TEMP1],WRK:[SCHENKENBERGER.TEMP2]
$  SET DEFAULT KUDOS
$  SHOW DEFAULT
 KUDOS:[SCHENKENBERGER]
 =   WRK:[SCHENKENBERGER.TEMP1]
 =   WRK:[SCHENKENBERGER.TEMP2]
```

In this example, the logical name KUDOS is defined as a search list containing the directories [SCHENKENBERGER.TEMP1] and [SCHENKENBERGER.TEMP2] on device WRK. The **SET DEFAULT** command equates this search list logical name with the logical name SYS$DISK. The subsequent **SHOW DEFAULT** command displays the search list logical name along with its equivalence strings.

Because the directory field has not been explicitly specified, the original [SCHENKENBERGER] directory remains in effect as the current default directory; however, unless the current default directory syntax ([]) is explicitly used, all file references are to those directories contained in the search list.

SHOW DEVICES

SHOW DEVICES — Displays the status of a device on the system. **See the qualifier descriptions for restrictions.**

Format

SHOW DEVICES [*device-name*[:]]

Parameter

***device-name*[:]**

Specifies the name of a device for which information is to be displayed. The device name displayed by the system uses the format *ddcu* where *dd* is the device code, *c* is the controller designation, and *u* is the unit number. If the system is part of an OpenVMS Cluster that is running with HSC controllers, the device name must include the node name using the format *node*$*ddcu* (where *node* is the node name).

You can specify a complete device name or only a portion of a device name. The **SHOW DEVICES** command provides defaults for nonspecified portions of device names, as follows:

- If you truncate a device name (for example, if you specify D), the command lists information about all devices whose device names begin with what you entered (in this case, D). However, if you specify a device in an OpenVMS Cluster that is running with HSC controllers, you must include the entire *node* portion of the device name.
- If you omit a controller designation, the **SHOW DEVICES** command lists all devices on all controllers with the specified unit number.
- If you omit a unit number, the **SHOW DEVICES** command lists all devices on the specified controller.

Note that the **/FILES** qualifier does not support defaults for nonspecified portions of device names; you must supply a complete device specification.

Description

When you enter the **SHOW DEVICES** command without specifying a device or using any qualifier, information about all devices on the system is displayed. If you specify a device name, the **SHOW DEVICES** command displays information about that device. If you use certain qualifiers with the **SHOW DEVICES** command, information is displayed about those devices that currently have volumes mounted or that have been allocated to processes, or both.

The device name displayed by the system uses the format *ddcu*, where *dd* is the device code, *c* is the controller designation, and *u* is the unit number. If the system is part of an OpenVMS Cluster that is running with HSC controllers, the device name must include the node name using the format *node*$*ddcu* (where *node* is the node name).

If you enter the **SHOW DEVICES** command and do not specify a device name parameter or a qualifier, the command provides a brief listing of characteristics of all devices on the system, with the exception of pseudodevices with the mailbox characteristic, such as MBA3:, NLA0:, NET9:, and so on. To display such a device, you must specify all or part of the device name.

Note

If a field width is too narrow to display a particular numeric field, asterisks are displayed in place of the data. Therefore, if asterisks appear in the Free Space field while displaying a particular device, use the **/UNITS** or **/SIZE** qualifier to display the free space in bytes.

Qualifiers

/ALLOCATED

Displays all devices currently allocated to processes.

If you specify a device name, the characteristics of only that device are displayed. If the device is not currently allocated, the command displays a message indicating that there is no such device. If you specify a generic device name, the characteristics of all allocated devices of that type are displayed.

/BRIEF (default)

Displays brief information about the specified devices.

/BITMAP *ddcu*:

Displays information regarding minicopy or HBMM (Host-Based Minimerge) bitmaps for the given shadow set.

Note that the **SHOW DEVICES/FULL *ddcu*:** display also displays status to indicate that bitmaps are active on the device.

/EXACT

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify a search string that must match the search string exactly and must be enclosed with quotation marks (" ").

If you specify the **/EXACT** qualifier without the **/SEARCH** qualifier, exact search mode is enabled when you set the search string with the Find (**E1**) key.

/FILES

Requires SYSPRV (system privilege) or BYPASS privileges to list read-protected files.

Displays a list of the names of all files open on a volume and their associated process name and process identification (PID). The specified device must be a mounted Files-11 volume. If the specified volume is a multivolume set, the files on each volume in the set are listed.

Note

The **SHOW DEVICES/FILES** command does not support defaults for nonspecified portions of device names. You must supply a complete device specification when using the **/FILES** qualifier.

If the **/SYSTEM** qualifier is also specified, only the names of installed files and files opened by the system are displayed. Files opened by the system are those that have been opened without the use of an ancillary control process (ACP), such as INDEXF.SYS and QUOTA.SYS, as well as system files such as JBCSYSQUE.EXE and SYSMSG.EXE.

If the **/NOSYSTEM** qualifier is specified, only those files opened by processes are displayed. To list files opened by a process in your group, your process must have at least GROUP privilege. If the process is not in your group, you need WORLD privilege.

If neither the **/SYSTEM** nor the **/NOSYSTEM** qualifier is specified, the names of all files currently opened on the system are displayed.

If a file is read-protected from your user identification code (UIC), the "No privilege" message is displayed instead of the file name. You must have SYSPRV (system privilege) privilege or BYPASS privilege to display the file name.

A space in place of a file name represents a work file (such as a temporary edit file) not entered in any directory. To display temporary file names, you must have BYPASS privilege in addition to GROUP or WORLD privilege.

Do not use the **/FILES** qualifier with the **/ALLOCATED**, **/BRIEF**, **/FULL**, or **/MOUNTED** qualifier. The functions of the **/FILES** qualifier and these qualifiers are mutually exclusive.

/FULL

Displays a complete list of information about the devices, as appropriate, except for rebuild status. You must use the **/REBUILD_STATUS** qualifier to get that information.

The display will be in blocks or bytes, depending on the current system default setting. You can use **SHOW PROCESS/UNITS** to display the current default. To change the default, execute the DCL command **SET PROCESS/UNITS=BYTES** or **SET PROCESS/UNITS=BLOCKS**.

For more information about fields related to Dissimilar Device Shadowing (DDS) or Dynamic Volume Expansion (DVE) in the display, see the discussion of these topics in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#d0e28741].

/HIGHLIGHT[=*keyword*]

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify the type of highlighting you want when a search string is found. When a string is found, the entire line is highlighted. You can use the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting.

/MOUNTED

Displays all devices that currently have volumes mounted on them.

If you specify a device name, only the characteristics of that device are displayed; however, if the device is not currently mounted, the command issues a message indicating there is no such device. If you specify a generic device name, the characteristics of all such devices that currently have volumes mounted are displayed.

/MULTIPATH

Displays all devices that are currently multipath sets.

If you specify a device name, only the characteristics of that device are displayed; however, if the device is not a multipath set, the command issues a message indicating there is no such device. If you specify a generic device name, the characteristics of all such devices that are currently multipath sets are displayed.

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter the **/OUTPUT** qualifier without a file specification, the output is sent to the current process default output stream or device, identified by the logical name SYS$OUTPUT.

If you enter the **/OUTPUT** qualifier with a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS the default file type. If you enter a file specification, it may not include the asterisk (*) and the percent sign (%) wildcard characters.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

/PAGE[=*keyword*]
/NOPAGE (default)

Controls the display of device information on the screen.

You can use the following keywords with the **/PAGE** qualifier:

Keyword	Description
CLEAR_SCREEN	Clears the screen before each page is displayed.
SCROLL	Displays information one line at a time.
SAVE[=*n*]	Enables screen navigation of information, where *n* is the number of pages to store.

The **/PAGE=SAVE** qualifier allows you to navigate through screens of information. The **/PAGE=SAVE** qualifier stores up to 5 screens of up to 255 columns of information. When you use the **/PAGE=SAVE** qualifier, you can use the following keys to navigate through the information:

Key Sequence	Description
Up arrow key, **Ctrl/B**	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.
Right arrow key	Scroll right one column.
Find (**E1**)	Specify a string to find when the information is displayed.
Insert Here (**E2**)	Scroll right one half screen.
Remove (**E3**)	Scroll left one half screen.
Select (**E4**)	Toggle 80/132 column mode.
Prev Screen (**E5**)	Get the previous page of information.
Next Screen (**E6**), **Return**, **Enter**, **Space**	Get the next page of information.
F10, **Ctrl/Z**	Exit. (Some utilities define these differently.)
Help (**F15**)	Display utility help text.
Do (**F16**)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.

The **/PAGE** qualifier is not compatible with the **/OUTPUT** qualifier.

/REBUILD_STATUS

Tells you whether volumes need rebuilding.

You may need to rebuild a volume if the volume was improperly dismounted. Volumes are improperly dismounted when, for example, the system crashes.

For each volume, **SHOW DEVICES/REBUILD_STATUS** returns one of these values:

Value	Meaning
Yes	Rebuild needed.
No	Rebuild not needed.
Not applicable	You cannot rebuild this volume. This value is returned if, for example, the volume is not a disk, or the volume is write-locked.
Information unavailable	Rebuild information is unavailable. This value is returned when, for example, the volume is not mounted, or mount verification is taking place.

To rebuild a volume, either:

- Use **SET VOLUME/REBUILD**
- Dismount the volume, then mount the volume again using **MOUNT/REBUILD**

You cannot use the **/REBUILD_STATUS** qualifier with any other **SHOW DEVICES** qualifiers, except **/OUTPUT**.

/SEARCH="*string*"

Use with the **/PAGE=SAVE** qualifier to specify a string that you want to find in the information being displayed. Quotation marks are required for the **/SEARCH** qualifier, if you include spaces in the text string.

You can also dynamically change the search string by pressing the Find key (**E1**) while the information is being displayed. Quotation marks are not required for a dynamic search.

/SIZE=*keyword*

Allows you to override the current default specified by **SET PROCESS/UNITS** so that you can display disk free space in your choice of blocks or bytes. (**/SIZE** is functionally equivalent to **/UNITS**.)

The following keywords are valid with the **/SIZE** qualifier: BLOCKS, BYTES.

If you specify **/SIZE** with no keyword, the default value is not changed.

/SYSTEM
/NOSYSTEM

Controls whether the names of installed files and files opened by the system are displayed. Files opened by the system are those that have been opened without the use of an ancillary control process (ACP), such as INDEXF.SYS and QUOTA.SYS.

If you specify the **/NOSYSTEM** qualifier with the **/FILES** qualifier, only files opened by processes are displayed. If you omit both the **/SYSTEM** and **/NOSYSTEM** qualifiers and specify the **/FILES** qualifier, the names of all files currently open on the system are displayed.

You can use this qualifier only with the **/FILES** qualifier. See the description of the **/FILES** qualifier for more details.

/UNITS=*keyword*

Allows you to override the current default specified by **SET PROCESS/UNITS** so that you can display disk free space in your choice of blocks or bytes. (**/UNITS** is functionally equivalent to **/SIZE**.)

The following keywords are valid with the **/UNITS** qualifier: BLOCKS, BYTES.

If you specify **/UNITS** with no keyword, the default value is not changed.

/WINDOWS

Displays the window count and total size of all windows for files open on a volume. The file name and related process name and process identification (PID) are also displayed. The letter C in a display indicates that the file is open with "cathedral windows" (segmented windows).

/WRAP
/NOWRAP (default)

Use with the **/PAGE=SAVE** qualifier to limit the number of columns to the width of the screen and to 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 and can be seen when you use the scrolling (left and right) features provided by the **/PAGE=SAVE** qualifier.

Examples

1.
```
$ SHOW DEVICES
Device          Device      Err.    Volume        Free  Trans Mount
 Name           Status     Count     Label       Blocks Count Count
DKA0:           Online mnt     0  VMS             47088   115     1
DKA1:           Online mnt     0  USERPACK1       45216     2     1
DKA2:           Online mnt     3  DOCUMENT         8068    20     1
DKA5:           Online mnt     0  MASTERP         28668     1     1
DKA6:           Online         0
DKA7:           Online mnt     0  PROJECT        110547     1     1
DMA0:           Online         0
DLA0:           Online         0
DYA0:           Online         0
DYA1:           Online         0
DRA3:           Online mnt     0  RES26APR        29317     1     1
```

In this example, the **SHOW DEVICES** command displays the following information for each device on the system:

- Device name
- Device status and characteristics (status indicates whether the device is on line; characteristics indicate whether the device is allocated, is spooled, has a volume mounted on it, or has a foreign volume mounted on it)
- Error count
- Volume label (for disk and tape volumes only)
- Number of free blocks on the volume
- Transaction count
- Number of mount requests issued for the volume (disk devices only)

2.
```
$ SHOW DEVICES /FULL $1$DGA30
Disk $1$DGA100: (EUROS), device type HPE MSA VOLUME, is online, mounted, file-
    oriented device, shareable, available to cluster, error logging is enabled.

    Error count                    0    Operations completed             17299
    Owner process                 ""    Owner UIC                      [SYSTEM]
    Owner process ID        00000000    Dev Prot            S:RWPL,O:RWPL,G:R,W
    Reference count                3    Default buffer size                512
    Current preferred CPU Id       0    Fastpath                             1
    Total blocks           488327805    Sectors per track                   96
    Total cylinders            52987    Tracks per cylinder                 96
    Logical Volume Size    488327805    Expansion Size Limit          941752320
    Host name                "EUROS"    Host type, avail HPE rx2600
 (900MHz/1.5MB), yes
    Allocation class               1

    Volume label       "DEBUG-DISK1"    Relative volume number              0
    Cluster size                 480    Transaction count                   2
    Free blocks             16558560    Maximum files allowed          507617
    Extend quantity                5    Mount count                         2
    Mount status              System    Cache name        "_$7$DKD100:XQPCACHE"
    Extent cache size             64    Maximum blocks in extent cache  1655856
```

```
   File ID cache size            64    Blocks in extent cache          581760
   Quota cache size               0    Maximum buffers in FCP cache      3514
   Volume owner UIC        [SYSTEM]    Vol Prot    S:RWCD,O:RWCD,G:RWCD,W:RWCD

  Volume Status:  ODS-2, subject to mount verification, file high-water marking,
      write-back caching enabled.
  Volume is also mounted on EUROS.
```

In this example, the **SHOW DEVICES** command requests a full listing of the status of the multipath device `$1$DGA30`. This device has four local Fibre Channel paths and one remote (MSCP-served) path.

3.
```
$ SHOW DEVICE/FULL DKB0:
Disk $DKB0:, device type RZ56, is online, mounted, file-oriented
    device, shareable, error logging is enabled.

Error count               0  Operations completed            392750
Owner process            ""  Owner UIC                        [1,4]
Owner process ID   00000000  Dev Prot           S:RWPL,O:RWPL,G:R,W
Reference count         317  Default buffer size                512
Total blocks        1299174  Sectors per track                   54
Total cylinders        1604  Tracks per cylinder                 15

Volume label  "SYSTEM_DISK"  Relative volume number               0
Cluster size              3  Transaction count                 278
Free blocks          367632  Maximum files allowed          162396
Extend quantity           5  Mount count                         1
Mount status         System  Cache name      "$DKB0:XQPCACHE"
Extent cache size        64  Maximum blocks in extent cache   36763
File ID cache size       64  Blocks currently in extent cache 36258
Quota cache size          0  Maximum buffers in FCP cache       210
Volume owner UIC      [1,1]  Vol Prot   S:RWCD,O:RWCD,G:RWCD,W:RWCD

Volume status:  ODS-2, subject to mount verification, protected
     subsystems enabled, file high-water marking, write-through
     XQP caching enabled, write-through XFC caching enabled.
```

In this example, the **SHOW DEVICES** command requests a full listing of the status of the `RZ56` device `DKB0`.

4.
```
$ SHOW DEVICES/REBUILD_STATUS

Device Name              Rebuild needed?

ADU15$DKA300:            Information unavailable
EDIV$DKA300:             Information unavailable
EMUL$DKB200:             No
EMUL$DKB300:             No
EMUL$DKB500:             Yes
FTA0:                    Not applicable
OPA0:                    Not applicable
```

In this example, the volume mounted on device `EMUL$DKB500` needs rebuilding.

5.
```
$ SHOW DEVICES/FULL NODE2$
Disk NODE2$DUA0:, device type RA81, is online, mounted,
error logging enabled
 Error count                0  Operations completed            24195
 Owner process name            Owner UIC                    [11,177]
 Owner process ID    20200000  Dev Prot  S:RWPL,O:RWPL,G:RWPL,W:RWPL
 Reference count           16  Default buffer size                512
 Volume label       VMSDOCLIB  Relative volume no.                  0
 Cluster size               3  Transaction count                   17
```

```
 Free blocks             525447  Maximum files allowed          111384
 Extend quantity              5  Mount count                         1
 Volume status           System  ACP process name
 Caching disabled
Volume is subject to mount verification, file high-water marking

Disk NODE2$DUA1:, device type RA81, is online, error logging enabled
 Error count                  0  Operations completed                0
 Owner UIC                [0,0]  Owner process name
 Owner process ID      20200000  Dev Prot  S:RWED,O:RWED,G:RWED,W:RWED
 Reference count              0  Default buffer size               512
.
.
.
```

In this example, the user requested a full display of information about each device on NODE2 in the OpenVMS Cluster. Information is shown here only for the first two devices: a mounted device and a device that is not mounted.

6.
```
$ SHOW DEVICES MUA1/FULL

Magtape $4$MUA1: (HSC70), device type TA79, is online, device is
 equipped with a stacker/loader, file-oriented device, available
 to cluster, error logging is enabled.

 Error count                  0  Operations completed                  0
 Owner process               ""  Owner UIC                      [SYSTEM]
 Owner process ID    00000000  Dev Prot            S:RWPL,O:RWPL,G:R,W
 Reference count              0  Default buffer size                2048
 Density                   1600  Format                         Normal-11
 Host name            "HSC70"  Host type, avail               HS70, yes
 Allocation class             4

Volume status:  no-unload on dismount, odd parity.
```

In this example, the user requested a full display of the tape device MUA1.

7.
```
$ SHOW DEVICE/FULL $1$DKA221

Disk $1$DKA221: (HSC952), device type RZ29, is online, mounted,
    file-oriented device, shareable, available to cluster, error
    logging is enabled.

 Error count                  0  Operations completed                6509
 Owner process               ""  Owner UIC                       [SYSTEM]
 Owner process ID    00000000  Dev Prot            S:RWPL,O:RWPL,G:R,W
 Reference count            108  Default buffer size                 512
 Current preferred CPU Id   1  FastPath                               1
 Total blocks          8380080  Sectors per track                     4
 Total cylinders         65470  Tracks per cylinder                  32
 Host name            "HSC952"  Host type, avail               HS95, yes
 Allocation class             1

 Volume label  "GALAXY_X6L5"  Relative volume number                 0
 Cluster size                 4  Transaction count                  302
 Free blocks            4691760  Maximum files allowed           838008
 Extend quantity              5  Mount count                         2
 Mount status            System  Cache name        "_$1$DKA221:XQPCACHE"
 Extent cache size           64  Maximum blocks in extent cache   469176
 File ID cache size          64  Blocks currently in extent cache 237504
 Quota cache size             0  Maximum buffers in FCP cache       1734
 Volume owner UIC        [1,1]  Vol Prot    S:RWCD,O:RWCD,G:RWCD,W:RWCD

Volume Status:  ODS-2, subject to mount verification, protected
```

```
    subsystems enabled, file high-water marking, write-through
    XQP caching enabled, write-through XFC caching enabled.
Volume is also mounted on MEAD.
```

This disk is a device that performs its I/O through the Fast Path port PNA0. Fast Path is enabled for this disk, and its I/O will be processed on the preferred CPU, CPU 1, which is associated with its Fast Path port PNA0.

The **/FULL** qualifier also displays additional information about Fast Path devices. A Fast Path device may be either a class device or a port device. An example of a class device is an HSC disk, and an example of a port device is the CI. For both Fast Path class devices and ports, the display indicates that Fast Path is enabled and lists the current CPU ID to which that device is assigned. Additionally, for a Fast Path port, the display indicates the port's user-established CPU preference (if any).

The current preferred CPU and the User Preferred CPU can be set by using the DCL command, **SET DEVICE/PREFERRED_CPU**, or by issuing a $QIO.

For more information about using Fast Path features to improve I/O performance, see the relevant section in the *VSI OpenVMS I/O User's Reference Manual* [https://docs.vmssoftware.com/vsi-openvms-io-user-s-reference-manual/#d0e29396].

8.
```
$ SHOW DEVICE/FULL PNA0

Device PNA0:, device type CIXCD, is online, shareable, error logging
    is enabled.

 Error count                  0  Operations completed              0
 Owner process               ""  Owner UIC                   [SYSTEM]
 Owner process ID     00000000  Dev Prot          S:RWPL,O:RWPL,G,W
 Reference count              0  Default buffer size               0
 Current preferred CPU Id     1  FastPath                          1
```

The above display shows that port PNA0 is assigned to CPU 1 and that Fast Path is enabled.

9.
```
$ SHOW DEVICE/FULL PNA0

Device PNA0:, device type CIXCD, is online, shareable, error logging
is enabled.

 Error count                    0  Operations completed              0
 Owner process                 ""  Owner UIC                   [SYSTEM]
 Owner process ID       00000000  Dev Prot          S:RWPL,O:RWPL,G,W
 Reference count                0  Default buffer size               0
 Current preferred CPU Id       1  FastPath                          1
 User preferred CPU Id          2
```

This example shows a Fast Path assignment made by a user.

Normally, the User Preferred CPU ID matches the current preferred CPU ID. The reason for the mismatch in the above display is because CPU 2 was stopped. The port was previously assigned to CPU 2. But when CPU 2 was stopped, the operating system moved the port assignment from CPU 2 to CPU 1.

If CPU 2 were started, the operating system would assign the device to the User Preferred CPU.

10.
```
$ SHOW DEVICE/FULL DSA0

Disk DSA0:, device type RAM Disk, is online, mounted, file-oriented
  device, shareable, available to cluster, error logging is enabled,
```

```
  device supports bitmaps (no bitmaps active).

 Error count                0  Operations completed              47
 Owner process             ""  Owner UIC                     [SYSTEM]
 Owner process ID     00000000  Dev Prot          S:RWPL,O:RWPL,G:R,W
 Reference count             2  Default buffer size              512
 Total blocks             1000  Sectors per track                 64
 Total cylinders             1  Tracks per cylinder               32
Volume label           "TST0"  Relative volume number             0
 Cluster size                1  Transaction count                 1
 Free blocks               969  Maximum files allowed           250
 Extend quantity             5  Mount count                       1
 Mount status           System  Cache name    "_$252$DUA721:XQPCACHE"
 Extent cache size          64  Maximum blocks in extent cache    96
 File ID cache size         64  Blocks currently in extent cache   0
 Quota cache size            0  Maximum buffers in FCP cache     404
 Volume owner UIC     [SYSTEM]  Vol Prot  S:RWCD,O:RWCD,G:RWCD,W:RWCD

 Volume Status:  ODS-2, subject to mount verification, file high-water
     marking, write-back XQP caching enabled, write-through XFC caching
     enabled.

Disk $252$MDA0:, device type RAM Disk, is online, member of shadow set DSA0:.

 Error count                0  Shadow member operation count    128
 Allocation class         252


Disk $252$MDA1:, device type RAM Disk, is online, member of shadow set DSA0:.

 Error count                0  Shadow member operation count    157
 Allocation class         252
```

This example shows that a write bitmap exists for a shadow set, and that no write bitmap is active. If a shadow set supports write bitmaps, "device supports bitmaps" is displayed along with either "bitmaps active" or "no bitmaps active". If the device does not support write bitmaps, no message pertaining to write bitmaps is displayed.

11.
```
$ SHOW DEVICE/BITMAP DSA13
Device          BitMap     Size      Percent      Type of    Master   Active
Name             ID       (Bytes)    Populated    Bitmap     Node
DSA13:        00010007      8248        0.01%    Minimerge  IPA        Yes
```

This example shows a **SHOW DEVICE/BITMAP** display. In this example, the bitmap is an HBMM bitmap that is active on node `IPA`.

The **/BITMAP** qualifier is similar to the **/REBUILD** qualifier in that neither can be combined with most other **SHOW DEVICE** qualifiers. The **SHOW DEVICE/BITMAP** display can be brief or full; brief is the default. Use the **/FULL** qualifier to view the entire display.

If no bitmap is active (or the device is not mounted), no bitmap file ID is displayed. The phrase "no bitmaps active" is displayed.

12.
```
$ SHOW DEVICE/FULL EWA0
Device EWA0:, device type DEGXA, is online, network device, error logging is
    enabled, device is a template only.

    Error count                    1    Operations completed                0
    Owner process                 ""    Owner UIC                      [SYSTEM]
    Ownerprocess ID         00000000    Dev Prot               S:RWPL,O:RWPL,G,W
    Reference count                0    Default buffer size                512
    Current preferred CPU ID       0    Fastpath                             1
    Current Interrupt CPU ID       0
```

```
Operating characteristics: Link up, Full duplex, Autonegotiation, Jumbo frames.
    Speed (Mbits/sec)           1000
    Def. MAC addr  00-D0-59-61-6A-B2    Current MAC addr       00-D0-59-61-6A-B2
```

This example shows information regarding the LAN template device `EWA0`. Only the template device displays the state of the link. In this example, the link is up.

13.
```
SHOW DEVICE/FULL EIA4
Device EIA4:, device type i82559, is online, network device, error logging is
    enabled.

    Error count                     0    Operations completed             375225
    Owner process           "NETACP"    Owner UIC                       [SYSTEM]
    Owner process ID        39800425    Dev Prot              S:RWPL,O:RWPL,G,W
    Reference count                2    Default buffer size                1498

Operating characteristics: Full duplex, Autonegotiation.
    Speed (Mbits/sec)            100
    Def. MAC addr  00-08-02-AF-EC-D4    Current MAC addr      AA-00-04-00-CD-4D
    Protocol name             DECNET    Protocol type                     60-03
```

This example show information regarding the LAN device `EIA4`. As this device is specific to a particular LAN protocol, information regarding that protocol is displayed; this information is not displayed for the template device in the previous example, as there is no protocol associated with the template service.

SHOW DEVICES/SERVED

SHOW DEVICES/SERVED — Displays information on devices served by the mass storage control protocol (MSCP) server on this node. The **/SERVED** qualifier is required.

Format

SHOW DEVICES/SERVED

Description

The **SHOW DEVICES/SERVED** command displays information about the MSCP server and the devices it serves. This information is used mostly by system managers. The following message displays when the user issues a **SHOW DEVICE/SERVED** command when the tape server is not loaded:

```
%SHOW-E-TMSCPNOTLD, TMSCP-Server code not loaded
```

Qualifiers

/ALL

This qualifier displays the information displayed by all of the qualifiers listed below except the **/OUTPUT** qualifier.

/COUNT

Displays the number of transfer operations completed, sorted by the size of the transfers, and the number of MSCP operations that have taken place since the MSCP server was started.

/EXACT

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify a search string that must match the search string exactly and must be enclosed with quotation marks (" ").

If you specify the **/EXACT** qualifier without the **/SEARCH** qualifier, exact search mode is enabled when you set the search string with the Find (**E1**) key.

/HIGHLIGHT[=*keyword*]

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify the type of highlighting you want when a search string is found. When a string is found, the entire line is highlighted. You can use the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting.

/HOST

Displays the names of the processors that have MSCP served devices on line. The System Generation utility (SYSGEN) command **MSCP/HOST** determines how many hosts in the OpenVMS Cluster can connect to the MSCP server at one time.

/OUTPUT=[*filespec*]

Redirects output from your terminal to the specified file. If you do not specify a file, or if you do not use this qualifier, output is sent to SYS$OUTPUT.

/PAGE[=*keyword*]
/NOPAGE (default)

Controls the display of device information on the screen.

You can use the following keywords with the **/PAGE** qualifier:

Keyword	**Description**
CLEAR_SCREEN	Clears the screen before each page is displayed.
SCROLL	Displays information one line at a time.
SAVE[=*n*]	Enables screen navigation of information, where *n* is the number of pages to store.

The **/PAGE=SAVE** qualifier allows you to navigate through screens of information. The **/PAGE=SAVE** qualifier stores up to 5 screens of up to 255 columns of information. When you use the **/PAGE=SAVE** qualifier, you can use the following keys to navigate through the information:

Key Sequence	**Description**
Up arrow key, **Ctrl/B**	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.
Right arrow key	Scroll right one column.

Key Sequence	Description
Find (**E1**)	Specify a string to find when the information is displayed.
Insert Here (**E2**)	Scroll right one half screen.
Remove (**E3**)	Scroll left one half screen.
Select (**E4**)	Toggle 80/132 column mode.
Prev Screen (**E5**)	Get the previous page of information.
Next Screen (**E6**), **Return**, **Enter**, **Space**	Get the next page of information.
F10, **Ctrl/Z**	Exit. (Some utilities define these differently.)
Help (**F15**)	Display utility help text.
Do (**F16**)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.

The **/PAGE** qualifier is not compatible with the **/OUTPUT** qualifier.

/RESOURCE

Displays information about the resources available to the MSCP server for use in processing I/O requests for the devices it serves.

You make these resources available by using the following system parameters: MSCP_BUFFER, MSCP_CREDITS, MSCP_LOAD, and MSCP_SERVE_ALL.

See the *VSI OpenVMS Cluster Systems Manual* [https://docs.vmssoftware.com/vsi-openvms-cluster-systems/] or SYSGEN online help for more information.

/SEARCH="*string*"

Use with the **/PAGE=SAVE** qualifier to specify a string that you want to find in the information being displayed. Quotation marks are required for the **/SEARCH** qualifier, if you include spaces in the text string.

You can also dynamically change the search string by pressing the Find key (**E1**) while the information is being displayed. Quotation marks are not required for a dynamic search.

Examples

1.
```
$ SHOW DEVICES/SERVED
        MSCP-Served Devices on BIAK 28-MAY-2001 13:48:01.32

                                                 Queue Requests
  Device:           Status     Total Size     Current     Max     Hosts
     $11$DUA8       Online        2376153           0       2         6
     $11$DUA9        Avail        2376153           0       0         0
     $11$DUA10      Online        2376153           0       2         8
     $11$DUA11      Online        2376153           0       2         7
     $11$DUA13      Online        2376153           0       2         7
     $11$DUA14       Avail        2376153           0       0         0
     $11$DUA16       Avail        2376153           0       0         0
```

```
   $11$DUA17       Avail        2376153           0      0        0
   $11$DUA18      Online        2376153           0      1        4
   $11$DUA19      Online        2376153           0      4        7
   $11$DUA20      Online        2376153           0      1        7
   $11$DUA21      Online        2376153           0     17       12
```

This example shows the output generated by the command **SHOW DEVICES/SERVED**. The first column in the display shows the names of the devices that are served by the MSCP server. The second column shows the status of the devices. The third column shows the size, in blocks, of the device.

The Queue Requests columns show the number of I/O requests currently awaiting processing by that device and the maximum number of I/O requests that have ever been concurrently awaiting processing by that device. The last column in the display shows the number of hosts that have the device on line.

2.
```
$ SHOW DEVICES/SERVED/COUNT
       MSCP-Served Devices on BIAK 28-MAY-2001 13:49:52.41
  .
  .
  .
Request Count:
     0-7:   951154        32-39:     2168        88-103:     1618
    8-15:   197224        40-55:     2543       104-127:      189
   16-23:   137707        56-71:     8343
   24-31:      982        72-87:      141

Operations Count:
   ABORT             0   ERASE          22772   READ        1042206
   ACCESS            0   FLUSH              0   REPLACE           0
   AVAILABLE       611   GET COM STS        0   SET CTL CHR     176
   CMP CTL DAT       0   GET UNT STS  4026024   SET UNT CHR    3630
   CMP HST DAT       0   ONLINE           427   WRITE        259953
   Total       5355799
```

This example shows the information displayed by the **SHOW DEVICES/SERVED/COUNT** command. The numbers to the left of the colon (:), separated by a hyphen (-), are the size, in pages, of the requests. The numbers to the right of the colon are the number of requests of that size that have been processed by the MSCP server.

The section of the display headed by the label Operations Count shows the number of times the MSCP server has performed the MSCP operations listed. In the example, this MSCP server has performed `176` set-controller-characteristics (`SET CTL CHR`) operations, and has performed `3630` set-unit-characteristics (`SET UNT CHR`) operations.

3.
```
$ SHOW DEVICES/SERVED/RESOURCE
       MSCP-Served Devices on BIAK 28-MAY-2001 13:51:32.01
  .
  .
  .
Resources:          Total      Free      In Use
    Buffer Area:      400       400           0
    I/O Packets:        0         0

                  Current   Maximum
    Buffer Wait:        0         0
```

This example shows the information displayed by the **SHOW DEVICES/SERVED/RESOURCE** command. The Total column shows the total number of pages in the buffer area and in the number of I/O-request packets set aside for use by the MSCP server. The Free column shows the number of pages in the buffer and the number of I/O-request packets that are available for use.

The In Use column shows the number of pages within the buffer area that are in use.

The line labeled Buffer Wait shows the number of I/O requests that are currently waiting for buffer space to become available for their use, and the maximum number of I/O requests that have waited concurrently to obtain a buffer.

4.
```
(wide)$ SHOW DEVICES/SERVED/HOST
       MSCP-Served Devices on BIAK 28-NOV-2001 13:54:41.99
   .
   .
   .
                                              Queue Requests
Host:                Time of Connection      Current    Max   Devices
    IPL31        25-MAY-2001 21:44:06.44           0      1         0
    DELAND       25-MAY-2001 21:44:09.98           0      1         0
    HEAVEN       25-MAY-2001 22:03:15.67           0      7        10
    VIVA         26-MAY-2001 09:44:11.96           0      1         0
   .
   .
   .
```

This example shows the information displayed by the **SHOW DEVICES/SERVED/HOST** command. The first column contains the names of the hosts that have class drivers connected to the MSCP server. The next column contains the times at which these connections were made.

The columns under the heading Queue Requests show the number of requests the MSCP server currently has outstanding for I/O activity on the devices it serves, the maximum number of such requests that have been outstanding at one time, and the number of MSCP server devices that the listed hosts have on line.

5.
```
$ SHOW DEVICES/SERVED
       MSCP-Served Devices on HEN  3-DEC-2001 09:09:08.49
                                          Queue Requests
Device:          Status    Total Size    Current   Max  Hosts
   254$DJB1       Avail             0          0     0      0
   254$DUA2      Online       1216665          0     0      1
   254$DUA4006    Avail             0          0     0      0

      TMSCP-Served Devices on HEN  3-DEC-2001 09:09:08.74
                                          Queue Requests
Device:          Status     Position     Current   Max  Hosts
   90$MUA7        Avail             0          0     0      0
   90$MUA8        Avail             0          0     0      0
   90$MUA50      Online          3804          0     0      0
```

This example displays the output of the **SHOW DEVICES/SERVED** command from a node that has both MSCP server and TMSCP server devices. In the display, the third column for MSCP server disk devices shows the size of the disk device. The same column for TMSCP server device shows the location where each tape is currently positioned.

SHOW DISPLAY

SHOW DISPLAY — Indicates the node where output from a DECwindows application will be displayed.

Format

SHOW DISPLAY [*display-device*]

Parameter

display-device

Refers to the display-device parameter specified with the **SET DISPLAY** command. If you are directing application output to multiple workstations in the same session, you can use logical names to point to each workstation. Using the **SHOW DISPLAY** command, you can specify this logical name as the display-device parameter to see where application output will be displayed.

If you do not specify a display-device string, the logical name DECW$DISPLAY is used.

Description

DECwindows gives you the ability to run applications across a network. The **SET DISPLAY** command enables you to direct the output of client applications across the network to an alternate workstation (X display server) for viewing. For example, the **SET DISPLAY** command allows you to:

- Redirect the output from local client applications to a remote workstation for display.
- Redirect the output from remote client applications to your local workstation for display. Although the application runs on another processor, it looks the same as any other application running locally on your workstation.

By running applications on a remote processor for local display on your workstation, you can take advantage of larger computers that might be better suited to a specific computing task. By default, applications running on your workstation are displayed on your workstation.

You use the **SET DISPLAY** command to direct the output from applications to other workstations. The **SHOW DISPLAY** command lets you see where the output from these applications will be displayed.

Sample output from the **SHOW DISPLAY** command looks like the following:

```
Device:    WSA2:  [super]
Node:      0
Transport: LOCAL
Server:    0
Screen:    0
```

The description of each item follows:

- Device is your workstation device. A new WSA*n* device is created each time you use the **SET DISPLAY/CREATE** command.
- Node is the network system on which the output from applications is displayed. When you are running and displaying applications on your node, Node is 0, which is the standard shorthand notation for representing your node.

- Transport refers to the mechanism, for example, DECNET or LOCAL, that passes information between the application, the client, and the server. The server sends input from the user to the application and output from the application to the display.
- Server is 0.
- Screen is 0.

On DECwindows workstations, the Session Manager creates a default workstation device for use by DECwindows processes (like DECterm). When you use the **SET HOST** command to connect to a remote node, no workstation device is created for that process and DECW$DISPLAY is not defined. You must specifically create new display devices with the **SET DISPLAY/CREATE** command.

If no definition for DECW$DISPLAY exists, entering the **SHOW DISPLAY** command returns an error.

Qualifiers to the **SHOW DISPLAY** command enable you to display properties related to the display device and extract authorization information, as described in the following sections.

Displaying Named Properties (Alpha/IA-64 only)

Named property values are designed to store configuration information associated with the display device, such as the network address of a session manager. By using the **/ALL** and **/SYMBOLS** qualifiers, you can display all the named properties related to the display device and use them to define one or more global symbols.

See the description of the **SET DISPLAY** command for more information.

Extracting Authorization Information (Alpha/IA-64 only)

If the display device is connected to a workstation that is using either Magic Cookie or Kerberos access control, you can use the **/EXTRACT** qualifier to obtain authorization information related to the display device from the current X authority file.

See the description of the **SET DISPLAY** command for more information.

Qualifiers

/ALL (Alpha/IA-64 only)

Displays all named properties and their values related to the current display device.

/EXTRACT (Alpha/IA-64 only)

Obtains the authorization data for the display device and writes that data to SYS$OUTPUT. The authorization data is obtained from the current X authority file and is in the format expected by the X Authority utility (xauth).

Note that when using an LBX proxy server, the extracted authorization information references the address of the proxy server and not the X display server.

Do not use the **/EXTRACT** qualifier with any other **SHOW DISPLAY** qualifier.

See the description of the **SET DISPLAY** command for more information.

/QUOTA (Alpha/IA-64 only)

Displays the current name count and data space quota values for the display device. This display includes the set limit and the amount of space currently available for use.

/SYMBOLS (Alpha/IA-64 only)

Defines one or more global DCL symbols for each property displayed by the **SHOW DISPLAY** command. You can then use these property symbols in DCL command procedures.

Each global symbol name follows the form:

```
DECW$DISPLAY_name
```

The following are the symbols for predefined properties:

- DECW$DISPLAY_NODE
- DECW$DISPLAY_TRANSPORT
- DECW$DISPLAY_SCREEN
- DECW$DISPLAY_SERVER
- DECW$DISPLAY_XAUTH

Note that symbols for user-defined properties have a double underscore in the name. For example, the user-defined symbol for the value DISK$USER:[JONES] would be defined as follows:

```
$ SHOW SYMBOL DECW$DISPLAY*

  DECW$DISPLAY_NODE == "101.124.99.119"
  DECW$DISPLAY_SCREEN == "0"
  DECW$DISPLAY_SERVER == "0"
  DECW$DISPLAY_TRANSPORT == "TCPIP"
  DECW$DISPLAY_XAUTH == "DISK$USER:[SMITH]SERVER1.DECW$XAUTH"
  DECW$DISPLAY__JONES == "DISK$USER:[JONES]"
```

If a symbol name or value exceeds or violates any DCL limits or naming conventions, the symbol is not set and an error message is displayed.

If the display device uses a proxy server, the symbols DECW$DISPLAY_NODE, DECW$DISPLAY_TRANSPORT, and DECW$DISPLAY_SERVER reflect values associated with the proxy server and not the X display server.

/VALUES=(*property-name*[,...]) (Alpha/IA-64 only)

Displays the value for the specified property. If the value consists of one or more items, they are displayed as a comma-delimited list.

Examples

1.
```
$ SHOW DISPLAY
   Device:     WSA1:  [super]
   Node:       0
   Transport:  LOCAL
```

```
    Server:     0
    Screen:     0

$ SET DISPLAY/CREATE/NODE=ZEPHYR
$ SHOW DISPLAY
    Device:     WSA2:  [super]
    Node:       ZEPHYR
    Transport:  DECNET
    Server:     0
    Screen:     0
$ SPAWN/NOWAIT/INPUT=NL: RUN SYS$SYSTEM:DECW$CLOCK
```

In this example, you are logged in to your workstation, here referred to as node `0` (0 is the standard shorthand notation for representing your node.) You want to run the DECwindows Clock on your workstation and display it on another node, `ZEPHYR`.

Assuming you are authorized to display applications on `ZEPHYR`, you redirect the application's output to `ZEPHYR` with the **`SET DISPLAY`** command and enter the **`SHOW DISPLAY`** command to verify the location of the redirected display. You then run Clock. Note that a new workstation display device, `WSA2`, is created when you enter the **`SET DISPLAY/CREATE`** command.

2.
```
$ SET DISPLAY/CREATE/VALUE=(NAME=DECW$SESSION_MANAGER, -
_$ SET="tcpip/zephyr:9510")
$ SHOW DISPLAY/SYMBOLS/ALL
    Device:     WSA23:  [super]
    Node:       0
    Transport: DECNET
    Server:     0
    Screen:     0
  User-defined values:
    "DECW$SESSION_MANAGER" = "tcpip/zephyr:9510"
$ SHOW SYMBOL DECW$DISPLAY__DECW$SESSION_MANAGER
  DECW$DISPLAY__DECW$SESSION_MANAGER == "tcpip/zephyr:9510"
```

In this example, you create a display device, and set the `DECW$SESSION_MANAGER` property to the network address of a session manager on remote node `ZEPHYR` using port number `9510`. The **`SHOW DISPLAY/SYMBOLS`** command then defines a DCL symbol for the port value.

SHOW DUMP_OPTIONS (x86-64 Only)

SHOW DUMP_OPTIONS (x86-64 Only) — Displays the currently defined list of alternate devices where system crash dumps can be written.

Format

`SHOW DUMP_OPTIONS`

Parameters

None.

Example

```
$ SHOW DUMP_OPTIONS
```

```
Dump configuration data is:
        Device=$55$DKA100:
```

SHOW ENTROPY (x86-64 Only)

SHOW ENTROPY (x86-64 Only) — Displays information about derived entropy sources and the status of hardware entropy sources.

Format

SHOW ENTROPY

Description

The **SHOW ENTROPY** command displays information about derived entropy sources and the status of hardware entropy sources.

Note

Not all listed sources of entropy may be implemented. In particular, hardware sources of entropy may vary by host hardware and hypervisor.

Qualifiers

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the **SHOW ENTROPY** command is sent. By default, the output is sent to the current SYS$OUTPUT device (usually your terminal). To send the output to a file, use the **/OUTPUT** qualifier followed by a file specification.

The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification. If you enter a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS is the default file type.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

/SYMBOLS
/NOSYMBOLS (default)

Define symbols to hold information about the availability of hardware sources of entropy. The symbol names are as follows:

SHOW$RDRAND_IN_HARDWARE	TRUE if the host hardware supports the RDRAND instruction as determined by the results of the CPUID instruction. Otherwise the symbol will be set to FALSE.
SHOW$RDRAND_AVAILABLE	TRUE if the host hardware supports the RDRAND instruction, and an attempt to execute

	the instruction returned success. Otherwise the symbol will be set to FALSE. Not all hypervisor configurations support the RDRAND instruction even if the host does. Consult the documentation for your hypervisor regarding support.

Example

```
$ SHOW ENTROPY/SYMBOLS
RANDOM_SOURCES:     FFFFFFFF
Enabled Sources:    000007FF
Disabled Sources:   FFFFF800

Sources enabled by RANDOM_SOURCES:

Bit  Description
---  -----------------
 0   Cached Random Data
 1   Device Attachment
 2   Per-CPU counters
 3   Lock Manager
 4   Cluster events
 5   Tunnel sources
 6   Ethernet Packets
 7   Device Interrupts
 8   Software Interrupts
 9   File System
10   Pool Packet Deallocation
16   RDRAND Instruction (disabled)
17   VIRTIO Random Device (disabled)

System hardware supports the RDRAND instruction.
The hypervisor does not support the RDRAND instruction.

$ SHOW SYMBOL SHOW*
  SHOW$RDRAND_AVAILABLE = "FALSE"
  SHOW$RDRAND_IN_HARDWARE = "TRUE"
```

SHOW ENTRY

SHOW ENTRY — Displays information about a user's batch and print jobs or about specific job entries. **Requires read (R) access to the queue.**

Format

SHOW ENTRY [*entry-number*[,...]],[*job-name*[,...]]

Parameters

***entry-number*[,...]**

Specifies the entry number of the job you want displayed. If no entry number or job name is specified, all your own jobs (or those owned by the user specified with the **/USER_NAME** qualifier) are displayed.

***job-name*[,...]**

Specifies the name of the jobs you want displayed. The asterisk (*) and the percent sign (%) wildcard characters are allowed. If no job name or entry number is specified, all your own jobs (or those owned by the user specified with the **/USER_NAME** qualifier) are displayed.

Description

The **SHOW ENTRY** command displays information about a user's batch and print jobs or about specific job entries.

The information displayed includes the entry number, the job name, the owner, job size in blocks (for print jobs), and the job status. The name, status, and queue type of the queue in which the job is located are also displayed.

If you are only interested in information about your own batch and print jobs, the **SHOW ENTRY** command produces a better display than the **SHOW QUEUE** command. Because the **SHOW QUEUE** command displays queue information regardless of whether your jobs are in these queues, you must scan the display to locate your jobs. By contrast, the **SHOW ENTRY** command displays only information relevant to your jobs (or those owned by the user you specify with the **/USER_NAME** qualifier.) If the **/USER_NAME** qualifier is not specified, information about your own jobs is displayed.

Qualifiers

/BATCH

Selects batch jobs for display.

/BRIEF (default)

Displays the following information for each job: entry number, job name, user name, job size in blocks (for print jobs), job status, queue name, queue status, and queue type. The **/FULL** and **/FILES** qualifiers override the **/BRIEF** qualifier.

Specify the **/FULL** qualifier to obtain more job information.

/BY_JOB_STATUS[=(*keyword*,...)]

Selects for display only those jobs with the specified status. Specify the status with one or more of the following keywords:

<table>
<tr><th>Keyword</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">EXECUTING</td><td colspan="2">Requests the display of jobs in any of the following states:</td></tr>
<tr><td>ABORTING
EXECUTING
PRINTING
PROCESSING</td><td>STALLED
STARTING
SUSPENDED</td></tr>
<tr><td colspan="2">ABORTING means that the job is halting prior to normal completion and will not continue processing. A SUSPENDED state means that the job stopped during processing, but should continue when the cause of the SUSPENDED state is resolved. A STALLED or SUSPENDED state means that the job stopped during processing, but should continue when the cause of the STALLED</td></tr>
</table>

Keyword	Description
	or SUSPENDED state is resolved. PROCESSING is the executing state for server queues. PRINTING is the executing state for printer and terminal queues. STARTING means the job is beginning to be processed.
HOLDING	Requests the display of jobs on hold. Holding status indicates that the job is being held in the queue indefinitely.
PENDING	Requests the display of jobs with pending status. Pending status indicates that the job is waiting its turn to execute.
RETAINED	Requests the display of jobs retained in the queue after execution. Retained status indicates that the job has completed but remains in the queue.
TIMED_RELEASE	Requests the display of jobs on hold until a specified time. Timed-release status indicates that the job is being held in the queue for execution at a specified time.

If no keyword is specified, the **/BY_JOB_STATUS** qualifier displays the status of all jobs.

/DEVICE[=(*keyword*,...)]

Selects for display only those print jobs in the queue types specified. Specify the queue type with one or more of the following keywords:

Keyword	Description
PRINTER	Requests the display of jobs in print queues.
SERVER	Requests the display of jobs in server queues.
TERMINAL	Requests the display of jobs in terminal queues.

If no keyword is specified, the **/DEVICE** qualifier displays all jobs in printer, terminal, or server queues.

/EXACT

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify a search string that must match the search string exactly and must be enclosed with quotation marks (" ").

If you specify the **/EXACT** qualifier without the **/SEARCH** qualifier, exact search mode is enabled when you set the search string with the Find (**E1**) key.

/FILES

Adds to the default display the list of full file specifications for each file in each job. The **/FILES** qualifier overrides the default **/BRIEF** qualifier.

/FULL

Displays the following information for each job: entry number, job name, user name, job status, full file specification associated with each job, date and time of submission, settings specified for the job, queue name, queue status, and queue type.

The **/FULL** qualifier overrides the default **/BRIEF** qualifier.

/GENERIC

Selects for display only those jobs contained in generic queues. A generic queue holds jobs of a particular type (for example, batch or line printer jobs) and directs them to execution queues for processing.

/HIGHLIGHT[=*keyword*]

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify the type of highlighting you want when a search string is found. When a string is found, the entire line is highlighted. You can use the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting.

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the **SHOW ENTRY** command is sent. By default, the output is sent to the current SYS$OUTPUT device (usually your terminal). To send the output to a file, use the **/OUTPUT** qualifier followed by a file specification.

The asterisk (*) and the percent sign (%)wildcard characters are not allowed in the file specification. If you enter a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS is the default file type.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

/PAGE[=*keyword*]
/NOPAGE (default)

Controls the display of information on the screen.

You can use the following keywords with the **/PAGE** qualifier:

Keyword	**Description**
CLEAR_SCREEN	Clears the screen before each page is displayed.
SCROLL	Displays information one line at a time.
SAVE[= *n*]	Enables screen navigation of information, where *n* is the number of pages to store.

The **/PAGE=SAVE** qualifier allows you to navigate through screens of information. The **/PAGE=SAVE** qualifier stores up to 5 screens of up to 255 columns of information. When you use the **/PAGE=SAVE** qualifier, you can use the following keys to navigate through the information:

Key Sequence	**Description**
Up arrow key, **Ctrl/B**	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.

Key Sequence	Description
Right arrow key	Scroll right one column.
Find (**E1**)	Specify a string to find when the information is displayed.
Insert Here (**E2**)	Scroll right one half screen.
Remove (**E3**)	Scroll left one half screen.
Select (**E4**)	Toggle 80/132 column mode.
Prev Screen (**E5**)	Get the previous page of information.
Next Screen (**E6**), **Return**, **Enter**, **Space**	Get the next page of information.
F10, **Ctrl/Z**	Exit. (Some utilities define these differently.)
Help (**F15**)	Display utility help text.
Do (**F16**)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.

The **/PAGE** qualifier is not compatible with the **/OUTPUT** qualifier.

/SEARCH="*string*"

Use with the **/PAGE=SAVE** qualifier to specify a string that you want to find in the information being displayed. Quotation marks are required for the **/SEARCH** qualifier, if you include spaces in the text string.

You can also dynamically change the search string by pressing the Find key (**E1**) while the information is being displayed. Quotation marks are not required for a dynamic search.

/USER_NAME=*username*

Selects for display those jobs owned by the specified user. If the **/USER_NAME** qualifier is not specified, information about your own jobs is displayed. The name must be 1 to 12 alphanumeric characters.

/WRAP
/NOWRAP (default)

Use with the **/PAGE=SAVE** qualifier to limit the number of columns to the width of the screen and to 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 and can be seen when you use the scrolling (left and right) features provided by the **/PAGE=SAVE** qualifier.

Examples

1.
```
$ SHOW ENTRY/BY_JOB_STATUS=RETAINED
Entry    Jobname       Username    Blocks      Status
-----    -------       --------    ------      ------
  422    FORECAST      JONES                   Retained on completion
         On stopped generic batch queue CLUSTER_BATCH
         Completed 21-DEC-2001 16:18 on queue NODE_BATCH
```

In this example, the **SHOW ENTRY** command displays a job entry that was retained on completion.

2. (wide)

```
$ SHOW ENTRY/FULL S*
Entry     Jobname      Username    Blocks     Status
-----     -------      --------    ------     ------
  625     STAFF        ESTES          112     Pending (queue stopped)
          On stopped printer queue LN01$PRINT
          Submitted 19-DEC-2001 12:14 /FORM=LN01$PORTRAIT
          (stock=DEFAULT) /NOTIFY /PRIORITY=100
          File: _DKA1:[ESTES]STAFF.DIS;3

  629     SEARCH         ESTES          94      Printing
          On printer queue LINE$PRINT
          Submitted  19-DEC-2001 12:16 /FORM=DEFAULT /NOTIFY
          /PRIORITY=100 /COPIES=2 /NOFEED /PASSALL
          File: _DKA1:[ESTES.DAILY]SEARCH.TXT;2 (printing copy 2)
```

In this example, the **SHOW ENTRY** command displays the current job name entries beginning with the letter "S" owned by ESTES on all queues. The **/FULL** qualifier lists the submission information, the full file specification, and the current settings for the job.

Note that entry 629 specified the **/COPIES**, **/NOFEED**, and **/PASSALL** qualifiers to the **PRINT** command. The **/NOFEED** qualifier suppresses automatic form feeds. The **/PASSALL** qualifier suppresses formatting (including form feeds and carriage return characters) performed by the print symbiont.

3.

```
$ SUBMIT ASSIGNMENTS.COM
Job ASSIGNMENTS.COM (queue SYS$BATCH, entry 199) pending
  .
  .
  .
$
SHOW ENTRY 199/FULL
Entry    Jobname       Username     Status
-----    -------       --------     ------
  199    ASSIGNMENTS   JONES        Executing
         Submitted 19-DEC-2001 08:36 /KEEP /NOLOG/ NOPRINT
         /PRIORITY=100
         File: DUA2:[JONES]ASSIGNMENTS.COM;34
```

In this example, JONES submits ASSIGNMENTS.COM for batch processing. Because JONES does not specify a specific queue, the job is entered into the generic SYS$BATCH queue to await processing. After performing other tasks, JONES checks the job status and sees that the file is now executing. Note that the job entry migrated from a generic to an execution batch queue and that JONES was able to check the job status without knowing the specific batch queue name. If the user did not specify an entry number, all jobs would have been displayed.

SHOW ERROR

SHOW ERROR — Displays the error count for all devices with error counts greater than zero.

Format

SHOW ERROR

Parameters

None.

Qualifiers

/EXACT

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify a search string that must match the search string exactly and must be enclosed with quotation marks (" ").

If you specify the **/EXACT** qualifier without the **/SEARCH** qualifier, exact search mode is enabled when you set the search string with the Find (**E1**) key.

/FULL

Displays the error count for all devices, including those with no errors (The error count is either zero or a number greater than zero).

/HIGHLIGHT[=*keyword*]

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify the type of highlighting you want when a search string is found. When a string is found, the entire line is highlighted. You can use the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting.

/OUTPUT[=*filespec*]
/OUTPUT=SYS$OUTPUT (default)

Specifies the file to which the display is written. By default, the display is written to the current SYS$OUTPUT device.

If you enter the **/OUTPUT** qualifier with a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS the default file type. The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

/PAGE[=*keyword*]
/NOPAGE (default)

Controls the display of error information on the screen.

You can use the following keywords with the **/PAGE** qualifier:

Keyword	**Description**
CLEAR_SCREEN	Clears the screen before each page is displayed.
SCROLL	Displays information one line at a time.
SAVE[= *n*]	Enables screen navigation of information, where *n* is the number of pages to store.

The **/PAGE=SAVE** qualifier allows you to navigate through screens of information. The **/PAGE=SAVE** qualifier stores up to 5 screens of up to 255 columns of information. When you use the **/PAGE=SAVE** qualifier, you can use the following keys to navigate through the information:

Key Sequence	Description
Up arrow key, **Ctrl/B**	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.
Right arrow key	Scroll right one column.
Find (**E1**)	Specify a string to find when the information is displayed.
Insert Here (**E2**)	Scroll right one half screen.
Remove (**E3**)	Scroll left one half screen.
Select (**E4**)	Toggle 80/132 column mode.
Prev Screen (**E5**)	Get the previous page of information.
Next Screen (**E6**), **Return**, **Enter**, **Space**	Get the next page of information.
F10, **Ctrl/Z**	Exit. (Some utilities define these differently.)
Help (**F15**)	Display utility help text.
Do (**F16**)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.

The **/PAGE** qualifier is not compatible with the **/OUTPUT** qualifier.

/SEARCH="*string*"

Use with the **/PAGE=SAVE** qualifier to specify a string that you want to find in the information being displayed. Quotation marks are required for the **/SEARCH** qualifier, if you include spaces in the text string.

You can also dynamically change the search string by pressing the Find key (**E1**) while the information is being displayed. Quotation marks are not required for a dynamic search.

/WRAP
/NOWRAP (default)

Use with the **/PAGE=SAVE** qualifier to limit the number of columns to the width of the screen and to 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 and can be seen when you use the scrolling (left and right) features provided by the **/PAGE=SAVE** qualifier.

Example

```
$ SHOW ERROR
Device                          Error Count
PNA0:                                 2
```

```
$1$DGA22: (AJAM PGA, ALPHAB)             1
$11$DUA1915: (HSJ018, HSJ019)           14
```

The **SHOW ERROR** command displays the error count for all devices with error counts greater than zero.

SHOW FASTPATH (Alpha/IA-64 only)

SHOW FASTPATH — Displays fast path port assignment and usage. **SHOW FASTPATH** is useful for evaluating the distribution of interrupts among the various CPUs in complex configurations.

Format

SHOW FASTPATH

Qualifiers

/CPU=*n*[,...]

Requests fast path information for one or more specific CPUs. By default, information is displayed for all CPUs.

/OUTPUT[=*filename*]
/NOOUTPUT

Controls where output from the command is sent. If you omit the **/OUTPUT** qualifier or omit the file specification with the qualifier, the display is written to the current SYS$OUTPUT device by default. If you enter the **/NOOUTPUT** qualifier, output is suppressed.

Examples

1.
```
$ SHOW FASTPATH
Fast Path preferred CPUs on NODE1 23-MAY-2004 16:47:58.10
HPE rx2600  (900MHz/1.5MB) with 2 CPUs

Device:           Fastpath CPU:
PKA0                     1
PEA0                     0
PKB0                     0
EIA0                     0
EWA0                     1

OpenVMS TCP/IP is currently running on CPU 1
```

   This example shows all the devices on NODE1 and the CPUs to which they are assigned.

2.
```
$ SHOW FASTPATH/CPU=(5,7,9)
Fast Path preferred CPUs on NODE2 23-MAY-2004 09:50:55.85
HPE AlphaServer GS140 6/525 with 6 CPUs

Device:           Fastpath CPU:
PKB0                     7
PKD0                     5
EBA0                     7
EWB0                     7
```

```
EWD0                    5
PGA0                    9

OpenVMS Lock Manager is currently running on CPU 5
```

This example shows all the devices on NODE2 that are assigned to CPUs 5, 7, and 9.

SHOW IMAGE

SHOW IMAGE — Displays information about the specified IA-64 image file. This command can be executed on an Alpha or IA-64 system, but it is designed only to display attributes of IA-64 images. These attributes can be modified using the **SET IMAGE** command.

Format

SHOW IMAGE ***image-filespec***

Parameter

image-filespec

Specifies the name of an OpenVMS IA-64 image file whose attributes you want to display. The asterisk (*) and percent sign (%) wildcard characters are allowed in the file specification. If you omit the file type extension, the default file type .EXE is applied.

Description

Displays information about an IA-64 image file. The information displayed includes the image identification, the linker identification, and information indicating whether the image has been modified since the original link.

Qualifier

/OUTPUT[=*file-spec*]

Writes the **SHOW IMAGE** output to a specified file rather than to the current SYS$OUTPUT device. If you specify the **/OUTPUT** qualifier without a file specification, the output is directed to SYS$OUTPUT. If you omit the file type in the file specification, the default file type is .LIS. If you specify a file type and omit the file name, the default file name is SHOW. No wildcard characters are allowed in the file specification.

/PAGE[=*keyword*]
/NOPAGE (default)

Controls the display of **SHOW IMAGE** information on the screen.

You can use the following keywords with the **/PAGE** qualifier:

Keyword	Description
CLEAR_SCREEN	Clears the screen before each page is displayed.

Keyword	Description
SCROLL	Displays information one line at a time.
SAVE[=n]	Enables screen navigation of information, where *n* is the number of pages to store.

The **/PAGE** qualifier is not compatible with the **/OUTPUT** qualifier.

The **/PAGE=SAVE** qualifier allows you to navigate through screens of information. The **/PAGE=SAVE** qualifier can store up to 5 screens of up to 255 columns of information. When you use the **/PAGE=SAVE** qualifier, you can use the following keys to navigate through the output:

Key Sequence	Description
Up arrow key, **Ctrl/B**	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.
Right arrow key	Scroll right one column.
Find (**E1**)	Specify a string to find when the information is displayed.
Insert Here (**E2**)	Scroll right one half screen.
Remove (**E3**)	Scroll left one half screen.
Select (**E4**)	Toggle 80/132 column mode.
Prev Screen (**E5**)	Get the previous page of information.
Next Screen (**E6**),	Get the next page of information.
Return, **Enter**, **Space**, **F10**, and **Ctrl/Z**	Exit. (Some utilities define these differently.)
Help (**F15**)	Display utility help text.
Do (**F16**)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.

Examples

1.
```
$ SHOW IMAGE PATCH
Show Image Version 1.2                    29-SEP-2004 07:52:08.41
WORK1:[SWEENEY.CLIUTL]PATCH.EXE;65
This is an OpenVMS IA64 (Elf format) executable image file

Image Identification Information

Image name: PATCH
Global Symbol Table name: PATCH
Image file identification: X-6
Image build identification: <unavailable>
Link identification: Linker T02-14
Link Date/Time: 28-SEP-2004 07:49:17.57
Patch Date/Time: Never
Manipulation Date/Time: 29-SEP-2004 07:51:58.73
```

```
Image Dynamic Data Version: 1.1

Image Completion Code Value: SUCCESS State: Original

Image Link Flags state: Manipulated

 Current Image Flags            Original Link Flags
 -------------------            -------------------
 IMGSTA                         CALL_DEBUG
 MAIN                           IMGSTA
 TBK_IN_IMG                     MAIN
 DBG_IN_IMG                     TBK_IN_IMG
 DBG_IN_IMG
 Image Link Flag                Description
 ---------------                --------------------
 CALL_DEBUG : Call debugger
     IMGSTA : Call SYS$IMGSTA
       MAIN : Image has main transfer
 TBK_IN_IMG : Traceback records in image file
 DBG_IN_IMG : Debug records in image file
```

In this example, the **SHOW IMAGE** command displays the attributes of the file PATCH.EXE located in the current working directory. The image was modified from the original attributes on 29-SEP-2004.

The image link flags output indicates the image was changed to not call the debugger at image startup. The output also shows that the traceback symbols and the debugger symbols are present in the image file.

The first line of output displays the version of **SHOW IMAGE** installed on the system. The line titled "Image Dynamic Data Version" shows the version of **SHOW IMAGE** that was available when the image was linked. Future updates to the IA-64 ELF image format might cause the image dynamic data version to increase. When the image version does not match the **SHOW IMAGE** version, a new version of **SHOW IMAGE** might be required to display the new ELF format data.

2.
```
$ SHOW IMAGE SYS$LIBRARY:DCLTABLES
Show Image Version 1.2                   29-SEP-2004 07:59:48.07
SYS$COMMON:[SYSLIB]DCLTABLES.EXE;928
This is an OpenVMS IA64 (Elf format) executable image file

Image Identification Information

Image name: DCLTABLES
    Global Symbol Table name: DCLTABLES
   Image file identification: 0-0
  Image build identification: XAAU-BL1-0000
         Link identification: Linker T02-02
              Link Date/Time: 12-MAY-2004 14:39:38.30
             Patch Date/Time: Never
       Manipulation Date/Time: Never
  Image Dynamic Data Version: <unavailable>
 Image Completion Code Value: SUCCESS State: Original
      Image Link Flags state: <unavailable>
```

This example lists the image information for the DCLTABLES.EXE file in SYS$LIBRARY. The <unavailable> string in the image dynamic data version line indicates that this image cannot be manipulated with the **SET IMAGE** command.

SHOW INTRUSION

SHOW INTRUSION — Displays the contents of the intrusion database. **Requires SECURITY privilege.**

Format

SHOW INTRUSION

Description

The OpenVMS system stores information in the intrusion database about login failures that originate from a specific source and that result from any number of failure types (invalid password, account expired, unknown user name). A security manager can identify possible break-in attempts by using the **SHOW INTRUSION** command to display the contents of the intrusion database.

The entries in the intrusion database have the following format:

```
Intrusion      Type      Count      Expiration      Source
```

The information provided in the fields in each entry is as follows:

Field	Description
Intrusion	Class of intrusion. The type of evasive action that the OpenVMS system takes depends on the class of intrusion.
Type	Severity of intrusion as defined by the threshold count for login failures.
Count	Number of login failures associated with a particular source.
Expiration	Absolute time at which a login failure is no longer counted by OpenVMS. The system parameter LGI_BRK_TMO controls how long the OpenVMS system keeps track of a login failure.
Source	Origin of the login failure. The information provided in this field depends on the class of intrusion.

In the intrusion database, the operating system classifies login failures according to their source. The four classes of system intrusion are as follows:

Intrusion Class	Description
NETWORK	Login failure originating from a remote node, using a valid user name.
TERMINAL	Login failure originating from one terminal.
TERM_USER	Login failure originating from one terminal, using a valid user name.
USERNAME	Login failure attempting to create a detached process.

The class of intrusion determines the type of information presented in the source field of the entry. Information appears in the source field in one of the following formats:

Intrusion Class	Format of Source Field
NETWORK	node::user name
TERMINAL	terminal:

Intrusion Class	Format of Source Field
TERM_USER	terminal:user name
USERNAME	user name

The type of evasive action that a security manager can take is based on the type of information provided. For details on how to use this information, see the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#d0e11646].

The intrusion database contains two levels of intrusion entries: suspect and intruder. The severity level of an entry is displayed in the type field of the entry. When a login failure associated with a particular source occurs, the OpenVMS system classifies the login failure as suspect. Each succeeding login failure from the same source is counted. The login failure count is displayed in the count field of the entry. The absolute time at which the login failure ceases to be counted is displayed in the expiration field of the entry. When the number of login failures exceeds the number specified by the system parameter LGI_BRK_LIM, the entry is classified as an intruder. However, if the parameter LGI_BRK_LIM is set to zero, the first login failure is not classified as an intruder; the result is the same as if the parameter LGI_BRK_LIM were set to one.

When an entry is promoted to intruder, the OpenVMS system takes evasive action by blocking all login attempts from that particular source.

The duration of the evasive action is determined by the system parameter LGI_HID_TIM. The absolute time at which the evasive action ends is displayed in the expiration field of the entry.

For information on intrusion detection, prevention, and evasive actions, see the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/].

If you determine that an entry in the intrusion database resulted from a user error and not a break-in attempt, you can remove an entry from the intrusion database with the **DELETE/INTRUSION_RECORD** command. See the **DELETE/INTRUSION_RECORD** [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_66] command for more details.

Qualifiers

/NODE[=(*node-name*[,...])]

The **/NODE** qualifier displays each intrusion record with the supporting node information.

If you specify individual nodes, the supporting node information is displayed only for the nodes listed.

/OUTPUT[=*filespec*]

Directs the output from the **SHOW INTRUSION** command to the file specified with the qualifier. By default, output from the command is displayed to SYS$OUTPUT.

/TYPE=*keyword*

Selects the type of information from the intrusion database that is displayed. The valid keywords are as follows:

Keyword	Description
ALL	All entries. By default, all entries are displayed.

Keyword	Description
SUSPECT	Entries for login failures that have occurred but have not yet passed the threshold necessary to be identified as intruders.
INTRUDER	Entries for which the login failure rate was high enough to warrant evasive action.

Examples

1. `$ SHOW INTRUSION/OUTPUT=INTRUDER.LIS`

   The **SHOW INTRUSION** command in this example writes all the entries currently in the intrusion database to the file `INTRUDER.LIS`.

2. `$ SHOW INTRUSION/TYPE=INTRUDER`

```
Intrusion    Type       Count    Expiration    Source
TERMINAL     INTRUDER     9      10:29:39.16   AV34C2/LC-1-15:
NETWORK      INTRUDER     7      10:47:53.12   NODE22::RONNING
```

   In this example, the **SHOW INTRUSION** command displays all intruder entries currently in the intrusion database.

3.
```
$ SHOW INTRUSION/NODE
 NETWORK       SUSPECT        5   26-JUL-2001 08:51:25.66  POPEYE::WONG
    Node: TSAVO      Count:    2
    Node: FROGGY     Count:    2
    Node: KITTY      Count:    1
```

   This command displays each intrusion record for all nodes.

4.
```
$ SHOW INTRUSION/NODE=(FROGGY,KITTY)
 NETWORK       SUSPECT        5   26-JUL-2001 08:51:25.66  POPEYE::HAMMER
    Node: FROGGY     Count:    2
    Node: KITTY      Count:    2
```

   This command displays intrusion record information for nodes `FROGGY` and `KITTY`.

5.
```
$ SHOW INTRUSION/NODE=EVMSA
$
```

   This command shows that there are no intrusion records for node `EVMSA`.

SHOW KEY

SHOW KEY — Displays the key definitions created with the **DEFINE/KEY** command.

Format

SHOW KEY [*key-name*]

Parameter

key-name

Specifies the name of the key whose definition you want displayed. For a list of valid key names, see the **DEFINE/KEY** command in the *VSI OpenVMS DCL Dictionary: A–M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_60].

Description

After you have defined keypad keys by using the **DEFINE/KEY** command, you can use the **SHOW KEY** command to refresh your memory about a key definition. You can also use the **SHOW KEY** command with the **/DIRECTORY** qualifier to list the names of all the states in which you have created key definitions. Use the **SET KEY** command to change key states.

Qualifiers

/ALL

Displays all key definitions in the current state (or the state specified with the **/STATE** qualifier.) If you use the **/ALL** qualifier, do not specify a key name.

/BRIEF (default)
/NOBRIEF

Displays only the key definition and state. The **/BRIEF** and **/NOFULL** qualifiers are equivalent.

/DIRECTORY

Displays the names of all states for which keys have been defined. If you have not specified a state with a key definition, the **SHOW KEY/DIRECTORY** command displays DEFAULT for the state.

You cannot use the **/DIRECTORY** qualifier with any of the other **SHOW KEY** qualifiers.

/FULL
/NOFULL (default)

Displays all qualifiers associated with a definition. By default, only the state of the definition and the definition itself are displayed. The **/FULL** and **/NOBRIEF** qualifiers are equivalent.

/STATE=(*state-name*[,...])
/NOSTATE

Displays the key definitions for the specified state. If you specify only one state name, you can omit the parentheses. State names can be any appropriate alphanumeric string. State names are created with the **DEFINE/KEY** command.

If you omit the **/STATE** qualifier or use the **/NOSTATE** qualifier, key definitions in the current state are displayed.

Example

```
$ DEFINE/KEY/TERMINATE PF1 "ATTACH BOB"
%DCL-I-DEFKEY, DEFAULT key PF1 has been defined
$ SHOW KEY PF1
DEFAULT keypad definitions:
```

```
  PF1 = "ATTACH BOB"
$ SHOW KEY/FULL PF1
DEFAULT keypad definitions:
  PF1 = "ATTACH BOB"  (noecho,terminate,noerase,nolock)
```

The **SHOW KEY** command in this example displays both the definition and the state for the **PF1** key. This is the default display. The **SHOW KEY/FULL** command displays all qualifiers associated with the key definition.

SHOW LICENSE

SHOW LICENSE — Displays software product licenses active on the current node and lists the names attached to a license (known as the RESERVE list). The **SHOW LICENSE** command displays the license database information currently in your system's memory. Use the License Management utility command, LICENSE LIST, when you want to view the license database information that is on disk.

Format

SHOW LICENSE [*product-name* [,...]]

Parameter

product-name

Specifies the name or names of activated software product licenses to display. The asterisk (*) and the percent sign (%) wildcard characters are allowed. If you do not specify a product name, information is displayed about all active product name licenses. The *product-name* parameter is incompatible with the **/UNIT_REQUIREMENTS** qualifier.

Description

The DCL command **SHOW LICENSE** displays software product licenses active on the current node. An active license is one that has been registered in the LICENSE database *and* has been loaded into system memory. To register and activate software product licenses, use the License Management utility (LICENSE) or VMSLICENSE.COM. Some licenses are registered automatically during product installation.

For a complete description of this utility, see the *VSI OpenVMS License Management Utility Guide* [https://docs.vmssoftware.com/vsi-openvms-license-management-utility-guide/].

To display licenses registered in the LICENSE database, use the **LICENSE LIST** command.

Qualifiers

/BEFORE

Use with **/TERMINATION_DATE** and **/RELEASE_DATE** qualifiers. Selects only those licenses whose times are before the time specified with the other qualifiers.

The **/BEFORE** qualifier cannot be used with the **/SINCE** qualifier.

/BRIEF (default)

Displays a summary of information about the specified active product licenses. Use the **/FULL** qualifier to obtain a complete product license listing.

/CHARGE_TABLE

Synonym for the **/UNIT_REQUIREMENTS** qualifier.

/CLUSTER

Use with the **/UNIT_REQUIREMENTS** qualifier to display the license unit requirements for every node in an OpenVMS Cluster.

/EXACT

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify a search string that must match the search string exactly and must be enclosed with quotation marks (" ").

If you specify the **/EXACT** qualifier without the **/SEARCH** qualifier, exact search mode is enabled when you set the search string with the Find (**E1**) key.

/FULL

Displays a summary of information about the specified active product licenses, including Product Authorization Key (PAK) options and the reserve list (if any). On IA-64 systems, lists the products included in the operating environment currently active on the system.

/HIERARCHY (IA-64 only)

Displays the hierarchy of licenses for operating environments active on the current node.

/HIGHLIGHT[=*keyword*]

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify the type of highlighting you want when a search string is found. When a string is found, the entire line is highlighted. You can use the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting.

/OE[=*OE-name*] (IA-64 only)

When *OE-name* is specified, displays the settings for the named operating environment currently active on the node. Valid *OE-name*s are FOE, BOE, EOE, MCOE, and HAOE. When no *OE-name* is specified, displays the operating environment currently active on the node.

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the **SHOW LICENSE** command is sent. By default, the output of the **SHOW LICENSE** command is sent to the current SYS$OUTPUT device (usually your terminal). To send the output to a file, use the **/OUTPUT** qualifier followed by a file specification.

The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification. If you enter a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS is the default file type.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

/PAGE[=*keyword*]
/NOPAGE (default)

Controls the display of license information on the screen.

You can use the following keywords with the **/PAGE** qualifier:

Keyword	Description
CLEAR_SCREEN	Clears the screen before each page is displayed.
SCROLL	Displays information one line at a time.
SAVE[=*n*]	Enables screen navigation of information, where *n* is the number of pages to store.

The **/PAGE=SAVE** qualifier allows you to navigate through screens of information. The **/PAGE=SAVE** qualifier stores up to 5 screens of up to 255 columns of information. When you use the **/PAGE=SAVE** qualifier, you can use the following keys to navigate through the information:

Key Sequence	Description
Up arrow key, **Ctrl/B**	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.
Right arrow key	Scroll right one column.
Find (**E1**)	Specify a string to find when the information is displayed.
Insert Here (**E2**)	Scroll right one half screen.
Remove (**E3**)	Scroll left one half screen.
Select (**E4**)	Toggle 80/132 column mode.
Prev Screen (**E5**)	Get the previous page of information.
Next Screen (**E6**), **Return**, **Enter**, **Space**	Get the next page of information.
F10, **Ctrl/Z**	Exit. (Some utilities define these differently.)
Help (**F15**)	Display utility help text.
Do (**F16**)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.

The **/PAGE** qualifier is not compatible with the **/OUTPUT** qualifier.

/PRODUCER=*producer-name*

Displays software product licenses active on the current node and supplied by the specified producer. The asterisk (*) and the percent sign (%) wildcard characters are allowed for the *producer-name* parameter. You cannot use the **/PRODUCER** qualifier with the **/UNIT_REQUIREMENTS** qualifier.

/RELEASE_DATE=[*date_time*]

Allows listing licenses using release dates as selection criteria.

/SEARCH="*string*"

Use with the **/PAGE=SAVE** qualifier to specify a string that you want to find in the information being displayed. Quotation marks are required for the **/SEARCH** qualifier, if you include spaces in the text string.

You can also dynamically change the search string by pressing the Find key (**E1**) while the information is being displayed. Quotation marks are not required for a dynamic search.

/SINCE (default)

Use with the **/TERMINATION_DATE** and **/RELEASE_DATE** qualifiers. Selects only those licenses whose times are on or after the time specified with the other qualifiers.

The **/SINCE** qualifier cannot be used with the **/BEFORE** qualifier.

/TERMINATION_DATE=*date_time*

Allows listing licenses using termination dates as selection criteria.

/UNIT_REQUIREMENTS

On Alpha systems, displays information in the License Unit Requirement Table (LURT). On IA-64 systems, displays information about the type of system, the number of active cores, and the number of sockets. The **/UNIT_REQUIREMENTS** qualifier is incompatible with the *product-name* parameter and with the **/BRIEF** and **/PRODUCER** qualifiers.

/USAGE

Tells you how many license units are loaded, how many are currently allocated, and how many are currently available, as well as the license type for each product on the system. Use with the **/FULL** qualifier to display complete information – including the PID, process name, node, or user name – for each instance of use of the product. You need group privilege to see the list of users in your group who have allocated license units; you need world privilege to see the list of users in all groups.

In an OpenVMS Cluster, if you own multiple license types for a single product, you are limited to viewing the usage information for the license type loaded on the node from which you are executing the **SHOW LICENSE/USAGE** command. To find out the usage of the other license type loaded on another node, issue the command on that node. You can also use the System Management (SYSMAN) utility to do this.

In an OpenVMS Cluster, usage information is limited to the local license type. For example, VAX and Alpha availability licenses are considered by LMF to be different license types. If you are running both VAX and Alpha systems in a cluster, usage information for availability licenses is limited to the local system type. For example, if you have DEC C installed on all nodes in your OpenVMS Cluster, you can display DEC C license allocation on all the VAX nodes in the cluster from any VAX node with DEC C installed, but you cannot display the DEC C license allocation on the Alpha nodes.

Usage information is not available for unlimited licenses (a license with 0 units). Clusterwide usage information is not available for personal use or NO_SHARE licenses. See the *VSI OpenVMS License*

Management Utility Guide [https://docs.vmssoftware.com/vsi-openvms-license-management-utility-guide/] for more information on license types.

/WARNING_INTERVAL=*n*
NOWARNING_INTERVAL

Displays a warning stating the number of licenses that will terminate in *n* days. The default is 30 days.

/WRAP
/NOWRAP (default)

Use with the **/PAGE=SAVE** qualifier to limit the number of columns to the width of the screen and to 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 and can be seen when you use the scrolling (left and right) features provided by the **/PAGE=SAVE** qualifier.

Examples

1.
```
$ SHOW LICENSE/FULL

Active licenses on node WTPOOH:
DVNETEND
        Producer: DEC
        Units: 0
        Version: 0.0
        Release Date: (none)
        Termination Date: 31-DEC-2012
        Availability: 0
        Activity: 100
        MOD_UNITS
        Product Token:

OPENVMS-ALPHA
        Producer: DEC
        Units: 0
        Version: 0.0
        Release Date: (none)
        Termination Date: 31-DEC-2012
        Availability: 0
        Activity: 100
        MOD_UNITS
        Product Token:
```

   The **SHOW LICENSE** command in this example displays all the active licenses on the current Alpha node, WTPOOH.

2.
```
$ SHOW LICENSE/FULL

Active licenses on node MACCHU:
C
        Producer: HPE
        Units: 3
        Version: 0.0
        Release Date: (none)
        Termination Date: 31-DEC-2012
        Availability: 0
```

```
        Activity: 1
        MOD_UNITS
        IA64_ALPHA
        Product Token:

DVNETEXT
        Producer: HPE
        Units: 4
        Version: 0.0
        Release Date: (none)
        Termination Date: 31-DEC-2012
        Per Core License
        Activity: 0
        IA64
        Product Token:

OPENVMS-I64-BOE
        Producer: HPE
        Units: 4
        Version: 0.0
        Release Date: (none)
        Termination Date: 31-DEC-2012
        Per Core License
        Activity: 0
        IA64
        Product Token:
```

The **SHOW LICENSE** command in this example displays all the active licenses on the current IA-64 node, MACCHU.

3. `$ SHOW LICENSE/BRIEF`

```
Active licenses on node WTPOOH:

--- Product ID ----   ---- Rating ----- -- Version --
Product   Producer Units Avail Activ Version Release Termination
DVNETEND  DEC         0  0     100    0.0     (none)   (none)
VAX-VMS   DEC         0  0     100    0.0     (none)   (none)
```

The **SHOW LICENSE** command in this example displays a summary of all the active licenses on the current VAX node, WTPOOH.

4. `$ SHOW LICENSE/OUTPUT=SYS$LOGIN:ACTIVE_LICENSES_OCT30.DAT`

The **SHOW LICENSE** command in this example writes all the active licenses to the file named SYS $LOGIN:ACTIVE_LICENSES_OCT30.DAT.

5. `$ SHOW LICENSE/FULL PERSONAL`

```
Active licenses on node PICCHU:

PERSONAL
       Producer: DEC
       Units: 100
       Version: 0.0
       Release Date: (none)
       Termination Date: (none)
       Availability: 0
```

```
        Activity: 100
        RESERVE_UNITS
        Reserve: RANCE
```

The **SHOW LICENSE** command in this example displays information about the product PERSONAL, as well as the name RANCE attached to the product license (known as the RESERVE list).

6.
```
$ SHOW LICENSE/TERM=10-JAN-2014 test0%

Active licenses on node PICCHU:

--- Product ID ----  ---- Rating ----- -- Version --
Product Producer Units Avail Activ Version Release    Termination
TEST01  DEC          0  A     0      0.0  (none)      (none)
TEST02  DEC          0  B     0      0.0  10-JAN-2014 12-NOV-2014
TEST03  DEC          0  C     0      0.0  30-DEC-2014 (none)
TEST04  DEC          0  D     0      0.0  (none)      25-AUG-2015
TEST05  DEC          0  E     0      0.0  14-NOV-2016 14-AUG-2016



$ SHOW LICENSE/RELEASE=10-JAN-2014/SINCE test0%

Active licenses on node PICCHU:

--- Product ID ----  ---- Rating ----- -- Version --
Product Producer Units Avail Activ Version Release    Termination
TEST02  DEC          0  B     0      0.0  10-JAN-2014 12-NOV-2014
TEST03  DEC          0  C     0      0.0  30-DEC-2014 (none)
TEST05  DEC          0  E     0      0.0  14-NOV-2016 14-AUG-2016



$ SHOW LICENSE/RELEASE=10-JAN-2014/BEFORE test0%

Active licenses on node PICCHU:

--- Product ID ----  ---- Rating ----- -- Version --
Product Producer Units Avail Activ Version Release    Termination
TEST01  DEC          0   A     0      0.0  (none)      (none)
TEST04  DEC          0   D     0      0.0  (none)      25-AUG-2015
```

In these examples, the **SHOW LICENSE** command uses the **/TERM**, **/RELEASE**, **/SINCE**, and **/BEFORE** qualifiers.

7.
```
$ SHOW LICENSE/UNIT_REQUIREMENTS

VMS/LMF Charge Information for node PICCHU
This is a AlphaServer 8400 5/440, hardware model type 1567

Type: A, Units Required: 2700 (VAX/VMS Capacity or OpenVMS
                                    Unlimited or Base)
Type: B, * Not Permitted *    (VAX/VMS F&A Server)
Type: C, * Not Permitted *    (VAX/VMS Concurrent User)
Type: D, * Not Permitted *    (VAX/VMS Workstation)
Type: E, * Not Permitted *    (VAX/VMS System Integrated Products)
Type: F, * Not Permitted *    (VAX Layered Products)
```

```
Type: G, * Not Permitted *    (Reserved)
Type: H, Units Required: 1150 (Alpha Layered Products)
Type: I, Units Required: 1150 (Layered Products)
```

In this example, the **/UNIT_REQUIREMENTS** qualifier displays information in the License Unit Requirement Table (LURT) for the Alpha node PICCHU.

8.
```
$ SHOW LICENSE/CHARGE_TABLE
OpenVMS I64/LMF Charge Information for node MACCHU

This is an HPE rx2600(900MHz/1.5MB), with 2 CPUs active, 2 socket(s)
Type: PPL,   Units Required: 2  (I64 Per Processor)
Type: PCL,   Units Required: 2  (I64 Per Core)
```

This example displays the CHARGE_TABLE information for an IA-64 node MACCHU with two active processor cores.

9.
```
$ SHOW LICENSE/CHAR/CLUSTER

VMS/LMF Cluster License Unit Requirements Information 14-MAR-2010
                                                          06:39:41.54

Node          A     B     C     D     E     F     G     H     I  PCL

FISH         20     -     -     -     -     -     -  1050  1050    -
SWORD        15     -     -     -     -     -     -  1050  1050    -
SALMON       12     -     -     -     -     -     -  1050  1050    -
MONGER       12     -     -     -     -     -     -  1050  1050    -
GORDON       15     -     -     -     -     -     -  1050  1050    -
ARTIST        -     -     -     -     -     -     -     -     -    2
PAINTS        -     -     -     -     -     -     -     -     -    1

Total Cluster Unit Requirements
Type: A, Units Required: 74 (VAX/VMS Capacity or OpenVMS
                                        Unlimited or Base)
Type: B, * Not Permitted *    (VAX/VMS F&A Server)
Type: C, * Not Permitted *    (VAX/VMS Concurrent User)
Type: D, * Not Permitted *    (VAX/VMS Workstation)
Type: E, * Not Permitted *    (VAX/VMS System Integrated Products)
Type: F, * Not Permitted *    (VAX Layered Products)
Type: G, * Not Permitted *    (Reserved)
Type: H, Units Required: 5250 (Alpha Layered Products)
Type: I, Units Required: 5250 (Layered Products)

Type: PPL, Units Required: 3  (I64 Per Processor)
Type: PCL, Units Required: 3  (I64 Per Core)
```

In this example, the display shows how many license units are required for each license type (A, B, etc. on Alpha and PCL on IA-64) on each node in the cluster. If a row of three asterisks (***) is displayed for a node, it means that the node is in the process of booting.

10.
```
$ SHOW LICENSE/OE

Current Operating Environment on node MACCHU at 8-MAR-2010
                                                      16:12:51.72

--------- Operating Environment ---------- ------ Units ------
Name     Description            Type Level   Loaded      Total
```

```
HAOE     High Availability        H    5          4          4
```

This example shows the currently operating environment (HAOE) on an IA-64 node MACCHU.

11.
```
$ SHOW LICENSE/HIER/FULL
                    Operating Environment Hierarchy
                        -----------------------------
--------- Operating Environment ---------- ------ Units ------

Name     Description            Type Level   Loaded      Total
HAOE     High Availability        H    5          2          2
  GWLM
MCOE     Mission Critical         H    4          -          2
  RTR-SVR
  VMSCLUSTER
  VMSCLUSTER-CLIENT
EOE      Enterprise               H    3          -          2
  AVAIL-MAN
  RMSJNL
  VOLSHAD
BOE      Base                     H    2          -          2
  DECRAM
  OMS
FOE      Foundation               H    1          -          2
  OPENVMS-I64
  OPENVMS-USER
  DVNETEND
  DW-MOTIF
  UCX
  TDC
  X500-ADMIN-FACILITY
  X500-DIRECTORY-SERVER
  CIFS
```

This example displays information about the available operating environments, the hierarchy among them, and the products contained in each OE on an IA-64 system.

12.
```
$ SHOW LICENSE/OE=BOE/FULL

--------- Operating Environment ---------- ------ Units ------
Name     Description            Type Level   Loaded      Total
BOE      Base                     H      2          4          7
  DECRAM
  OMS
  OPENVMS-I64
  OPENVMS-USER
  DVNETEND
  DW-MOTIF
  UCX
  TDC
  X500-ADMIN-FACILITY
  X500-DIRECTORY-SERVER
  CIFS
```

This example shows all the products included in the Base Operating Environment (BOE) on an IA-64 node.

13.
```
$ SHOW LICENSE OPENVMS-I64-HAOE
```

```
Active licenses on node MACCHU:

------- Product ID --------    ---- Rating ----- -- Version --
Product        Producer Units PCL Activ Version Release Termination
OPENVMS-I64-HAOE HPE      4     1   0      0.0  (none)   10-MAR-2011
```

This example shows licensing information for the HAOE environment currently active on an IA-64 node MACCHU.

14.
```
$ SHOW LICENSE/WARNING_INTERVAL=8000 test0%
Active licenses on node PICCHU:

--- Product ID ----  ---- Rating ----- -- Version --
Product  Producer Units Avail Activ Version Release    Termination
TEST01   DEC          0  A     0      0.0  (none)      (none)
TEST02   DEC          0  B     0      0.0  10-JAN-2014 12-NOV-2014
TEST03   DEC          0  C     0      0.0  30-DEC-2014 (none)
TEST04   DEC          0  D     0      0.0  (none)      25-AUG-2015
TEST05   DEC          0  E     0      0.0  14-NOV-2016 14-AUG-2016
%SHOW-I-TERMIMM, 3 licenses will terminate in 8000 days
```

The **/WARNING_INTERVAL** qualifier in this example displays three licenses that will terminate in 8000 days.

15.
```
$ SHOW LICENSE/USAGE/FULL DECWRITE-USER

View of loaded licenses from node SLTG24 29-DEC-2001 13:36:22.23

ACTIVITY license DECWRITE-USER usage information:

Pid        Process Name      Units   Username        Node
416000E6   MACAHAY             100   MACAHAY         SLTG24
416000E7   MACAHIGH            100   MACAHIGH        SLTG24
416000E8   ALICE               100   ALICE           SLTG24
416000E9   MORGEN              100   MORGEN          SLTG24
416000F1   ANGEL               100   ANGEL           SLTG24
416000F2   ANGEL_1             100   ANGEL           SLTG24

Units loaded: 2000    Units allocated: 600    Units available: 1400
```

The **SHOW LICENSE** command in this example lists the current users of the activity license for the product DECwrite. For each instance of use of the product, the process identification (PID), process name, node, and user name are identified. The units column shows the number of units allocated for each particular invocation of the product. The last line displays the units loaded when the **LICENSE LOAD** command was given, the total number of units currently allocated, and the total of unused (available for others to use) units.

16.
```
$ SHOW LICENSE/USAGE/FULL TEST_PER

View of loaded licenses from node: SLTG24   30-DEC-2001 15:45:59

PERSONAL USE license DEC TEST_PER usage information:
Units   Reserved for:
  100   UNCLE
  100   AUNT
  100   NEPHEW
  100   NIECE
```

```
Units loaded: 600    Units reserved: 400 Units available: 200
```

This example shows a personal use license. The `DEC TEST_PER` product has enough units for six reservations with `100` units for each reservation. The license database (LDB) only has a total of four names in the reserve list attached to this product. If the license administrator (usually the system manager) wants to take full advantage of this license and adds two more names to the reserve list, he should use the following commands to update the product information:

```
$ LICENSE MODIFY TEST_PER/RESERVE=(NAME, ANOTHER_NAME)/ADD
$ LICENSE UNLOAD TEST_PER
$ LICENSE LOAD TEST_PER
```

If this product is used in a cluster environment, you may use the SYSMAN utility to unload and load the license.

17. `$ SHOW LICENSE/USAGE/FULL TEST_CAP`

```
View of loaded licenses from node: SLTG24 30-DEC-2001 15:45:59

Availability license DEC TEST_CAP usage information:
Units   Node
   10   SLTG24
   10   SLTG43
  600   TORN8O
  600   LTNUP

Units loaded: 620    Units allocated: 1220    Units available: ***
```

In this example, the number of units allocated appears to be greater than the total units loaded and the units available value is three asterisks (`***`).

When you see three asterisks (`***`) as the number of units available, it is generally not a cause for alarm. This situation might arise when the license database (LDB) has been updated on disk, but the new information has not been propagated to the license database in memory on all nodes in the cluster. This node, `SLTG24`, happens to be one of the nodes that has not received the latest LDB information.

To update the information in the license database in memory for the `TEST_CAP` product, enter the following commands:

```
$ LICENSE UNLOAD TEST_CAP
$ LICENSE LOAD TEST_CAP
```

The next time you issue the **`SHOW LICENSE/USAGE`** command the three asterisks (`***`) in display should disappear. If, however, you are using multiple LDB files in a cluster, you should read the section on the license database in the *VSI OpenVMS License Management Utility Guide* [https://docs.vmssoftware.com/vsi-openvms-license-management-utility-guide/#section_tjk_y5d_s3b].

18. `$ SHOW LICENSE/UNIT_REQUIREMENT/CLUSTER`

```
VMS/LMF Cluster License Unit Requirements Information 24-DEC-2001
                                                       14:05:51.65

Node         A     B     C     D     E     F     G     H     I
KARBO        -     -     -   100    50    10     -     -    10
JENJON       -     -     -   100    50    10     -     -    10
HELENA     143     -     -     -   600  2400     -     -  2400
```

```
SHAKTI      -     -     -   100    50    10    -     -     10
Total Cluster Unit Requirements
Type: A, Units Required: 143     (VMS Capacity)
Type: B, * Not Permitted *       (VMS Server)
Type: C, * Not Permitted *       (VMS Concurrent User)
Type: D, Units Required: 300     (VMS Workstation)
Type: E, Units Required: 750     (System Integrated Products)
Type: F, Units Required: 2430    (Layered Products)
Type: G, * Not Permitted *       (VMS Reserved)
Type: H, * Not Permitted *       (Alpha Layered Products)
Type: I, Units Required: 2430    (Layered Products)
```

In this example, the display shows how many license units are required for each license type (`A`, `B`, etc.) on each node in the cluster. If a row of three asterisks (***) is displayed for a node, it means that the node is in the process of booting.

19. `$ SHOW LICENSE/USAGE`

```
View of loaded licenses from node REDSOX    8-MAR-2010 16:20:11.14

------- Product ID ---- ---- Unit usage information ------
Product          Producer Loaded Allocated Available Compliance
C                HPE         250         0       250   Yes
DVNETEXT         HPE           4         3         1   Yes
OPENVMS-I64-BOE  HPE           2         2         0   Yes
OPENVMS-I64-HAOE HPE          20         8        12   Yes
VAXSET           HPE          10         8         2   Yes
```

This example shows how many license units are loaded, how many are currently allocated, and how many are available on `REDSOX`, an IA-64 system. The last column in the display shows that are products are in compliance with their license unit requirements.

20. `$ SHOW LICENSE/USAGE`

```
View of loaded licenses from node HOVMS2   8-MAR-2010 08:38:17.13
------- Product ID -------- ---- Unit usage information --------
Product          Producer Loaded  Allocated  Available  Compliance
OPENVMS-I64-HAOE HPE    Virtual Machine guest, no usage information
```

Issuing the **`SHOW LICENSE/USAGE`** command from an OpenVMS guest cluster member displays the text "`Virtual Machine guest, no usage information`" for PCL licenses loaded on the system. There is essentially no usage charge against the license units for OpenVMS guest nodes since multiple guests can run on the same host using the same license units.

SHOW LOGICAL

SHOW LOGICAL — Displays translations, the level of translation, and the logical name table for a specified logical name. The **`SHOW LOGICAL`** command performs iterative translations. **Requires read (R) access to the table in which a logical name is cataloged to display information about the logical name.**

Format

`SHOW LOGICAL [`*`logical-name`*`[:][,...]]`

Parameter

***logical-name*[:][,...]**

Specifies one or more logical names whose translations you want to display. The asterisk (*) and the percent sign (%) wildcard characters are allowed;however, if a wildcard character is used, iterative translation is not done.

The logical name is translated iteratively up to a number of times determined by the system (from 9 to 11). That is, translations are examined to see if they are also logical names.

Description

The **SHOW LOGICAL** command displays logical names.

The logical name LNM$DCL_LOGICAL contains the list of logical name tables and the order in which they are searched. Unless LNM$DCL_LOGICAL has been redefined, the process, job, group, and system tables are searched, in that order (To see how LNM$DCL_LOGICAL is defined for your process, enter the command **SHOW LOGICAL/TABLE=LNM$DIRECTORIES LNM$DCL_LOGICAL**).

If you specify a logical name, its translations are displayed. If you do not specify a logical name, all the logical names in the tables defined by the logical name LNM$DCL_LOGICAL are displayed.

You can specify the tables you want to search. If you do not specify a table, **SHOW LOGICAL** searches the tables specified by the logical name LNM$DCL_LOGICAL.

The **SHOW LOGICAL** command performs iterative translations. If a logical name has more than one translation, then all translations at a level are displayed before going to the next level. Use the **SHOW TRANSLATION** command to display only the first translation found for a specified logical name.

The **SHOW LOGICAL** command executes an image and causes the current image (if any) to exit. Use the **SHOW TRANSLATION** command (which is built into the command interpreter) when you do not want to exit the current image.

If a logical name contains control characters, the **SHOW LOGICAL** command replaces them with periods (.) for display.

Qualifiers

/ACCESS_MODE=*mode*

Displays names defined in the specified access mode and any inner access modes. You can specify one of the following keywords to indicate the access mode: USER_MODE, SUPERVISOR_MODE, EXECUTIVE_MODE, or KERNEL_MODE.

The default value for this qualifier is USER_MODE; by default any definitions in all four access modes are displayed.

/ALL (default)

Indicates that all logical names in the specified logical name tables are to be displayed. If you do not enter the **/PROCESS**, **/JOB**, **/GROUP**, **/SYSTEM**, or **/TABLE** qualifier, all logical names in the tables specified by the logical name LNM$DCL_LOGICAL are displayed.

/CLUSTER

Displays all the logical names in all clusterwide tables (for example, the LNM$CLUSTER and LNM$SYSCLUSTER tables).

/DESCENDANTS
/NODESCENDANTS (default)

Controls whether the system displays names from the specified logical name table and any descendant tables. A descendant table is created by the **CREATE/NAME_TABLE** command, with the **/PARENT_TABLE** qualifier specifying its parent table. If you use the **/DESCENDANTS** qualifier, you must also use the **/TABLE** qualifier.

/EXACT

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify a search string that must match the search string exactly and must be enclosed with quotation marks (" ").

If you specify the **/EXACT** qualifier without the **/SEARCH** qualifier, exact search mode is enabled when you set the search string with the Find (**E1**) key.

/FULL

Displays more detailed information for the specified logical name. The information includes the access mode, attributes, the translation, and the logical name table.

/GROUP

Indicates that only the group logical name table is to be searched. The **/GROUP** qualifier is synonymous with the **/TABLE=LNM$GROUP** qualifier. If you specify the **/GROUP** qualifier and you do not also specify a logical name, all names in the group table are displayed.

/HIGHLIGHT[=*keyword*]

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify the type of highlighting you want when a search string is found. When a string is found, the entire line is highlighted. You can use the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting.

/JOB

Indicates that only the job logical name table is to be searched. The **/JOB** qualifier is synonymous with the **/TABLE=LNM$JOB** qualifier. If you specify the **/JOB** qualifier and you do not also specify a logical name, all names in the job logical name table are displayed.

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the **SHOW LOGICAL** command is sent. By default, the output of the **SHOW LOGICAL** command is sent to the current SYS$OUTPUT device (usually your terminal). To send the output to a file, use the **/OUTPUT** qualifier followed by a file specification.

The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification. If you enter a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS is the default file type.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

/PAGE[=*keyword*]
/NOPAGE (default)

Controls the display of information on the screen.

You can use the following keywords with the **/PAGE** qualifier:

Keyword	Description
CLEAR_SCREEN	Clears the screen before each page is displayed.
SCROLL	Displays information one line at a time.
SAVE[= *n*]	Enables screen navigation of information, where *n* is the number of pages to store.

The **/PAGE=SAVE** qualifier allows you to navigate through screens of information. The **/PAGE=SAVE** qualifier stores up to 5 screens of up to 255 columns of information. When you use the **/PAGE=SAVE** qualifier, you can use the following keys to navigate through the information:

Key Sequence	Description
Up arrow key, **Ctrl/B**	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.
Right arrow key	Scroll right one column.
Find (**E1**)	Specify a string to find when the information is displayed.
Insert Here (**E2**)	Scroll right one half screen.
Remove (**E3**)	Scroll left one half screen.
Select (**E4**)	Toggle 80/132 column mode.
Prev Screen (**E5**)	Get the previous page of information.
Next Screen (**E6**), **Return**, **Enter**, **Space**	Get the next page of information.
F10, **Ctrl/Z**	Exit. (Some utilities define these differently.)
Help (**F15**)	Display utility help text.
Do (**F16**)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.

The **/PAGE** qualifier is not compatible with the **/OUTPUT** qualifier.

/PROCESS

Indicates that only the process logical name table is to be searched. The **/PROCESS** qualifier is synonymous with the **/TABLE=LNM$PROCESS** qualifier. If you specify the **/PROCESS** qualifier and you do not also specify a logical name, all names in the process table are displayed.

/SEARCH="*string*"

Use with the **/PAGE=SAVE** qualifier to specify a string that you want to find in the information being displayed. Quotation marks are required for the **/SEARCH** qualifier, if you include spaces in the text string.

You can also dynamically change the search string by pressing the Find key (**E1**) while the information is being displayed. Quotation marks are not required for a dynamic search.

/STRUCTURE
/NOSTRUCTURE (default)

Controls whether the system displays the "family tree" of all accessible logical name tables. The display includes the two logical name directory tables (process and system) and all logical name tables cataloged in these directory tables. Any descendant logical name tables are shown under their parent tables.

If you specify the **/STRUCTURE** qualifier, you cannot use any other qualifiers except **/ACCESS_MODE**, **/FULL**, and **/OUTPUT**.

/SYSTEM

Indicates that only the system logical name table is to be searched. The **/SYSTEM** qualifier is synonymous with the **/TABLE=LNM$SYSTEM** qualifier. If you specify the **/SYSTEM** qualifier and you do not also specify a logical name, all names in the system table are displayed.

/TABLE=(*name*[,...])

Specifies the tables you want to search. If you specify only one table, you can omit the parentheses. The asterisk (*) and the percent sign (%) wildcard characters are allowed. Names with wildcards are used to match table names. Names without wildcard characters are treated both a stable names and table search lists (whichever is appropriate).

You can use the **/TABLE** qualifier to specify the following:

- A user-defined logical name table (created with the **CREATE/NAME_TABLE** command)
- The process, group, or system logical name tables
- The process or system directory tables

If you specify the table name by using a logical name that translates to more than one table, then each table is searched in the order specified. For example, if you specify **SHOW LOGICAL/TABLE=LNM$FILE_DEV**, and LNM$FILE_DEV is equated to LNM$PROCESS, LNM$JOB, LNM$GROUP, and LNM$SYSTEM, then the process, job, group, and system tables are searched, in that order.

If you do not specify the **/TABLE** qualifier, the default is **/TABLE=LNM$DCL_LOGICAL**.

/WRAP
/NOWRAP (default)

Use with the **/PAGE=SAVE** qualifier to limit the number of columns to the width of the screen and to 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 and can be seen when you use the scrolling (left and right) features provided by the **/PAGE=SAVE** qualifier.

Examples

1.
```
$ SHOW LOGICAL/PROCESS
(LNM$PROCESS_TABLE)
  "SYS$COMMAND" = "_TTB4:"
```

```
"SYS$DISK" = "WORK6:"
"SYS$DISK" = "WORK6:"
"SYS$ERROR" = "_TTB4:"
"SYS$INPUT" = "_TTB4:"
"SYS$LOGIN" = "WORK6:[ODONNELL]"
"SYS$LOGIN_DEVICE" = "WORK6:"
"SYS$OUTPUT" = "_TTB4:"
"SYS$OUTPUT" = "DKA2:"
"SYS$SCRATCH" = "WORK6:[ODONNELL]"
```

The **SHOW LOGICAL** command in this example displays all process logical names and their translations. Note that **/TABLE=LNM$PROCESS** would produce the same display as **/PROCESS**.

2.
```
$ SHOW LOGICAL INFILE
  "INFILE" = "WORK6:[LOGAN]PAYROLL.EXE" (LNM$PROCESS_TABLE)
```

   The **SHOW LOGICAL** command in this example displays the translation for the logical name INFILE. The response indicates that the logical name was found in the process logical name table.

3.
```
$ SHOW LOGICAL/GROUP
   .
   .
   .
```

   The **SHOW LOGICAL** command in this example displays all group logical names and their translations. Note that **/TABLE=LNM$GROUP** would produce the same display as **/GROUP**.

4.
```
$ SHOW LOGICAL/TABLE=SYSTEM  SYS$LIBRARY
  "SYS$LIBRARY" = "SYS$SYSROOT:[SYSLIB]" (LNM$SYSTEM_TABLE)
                = "DOCD$:[SYSC.SYSLIB]"
```

   The **SHOW LOGICAL** command in this example displays the translation of the logical name SYS$LIBRARY in the system table. The response indicates that SYS$LIBRARY is defined in the system table, and that the logical name has two translations.

5.
```
$ SHOW LOGICAL/TABLE=LNM$GROUP/TABLE=LNM$SYSTEM SYS$DISK
  "SYS$DISK" = "ZZZ3:" (LNM$SYSTEM_TABLE)
```

   The **SHOW LOGICAL** command in this example is qualified by both the **/TABLE=LNM$GROUP** and **/TABLE=LNM$SYSTEM** qualifiers. The response indicates that the logical name SYS$DISK was found in the system logical name table. When you enter two conflicting qualifiers, as in this example, only the last qualifier you specify is used.

6.
```
$ SHOW LOGICAL/TABLE=LNM$PROCESS_DIRECTORY
```

   The **SHOW LOGICAL** command in this example displays the logical names in the process directory table. Each name is either a table name, or a name that translates iteratively to a table.

SHOW MEMORY

SHOW MEMORY — Displays the availability and usage of those system resources that are related to memory.

Format

SHOW MEMORY

Parameters

None.

Description

The information provided by the **SHOW MEMORY** command can help you determine whether to change certain system memory resources to improve system performance. The system memory resources are as follows:

Physical memory
Process entry slots and balance slots
Nonpaged and paged dynamic memory
Space in paging and swapping files
File data cache memory usage

When the **SHOW MEMORY** command is executed, a display is written to SYS$OUTPUT. Depending on which qualifiers you specify, the display shows the following memory resource statistics:

Physical memory usage
Bad page list
Number of pages allocated to an OpenVMS system
Slot usage
Dynamic memory usage (in bytes)
Paging file usage (in blocks)
Dynamic memory
Granularity hint regions for code and data (Alpha/IA-64 only)
Extended File Cache facility (Alpha/IA-64 only)
Write bitmap (WBM) memory use (Alpha/IA-64 only)

For a detailed description of each item displayed, see the Examples section.

Qualifiers

/ALL (default)

Displays all available information, that is, information displayed by the **/FILES**, **/PHYSICAL_PAGES**, **/POOL**, and **/SLOTS** qualifiers.

/BITMAP (Alpha/IA-64 only)

Displays information about memory use by write bitmaps (WBM) used with Host-Based Volume Shadowing (HBVS). If the system parameter SHADOWING is set to 0, no information about bitmaps is displayed because WBM is used only with HBVS.

Note that the memory consumption displayed by **SHOW MEMORY** is often greater than that displayed by **SHOW DEVICE/BITMAP**, which displays the exact size (in bytes) of the bitmap. The memory used by a bitmap is allocated in units of pages; if a bitmap uses part of a page of memory, the rest of that page is unavailable for use.

/BUFFER_OBJECTS (Alpha/IA-64 only)

Displays information about system resources used by buffer objects.

/CACHE[=(*keyword*[,...])]

Displays information about the Virtual I/O Cache facility or the Extended File Cache facility, depending on which is loaded. The Cache facility information is displayed as part of the **SHOW MEMORY** and **SHOW MEMORY/CACHE/FULL** commands.

If the Extended File Cache is loaded, you can specify the following keywords and options:

Keyword	Description	
VOLUME[=*device*]	Requests display of volume-level statistics for the specified volume(s). If no device is specified, defaults to SYS$DISK. Device specifications may be wildcarded; if so, the device specification must be based on the physical device name. Specify VOLUME=* to see statistics for all volumes in cache.	
FILE=*filespec*	Requests display of file-level statistics for the specified file(s) in cache. The filespec must contain an explicit or implicit device specification. The filespec may be wildcarded according to RMS rules. Specify FILE=dev:*.* to see statistics for all files in cache for a device. Partially wild file specifications that are satisfied by large numbers of files on disk may incur significant processing overhead during display. By default, both open and closed files are displayed (use the OPEN and CLOSED keywords to further restrict which files are displayed). Note that this command may fail if you select a large number of files for display. If you receive an error message, reduce the number of selected files and reissue the command.	
TOPQIO[=*n*]	Requests display of file-level statistics for the *n* files in cache with the highest overall QIO count (*n* defaults to 10).	
TOPHITRATE[=*n*]	Requests display of file-level statistics for the *n* files in cache with the highest overall cache hit rate (*n* defaults to 10).	
	OPEN	Include only open files in the display (default for TOPQIO and TOPHITRATE).
	CLOSED	Include only closed files in the display.

/EXACT

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify a search string that must match the search string exactly and must be enclosed with quotation marks (" ").

If you specify the **/EXACT** qualifier without the **/SEARCH** qualifier, exact search mode is enabled when you set the search string with the Find (**E1**) key.

/FILES

Displays information about the use of each paging and swapping file currently installed.

/FULL

When used with **/POOL**, displays additional information about pool area.

When used with **/CACHE**, displays additional information about the file data cache that is loaded (either VIOC or XFC).

This qualifier is ignored unless the **/CACHE** or **/POOL** qualifier is specified explicitly.

/GH_REGIONS

On Alpha, displays information about the granularity hint regions (GHR) that have been established. For each of these regions, information is displayed about the size of the region, the amount of free memory, the amount of memory in use, and the amount of memory released to OpenVMS from the region. The granularity hint regions information is also displayed as part of **SHOW MEMORY**, **SHOW MEMORY/ALL**, and **SHOW MEMORY/FULL** commands.

/HIGHLIGHT[=*keyword*]

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify the type of highlighting you want when a search string is found. When a string is found, the entire line is highlighted. You can use the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting.

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter the **/OUTPUT** qualifier without a file specification, the output is sent to the current process default output stream or device, identified by the logical name SYS$OUTPUT.

If you enter the **/OUTPUT** qualifier with a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS the default file type. If you enter a file specification, it cannot include asterisk (*) and the percent sign (%) wildcard characters.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

/PAGE[=*keyword*]
/NOPAGE (default)

Controls the display of memory information on the screen.

You can use the following keywords with the **/PAGE** qualifier:

Keyword	**Description**
CLEAR_SCREEN	Clears the screen before each page is displayed.
SCROLL	Displays information one line at a time.
SAVE[=*n*]	Enables screen navigation of information, where *n* is the number of pages to store.

The **/PAGE=SAVE** qualifier allows you to navigate through screens of information. The **/PAGE=SAVE** qualifier stores up to 5 screens of up to 255 columns of information. When you use the **/PAGE=SAVE** qualifier, you can use the following keys to navigate through the information:

Key Sequence	**Description**
Up arrow key, **Ctrl/B**	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.
Right arrow key	Scroll right one column.
Find (**E1**)	Specify a string to find when the information is displayed.

Key Sequence	Description
Insert Here (**E2**)	Scroll right one half screen.
Remove (**E3**)	Scroll left one half screen.
Select (**E4**)	Toggle 80/132 column mode.
Prev Screen (**E5**)	Get the previous page of information.
Next Screen (**E6**), **Return**, **Enter**, **Space**	Get the next page of information.
F10, **Ctrl/Z**	Exit. (Some utilities define these differently.)
Help (**F15**)	Display utility help text.
Do (**F16**)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.

The **/PAGE** qualifier is not compatible with the **/OUTPUT** qualifier.

/PHYSICAL_PAGES

Displays information about the amount of physical memory and the number of free and modified pages.

/POOL

Displays information about the usage of each dynamic memory (pool) area, including the amount of free space and the size of the largest contiguous block in each area.

/RESERVED (Alpha/IA-64 only)

Displays information about memory reservations.

/SEARCH="*string*"

Use with the **/PAGE=SAVE** qualifier to specify a string that you want to find in the information being displayed. Quotation marks are required for the **/SEARCH** qualifier, if you include spaces in the text string.

You can also dynamically change the search string by pressing the Find key (**E1**) while the information is being displayed. Quotation marks are not required for a dynamic search.

/SLOTS

Displays information about the availability of partition control block (PCB) vector slots and balance slots.

/UNITS=*keyword*

Allows you to override the current default specified by **SET PROCESS/UNITS** so that you can display memory in your choice of blocks or bytes.

Keyword options are:

Option	Description
BLOCKS	Displays memory in blocks (equivalent to PAGES).

Option	Description
BYTES	Displays memory in bytes.
PAGES	Displays memory in page format (equivalent to BLOCKS).

/WRAP
/NOWRAP (default)

Use with the **/PAGE=SAVE** qualifier to limit the number of columns to the width of the screen and to 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 and can be seen when you use the scrolling (left and right) features provided by the **/PAGE=SAVE** qualifier.

Examples

1.
```
$ SHOW MEMORY
System Memory Resources on 20-JAN-2001 15:22:04.11
Physical Memory Usage (pages):   Total      Free     In Use  Modified
 Main Memory (2.00Gb)           262144    201249      60162       733
Extended File Cache  (Time of last reset: 18-JAN-2001 17:15:14.33)
  Allocated (Mbytes)         370.00  Maximum size (Mbytes)    1024.00
  Free (Mbytes)              317.79  Minimum size (Mbytes)     370.00
  In use (Mbytes)             52.20  Percentage Read I/Os          72%
  Read hit rate                  96% Write hit rate                 0%
  Read I/O count             119363  Write I/O count            46259
  Read hit count             115283  Write hit count                0
  Reads bypassing cache          45  Writes bypassing cache         0
  Files cached open             372  Files cached closed          392
  Vols in Full XFC mode           0  Vols in VIOC Compatible mode   3
  Vols in No Caching mode         0  Vols in Perm. No Caching mode  0
Granularity Hint Regions (pages): Total    Free     In Use   Released
 Execlet code region               1024       0        586        438
 Execlet data region                256       0        256          0
 S0/S1 Executive data region        576       0        576          0
 Resident image code region        1024       0        830        194
Slot Usage (slots):               Total    Free   Resident    Swapped
 Process Entry Slots                647     614         33          0
 Balance Set Slots                  645     614         31          0
Dynamic Memory Usage:             Total    Free     In Use    Largest
 Nonpaged Dynamic Memory (Mb)      4.28    2.38       1.89       2.25
 Bus Addressable Memory  (Kb)    168.00  128.00      40.00     128.00
 Paged Dynamic Memory    (Mb)      5.68    3.42       2.26       3.41
 Lock Manager Dyn Memory (Mb)      1.13    0.00       1.12
Buffer Object Usage (pages):                In Use       Peak
 32-bit System Space Windows (S0/S1)            3          3
 64-bit System Space Windows (S2)               0          0
 Physical pages locked by buffer objects        3          3
Memory Reservations (pages):     Group  Reserved    In Use       Type
 VCC$MIN_CACHE_SIZE                ---     47360     47360  Allocated
 Total (370.00 Mbytes reserved)              47360     47360
Swap File Usage (8KB pages):                Index      Free       Size
 DISK$X6XV_BRAHMA:[SYS0.SYSEXE]SWAPFILE.SYS
                                               1      5168       5168
Paging File Usage (8KB pages):              Index      Free       Size
 DISK$X6XV_BRAHMA:[SYS0.SYSEXE]PAGEFILE.SYS
                                             254    215288     215288
```

```
 Total committed paging file usage:                              7418
Of the physical pages in use, 4951 pages are permanently allocated to
  OpenVMS.
```

This example shows reservable pages. If the reservable pages are a negative value, this indicates that you may want to consider adding to the fixed amount of page file space available.

2.
```
$ SHOW MEMORY/CACHE/FULL
             System Memory Resources on 3-NOV-2001 18:36:12.79

Extended File Cache  (Time of last reset: 31-OCT-2001 10:18:07.39)
Allocated (Mbytes)     ❶370.00   Maximum size (Mbytes)   ❷1024.00
Free (Mbytes)          ❸319.75   Minimum size (Mbytes)   ❹  370.0
In use (Mbytes)        ❺ 50.25   Percentage Read I/Os    ❻     98%
Read hit rate          ❼    79%  Write hit rate          ❽      0%
Read I/O count         ❾ 18110   Write I/O count         ❿    310
Read hit count         ⓫ 14343   Write hit count         ⓬      0
Reads bypassing cache  ⓭    37   Writes bypassing cache  ⓮      0
Files cached open      ⓯   351   Files cached closed      ⓰    307
Vols in Full XFC mode  ⓱     0   Vols in VIOC Compatible mode ⓱ 0
Vols in No Caching mode⓱     0   Vols in Perm No Caching mode ⓱ 0

I/O Statistics ⓲ - Distributions (MAX_IO_SIZE: 127 ⓳)
------------------------------------------------------
Transfer Size:      Reads    Read Hits     Writes
   1 Block IO:      35053 ⓴     34553 ㉑      87 ㉒
   2 Block IO:        469          284         9
   3 Block IO:      26516        26346        24
   4 Block IO:        348          205         6
   5 Block IO:      13205        13160         0
   6 Block IO:        161          111         0
   7 Block IO:        185          152         0
       .
       .
       .
 114 Block IO:          1            0         0
 116 Block IO:          2            1         0
 121 Block IO:          1            0         0

>127<256 Block IO:     12                     0
>255     Block IO:     14                     0
------------------------------------------------------
Totals:             84411        80535       630
```

Note

This example shows the output for the **SHOW MEMORY/CACHE/FULL** command on an Alpha system with the Extended File Cache loaded. The **SHOW MEMORY/CACHE/FULL** command will display different fields on an Alpha system with the Virtual I/O Cache loaded.

Extended File Cache Usage

Shows the use of the Extended File Cache facility on an Alpha system.

❶ Allocated	Total amount of system memory in megabytes allocated to the Extended File Cache.

❸ Free	Number of allocated megabytes not currently in use by the cache.
❺ In use	Number of allocated megabytes that contain valid cached data.
❼ Read hit rate	Shows cache read hits as a percentage of read I/Os.
❾ Read I/O count	Total number of read I/Os that have been seen by the cache since the last statistics reset.
❶❶ Read hit count	Total number of read I/Os that did not perform a physical I/O, because the data for these was found in the cache since the prior statistics reset.
❶❸ Reads bypassing cache	Total number of reads that bypassed the cache and went directly to disk since the last statistics reset.
❶❺ Files cached open	Total number of open files in cache.
❷ Maximum size (Mbytes)	Maximum size of the cache in megabytes.
❹ Minimum size (Mbytes)	Minimum size of the cache in megabytes. This displays the value of the memory reservation, VCC$MIN_CACHE_SIZE. See the relevant section in the *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/#CACHE_MIN_SIZE_SEC] for additional information.
❻ Percentage Read I/Os	Percentage of read I/Os that occur. For example, a read/write I/O ratio of 100% implies that all were read I/Os.
❽ Write hit rate	(Always zero for Extended File Cache, Version 1.)
❿ Write I/O count	Total number of write I/Os that have been seen by the cache since the last statistics reset.
❿❷ Write hit count	(Always zero for Extended File Cache, Version 1.)
❿❹ Writes bypassing cache	Total number of writes that bypassed the cache and went directly to disk since the last statistics reset.
❿❻ Files cached closed	Total number of closed files in cache.
❿❼ Volume modes	Total number of volumes in each caching state. Note that the initial release of XFC only supports VIOC compatible mode. Future releases will support full mode.
❿❽ I/O Statistics	Systemwide distribution of I/O sizes (rounded up).
❿❾ MAX_IO_SIZE	Current setting of the system parameter VCC_MAX_IO_SIZE in blocks. The extended file cache (XFC) will only cache files up to this size.
❷❿ Reads	Number of reads for the entire system.
❷❶ Read Hits	Number of reads satisfied from the cache.
❷❷ Writes	Number of writes for the entire system.

3.
```
$ SHOW MEMORY/CACHE=(VOLUME=*)
                System Memory Resources on 20-JAN-2001 15:35:32.51

Extended File Cache Volume Statistics:
```

```
_$7$DKA200: (DISK$X6XV_BRAHMA), Caching mode is VIOC Compatible
Open files        ❶    372      Closed files        ❷    395
Files ever opened ❸   1007      Files ever deposed  ❹    240
Allocated pages   ❺   6405      Locks acquired      ❻    736
Total QIOs        ❼ 166552      Read hits           ❽ 115984
Virtual reads     ❾ 120068      Virtual writes      ❿  46484
Hit rate          ⓫     69 %    Read aheads         ⓬  11766
Read throughs     ⓭ 120021      Write throughs      ⓮  46484
Read arounds      ⓯     47      Write arounds       ⓰      0
Ave Disk I/O Resp Time incl cache hits (microseconds)⓱   3497

Total of 1 volumes in cache
```

Extended File Cache Usage

Shows the use of the Extended File Cache facility on an Alpha system.

❶ Open files	Number of files that are currently open and cached by the extended file cache on this volume at the instant the SHOW command was issued.
❸ Files ever opened	Number of times the extended file cache has been requested to open a file on this volume.
❺ Allocated pages	Number of main memory data pages currently being used by the extended file cache to contain actual file data, for this volume.
❼ Total QIOs	Total number of QIO and PAGE I/O calls received by the extended file cache on this volume.
❾ Virtual reads	Cumulative number of Virtual Read requests received by the extended file cache on this volume.
⓫ Hit rate	Percentage of Total QIOs that were completed by copying data from cache, rather than issuing real I/O to a volume.
⓭ Read throughs	Number of Virtual Reads that are capable of being satisfied by the extended file cache.
⓯ Read arounds	Number of Virtual Reads that cannot be satisfied by the extended file cache. Reasons for this include the presence of a non-XFC compliant node in an OpenVMS cluster, and the receipt of an I/O that is larger than VCC_MAX_IO_SIZE.
❷ Closed files	Number of closed files the extended file cache has in memory. If a subsequent file access occurs, data will be accessible from main memory rather than having to issue I/O to the real volume.
❹ Files ever deposed	Files removed from memory. Reasons the extended file cache may remove a file from its memory include requests for memory from the rest of the system, and conflicting access modes in a cluster.
❻ Locks acquired	Current number of file access locks that are in use on this volume.
❽ Read hits	Cumulative number of times a read has been satisfied from the data cache.
❿ Virtual writes	Cumulative number of virtual write I/Os received on this volume by the cache.

⓬ Read aheads	Cumulative number of times the extended file cache issued read ahead I/Os on this volume after detecting sequential file I/O virtual reads.
⓮ Write throughs	Cumulative number of times a virtual write has been capable of modifying cached data.
⓰ Write arounds	Cumulative number of times a virtual write has been detected that cannot update data in cache memory.
⓱ Average Disk I/O Response Time	Average disk I/O response time, including cache hits, in microseconds. Note that on a symmetric multiprocessing (SMP) system, the response time only includes that of I/Os starting or finishing on the same CPU. Any physical I/Os are ignored.

4.
```
$ SHOW MEMORY/CACHE=(VOL=DKA100,TOPQIO=3)

System Memory Resources on 3-NOV-2001 18:36:12.79

Extended File Cache Top QIO File Statistics:

_$7$DKA200: (DISK$X6Y0_RESD), Caching mode is VIOC Compatible

_$7$DKA200:[SYS0.SYSEXE]SYSDUMP.DMP;2 (open)
Caching is enabled, active caching mode is Write Through
Allocated pages    ❶  334      Total QIOs       ❷ 1712
Read hits          ❸ 1559      Virtual reads    ❹ 1712
Virtual writes     ❺    0      Hit rate         ❻   91 %
Read aheads        ❼    0      Read throughs    ❽    0
Write throughs     ❾    0      Read arounds     ❿    0
                               Write arounds    ⓫    0

_$7$DKA200:[VMS$COMMON.SYSEXE]SYS$QUEUE_MANAGER.QMAN$JOURNAL;1 (open)
Caching is enabled, active caching mode is Write Through
Allocated pages       931      Total QIOs         1057
Read hits               5      Virtual reads      1044
Virtual writes         13      Hit rate              0 %
Read aheads             2      Read throughs      1044
Write throughs         13      Read arounds          0
                               Write arounds         0

_$7$DKA200:[VMS$COMMON.SYSLIB]DECC$SHR.EXE;1 (open)
Caching is enabled, active caching mode is Write Through
Allocated pages       135      Total QIOs          730
Read hits             673      Virtual reads       730
Virtual writes          0      Hit rate              0 %
Read aheads             2      Read throughs         0
Write throughs          0      Read arounds          0
                               Write arounds         0

Total of 3 files for this volume
```

Extended File Cache Top QIO File Statistics Usage

Shows the use of the Extended File Cache Top QIO File Statistics facility on an Alpha system.

❶ Allocated pages	Number of main memory data pages currently being used by the extended file cache to contain actual file data, for this volume
❸ Read hits	Cumulative number of times a read has been satisfied from the data cache.
❺ Virtual writes	Cumulative number of virtual write I/Os received on this volume by the cache.
❼ Read aheads	Cumulative number of times the extended file cache issued read ahead I/Os on this volume after detecting sequential file I/O virtual reads.
❾ Write throughs	Cumulative number of times a virtual write has been capable of modifying cached data.
❷ Total QIOs	Total number of QIO and PAGE I/O calls received by the extended file cache on this volume.
❹ Virtual reads	Cumulative number of Virtual Read requests received by the extended file cache on this volume.
❻ Hit rate	Percentage of Total QIOs that were completed by copying data from cache, rather than issuing real I/O to a volume.
❽ Read throughs	Number of Virtual Reads that are capable of being satisfied by the extended file cache.
❿ Read arounds	Number of Virtual Reads that cannot be satisfied by the extended file cache. Reasons for this include the presence of a non-XFC compliant node in an OpenVMS cluster, and the receipt of an I/O that is larger than VCC_MAX_IO_SIZE.
⓫ Write arounds	Cumulative number of times a virtual write has been detected that cannot update data in cache memory.

5.
```
$ SHOW MEMORY/CACHE=(VOL=DKA100,TOPHITRATE=3)

System Memory Resources on 3-NOV-2001 18:36:12.79

Extended File Cache Top Hitrate File Statistics:

_$7$DKA200: (DISK$X6Y0_RESD), Caching mode is VIOC Compatible

_$7$DKA200:[VMS$COMMON.SYSLIB]SMGSHR.EXE;1 (open)
Caching is enabled, active caching mode is Write Through
Allocated pages   ❶   16      Total QIOs      ❷  277
Read hits         ❸  267      Virtual reads   ❹  277
Virtual writes    ❺    0      Hit rate        ❻   96 %
Read aheads       ❼    0      Read throughs   ❽  277
Write throughs    ❾    0      Read arounds    ❿    0
                              Write arounds   ⓫    0

_$7$DKA200:[VMS$COMMON.SYSLIB]UTIL$SHARE.EXE;1 (open)
Caching is enabled, active caching mode is Write Through
Allocated pages         8      Total QIOs          185
Read hits             178      Virtual reads       185
Virtual writes          0      Hit rate             96 %
Read aheads             1      Read throughs       185
```

```
Write throughs           0      Read arounds           0
                                Write arounds          0

_$7$DKA200:[VMS$COMMON.SYSLIB]CMA$TIS_SHR.EXE;1 (open)
Caching is enabled, active caching mode is Write Through
Allocated pages          6      Total QIOs           165
Read hits              159      Virtual reads        165
Virtual writes           0      Hit rate              96 %
Read aheads              2      Read throughs        165
Write throughs           0      Read arounds           0
                                Write arounds          0

Total of 3 files for this volume
```

Extended File Cache Top Hit rate File Statistics Usage

Shows the use of the Extended File Cache Top Hit rate File Statistics facility on an Alpha system.

❶ Allocated pages	.Number of main memory data pages currently being used by the extended file cache to contain actual file data, for this volume
❸ Read hits	Cumulative number of times a read has been satisfied from the data cache.
❺ Virtual writes	Cumulative number of virtual write I/Os received on this volume by the cache.
❼ Read aheads	Cumulative number of times the extended file cache issued read ahead I/Os on this volume after detecting sequential file I/O virtual reads.
❾ Write throughs	Cumulative number of times a virtual write has been capable of modifying cached data.
❷ Total QIOs	Total number of QIO and PAGE I/O calls received by the extended file cache on this volume.
❹ Virtual reads	Cumulative number of Virtual Read requests received by the extended file cache on this volume.
❻ Hit rate	Percentage of Total QIOs that were completed by copying data from cache, rather than issuing real I/O to a volume.
❽ Read throughs	Number of Virtual Reads that are capable of being satisfied by the extended file cache.
❿ Read arounds	Number of Virtual Reads that cannot be satisfied by the extended file cache. Reasons for this include the presence of a non-XFC compliant node in an OpenVMS cluster, and the receipt of an I/O that is larger than VCC_MAX_IO_SIZE.
⓫ Write arounds	Cumulative number of times a virtual write has been detected that cannot update data in cache memory.

6.
```
$ SHOW MEMORY/CACHE=(FILE=SYS$LOGIN:*.*,OPEN,CLOSED)

System Memory Resources on 3-NOV-2001 18:36:12.79

Extended File Cache Statistics:
```

```
_$7$DKA200:[SYS0.SYSMGR]ACCOUNTNG.DAT;1 (open)
Caching is enabled, active caching mode is Write Through
Allocated pages   ❶    1      Total QIOs      ❷   17
Read hits         ❸    0      Virtual reads   ❹    1
Virtual writes    ❺   16      Hit rate        ❻    0 %
Read aheads       ❼    0      Read throughs   ❽    0
Write throughs    ❾   16      Read arounds    ❿    0
                             Write arounds   ⓫    0

_$7$DKA200:[SYS0.SYSMGR]ACME$SERVER.LOG;129 (open)
Caching is enabled, active caching mode is Write Through
Allocated pages         1      Total QIOs            2
Read hits               0      Virtual reads         1
Virtual writes          1      Hit rate              0 %
Read aheads             0      Read throughs         1
Write throughs          1      Read arounds          0
                               Write arounds         0

_$7$DKA200:[SYS0.SYSMGR]CLUE$STARTUP_BRAHMA.LOG;935 (closed)
Caching is enabled, active caching mode is Write Through
Allocated pages         2      Total QIOs           32
Read hits               0      Virtual reads         0
Virtual writes         32      Hit rate              0 %
Read aheads             0      Read throughs         0
Write throughs         32      Read arounds          0
                               Write arounds         0
        .
        .
        .

_$7$DKA200:[VMS$COMMON.SYSMGR]VMS$AUDIT_SERVER.DAT;1 (open)
Caching is enabled, active caching mode is Write Through
Allocated pages         1      Total QIOs            8
Read hits               3      Virtual reads         8
Virtual writes          0      Hit rate              37 %
Read aheads             1      Read throughs         8
Write throughs          0      Read arounds          0
                               Write arounds         0

Total of 50 files for this volume
```

Extended File Cache Statistics Usage

Shows the use of the Extended File Cache Statistics facility on an Alpha system.

❶ Allocated pages	.Number of main memory data pages currently being used by the extended file cache to contain actual file data, for this volume
❸ Read hits	Cumulative number of times a read has been satisfied from the data cache.
❺ Virtual writes	Cumulative number of virtual write I/Os received on this volume by the cache.
❼ Read aheads	Cumulative number of times the extended file cache issued read ahead I/Os on this volume after detecting sequential file I/O virtual reads.

❾ Write throughs	Cumulative number of times a virtual write has been capable of modifying cached data.
❷ Total QIOs	Total number of QIO and PAGE I/O calls received by the extended file cache on this volume.
❹ Virtual reads	Cumulative number of Virtual Read requests received by the extended file cache on this volume.
❻ Hit rate	Percentage of Total QIOs that were completed by copying data from cache, rather than issuing real I/O to a volume.
❽ Read throughs	Number of Virtual Reads that are capable of being satisfied by the extended file cache.
❿ Read arounds	Number of Virtual Reads that cannot be satisfied by the extended file cache. Reasons for this include the presence of a non-XFC compliant node in an OpenVMS cluster, and the receipt of an I/O that is larger than VCC_MAX_IO_SIZE.
⓫ Write arounds	Cumulative number of times a virtual write has been detected that cannot update data in cache memory.

7. `$ SHOW MEMORY/PHYSICAL_PAGES`

```
                System Memory Resources on  3-NOV-2001 16:11:30.76
Physical Memory Usage (pages): Total ❶  Free ❷   In Use ❸  Modified ❹
Main Memory (32.00Mb)          65536   44233      20955         308

Of the physical pages in use, 10970 pages are permanently allocated
to VMS.
```

Physical Memory Usage

Shows the use of physical memory.

❶ Total	Number of physical memory pages available for general system use. Multiport memory pages used for shared memory global sections, mailboxes, and common event blocks are not included in this number.
❷ Free	Number of pages on the free page list.
❸ In Use	Number of pages currently being used. This number is calculated by adding the number of pages on the free, modified, and bad lists and then subtracting that sum from the total number of available pages.
❹ Modified	Number of pages on the modified page list.

Bad Page List

Note

This display is written only when there are pages on the bad page list.

Shows the contents of the bad page list.

Total	Number of pages on the bad page list.
Dynamic	Number of memory errors detected after the system was booted.
I/O Errors	Number of errors detected during page fault handling.
Static	Number of memory errors detected during boot-time scan.

By default, either single-bit or double-bit errors cause the pages to be removed during the boot-time scan.

Pages Allocated to OpenVMS

Any **SHOW MEMORY** display that includes the physical memory display concludes with the number of pages permanently allocated to the OpenVMS system. These pages include nonpaged executive code and data, the PFN database, nonpaged dynamic memory, the interrupt stack, and the system page table.

8. `$ SHOW MEMORY/SLOTS`

```
                System Memory Resources on 3-NOV-2001 16:11:35.31
Slot Usage (slots):       Total ❶   Free ❷  Resident ❸  Swapped ❹
Process Entry Slots          75       28          46          1
Balance Set Slots            70       26          44          0
```

Slot Usage (slots)

Displays the use of process entry slots and balance set slots.

❶ Total	Number of process entry slots (the value of the system parameter MAXPROCESSCNT) and balance set slots (the value of the system parameter BALSETCNT) permanently allocated when the system was bootstrapped.
❷ Free	Number of slots currently available.
❸ Resident	Number of slots currently used by memory-resident processes. The number of balance set slots in use can never be any larger than the number of process entry slots in use because the SWAPPER and NULL processes have process entry slots but do not require balance set slots.
❹ Swapped	Number of slots used by outswapped processes. For process entry slots, this number includes all processes that have been partially outswapped. For balance set slots, this number includes those processes that have had their process bodies outswapped but have process headers that are still resident.

9. `$ SHOW MEMORY/POOL`

```
                System Memory Resources on  5-DEC-2001 03:59:35.29

Dynamic Memory Usage (bytes):   Total ❶    Free ❷ In Use ❸ Largest ❹
  Nonpaged Dynamic Memory    24993792   23511872   1481920   23363776
  Bus Addressable Memory       131072     114688     16384     114688
  Paged Dynamic Memory        5799936    3530784   2269152    3530176
```

```
  Lock Manager Dynamic Memory 1048576     254976    793600
```

Dynamic Memory Usage (in bytes)

Shows the use of the nonpaged and paged pool areas that allocate variably sized blocks.

❶ Total	Total number of bytes set aside for each area.
❷ Free	Total amount of free space in each dynamic memory area.
❸ In Use	Amount of space currently allocated from each area. This number is the total size minus the number of free bytes.
❹ Largest	Size of the largest contiguous block in each area. For the paged pool, this number represents the largest block that can be allocated. For the nonpaged pool, an allocation request larger than this number will cause the nonpaged pool to grow (if other constraints allow growth). This attribute does not apply to the lock manager, as all items for the lock manager are the same size.

10. `$ SHOW MEMORY/POOL/FULL`

```
                System Memory Resources on  5-DEC-2001 03:59:35.29

Nonpaged Dynamic Memory                        (Lists + Variable)
 Current Size (bytes)     ❶ 24993792 Current Size (pagelets)  ❶  48816
 Initial Size (NPAGEDYN)  ❷ 24993792 Initial Size (pagelets)  ❷  48816
 Maximum Size (NPAGEVIR)  ❸499998720 Maximum Size (pagelets)  ❸ 976560
 Free Space (bytes)       ❹ 22230144 Space in Use (bytes)     ❺2763648
 Largest Variable Block   ❻ 19676672 Smallest Variable Block  ❼   2176
 Number of Free Blocks    ❽    38334 Free Blocks LEQU 64 Bytes❾      0
 Free Blocks on Lookasides❿  38330 Lookaside Space (bytes)     2537472

Bus Addressable Memory                         (Lists + Variable)
 Current Size (bytes)     ❶   131072 Current Size (pagelets)  ❶    256
 Initial Size (NPAGEDYN)  ❷   131072 Initial Size (pagelets)  ❷    256
 Free Space (bytes)       ❹   114688 Space in Use (bytes)     ❺  16384
 Largest Variable Block   ❻   114688 Smallest Variable Block  ❼ 114688
 Number of Free Blocks    ❽        1 Free Blocks LEQU 64 Bytes❾      0
 Free Blocks on Lookasides❿      0  Lookaside Space (bytes)          0

Paged Dynamic Memory                           (Lists + Variable)
 Current Size (MB)        ❶    22.88 Current Size (pagelets)   ❶ 46864
 Free Space (MB)          ❹    21.96 Space in Use (MB)         ❺  0.91
 Largest Variable Block   ❻    21.77 Smallest Variable Block(MB)❼21.77
 Number of Free Blocks    ❽      166 Free Blocks LEQU 64 Bytes  ❾    4
 Free Blocks on Lookasides❿     165 Lookaside Space (KB)        201.85
 Over-limit Lookaside Blocks⓫     3 Over-limit Space (KB)         7.45

Lock Manager Dynamic Memory
 Current Size (Mbytes)    ❶    13.85 Current Size (pages)     ❶    1774
 Free Space (Mbytes)      ❹    10.06 Hits                     ⓬   54050
 Space In Use             ❺     3.79 Misses                   ⓭    1664
 Number of Empty Pages    ⓮      26 Expansions                ⓯    1792
 Number of Free Packets   ⓰   41244 Packet Size               ⓱     256
```

Dynamic Memory

Shows the use of dynamic memory areas consisting of variably sized blocks.

❶ Current Size	Current size (in 512-byte pagelets on Alpha) of each dynamic memory area. For the lock manager display, the unit of measurement is megabytes.
❷ Initial Size	Initial size of nonpaged dynamic memory and the name of the system parameter (NPAGEDYN) that controls this size.
❸ Maximum Size	Maximum size to which nonpaged dynamic memory can grow and the name of the system parameter (NPAGEVIR) that controls this size.
❹ Free Space	Amount of free space in each dynamic memory area.
❺ Space in Use	Amount of space currently allocated from each area.
❻ Largest Block	Size of the largest contiguous area in each pool area.
❼ Smallest Block	Size of the smallest hole (free block) in each pool area.
❽ Free Blocks	Total number of free blocks in each pool area. The size of this number is a measure of pool fragmentation.
❾ Blocks LEQU 64	Number of free blocks that are 64 bytes long or shorter. This number is another measure of pool fragmentation because while allocation of these small blocks is unlikely, they contribute to the allocation time whenever an allocation request is made.
❿ Empty Pages	Number of pages that do not have any packets allocated.
⓰ Free Packets	Total number of packets not in use. This number is the sum of free packets from partially used pages and the packets of all Empty Pages.
⓬ Hits	Total number of packet allocations.
⓭ Misses	Total number of packet allocation requests that could not be satisfied because the number of packets requested were not available. A miss causes dynamic memory to expand by one page to satisfy the packet allocation request.
⓯ Expansions	Total number of times that dynamic memory increased by a page to satisfy packet allocation requests.
⓱ Packet Size	Packet size of 256 bytes, which is used by the lock manager to hold either a lock block data structure or a resource block data structure.
❿ Lookasides Free Blocks	Total number of memory packets placed on "lookaside lists". If a packet of the correct size is on a lookaside list, it can be allocated very quickly and efficiently, however, it cannot then be used as part of any other size packet. An excess of packets on the lookaside list can be a cause for the inability to allocate other sizes.
⓫ Over-limit Lookaside Blocks	Total number of memory packets that exist on "lookaside lists" that are no longer actively used because the SYSGEN parameter PAGED_LAL_SIZE was lowered. These packets will be available for use again if the SYSGEN parameter is increased to include them, or if there was a paged pool shortage and they are reclaimed. If there

	are no lookaside list packets on these "over-limit" lists then this line will not appear.

11. `$ SHOW MEMORY/FILES`

```
        System Memory Resources on 22-MAY-2001 19:04:19.67
Swap File Usage (8KB pages):              Index ❶     Free     Size
DISK$ALPHASYS:[SYS48.SYSEXE]SWAPFILE.SYS      1         904      904
DISK$SWAP:[SYS48.SYSEXE]SWAPFILE.SYS;1        2        1048     1048

  Total size of all swap files:                                 1952

Paging File Usage (8KB pages):            Index ❷     Free     Size
DISK$PAGE:[SYS48.SYSEXE]PAGEFILE.SYS;1      253       16888    16888
DISK$ALPHASYS:[SYS48.SYSEXE]PAGEFILE.SYS    254       16888    16888

  Total size of all paging files:                              33776
  Total committed paging file usage:  ❸                         1964
```

Paging File Usage (in blocks)

Shows the usage of paging and swapping files.

❶ Swap File Usage	Number of swap files. Begins with an index value of 1 and increases in count.
❷ Paging File Usage	Number of page files. Begins with an index value of 254 and decreases in count.
❸ Total Paging File Usage	Total committed page file usage. As in previous releases, more pages can reside in page-file sections systemwide than would fit into installed page files.

SHOW NETWORK

SHOW NETWORK — Displays information about network services registered on a node.

Format

SHOW NETWORK [*network-service*]

Parameter

network-service

Specifies a network service for which you want to display information.

If you do not specify a service, the **SHOW NETWORK** command displays information about all services that are currently registered on the local node.

Description

The **SHOW NETWORK** command displays information about network services available on an OpenVMS system.

Qualifiers

/FULL

Displays all the information for the specified network service.

/OLD

If you are running DECnet Phase IV, retains the behavior of the **SHOW NETWORK** command prior to OpenVMS Version 7.0. If the local node is a routing node, the **/OLD** qualifier provides routing information.

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the command is sent. By default, the output of the **SHOW NETWORK** command is sent to the current SYS$OUTPUT device (usually your terminal).

The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

Examples

1.
```
$ SHOW NETWORK

Product:  DECNET Node:  VADIR             Address(es):  24.365
Product:  TCP/IP Node:  vadir.lkr.dec.com Address(es):  161.114.99.17
```

   This command was executed on a system running DECnet Phase V and TCP/IP. It indicates that the `DECNET` and "`TCP/IP`" network-service parameters are valid for this system.

2.
```
$ SHOW NETWORK "TCP/IP"

Product: TCP/IP  Node: vadir.lkr.dec.com  Address(es):  161.114.99.17
```

   When the `TCP/IP` network-service parameter is used, it must be enclosed in enclosed in quotes because it contains a slash.

3.
```
$ SHOW NETWORK/FULL DECNET
 Product:  DECNET                Manufacturer:  Hewlett-Packard
 Enterprise Company
Node:  CRAP1                    Address(es):  17.51
Network Type:  DNA IV           Interface(s):  net 0


Node Volatile Characteristics as of  8-OCT-2009 09:15:12

Executor node = 17.51 (CRAP1)

Identification           = HPE DECnet for OpenVMS I64
Management version       = V4.0.0
Incoming timer           = 45
Outgoing timer           = 60
Incoming Proxy           = Enabled
Outgoing Proxy           = Enabled
NSP version              = V4.1.0
```

```
Maximum links            = 32
Delay factor             = 80
Delay weight             = 5
Inactivity timer         = 60
Retransmit factor        = 10
Routing version          = V2.0.0
Type                     = nonrouting IV
Routing timer            = 600
Broadcast routing timer  = 180
Maximum address          = 1023
Maximum circuits         = 16
Maximum cost             = 1022
Maximum hops             = 30
Maximum visits           = 63
Maximum area             = 63
Max broadcast nonrouters = 64
Max broadcast routers    = 32
Maximum path splits      = 1
Area maximum cost        = 1022
Area maximum hops        = 30
Maximum buffers          = 100
Buffer size              = 576
Default access           = incoming and outgoing
Pipeline quota           = 4032
Alias maximum links      = 32
Path split policy        = Normal
Maximum Declared Objects = 31



Known Link Volatile Summary as of  8-OCT-2009 09:15:12

No information in database



Node Counters as of  8-OCT-2009 09:15:12

Executor node = 17.51 (CRAP1)

         856  Seconds since last zeroed
         471  Bytes received
         468  Bytes sent
          27  Messages received
          28  Messages sent
           1  Connects received
           1  Connects sent
           0  Response timeouts
           0  Received connect resource errors
           2  Maximum logical links active
           0  Aged packet loss
           0  Node unreachable packet loss
           0  Node out-of-range packet loss
           0  Oversized packet loss
           0  Packet format error
           0  Partial routing update loss
           0  Verification reject
```

The DECNET network-service parameter is used to signify either DECnet Phase IV or DECnet Phase V. Use the **/FULL** qualifier to determine the DECnet phase on the node. This is an example from a system running DECnet Phase IV.

4.
```
$ SHOW NETWORK/FULL DECNET

Product:  DECNET      Manufacturer:  HPE
Node:  VADIR          Address(es):  24.365
Network Type:  DNA V  Interface(s):  net 0

Node 0
at 2002-01-16-12:50:30.528-05:00Iinf

Identifiers

  Name                          = DEC:.XPR.VADIR
  Address                       =
    {
(
    [ DNA_CMIP-MICE ] ,
    [ DNA_SessionControlV3 , number = 19 ] ,
    [ DNA_OSItransportV1 , 'DEC0'H ] ,
    [ DNA_OSInetwork , 49::00-18:AA-00-04-00-6D-61:21 ]
) ,
(
    [ DNA_CMIP-MICE ] ,
    [ DNA_SessionControlV3 , number = 19 ] ,
    [ DNA_OSItransportV1 , 'DEC0'H ] ,
    [ DNA_OSInetwork , 49::00-18:AA-00-04-00-12-60:21
 (DEC:.XPR.BULEAA) ]
) ,
(
    [ DNA_CMIP-MICE ] ,
    [ DNA_SessionControlV2 , number = 19 ] ,
    [ DNA_OSItransportV1 , 'DEC0'H ] ,
    [ DNA_IP , 161.114.99.17 ]
) ,
(
    [ DNA_CMIP-MICE ] ,
    [ DNA_SessionControlV3 , number = 19 ] ,
    [ DNA_NSP ] ,
    [ DNA_OSInetwork , 49::00-18:AA-00-04-00-6D-61:20 ]
) ,
(
    [ DNA_CMIP-MICE ] ,
    [ DNA_SessionControlV3 , number = 19 ] ,
    [ DNA_NSP ] ,
    [ DNA_OSInetwork , 49::00-18:AA-00-04-00-12-60:20
 (DEC:.XPR.BULEAA) ]
)
    }

Status

 UID                        = 562B4000-43CF-11CF-8001-AA0004007E10
 State                      = On
 Functions Enabled          =
```

```
      {
      Address Watcher ,
      CMIP Listener
      }
   ID                          = 00-00-F8-01-FD-62

  Characteristics

   Version                     = T5.0.3
   Implementation              =
      {
      [
      Name = OpenVMS AXP ,
      Version = "V7.2-1  "
      ] ,
      [
      Name = HPE DECnet-Plus for OpenVMS ,
      Version = "V7.2-1 ECO03 31-MAY-2001 16:54:53.36"
      ]
      }
    Script Location             = <Default value>
    Maximum Listeners           = 0
    Listener Template           = <Default value>
    Secondary Names             =
      {
      }
```

This example is from a system running DECnet Phase V.

5.
```
$ SHOW NETWORK/OLD

OpenVMS Network status for local node 12.378 EMARET on 16-JAN-2002
 14:46:35.64

This is a nonrouting node, and does not have any network information.
The designated router for EMARET is node 12.1023 A12LKR.
```

The **/OLD** qualifier can be used only on a system running DECnet Phase IV.

SHOW PRINTER

SHOW PRINTER — Displays the current settings for a printer.

Format

SHOW PRINTER *device-name*[:]

Parameter

***device-name*[:]**

Specifies the name of the printer for which settings are to be displayed.

Description

The **SHOW PRINTER** command displays the settings that are currently defined for the specified printer. Settings include the following:

Printer type	Form feed
Page width	Case
Page length	Character translation
Line termination type	Wrapping

Printer settings are established by the **SET PRINTER** command. You must have OPER (operator) privilege to use the **SET PRINTER** command.

Qualifiers

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the command is sent. By default, the output of the **SHOW PRINTER** command is sent to the current SYS$OUTPUT device (usually your terminal). To send the output to a file, use the **/OUTPUT** qualifier followed by a file specification.

The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification. If you enter a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS is the default file type.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

/IDSTRING

Returns the USB printer ID string as a text string and displays it on your screen. This information appears at the end of the typical **SHOW PRINTER** command display.

/SERIALNUM

Returns the USB printer serial number as a text string and displays to the screen. The USB printer returns the serial number in Unicode format, and this is converted to ASCII text before being displayed.

Examples

1.
```
$ SHOW PRINTER LPA0:
 Printer LPA0:, device type LP11, is online, allocated, spooled
 Error count                  0  Operations completed             880
 Owner process  "SYMBIONT_0001"  Owner UIC                      [0,0]
 Owner process ID      21C0008D  Dev Prot  S:RWLP,O:RWLP,G:RWLP,W:RWLP
 Reference count              2  Default buffer size               132
 Page width                 132  Page Length                       66
 No Carriage_return     Formfeed        Lowercase
 No Passall             No Wrap         Printall
 No Fallback            No Tab          Truncate
 No Sixels              No Bitmapped           No Polled
 Unidirectional
 Intermediate device: STAR$DUA1:
 Associated queue: LN01$PRINT
```

The **SHOW PRINTER** command in this example displays the settings for the printer LPA0.

2.
```
$ SHOW PRINTER LPA0/IDSTRING
 Printer LPA0:, device type PC printer, is online, record-oriented
 device, carriage control.
 Error count               0    Operations completed             0
 Owner process            ""    Owner UIC                  [SYSTEM]
 Owner process ID    00000000    Dev Prot          S:RWPL,O:RWPL,G,W
 Reference count           0    Default buffer size            132
 Page width              132    Page Length                     66
 No Carriage_return  Formfeed       Uppercase
 No Passall          No Wrap        No Printall
 No Fallback         No Tab         Truncate
 No Sixels           No Bitmapped           Polled
 Unidirectional
 IDString:
     MFG:            HEWLETT-PACKARD;
     MDL:            DESKJET 895C;
     CMD:            MLC,PCL,PML;
     CLASS:          PRINTER;
     DESCRIPTION:    Hewlett-Packard DeskJet 895C;
     SERN:           MX88P2V0VWGO;
     VSTATUS:               $HB0$FC0,ff,DN,IDLE,CUT;
```

The **SHOW PRINTER/IDSTRING** command in this example displays the ID string for the printer LPA0.

3.
```
$ SHOW PRINT LPA0/IDSTRING
 Printer LRA0:, device type LP11, is online, record-oriented device,
 carriage control.
 Error count               0    Operations completed             0
 Owner process            ""    Owner UIC                  [SYSTEM]
 Owner process ID    00000000    Dev Prot          S:RWPL,O:RWPL,G,W
 Reference count           0    Default buffer size            132
 Page width              132    Page Length                     66
 No Carriage_return  Formfeed       Uppercase
 No Passall          No Wrap        No Printall
 No Fallback         No Tab         Truncate
 No Sixels           No Bitmapped           No Polled
 Unidirectional
 %SYSTEM-F-DEVOFFLINE, device is not in configuration or not available
```

If the device has been disconnected and marked offline, the **SHOW PRINTER/IDSTRING** command displays this information.

If the attempt to read the ID string should fail (for an unknown reason other than device disconnection), the reference to the ID string appears as:

```
%SHOW-W-NOIDSTRING, ID String could not be retrieved
```

4.
```
$ SHOW PRINT LPA0/SERIALNUM
 Printer LPA0:, device type PC printer, is online, record-oriented
 device, carriage control.
 Error count               0    Operations completed             0
 Owner process            ""    Owner UIC                  [SYSTEM]
 Owner process ID    00000000    Dev Prot          S:RWPL,O:RWPL,G,W
 Reference count           0    Default buffer size            132
```

```
Page width               132    Page Length                      66
No Carriage_return  Formfeed        Uppercase
No Passall          No Wrap         No Printall
No Fallback         No Tab          Truncate
No Sixels           No Bitmapped            Polled
Unidirectional
Serial Number:      MX88P2V0VWGO
```

The **SHOW PRINTER/SERIALNUM** command in this example displays the serial number for the printer LPA0.

If the device has been disconnected and marked offline, the **SHOW PRINTER/SERIALNUM** command replaces the reference to the serial number with the following message:

```
%SYSTEM-F-DEVOFFLINE, device is not in configuration or not available
```

If the ID string is not supported by the USB printer in question, the reference to the ID string will appear as:

```
%SHOW-W-NOSERIALNUM, Serial number could not be retrieved
```

SHOW PROCESS

SHOW PROCESS — Displays information about a process and its subprocesses. **Requires GROUP privilege to show other processes in the same group. Requires WORLD privilege to show processes outside your group.**

Format

SHOW PROCESS [[*node-name*::]*process-name*]

Parameters

***node-name*::**

The name of the node on which the specified process is running.

You must specify a node name on the same OpenVMS Cluster on which the current process is running.

process-name

The name of the process about which information is to be displayed. The process name can have up to 15 alphanumeric characters.

The specified process name must be in the same group as the current process.

Description

The **SHOW PROCESS** command displays information about a process and its subprocesses. If no qualifier is entered, only the following subset of information is displayed:

- Time
- Process terminal
- User name and user identification code (UIC)

- Node name
- Process name and process identification (PID)
- Priority
- Default directory
- Allocated devices
- Number of kernel threads

A local process name can look like a remote process name; therefore, if you specify ATHENS::SMITH, the system checks for a process named ATHENS::SMITH on the local node before checking node ATHENS for a process named SMITH.

You also can use the **/IDENTIFICATION=*pid*** qualifier to specify a process name. If you use the **/IDENTIFICATION** qualifier and the *process-name* parameter together, the qualifier overrides the parameter. If you do not specify either the *process-name* parameter or the **/IDENTIFICATION** qualifier, information is displayed for the current process.

If you have GROUP or WORLD privilege, you can display information about processes other than your own. With GROUP privilege you can look at other processes in your group. With WORLD privilege you can look at processes outside of your group. You must use the **/IDENTIFICATION** qualifier to look at processes outside of your group.

Qualifiers

/ACCOUNTING

Shows the resources used by the process.

/ALL

Displays the basic subset of information as well as information about:

- Accounting
- Dynamic memory use
- Privileges
- Quotas
- Shelving
- Subprocesses
- Wildcard status

/CASE_LOOKUP

Lists the active case sensitivity setting in your process (either BLIND or SENSITIVE).

VSI strongly recommends that you use caution when enabling case sensitivity in your processes.

See the relevant section in the *Guide to OpenVMS File Applications* [https://docs.vmssoftware.com/guide-to-openvms-file-applications/#UPPER_LOWER_LETTERS] for additional information.

/CONTINUOUS

Displays continuously updated information about the local process in an OpenVMS environment. You cannot use the **/CONTINUOUS** qualifier to display information about a process on another node in a cluster environment.

While the continuous display is running, you can press the following keys to customize the output display:

Key	Action
E	Exits the continuous display.
Q	Dynamically displays process quotas.
T	Displays information for threads in the process. Press the **T** key repeatedly to cycle through multiple threads.
V	Displays a map of the pages in the virtual address space of the process. Each character displayed in the map represents the type of page. If the current program counter is in the page, the page type is indicated by an at sign (@). Pages locked in the working set are indicated by the letter L. Global pages are indicated by the letter G. Other valid pages in the working set are indicated by an asterisk (*).
Space	Returns to the default display.

The **/CONTINUOUS** qualifier cannot be used with the **/OUTPUT** qualifier.

/DUMP

Displays the image dump setting which is set or cleared by the **SET PROCESS/DUMP** command.

/EXACT

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify a search string that must match the search string exactly and must be enclosed with quotation marks (" ").

If you specify the **/EXACT** qualifier without the **/SEARCH** qualifier, exact search mode is enabled when you set the search string with the Find (**E1**) key.

/HIGHLIGHT[=*keyword*]

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify the type of highlighting you want when a search string is found. When a string is found, the entire line is highlighted. You can use the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting.

/IDENTIFICATION=*pid*

Requires GROUP or WORLD privilege to access processes other than your own.

Displays information about the process with the specified process identification (PID). The PID is assigned by the system when the process is created. When you specify a PID, you can omit the leading zeros.

If you specify the **/IDENTIFICATION** qualifier, it overrides the *process-name* parameter. If, in addition, you specify the **/MEMORY** qualifier, the PID value must be that of the current process.

/INTERVAL=*n*

Use with the **/CONTINUOUS** qualifier to update the information at the specified number of seconds.

/MEMORY

Displays the process's use of dynamic memory areas. The **/MEMORY** qualifier is allowed only for the current process.

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the command is sent. By default, the output of the **SHOW PROCESS** command is sent to the current SYS$OUTPUT device (usually your terminal). To send the output to a file, use the **/OUTPUT** qualifier followed by a file specification.

The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification. If you enter a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS is the default file type.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

The **/OUTPUT** qualifier is incompatible with the **/CONTINUOUS** qualifier.

/PAGE[=*keyword*]
/NOPAGE (default)

Controls the display of process information on the screen.

You can use the following keywords with the **/PAGE** qualifier:

Keyword	Description
CLEAR_SCREEN	Clears the screen before each page is displayed.
SCROLL	Displays information one line at a time.
SAVE[=*n*]	Enables screen navigation of information, where *n* is the number of pages to store.

The **/PAGE=SAVE** qualifier allows you to navigate through screens of information. The **/PAGE=SAVE** qualifier stores up to 5 screens of up to 255 columns of information. When you use the **/PAGE=SAVE** qualifier, you can use the following keys to navigate through the information:

Key Sequence	Description
Up arrow key, **Ctrl/B**	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.
Right arrow key	Scroll right one column.
Find (**E1**)	Specify a string to find when the information is displayed.
Insert Here (**E2**)	Scroll right one half screen.
Remove (**E3**)	Scroll left one half screen.
Select (**E4**)	Toggle 80/132 column mode.
Prev Screen (**E5**)	Get the previous page of information.
Next Screen (**E6**), **Return**, **Enter**, **Space**	Get the next page of information.
F10, **Ctrl/Z**	Exit. (Some utilities define these differently.)
Help (**F15**)	Display utility help text.

Key Sequence	Description
Do (**F16**)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.

The **/PAGE** qualifier is not compatible with the **/OUTPUT** qualifier.

/PRIVILEGES

Displays current privileges and rights for the process. To display only privileges, also specify the **/NORIGHTS** qualifier. Use the **SET PROCESS/PRIVILEGES** command to enable process privileges on a system.

/QUOTAS

Displays, for each resource, either a quota or a limit. The values displayed for quotas reflect any quota reductions resulting from subprocess creation. The values displayed for limits reflect the resources available to a process at creation.

/RAD

Displays home resource affinity domain (RAD).

RAD is supported on AlphaServer GS series systems and NUMA-capable IA-64 systems.

/RIGHTS (default)
/NORIGHTS

Displays the current rights for the specified process. Use the **/NORIGHTS** qualifier with the **/PRIVILEGES** qualifier to suppress the rights display.

/SCHEDULING_CLASS

Displays the scheduling class of the specified process.

/SEARCH="*string*"

Use with the **/PAGE=SAVE** qualifier to specify a string that you want to find in the information being displayed. Quotation marks are required for the **/SEARCH** qualifier, if you include spaces in the text string.

You can also dynamically change the search string by pressing the Find key (**E1**) while the information is being displayed. Quotation marks are not required for a dynamic search.

/SHELVING

Displays whether the process automatically unshelves files.

/SUBPROCESSES

Displays the current subprocesses in hierarchical order.

/SYMLINK

The **SHOW PROCESS/ALL** and **SHOW PROCESS/SYMLINK** commands display the status of the symlink wildcard search mode.

/TOKEN (Alpha/IA-64 only)

Displays the current token size as either Traditional (255 bytes) or Extended (4000 bytes). (A token is any element in a command line that is bounded by spaces. For example, the command **COPY X.TXT Y.TXT** contains three tokens.) The token size is determined by the setting of bit 1 in the DCL_CTLFLAGS system parameter. You can use the **SET PROCESS/TOKEN** command to change the token size.

/UNITS

Displays whether the process currently reports disk space in bytes or blocks. You can change this setting using **SET PROCESS/UNITS**.

/WRAP
/NOWRAP (default)

Use with the **/PAGE=SAVE** qualifier to limit the number of columns to the width of the screen and to 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 and can be seen when you use the scrolling (left and right) features provided by the **/PAGE=SAVE** qualifier.

Examples

1.
```
$ SHOW PROCESS OCALA::MALIK

 14-DEC-2001 15:35:19.39  User: MALIK   Process ID: 28200364
                          Node: OCALA   Process name: MALIK
Terminal:        RTA5:
User identifier: [VMS,MALIK]
Base  priority:   4
Default file spec: WORK5:[MALIK]
Devices allocated: RTA5:
```

   The **SHOW PROCESS** command in this example is entered by the user MALIK. The system displays the subset of information for the owned process on node OCALA. The information includes the following:

   Date and time the **SHOW PROCESS** command is entered
   User name
   Process identification (PID) number
   Node name
   Process name
   Device name of the current SYS$INPUT device
   User identification code (UIC)
   Base execution priority
   Default device (only for processes on the same node)
   Default directory (only for current processes)
   Devices allocated to the process and volumes mounted, if any

2.
```
$ SHOW PROCESS/ACCOUNTING
 14-DEC-2001 14:48:01.31   User: MALIK  Process ID:   28200364
                           Node: OCALA  Process name: "MALIK"

 Accounting information:
  Buffered I/O count:      4878  Peak working set size:      844
```

```
 Direct I/O count:         1284  Peak virtual size:          1176
 Page faults:              6100  Mounted volumes:               0
 Images activated:           22
 Elapsed CPU time:       0 00:01:20.51
 Connect time:           0 04:06:03.75
```

The **SHOW PROCESS** command in this example displays the accounting statistics for the process. The values under "`Peak working set size`" and "`Peak virtual size`" are rendered and in pagelets on Alpha.

3. `$ SHOW PROCESS/PRIVILEGES`

```
 14-DEC-2001 14:59:28.53  User: MALIK   Process ID:    28200364
                          Node: OCALA   Process name: "MALIK"

 Process privileges:
  GROUP                may affect other processes in same group
  TMPMBX               may create temporary mailbox
  NETMBX               may create network device

 Process rights identifiers:
  INTERACTIVE
  DIALUP
```

The **SHOW PROCESS** command in this example displays the current privileges for the process.

4. `$ SHOW PROCESS/QUOTAS`

```
 14-DEC-2001 15:00:28.79   User: MALIK  Process ID:    28200364
                           Node: OCALA  Process name:  "MALIK"

 Process Quotas:
 Account name: VMS
  CPU limit:                       Infinite  Direct I/O limit:       6
  Buffered I/O byte count quota:      17904  Buffered I/O limit:     6
  Timer queue entry quota:               10  Open file quota:       31
  Paging file quota:                  24945  Subprocess quota:       8
  Default page fault cluster:            64  AST limit:             14
  Enqueue quota:                         30  Shared file limit:      9
  Max detached processes:                11  Max active jobs:       14
```

The **SHOW PROCESS** command in this example displays the available quotas and limits. The value under "`Paging file quota`" is rendered in pagelets on Alpha.

5. `$ SHOW PROCESS/SUBPROCESSES`

```
 14-DEC-2001 15:44:59.39  User: MALIK  Process ID:   28200364
                          Node: OCALA  Process name: "MALIK_1"

 Processes in this tree:

 MALIK
   MALIK_1 (*)
     MALIK_2
```

The **SHOW PROCESS** command in this example shows the current process tree. The current process is indicated by the asterisk `(*)`. Processes both below and above the current process are shown.

6.
```
$ SHOW PROCESS/CONTINUOUS BOB
                            Process BOB                         12:26:53

    State               CUR             Working set              269
    Cur/base priority   8/4             Virtual memory          1713
    Current PC          7FFEE07E        CPU time          00:00:13.82
    Current PSL         03C00000        Direct I/O               246
    Current user SP     7FF785A4        Buffered I/O             646
    PID                 226006C0        Page faults             3417
    UIC                 [VMS,BOB]       Event flags          C8000007
                                                             C0000000
    SYS$SYSROOT:[SYSEXE]SHOW.EXE
```

   In this example, the **/CONTINUOUS** qualifier causes the display of information about process BOB to be updated continuously. Note that the Virtual pages heading appears on Alpha indicating the virtual memory used as a number of CPU-specific pages. The value displayed for "Working set" is also a number of CPU-specific pages.

7.
```
$ SHOW PROCESS/CONTINUOUS CIMSERVER
                            Process CIMSERVER                   07:38:02

    State               LEF                 Working set              2599
    Cur/base priority   9/4                 Virtual pages           15135
    Current PC          8065BDC0            CPU time         0 00:00:14.33
    Current PSL         00000000            Direct I/O               5889
    Current user SP     7ACE6680            Buffered I/O            88641
    PID                 20200822            Page faults              1487
    UIC                 [SYSTEM]            Event flags           C0000001
                                                              80000000

    $1$DGA02:[SYS0.SYSCOMMON.][WBEM_SERVICES]CIMSERVER.EXE;1


[Press Q key]

                            Process CIMSERVER                   07:38:18

    State               LEF                 Working set              2599
    PID                 20200822            Page faults              1487
    UIC                 [SYSTEM]            Event flags           C0000001
                                                              80000000

    # open files remaining              125/128                 ( 97%)
    Direct I/O count/limit              100/100                 (100%)
    Buffered I/O count/limit             99/100                 ( 99%)
```

```
   BUFIO byte count/limit            251840/251840              (100%)
   ASTs remaining                       98/100                  ( 98%)
   Timer entries remaining              15/16                   ( 93%)
   PGFL quota count/limit             42408/43750               ( 96%)
   ENQ quota count/limit               2048/2048                (100%)

   $1$DGA02:[SYS0.SYSCOMMON.][WBEM_SERVICES]CIMSERVER.EXE;1
```

In this example, the standard continuous display is interrupted when the **Q** key is pressed. The display then dynamically changes to show the quotas in two columns: one column with the actual counts and another with percentages.

8. `$ SHOW PROCESS/MEMORY`

```
 14-DEC-2001 14:59:04.48   User: MALIK  Process ID:   28200364
                           Node: OCALA  Process Name: "MALIK"

 Process Dynamic Memory Area
     Current Size (bytes)    25600   Current Total Size (pages)    50
     Free Space (bytes)      22698   Space in Use (bytes)        2902
     Size of Largest Block   22496   Size of Smallest Block        15
     Number of Free Blocks       7   Free Blocks LEQU 32 Bytes      3
```

The **SHOW PROCESS** command in this example displays the use of dynamic memory areas for the current process, MALIK. These areas are described as follows:

Current size	Displays the current size in bytes and pagelets on Alpha of each dynamic memory area.
Free space	Displays the amount of free space in each dynamic memory area.
Space in use	Displays the amount of space currently allocated from each area.
Largest block	Displays the size of the largest contiguous area in each pool area.
Smallest block	Displays the size of the smallest free block in each pool area.
Free blocks	Displays the total number of free blocks in each pool area. The size of this number is a measure of pool fragmentation.
Blocks LEQU 32 Blocks LEQU 64 (Alpha)	Displays the number of free blocks that are 32 bytes or 64 bytes or shorter on Alpha. This number is another measure of pool fragmentation, because while allocation of these small blocks is unlikely, they contribute to the allocation time whenever an allocation request is made.

9. `$ SHOW PROCESS REDSOX::RAMIREZ`

```
  27-OCT-2004 23:40:20.04  User: RAMIREZ   Process ID: 19182004
                           Node: REDSOX    Process name: RAMIREZ

Terminal:          RTA5:
User identifier:   [VMS,RAMIREZ]
Base  priority:    4
Default file spec: WORK5:[RAMIREZ]
Devices allocated: RTA5:
```

The **SHOW PROCESS** command in this example displays the basic subset of information for the process RAMIREZ on node REDSOX.

10. `$ SHOW PROCESS/SHELVING LYONS`

```
15-JUNE-2001  10:07:25.11   User: LYONS       Process ID:  20206A50
                            Node: BANANA      Process name: "LYONS"

Auto-unshelve: on
```

The **SHOW PROCESS** command in this example shows that the LYONS process automatically unshelves files.

11. `$ SHOW PROCESS/DUMP RALSTON`

```
31-DEC-2001 10:38:32.48   User: RALSTON       Process ID:   20C0011B
                          Node: AAAAAA        Process name: "_RTA1:"

Image Dump: on
```

The **SHOW PROCESS** command in this example shows that the image dump capability for the RALSTON process is enabled. This causes the contents of the address space to be written to the file named SYS$LOGIN:(*image-name*).DMP (where the image name is the same as the file name) when an image terminates due to an unhandled error.

See the **SET PROCESS/DUMP** command for additional information.

12. `$ SHOW/SYMLINK`

```
$ 17-SEP-2009 05:31:34.92   User: SYSTEM           Process ID:
 20A0045C
                          Node: LENOX1          Process name: "SYSTEM"

Symlink search mode: No wildcard
```

This example shows that wildcard is not used for symlink search.

SHOW PROTECTION

SHOW PROTECTION — Displays the current default process protection to be applied to all new files created during the terminal session or batch job.

Format

SHOW PROTECTION

Parameters

None.

Description

The **SHOW PROTECTION** command displays the current default file protection to be applied to all new files created during the terminal session or batch job. You can change the default protection at any time with the **SET PROTECTION/DEFAULT** command.

Use the **DIRECTORY/PROTECTION** command to see the current protection for one or more specific files.

Example

```
$ SHOW PROTECTION
SYSTEM=RWED, OWNER=RWED, GROUP=RE, WORLD=NO ACCESS
$ SET PROTECTION=(GROUP:RWED,WORLD:RE)/DEFAULT
$ SHOW PROTECTION
SYSTEM=RWED, OWNER=RWED, GROUP=RWED, WORLD=RE
```

The **SHOW PROTECTION** command in this example requests a display of the current protection defaults and the user identifiers; the **SET PROTECTION/DEFAULT** command changes the file access allowed to other users in the same group and to miscellaneous system users. The next **SHOW PROTECTION** command shows the modified protection defaults.

SHOW QUEUE

SHOW QUEUE — Displays information about queues and the jobs that are currently in queues. **Requires read (R) access to the job.**

Format

SHOW QUEUE [*queue-name*]

Parameter

queue-name

Specifies the name of the queue for which you want information displayed. The asterisk (*) and the percent sign (%) wildcard characters are allowed. The default value for the *queue-name* parameter is the asterisk (*) wildcard character. If no queue name is specified, information on all queues is displayed.

Description

The **SHOW QUEUE** command displays the name, type, status, and attributes of batch and output queues as well as information about jobs in the queues. When you enter the **SHOW QUEUE** command with no qualifiers, the OpenVMS system lists the names, types, and status of all the specified queues along with information about your jobs in those queues. The **SHOW QUEUE** command assumes the **/BRIEF** qualifier by default.

In an OpenVMS Cluster, the **SHOW QUEUE** command displays information on all the queues available for the cluster.

If you want information about queues or a complete list of the jobs in those queues, use the **SHOW QUEUE** command. For information about specific jobs, use the **SHOW ENTRY** command.

Qualifiers

/ALL_JOBS

Displays all the jobs in the specified queues. You must have read (R) access to the job to show information about any jobs.

/BATCH

Displays only batch queues and jobs in those queues.

/BRIEF (default)

Displays a one-line description of each queue and jobs that are in it. This information includes the name, type, and status of each queue. It also includes the entry number, the job name, the user name, and the status for each job shown. The **/FULL** and **/FILES** qualifiers override the **/BRIEF** qualifier.

/BY_JOB_STATUS=(*keyword-list*)

Displays queues that contain jobs that match the specified status. By default, jobs owned by the user are displayed. Specify the status with one or more of the following keywords:

Keyword	**Description**	
EXECUTING	Requests the display of jobs in any of the following states:	
	ABORTING EXECUTING PRINTING PROCESSING	STALLED STARTING SUSPENDED
	ABORTING means that the job is halting prior to normal completion and will not continue processing. A SUSPENDED state means that the job stopped during processing, but should continue when the cause of the SUSPENDED state is resolved. PROCESSING is the executing state for server queues. PRINTING is the executing state for printer and terminal queues. STARTING means the job is beginning to be processed or printed. A STALLED or SUSPENDED state means that the job stopped during processing, but should continue when the cause of the STALLED or SUSPENDED state is resolved.	
HOLDING	Requests the display of jobs on hold. Holding status indicates that the job is being held in the queue indefinitely.	
PENDING	Requests the display of jobs with pending status. Pending status indicates that the job is waiting its turn to execute.	
RETAINED	Requests the display of jobs retained in the queue after execution. Retained status indicates that the job has completed, but it remains in the queue.	
TIMED_RELEASE	Requests the display of jobs on hold until a specified time. Timed release status indicates that the job is being held in the queue for execution at a specified time.	

Note that if you specify the qualifier without a keyword, the system will only display queues that actually contain jobs.

/DEVICE[=(*keyword-list*)]

Displays a particular type of queue and jobs in that queue.

Specify the type of device queue with one or more of the following keywords:

Keyword	**Description**
PRINTER	Requests the display of print queues.

Keyword	Description
SERVER	Requests the display of server queues.
TERMINAL	Requests the display of terminal queues.

You can specify more than one keyword. If you do not specify a keyword with the /DEVICE qualifier, all printer, terminal, and server queues are displayed.

/EXACT

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify a search string that must match the search string exactly and must be enclosed with quotation marks (" ").

If you specify the **/EXACT** qualifier without the **/SEARCH** qualifier, exact search mode is enabled when you set the search string with the Find (**E1**) key.

/FILES

Adds to the display the list of files associated with each job. The **/FILES** qualifier overrides the **/BRIEF** qualifier.

/FULL

Displays complete information about queues, jobs contained in queues, and the files associated with the jobs. The **/FULL** qualifier overrides the **/BRIEF** qualifier.

/GENERIC

Displays only generic queues and jobs in those queues. A generic queue is not an execution queue. Its function is to hold jobs of a particular type (line printer jobs, for example) and direct them to execution queues for processing.

/HIGHLIGHT[=*keyword*]

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify the type of highlighting you want when a search string is found. When a string is found, the entire line is highlighted. You can use the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting.

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the command is sent. By default, the output of the **SHOW QUEUE** command is sent to the current SYS$OUTPUT device (usually your terminal). To send the output to a file, use the **/OUTPUT** qualifier followed by a file specification.

The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification. If you enter a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS is the default file type.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

/PAGE[=*keyword*]
/NOPAGE (default)

Controls the display of information on the screen.

You can use the following keywords with the **/PAGE** qualifier:

Keyword	Description
CLEAR_SCREEN	Clears the screen before each page is displayed.
SCROLL	Displays information one line at a time.
SAVE[= *n*]	Enables screen navigation of information, where *n* is the number of pages to store.

The **/PAGE=SAVE** qualifier allows you to navigate through screens of information. The **/PAGE=SAVE** qualifier stores up to 5 screens of up to 255 columns of information. When you use the **/PAGE=SAVE** qualifier, you can use the following keys to navigate through the information:

Key Sequence	Description
Up arrow key, **Ctrl/B**	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.
Right arrow key	Scroll right one column.
Find (**E1**)	Specify a string to find when the information is displayed.
Insert Here (**E2**)	Scroll right one half screen.
Remove (**E3**)	Scroll left one half screen.
Select (**E4**)	Toggle 80/132 column mode.
Prev Screen (**E5**)	Get the previous page of information.
Next Screen (**E6**), **Return**, **Enter**, **Space**	Get the next page of information.
F10, **Ctrl/Z**	Exit. (Some utilities define these differently.)
Help (**F15**)	Display utility help text.
Do (**F16**)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.

The **/PAGE** qualifier is not compatible with the **/OUTPUT** qualifier.

/SEARCH="*string*"

Use with the **/PAGE=SAVE** qualifier to specify a string that you want to find in the information being displayed. Quotation marks are required for the **/SEARCH** qualifier, if you include spaces in the text string.

You can also dynamically change the search string by pressing the Find key (**E1**) while the information is being displayed. Quotation marks are not required for a dynamic search.

/SUMMARY

Displays the total number of executing jobs, pending jobs, holding jobs, retained jobs, and timed release jobs for each queue. For output queues, the total block count for pending jobs is also shown.

/WRAP
/NOWRAP (default)

Use with the **/PAGE=SAVE** qualifier to limit the number of columns to the width of the screen and to 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 and can be seen when you use the scrolling (left and right) features provided by the **/PAGE=SAVE** qualifier.

Examples

1.
```
$ SHOW QUEUE/ALL/BY_JOB_STATUS=pending

Printer queue KLEE$LCA0, stopped, mounted form DEFAULT
  /BASE_PRIORITY=4 /DEFAULT=(FEED,FORM=ELITE (stock=DEFAULT))
  /OWNER=[SYSTEM] /PROTECTION=(S:M,O:D,G:R,W:S)

  Entry     Jobname         Username        Blocks  Status
  -----     -------         --------        ------  ------
   2045     APPLICA         MARTIN             102  Pending

Batch queue SYS_TEX, available, on NODE22::
  /BASE_PRIORITY=3 /JOB_LIMIT=25 /OWNER=[SYSTEM]
  /PROTECTION=(S:M,O:D,G:R,W:S)

  Entry     Jobname         Username                Status
  -----     -------         --------                ------
   1388     CHAPTER1.JOB    MARTIN                  Pending

Generic batch queue CLUSTER_BATCH
  /GENERIC=(HMSCMS_BATCH,NODE22_BATCH) /OWNER=[SYSTEM]
  /PROTECTION=(S:M,O:D,G:R,W:S)

  Entry     Jobname         Username                Status
  -----     -------         --------                ------
     936    no privilege                            Pending
```

   The **SHOW QUEUE** command in this example displays only the queues in which jobs are currently pending. The **/ALL** qualifier requests the display of all pending jobs. Notice that you cannot see job information for entry `936`. You must have GROUP privilege to show all jobs in your group or OPER privilege to show all jobs in all groups.

2.
```
$ SHOW QUEUE/FULL ARMADA*

Batch queue ARMADA_BATCH, idle, on ARMADA::
  /AUTO_START_ON=(ARMADA::) /BASE_PRIORITY=3 /JOB_LIMIT=3
  /OWNER=[SYSTEM] /PROTECTION=(S:M,O:D,G:R,W:S)

Printer queue ARMADA$PRINT, stopped, on ARMADA::LPA0,
 mounted form DEFAULT
  /BASE_PRIORITY=4 /DEFAULT=(FEED,FORM=ELITE (stock=DEFAULT))
  /OWNER=[SYSTEM] /PROTECTION=(S:E,O:D,G:R,W:W)

  Entry  Jobname         Username     Blocks  Status
  -----  -------         --------     ------  ------
    605  SET             DUDSON          140  Pending (queue stopped)
       Submitted 21-DEC-2001 16:14 /FORM=ELITE (stock=DEFAULT)
```

```
          /PRIORITY=200
          File: _$5$DUA1:[DUDSON]SET.CLD;5

     606  CLOSE           DUDSON          140  Pending (queue stopped)
          Submitted 21-DEC-2001 16:16 /FORM=DEFAULT /PRIORITY=200
          File: _$5$DUA1:[DUDSON]CLOSE.TXT;2

Printer queue ARMADA$PRINTER_1, idle, on ARMADA::TTA3,
 mounted form DEFAULT

<Printer queue on node ARMADA for a LN03R printer>
  /BASE_PRIORITY=4 /DEFAULT=(FEED,FORM=DEFAULT) /LIBRARY=SMU$DEVCTL
  Lowercase /OWNER=[SYSTEM] /PROCESSOR=CPS$SMB
  /PROTECTION=(S:M,O:D,G:R,W:S) /SEPARATE=(FLAG)
```

The **SHOW QUEUE** command in this example lists information on all of your current job entries. Complete information about the queues, the jobs contained in those queues, and the files associated with the jobs is displayed.

3.
```
$ SHOW QUEUE/SUMMARY

Batch queue HMSCMS_BATCH, available, on HMSCMS::

    Job summary:  1 executing,  5 holding

Server queue NM$QUE01, available, on NODE22::, mounted form DEFAULT

    Job summary:  2 executing

Generic server queue NM$QUEUE

    Job summary:  1 timed release

Batch queue PIZZA_BATCH, idle, on PIZZA::

    Job summary:  queue is empty

Batch queue NODE22_BATCH, available, on NODE22::

    Job summary:  1 executing,  1 timed release

Printer queue ARMADA$PRINT, stopped, on ARMADA::LPA0, mounted
  form DEFAULT

    Job summary:  8 pending (565 blocks)

Terminal queue ALLBQ, stopped, on KATYA::TTA2, mounted form DEFAULT

    Job summary:  2 pending (445 blocks)
```

The **SHOW QUEUE** command in this example lists all queues and displays the total number of jobs on the queues.

4.
```
$ SHOW QUEUE DOC$LN03

Server queue DOC$LN03, stopped, on NEWTON::, mounted form DEFAULT

 Entry  Jobname   Username  Blocks  Status
```

```
_____  _______   ________  ______  ______
  436  DOCPLAN   HERSHEY         8  Retained until 11-JAN-2001 16:56
     %JBC-F-JOBABORT, job aborted during execution
       Completed  11-JAN-2001 15:56 on queue DOC$LN03
```

This **SHOW QUEUE** display includes the date and time at which a retained job completed and the queue on which it executed. You can specify the **/RETAIN** qualifier with the **PRINT**, **SUBMIT**, or **SET ENTRY** command to specify the circumstances under which you want your job to be retained in a queue. In this example, the user set job retention to be one hour after job completion. Depending on the queue's job retention policy, the job might be deleted from the queue at `16:56`.

See the **PRINT**, **SUBMIT**, or **SET ENTRY** command descriptions for more information on the **/RETAIN** qualifier.

SHOW QUEUE/CHARACTERISTICS

SHOW QUEUE/CHARACTERISTICS — Displays information about queue characteristics defined for the system. A characteristic is a user-defined attribute of a batch or output queue, such a sink color.

Format

SHOW QUEUE/CHARACTERISTICS [*characteristic-name*]

Parameter

characteristic-name

Specifies the name of a characteristic. The asterisk (*) and the percent sign (%) wildcard characters are allowed. The default value for the *characteristic-name* parameter is the asterisk wildcard (*). Thus, information about all characteristics is displayed when you do not specify a characteristic name.

You create a characteristic name with the **DEFINE/CHARACTERISTIC** command. You can then assign it to a queue by using the **INITIALIZE/QUEUE**, the **SET QUEUE**, or the **START/QUEUE** command with the **/CHARACTERISTICS** qualifier. You can also specify characteristics for a job by using the **/CHARACTERISTICS** qualifier with the **PRINT**, the **SUBMIT**, or the **SET ENTRY** command.

Description

The **SHOW QUEUE/CHARACTERISTICS** command displays the characteristic name and number of characteristics defined for the system. To see which characteristics are associated with a particular queue, use the **SHOW QUEUE** command with the **/FULL** qualifier.

Qualifier

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the command is sent. By default the output of the **SHOW QUEUE/CHARACTERISTICS** command is sent to the current SYS$OUTPUT device(usually your terminal). To send the output to a file, use the **/OUTPUT** qualifier followed by a file specification.

The file specification may not include any wildcard characters. If you enter a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS is the default file type.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

Examples

1.
```
$ SHOW QUEUE/CHARACTERISTICS
Characteristic name                  Number
-------------------                  ------
BLUEINK                                   6
BROWNINK                                 25
COLOR_CHART                               1
REDINK                                    0
TEXMAC_JOB                                2
```

   The **SHOW QUEUE/CHARACTERISTICS** command in this example displays all the characteristics that have been defined for this system.

2.
```
(wide)
$ SHOW QUEUE/CHARACTERISTICS *INK
Characteristic name                  Number
-------------------                  ------
BLUEINK                                   6
BROWNINK                                 25
REDINK                                    0
```

   The **SHOW QUEUE/CHARACTERISTICS** command in this example displays the name and number of all characteristics that end with `INK`.

SHOW QUEUE/FORM

SHOW QUEUE/FORM — Displays information about forms defined for the system. Forms define the size and type paper and the layout of text that are used for print jobs.

Format

SHOW QUEUE/FORM [*form-name*]

Parameter

form-name

Specifies the name of the form. The asterisk (*) and the percent sign (%) wildcard characters are allowed. The default value for the *form-name* parameter is an asterisk (*), which means that the names of all forms on the system are displayed.

Description

The **SHOW QUEUE/FORM** command displays the name and number of forms defined for the system. To display the attributes associated with forms, use the **/FULL** qualifier.

Use the **SHOW QUEUE/FORM** command to find out which forms are available on the system. To see which forms are available on one or more queues, use the **SHOW QUEUE** command with the **/FULL** qualifier.

Form names are created by the **DEFINE/FORM** command. You can specify a particular form for a print job by using the **/FORM** qualifier with the **PRINT** or **SET ENTRY** command. Forms are assigned to queues with the **INITIALIZE/QUEUE**, the **SET QUEUE**, or the **START/QUEUE** command.

Qualifiers

/BRIEF (default)

Displays a brief description (form names, numbers, and descriptions) about the forms on the system.

/FULL

Displays a full description (including paper size and margin settings) about the forms on the system.

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the command is sent. By default the output of the **SHOW QUEUE/FORM** command is sent to the current SYS$OUTPUT device (usually your terminal). To send the output to a file, use the **/OUTPUT** qualifier followed by a file specification.

The file specification cannot include the asterisk (*) and the percent sign (%) wildcard characters. If you enter a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS is the default file type.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

Examples

1.
```
$ SHOW QUEUE/FORM DEFAULT
Form name                          Number   Description
---------                          ------   -----------
DEFAULT                                 0   System-defined default
```

   The **SHOW QUEUE/FORM** command in this example displays only the default form.

2.
```
$ SHOW QUEUE/FORM LN01*
Form name                             Number   Description
---------                             ------   -----------
LN01_LANDSCAPE (stock=DEFAULT)           105   132 by 66 (landscape)
LN01_LANDSCAPE_INDENTED (stock=DEFAULT)
                                         107   132 by 65 (landscape)
LN01_PORTRAIT (stock=DEFAULT)            106   80 by 60 (portrait)
```

   The **SHOW QUEUE/FORM** command in this example displays the names of all forms, including the stock, that begin with `LN01`. The display includes the names, stock for each form, numbers, and brief descriptions of those forms.

3.
```
$ SHOW QUEUE/FORM/FULL
Form name                        Number   Description
```

```
---------                          ------   -----------
132_51_STD (stock=DEFAULT)           102   132 by 51 (standard short)
    /LENGTH=51 /MARGIN=(BOTTOM=6) /STOCK=DEFAULT /TRUNCATE
    /WIDTH=132
40_66_STD (stock=DEFAULT)            103   40 by 66 (standard labels)
    /LENGTH=66 /MARGIN=(BOTTOM=6) /STOCK=DEFAULT /WIDTH=40
BLUE_PAPER_STOCK (stock=DIGITAL_8X11_STOCK1412TEA)
                                   22222 blue paper, DEC order# 22222
    /LENGTH=66 /MARGIN=(BOTTOM=6) /STOCK=DIGITAL_8X11_STOCK1412TEA
    /TRUNCATE  /WIDTH=80
DEFAULT                              0   System-defined default
    /LENGTH=66 /MARGIN=(BOTTOM=6) /STOCK=DEFAULT /TRUNCATE
    /WIDTH=132
LN01_LANDSCAPE (stock=DEFAULT)       105   132 by 66 (landscape)
    /LENGTH=66 /STOCK=DEFAULT /WIDTH=132
LN01_LANDSCAPE_INDENTED (stock=DEFAULT)
                                     107   132 by 65 (landscape)
    /LENGTH=65 /SETUP=(LN01_TOP_MARGIN_150) /STOCK=DEFAULT
    /WIDTH=132
LN01_PORTRAIT (stock=DEFAULT)        106   80 by 60 (portrait)
    /LENGTH=60 /SETUP=(LN01_PORTRAIT) /STOCK=DEFAULT /WIDTH=80

MEMO (stock=DEFAULT)                 110   LN03 indented memo format
    /LENGTH=64 /MARGIN=(TOP=2,LEFT=5) /STOCK=DEFAULT /TRUNCATE
    /WIDTH=80
```

This **SHOW QUEUE/FORM** command also displays the names of all form types and stock for the system. By using the **/FULL** qualifier, you can see what image size has been set for each form type.

SHOW QUEUE/MANAGERS

SHOW QUEUE/MANAGERS — Provides information about the queue manager(s) on the system or OpenVMS Cluster.

Format

SHOW QUEUE/MANAGERS [*queue-manager-name*]

Parameter

queue-manager-name

Specifies the name of the queue manager for which you want information displayed. The asterisk (*) and the percent sign (%) wildcard characters are allowed. The default value for the *queue-manager-name* parameter is the asterisk (*) wildcard character. If no queue manager name is specified, information on all queue managers is displayed.

Description

The **SHOW QUEUE/MANAGERS** command displays information about the queue manager(s) on the system or OpenVMS Cluster. The command assumes the **/BRIEF** qualifier as the default, and displays a one-line description that include search queue manager's name, status, and process node name. The **/FULL** qualifier displays the following information:

- Full file specification of the queue database master file (disk, directory, and file name of the shared master file)
- Name of queue manager
- Status of queue manager
- The node name on which the queue manager process is running (if the queue manager process is currently running)
- Prioritized list of nodes on which the queue manager process can run
- Disk and directory containing the queue database files specific to the particular queue manager

If a queue manager process is not running, the current location node name will not be displayed for that queue manager.

The queue manager status can be one of the following:

State	Description
Start pending	The queue manager process is attempting to start up, however, it is waiting for a node on which it can run to enter the cluster.
Starting	The queue manager is in the process of starting up as a result of a **START/QUEUE/MANAGER** command or failing over from another node in the cluster.
Running	The queue manager is running.
Failing over	The queue manager is in the process of failing over to another node in the cluster.
Stopping	The queue manager is in the process of stopping.
Stopped	The queue manager is stopped and will not start until a **START/QUEUE/MANAGER** command is executed.

Qualifiers

/BRIEF

Displays a one-line description of queue manager(s) on the system or cluster. The information displayed includes the name and status of the queue manager and the node on which its process is running. The **/BRIEF** qualifier is present by default.

/FULL

Displays complete information about queue manager(s) on the system or cluster. The **/FULL** qualifier overrides the **/BRIEF** qualifier.

Examples

1.
```
$ SHOW QUEUE/MANAGERS/FULL
Master file:  SYS$COMMON:[SYSEXE]QMAN$MASTER.DAT;

Queue manager BATCHQ_MANAGER, running, on BBBBBB::
  /ON=(BBBBBB, AAAAAA, CCCCCC, EEEEEE, *)
```

```
    Database location:  DISK1:[QUEUES]

  Queue manager PRINTQ_MANAGER, starting, on AAAAAA::
    /ON=(AAAAAA, BBBBBB, CCCCCC, EEEEEE, *)
    Database location:  DISK2:[QUEUES]

  Queue manager SYS$QUEUE_MANAGER, running, on FFFFFF::
    /ON=(FFFFFF, EEEEEE, CCCCCC, BBBBBB, AAAAAA, *)
    Database location:  DISK1:[QUEUES]

  Queue manager TRANSFER_MANAGER, stopped
    /ON=(AAAAAA, BBBBBB, EEEEEE)
    Database location:  DISK1:[TRANSFER]
```

The **SHOW QUEUE/MANAGERS/FULL** command displays complete information about a queue manager on a system or cluster.

2.
```
$ SHOW QUEUE/MANAGERS
Queue manager BATCHQ_MANAGER, running, on BBBBBB::

Queue manager PRINTQ_MANAGER, starting, on AAAAAA::

Queue manager SYS$QUEUE_MANAGER, running, on FFFFFF::

Queue manager TRANSFER_MANAGER, stopped
```

3. The **SHOW QUEUE/MANAGERS** command in this example shows information about the queue manager in brief format. The **/BRIEF** qualifier is the default and does not need to be specified.

```
$ SHOW QUEUE/MANAGERS/FULL SYS$QUEUE_MANAGER
Master file:  SYS$COMMON:[SYSEXE]QMAN$MASTER.DAT;

Queue manager SYS$QUEUE_MANAGER, running, on FFFFFF::
  /ON=(FFFFFF, EEEEEE, CCCCCC, BBBBBB, AAAAAA, *)
  Database location:  DISK1:[QUEUES]
```

The command, **SHOW QUEUE/MANAGERS/FULL SYS$QUEUE_MANAGER**, in this example shows a request for a specific queue manager by name.

SHOW QUOTA

SHOW QUOTA — Displays the current disk quota that is authorized for a specific user on a specific disk. This display includes a calculation of the amount of space available and the amount of overdraft that is permitted. **Requires read (R) access to the quota file in order to display the quotas of other users.**

Format

SHOW QUOTA

Description

The **SHOW QUOTA** command indicates whether a quota exists for any specific user on a specific disk. The display that results gives the quotas used, authorized, and available in blocks. The amount of overdraft permitted is also shown.

Qualifiers

/DISK[=*device-name*[:]]

Specifies the disk whose quotas are to be examined. By default, the current default disk (defined by the logical name SYS$DISK) is examined.

/USER=*uic* OR *identifier*

Specifies which user's quotas are to be displayed. By default, the current user's quotas are displayed.

Examples

1. ```
   $ SHOW QUOTA
   User [360,010] has 2780 blocks used, 7220 available,
   of 10000 authorized and permitted overdraft of 500 blocks on DISK$
   ```

   The **SHOW QUOTA** command in this example displays the amount of disk space authorized, used, and still available on the current default disk for the present user. The permitted overdraft in this example is `500` blocks.

2. ```
   $ SHOW QUOTA /USER=[360,007]/DISK=XXX1:
   %SYSTEM-F-NODISKQUOTA, no disk quota entry for this UIC
   ```

   The **SHOW QUOTA** command in this example displays the fact that the user with user identification code (UIC) `[360,007]` has no disk quota allocation on device `XXX1`.

3. ```
   $ SHOW QUOTA /USER=[360,111]
   User [360,111] has 27305 blocks used, 2305 OVERDRAWN,
   of 25000 authorized and permitted overdraft of 4000 blocks on DISK$
   ```

   The **SHOW QUOTA** command in this example illustrates a user with an overdrawn quota.

SHOW RMS_DEFAULT

SHOW RMS_DEFAULT — Displays the current default values for the multiblock count, the multibuffer count, the network transfer size, the prolog level, the extend quantity, and the query lock option.

Format

SHOW RMS_DEFAULT

Parameters

None.

Description

The **SHOW RMS_DEFAULT** command displays information that OpenVMS Record Management Services (RMS) uses for file operations including the default values for the multibuffer count, the multiblock count, the network block count, the prolog level, the extend quantity, and the query lock option. The command provides these values at both the current process level and at the system level.

The **SHOW RMS_DEFAULT** command lists the multibuffer count values for each of the file types, including the values for the three classes of sequential files(disk, magtape and unit record).
   ```

Qualifier

/OUTPUT[=*filespec*]
/NOOUTPUT

Specifies the file to which the display is written (default is SYS$OUTPUT). The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification. If you enter the **/OUTPUT** qualifier with a partial file specification (for example, specifying only a directory), SHOW is the default output file name and .LIS the default output file type.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

Example

```
$ SHOW RMS_DEFAULT
        MULTI- |               MULTIBUFFER COUNTS             | NETWORK
        BLOCK  | Indexed Relative         Sequential          | BLOCK
        COUNT  |                   Disk  Magtape  Unit Record | COUNT
Process   0    |    0        0      0      0         0        |   0
System   16    |    0        0      0      0         0        |   8
         Prolog    Extend Quantity      VCC_DFW     QUERY_LOCK
Process    0              0                0        System
System     0              0                0        Enabled
         CONTENTION_POLICY
Process      System
System       Never
```

The **SHOW RMS_DEFAULT** command in this example shows a system multiblock count of 16 and a network block count of 8. These are typical values. Note that the placeholder, VCC_DFW, is currently reserved for use by VSI.

SHOW ROOT

SHOW ROOT — Displays the current value of the system root and, if defined, the process root.

Format

SHOW ROOT

Example

```
$ SHOW ROOT
DISK1:[000000]
$ SET ROOT DISK5:[ PSX$ROOT]
$ SHOW ROOT
DISK5:[ PSX$ROOT]
$
```

The **SHOW ROOT** command in this example displays the current root path. The **SET ROOT** command changes these defaults, and the next **SHOW ROOT** command displays the new default root.

SHOW SECURITY

SHOW SECURITY — Displays the name, class, and profile of a protected object.

Format

SHOW SECURITY *object-name*

Parameter

object-name

Specifies the name of an object whose security profile is to be displayed. If the object class is SECURITY_CLASS, you can specify an asterisk (*) as *object-name* to see a display of all security class objects. An object name of the FILE class (explicitly or implicitly specified) can include the asterisk (*) and the percent sign (%) wildcard characters; however, wildcard characters are not allowed in any class other than FILE or SECURITY_CLASS.

Description

The **SHOW SECURITY** command displays the name, class, and profile of a protected object. A profile includes an access control list (ACL), the protection code, and the owner of a protected object. For SECURITY_CLASS objects, **SHOW SECURITY** displays all the template profiles, which are the basis for profiles of new objects.

The **/CLASS** qualifier identifies the class *object-name* is a member of. If the class is DEVICE and the object is a disk device on which a volume is mounted as a Files-11 (rather than foreign) volume, then both the device and the volume profiles are displayed.

All qualifiers other than **/CLASS** apply only to files.

For a description of protected objects, see the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#SEC_OBJ_LIST].

Qualifiers

/BACKUP

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/BACKUP** qualifier selects files according to the dates of their most recent backups (rather than by the creation, expiration, or modification date). By default, **SHOW SECURITY** selects files according to their creation date.

/BEFORE[=*time*]

Selects only those files dated prior to the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify the **/CREATED** or the **/MODIFIED** qualifier to indicate the time attribute to be used as the basis for selection. The **/CREATED** qualifier is the default.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/BY_OWNER[=*uic*]

Selects files whose owner's UIC matches the UIC specified with *uic*. The default UIC is that of the current process.

Specify the UIC by using the standard format described in the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#UIC_SECTION].

/CLASS=*class*

Specifies the class of the object whose profile is to be displayed. By default, the command assumes the object class is FILE. Object class keywords are as follows:

CAPABILITY	LOGICAL_NAME_TABLE
COMMON_EVENT_CLUSTER	QUEUE
DEVICE	RESOURCE_DOMAIN
FILE	SECURITY_CLASS
GROUP_GLOBAL_SECTION	SYSTEM_GLOBAL_SECTION
ICC_ASSOCIATION	VOLUME

/CREATED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/CREATED** qualifier selects files according to the date they were created (rather than by the backup, expiration, or modification date). By default, **SHOW SECURITY** selects files according to their creation date.

/EXCLUDE=(*filespec*[,...])

Excludes the specified files from the **SHOW SECURITY** operation. You can include a directory, but not a device, in the file specification. You cannot use relative version numbers to exclude a specific version.

/EXPIRED

Modifies the time specified with the **/BEFORE** or the **/SINCE** qualifier. The **/EXPIRED** qualifier selects files according to their expiration dates rather than by the backup, creation, or modification date. (The expiration date is set with the **SET FILE/EXPIRATION_DATE** command.) By default, files are selected according to their creation date.

/MODIFIED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/MODIFIED** qualifier selects files according to the dates on which they were last modified, rather than by the backup, creation, or expiration date. By default, files are selected according to their creation date.

/SINCE[=*time*]

Selects only those files dated on or after the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, JOB_LOGIN, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify the **/CREATED** or the **/MODIFIED** qualifier to indicate the time attribute to be used as the basis for selection. The **/CREATED** qualifier is the default.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/SYMLINK=*keyword*

The valid keywords for this qualifier are [NO]WILDCARD, [NO]ELLIPSIS, and [NO]TARGET. Descriptions are as follows:

Keyword	Description
NOWILDCARD	Indicates that symlinks are disabled during directory wildcard searches.
WILDCARD	Indicates that symlinks are enabled during wildcard searches.
NOELLIPSIS	Indicates that symlinks are matched for all wildcard fields except for ellipsis.
ELLIPSIS	Equivalent to WILDCARD (included for command symmetry).
TARGET	Indicates that if the named file is a symlink, then the symlink is followed to operate on the symlink target.
NOTARGET	Indicates that the command operates on the named file whether it is an ordinary file or a symlink.

If the file named in the **SHOW SECURITY** command is a symlink, the command by default operates on the symlink itself.

Examples

1.
```
$  SHOW SECURITY LNM$SYSTEM_TABLE /CLASS=LOGICAL_NAME_TABLE

LNM$SYSTEM_TABLE object of class LOGICAL_NAME_TABLE

     Owner: [SYSTEM]
     Protection: (System: RWC, Owner: RWC, Group: R, World: R)
     Access Control List:
          (IDENTIFIER=[USER,SVENSEN],ACCESS=CONTROL)
```

   This example shows a typical request to display the security elements of an object. The logical name table `LNM$SYSTEM_TABLE` is displayed with the settings of the security elements owner, protection, and ACL.

2.
```
$ SHOW SECURITY/CLASS=DEVICE $99$DUA22

_$99$DUA22: object of class DEVICE
    Owner: [SALES,TSUTTER]
    Protection: (System: RWPL, Owner: RWPL, Group: R, World)
    Access Control List: <empty>

RES17SEP object of class VOLUME
    Owner: [FEAST,FY93]
    Protection: (System: RWCD, Owner: RWCD, Group: RWCD, World: RWCD)
    Access Control List: <empty>

$ SHOW DEVICE $99$DUA22

Device                 Device      Error    Volume      Free  Trans Mnt
 Name                  Status      Count     Label     Blocks Count Cnt
$99$DUA22:   (KUDOS)  Mounted         0  RES17SEP     649904     1   2
```

   This example shows a request for the security profile of a disk device. The resulting display provides both the profiles of the disk `$99$DUA22` and the volume `RES17SEP` that is mounted on it. The subsequent **SHOW DEVICE** command confirms that the volume is mounted on the device `$99$DUA22`.

3.
```
$  SHOW SECURITY LOGICAL_NAME_TABLE /CLASS=SECURITY_CLASS
```

```
LOGICAL_NAME_TABLE object of class SECURITY_CLASS

     Owner: [SYSTEM]
     Protection: (System: RWCD, Owner: RWCD, Group: R, World: R)
     Access Control List: <empty>

  Template: GROUP

     Owner: [SYSTEM]
     Protection: (System: RWCD, Owner: R, Group: R, World: R)
     Access Control List: <empty>

  Template: JOB

     Owner: [SYSTEM]
     Protection: (System: RWCD, Owner: RWCD, Group, World)
     Access Control List: <empty>

  Template: DEFAULT

     Owner: [SYSTEM]
     Protection: (System: RW, Owner: RW, Group: R, World: R)
     Access Control List: <empty>
```

This example shows the output for the special case of a security class object. The security class object LOGICAL_NAME_TABLE is displayed with the security profile. In addition, three templates are displayed.

4.
```
$  SHOW SECURITY * /CLASS=SECURITY_CLASS

SECURITY_CLASS object of class SECURITY_CLASS
     Owner: [SYSTEM]
     Protection: (System: RWCD, Owner: RWCD, Group: R, World: R)
     Access Control List:  <empty>

LOGICAL_NAME_TABLE object of class SECURITY_CLASS
     Owner: [SYSTEM]
     Protection: (System: RWCD, Owner: RWCD, Group: R, World: R)
     Access Control List:  <empty>
```

This example shows the output for the special case of showing all the security classes currently registered. The asterisk (*) wildcard character is used; any other form of wildcard characters is not accepted. Security profiles are shown for each security class. Note that template information is not shown.

SHOW SERVER ACME_SERVER

SHOW SERVER ACME_SERVER — Displays information about the ACME server and ACME agent. **Requires the SYSPRV privilege.**

Format

SHOW SERVER ACME_SERVER

Description

The **SHOW SERVER ACME_SERVER** command provides a system manager with the ability to display information about the ACME server and one or more ACME agents.

For related information, see the **SET SERVER ACME_SERVER** command.

Qualifiers

/AGENT[=(*agent*,...*agent*)] (default)
/NOAGENT

Specifies whether to display ACME agent information.

/BRIEF (default)

Displays summary information.

/CLUSTER

Executes the command on all cluster members.

/FULL

Displays complete information.

/NODE=(*node-name*[,...])

Specifies the cluster member or members for which information is to be displayed.

/OUTPUT=*filespec*

Sends the output information to the specified file.

/PAGE

Displays information one screen at a time.

/SERVER (default)
/NOSERVER

Specifies whether to display ACME server information.

Example

1. `$ SHOW SERVER ACME_SERVER`

   This example shows the output for **SHOW SERVER ACME/BRIEF** on an Alpha system.

```
ACME Information on node TSINOD  29-JUN-2004 10:08:12.32  Uptime
 000:05:45

ACME Server id: 5  State: Processing New Requests ❶
   Agents Loaded:         2   Active:       2 ❷
   Thread Maximum:        4   Count:        4
   Request Maximum:       8   Count:        0
```

```
ACME Agent id: 1  State: Active ❸
   Name: "VMS" ❹
   Image: "DISK$AXP_721:[VMS$COMMON.SYSLIB]VMS$VMS_ACMESHR.EXE;1" ❺
   Identification: "VMS ACME built  3-JUN-1999" ❻
   Information: "No requests completed since the last startup" ❼
   Domain of Interpretation: Yes ❽
   Execution Order:      1 ❾

ACME Agent id: 2  State: Active
   Name: "MSV1_0"
   Image: "DISK$AXP_721:[SYS0.SYSLIB]PWRK$MSV1_0_ACMESHR.EXE;2"
   Identification: "MSV1_0 ACME X-22"
   Information: "MSV1_0 ready to process requests."
   Domain of Interpretation: Yes
   Execution Order:      2
```

The following are the item descriptions keyed to the callouts in the example output:

❶ Server state
❷ Server agent/thread/request-slot statistics
❸ Agent state
❹ Agent name
❺ Agent file-specification
❻ Agent-defined identification string (static)
❼ Agent-defined information string (dynamic)
❽ Domain-of-interpretation (DOI) indicator
❾ Execution order (the order in which agents are called for each request)

2. `$ SHOW SERVER ACME_SERVER/FULL`

This example shows the output for **`SHOW SERVER ACME/FULL`** on an Alpha system.

```
ACME Information on node AUTUMN  25-SEP-2009 04:15:37.34  Uptime 0
 22:19:26

ACME Server id: 255  State: Processing New Requests
   Agents Loaded:        2   Active:      2
   Thread Maximum:       1   Count:       1
   Request Maximum:    700   Count:       0  ❶
   Requests awaiting service:             0  ❷
   Requests awaiting dialogue:            0  ❸
   Requests awaiting AST:                 0  ❹
   Requests awaiting resource:            0  ❺
   Logging status: Active

   Tracing status: Inactive
   Log file: "SYS$SYSROOT:[SYSMGR]ACME$SERVER.LOG;147"

ACME Agent id: 1  State: Active
   Name: "VMS"
   Image: "DISK$JEDIXC42:[VMS$COMMON.SYSLIB]VMS$VMS_ACMESHR.EXE;1"
   Identification: "VMS ACME built  20-SEP-2006"
   Information: "Unused max 1: 0u 0r 0a 1a 1w 0m 0o 0c 1h 1d" ❻

   Domain of Interpretation: Yes
   Execution Order:      1
```

```
   Credentials Type:     1   Name: "VMS"
   Resource wait count:                    0  ❼

ACME Agent id: 2  State: Active
   Name: "LDAP-STD"
   Image: "DISK$JEDIXC42:[VMS$COMMON.SYSLIB]LDAPACME$LDAP-
STD_ACMESHR.EXE;2"
   Identification: "LDAP ACME Standard V1.5"
   Information: "ACME_LDAP_DOI Agent is initialized"
   Domain of Interpretation: Yes
   Execution Order:      2
   Credentials Type:     3   Name: "LDAP"
   Resource wait count:                    0 ❽
$
```

The following are the item descriptions keyed to the callouts in the example output:

❶ Number of requests currently handled by the ACME agent.
❷ Number of requests waiting to get serviced.
❸ Number of requests waiting for user dialogue/inputs.
❹ Number of requests waiting for system services to send AST.
❺ Total number of requests waiting for the availability of ACME agents such as the VMS ACME and LDAP ACME agents.
❻ Internal counter values of VMS ACME agent.
❼ Number of requests waiting for the availability of resources such as the SYSUAF file from the VMS ACME agent.
❽ Number of requests waiting for the availability of resources such as the SYSUAF file from the LDAP ACME agent.

SHOW SERVER REGISTRY_SERVER (Alpha/IA-64 only)

SHOW SERVER REGISTRY_SERVER (Alpha/IA-64 only) — Displays information about the Registry server. **Requires the SYSPRV privilege.**

Format

SHOW SERVER REGISTRY_SERVER

Description

The **SHOW SERVER REGISTRY_SERVER** command provides a system manager with the ability to display information about the Registry server.

For related information, see the **SET SERVER REGISTRY_SERVER** command.

Qualifiers

/CLUSTER

Displays information about each server in the cluster.

Cannot be used with the **/MASTER** or **/NODE** qualifiers.

/MASTER

Requires the SYSLCK privilege.

Displays information about the Registry master server only.

Cannot be used with the **/CLUSTER** or **/NODE** qualifiers.

/NODE=(*node-name*[,...])

Displays information about the Registry servers on the specified nodes in the order in which the nodes are listed. The specified nodes must be in the current cluster.

Cannot be used with the **/CLUSTER** or **/MASTER** qualifiers.

/OUTPUT=*filespec*

Sends the output information to the specified file.

Examples

1. `$ SHOW SERVER REGISTRY_SERVER`

   This command displays information about the node you are on. For example:

   ```
   Registry Information on node OPNEYE  29-JUN-2004 16:37:26.52 -
   Uptime 20 04:10:55
   ```

2. `$ SHOW SERVER REGISTRY_SERVER/MASTER`

   This command displays information about the Registry master server. For example:

   ```
   Registry Master Server is running on OPNEYE, Pid: 20A0021A
   ```

SHOW SHADOW

SHOW SHADOW — Displays the status of a specified shadow set on the system. If no qualifiers are specified, displays a summary of the device.

Format

SHOW SHADOW [DSA*n*[:]]

Parameter

DSA*n*[:]

Specifies the virtual unit name of the shadow set for which information is to be displayed.

If no virtual unit is specified, information is displayed for all mounted shadow sets.

Qualifiers

/ACTIVE

Displays one of three possible states:

- Merge or Copy is not required
- Copy is in progress on node *node-name*, *n*% completed
- Merge is in progress on node *node-name*, *n*% completed

/BY_PRIORITY
/BY_PRIORITY DSA*n*:

Displays a list of shadow sets on the system, sorted in the order in which a recovery operation (copy, full merge, or minimerge) is scheduled to occur. If a copy or merge is active, the display shows on which system it is occurring. This list is specific to the system on which the command is executed, so the list might vary across systems in a cluster. Higher numbers indicate a higher priority.

Tip

When the command specifies a virtual unit (DSA*n*:), the display shows the current priority and status of just this virtual unit. If any copy or merge operations are in progress, the system on which the operation is occurring is also displayed, along with the operation's progress.

/COPY

Displays one of three possible states:

- Copy is not required
- Copy is pending
- Copy is in progress on node *node-name*, *n*% completed

/FULL

Displays complete information about the specified devices (or all devices if no device is specified). The only other qualifier that can be specified with **`/FULL`** is **`/OUTPUT`**.

/MERGE

Displays one of three possible states:

- Merge is not required
- Merge is pending
- Merge is in progress on node *node-name*, *n*% completed

/NAME[=*policy-name*]

Used with **`/POLICY=HBMM`** to display the characteristics of a specific, named host-based minimerge (HBMM) policy or, if no policy name is specified, all named HBMM policies on the cluster.

/OUTPUT[=*filespec*]

Outputs any messages to the specified file.

/POLICY=HBMM

Displays the host-based minimerge (HBMM) policy associated with the specified devices.

HBMM is the only value supported for **/POLICY** and must be included.

If the command is issued with no device specification, information is output for all devices. There might be a noticeable delay between the output of device information or before returning to the DCL command prompt. This delay is normal.

/SITE=*n*

Displays every shadow set member device (and the virtual device of which it is a member) that has the site value specified by *n*.

Examples

1.
```
$ SHOW SHADOW DSA112:
_DSA112:  Volume Label: DSA112
  Virtual Unit State:   Steady State
  No Enhanced Shadowing Features in use

  VU Timeout Value      1200    VU Site Value          0
  Copy/Merge Priority   5000    Mini Merge      Disabled
  Served Path Delay     30

  Device $1$DGA112              Master Member
    Read Cost             2     Site 0
    Member Timeout        180

  Device $1$DGA113
    Read Cost             2     Site 0
    Member Timeout        180
```

   This display is for a shadow set that is in a steady state. That is, there are no active or pending merges or copy operations for this shadow set and HBMM has not been enabled.

2.
```
$ SHOW SHADOW DSA112:
_DSA112:  Volume Label: DSA112
  Virtual Unit State: MiniMerge Active (5%), Copy Pending (32%) on NODE1
  Enhanced Shadowing Features in use:
      Host-Based Minimerge (HBMM)

  VU Timeout Value      1200    VU Site Value          0
  Copy/Merge Priority   5000    Mini Merge       Enabled
  Served Path Delay     30

  HBMM Policy
    HBMM Reset Threshold: 50000
    HBMM Master lists:
      Up to any 6 nodes in the cluster
    HBMM bitmaps are active on NODE1,NODE2
```

```
    Modified blocks since bitmap creation: 137033

  Device $1$DGA112              Master Member, Merging (5%)
    Read Cost             2     Site 0
    Member Timeout        180

  Device $1$DGA113              Copy Target (32%)
    Read Cost             2     Site 0
    Member Timeout        180
```

This display is for a shadow set that is enabled for HBMM. A shadow set member copy and a shadow set merge are required for this shadow set. The shadow member copy was `32%` complete when a merge became necessary. Because the merge was a minimerge, it preempted the copy and the status of the copy now displays as pending.

3. `$ SHOW SHADOW DSA12`

```
_DSA12:   Volume Label: DGA12
  Virtual Unit State: Merge Active (23%), Copy Pending (3%) on NODE1
  Enhanced Shadowing Features in use:
      Host-Based Minimerge (HBMM)


  VU Timeout Value      300     VU Site Value          5
  Copy/Merge Priority   5000    Mini Merge       Enabled
  Served Path Delay     0

  HBMM Policy
    HBMM Reset Threshold: 50000
    HBMM Master lists:
      Up to any 2 of the nodes: NODE1,NODE2
    HBMM bitmaps are active on the following nodes: NODE2,NODE1
  Modified blocks since bitmap creation: 254

  Device $1$DGA12               Master Member, Merging (23%)
    Read Cost             2     Site 5
    Member Timeout        180

  Device $1$DGA22               Copy Target (3%)
    Read Cost             2     Site 0
    Member Timeout        180


  Device $1$DGA32               Merging (23%)
    Read Cost             2     Site 0
    Member Timeout        180
```

This example displays status information for shadow set `DSA12`. A minimerge operation takes precedence over a copy operation, so the copy has been stopped at `3%` and the merge is now active and `23%` completed.

Note that the VU site value and the site value for device `$1$DGA12` are both `5`. Therefore, read I/O to `DSA12` on this system is directed to `$1$DGA12`.

4. `$ SHOW SHADOW /BY_PRIORITY`

```
Device    Mbr                                      Active
 Name     Cnt  Priority Virtual Unit State        on Node
```

```
_DSA30:    2     5000   Steady State
_DSA12:    3     4000   Steady State
_DSA234:         3000   Not Mounted on this node
Total of 2 Operational shadow sets; 0 in Mount Verification; 1 not
 mounted
```

The display resulting from this command shows that device DSA30 (with a priority of 5) will start a copy or merge operation on this system before DSA12 (with a priority of 4). It also shows that DSA30 is a two-member shadow set and that DSA12 is a three-member shadow set.

The last line of the display summarizes all the shadow sets and their states, including shadow sets that have exceeded their mount verification timeout limits. Operational shadow sets are shadow sets that are mounted with one or more members and that may or may not have copy or merge operations in progress. These shadow sets are available to applications for reads and writes.

5.
```
$ SHOW SHADOW /BY_PRIORITY
Device  Mbr                                                 Active
 Name   Cnt Priority Virtual Unit State                   on Node
_DSA30: 2+1   5000   Merge Pending (23%), Copy Active (5%) NODE1
_DSA12:  2    4000   Steady State
_DSA234:      3000   Not Mounted on this node
Total of 2 Operational shadow sets; 0 in Mount Verification; 1 not
 mounted
```

This display shows that a copy is being performed on system NODE1 and is 5% done. The member count (Mbr Cnt) of 2+1 indicates that there are two source members in the shadow set and another member is being added. The shadow set had completed 23% of a merge when the third member was added. When the copy operation is complete, the merge will resume from that point.

6.
```
$ SHOW SHADOW /BY_PRIORITY DSA11
Device    Mbr                                          Active
 Name     Cnt  Priority  Virtual Unit State           on Node
_DSA11:    3     5000    Merge Active (29%)             NODE3
```

This command displays the status of only DSA11, which is 29% through a merge operation on NODE3.

7.
```
$ SHOW SHADOW /MERGE

Device    Volume Name Status

_DSA1010: FOOBAR      Merge Pending (23%), Copy Active (77%) on NODE1
```

This command requests the status of a merge operation. The display provides the status of the merge, but also reveals that a copy operation is currently active.

8.
```
$ SHOW SHADOW DSA1010 /ACTIVE

Device     Volume Name   Status

_DSA1010:  FOOBAR        Merge Active (25%) on NODE1
```

The copy shown to be in progress in the previous example has completed; now the merge has resumed and is 25% done. NODE1 is doing the merge.

9.
```
$ SHOW SHADOW DSA1010
```

```
_DSA1010: Volume Label: FOOBAR
  Virtual Unit State:   Merge Pending (0%)
  Enhanced Shadowing Features in use:
      Host-Based Minimerge (HBMM)


  VU Timeout Value      3600    VU Site Value          0
  Copy/Merge Priority   5000    Mini Merge      Disabled
  Served Path Delay     30

  Device $252$DUA610            Master Member Merging (0%)
    Read Cost             2     Site 0
    Member Timeout        10

  Device $252$DUA1010     Merging (0%)
    Read Cost             2     Site 0
    Member Timeout        10

  Device $252$DUA716      Merging (0%)
    Read Cost             2     Site 0
    Member Timeout        10
```

In this example, another event has caused the set to go into a merge state. However, this set is not currently being acted on – perhaps because the system does not have any more SHADOW_MAX_COPY threads or because the relative priority of this shadow set is lower, causing other merges to be completed first. When a system is able to perform the merge, the word Active appears in the display and the Merging percent begins to increase.

10. `$ SHOW SHADOW DSA12 /POLICY=HBMM`

```
HBMM Policy for device _DSA12:
  HBMM Reset Threshold: 50000
  HBMM Master lists:
    Up to any 2 of the nodes: NODE1,NODE2,NODE3
  HBMM bitmaps are active on the following nodes: NODE2,NODE1
HBMM Reset Count      4      Last Reset  9-JAN-2004 13:40:11.45
Modified blocks since last bitmap reset: 142367
```

This command displays host-based minimerge (HBMM) information for device `DSA12`.

In this example, an HBMM policy has been defined for `DSA12`. No more than two of the systems `NODE1` and `NODE2` can have master bitmaps for `DSA12` at any time. The bitmap is reset when the bitmap has bits set to represent a total of `50000` blocks written to `DSA12`. When this command was entered, both `NODE1` and `NODE2` had master bitmaps.

Note that the "`modified blocks`" value exceeds that of the reset threshold. This can happen because the bitmap is not reset immediately upon crossing the reset threshold. The reset threshold is checked periodically, and the interval for checking the reset threshold is controlled by the system parameter SHADOW_HBMM_RTC; the default is 150 seconds.

11. `$ SHOW SHADOW/POLICY=HBMM/NAME=PEAKS_ISLAND`

```
 HBMM Policy PEAKS_ISLAND
   HBMM Reset Threshold: 50000
   HBMM Master lists:
     Up to any 2 of the nodes: NODE1,NODE2,NODE3
```

```
        Any 1 of the nodes: NODE4,NODE5
        Up to any 2 of the nodes: NODE6,NODE7,NODE8
```

This command displays the HBMM named policy `PEAKS_ISLAND`.

12. `$ SHOW SHADOW /POLICY=HBMM`

```
 HBMM Policy for device _DSA12:
   HBMM Reset Threshold: 50000
   HBMM Master lists:
     Up to any 2 of the nodes: NODE1,NODE2
   HBMM bitmaps are active on NODE1,NODE2
 Modified blocks since bitmap creation: 254

 HBMM Policy for device _DSA30:
   HBMM Reset Threshold: 50000
   HBMM Master lists:
     Up to any 2 of the nodes: FLURRY,FREEZE,HOTTUB

 HBMM Policy for device _DSA99:
   HBMM Reset Threshold: 50000
   HBMM Master lists:
     Up to any 2 of the nodes: NODE1,NODE2,NODE3
     Any 1 of the nodes: NODE4,NODE5
     Up to any 2 of the nodes: ICEBAR,SWEDEN,MILANO

 HBMM Policy for device _DSA999:
   HBMM Reset Threshold: 50000
   HBMM Master lists:
     Up to any 2 of the nodes: NODE1,NODE2,NODE3
     Any 1 of the nodes: NODE4,NODE5
     Up to any 2 of the nodes: ICEBAR,SWEDEN,MILANO
```

The command in this example displays a list of all shadow sets that have HBMM policies.

13. `$ SHOW SHADOW /POLICY=HBMM /NAME`

```
HBMM Policy PEAKS_ISLAND
  HBMM Reset Threshold: 50000
  HBMM Master lists:
    Up to any 2 of the nodes: NODE1,NODE2,NODE3
    Any 1 of the nodes: NODE4,NODE5
    Up to any 2 of the nodes: NODE6,NODE7,NODE8

HBMM Policy POLICY_1
  HBMM Reset Threshold: 50000
  HBMM Master lists:
    Up to any 2 of the nodes: NODE1,NODE2,NODE3
    Any 1 of the nodes: NODE4,NODE5

HBMM Policy ICE_HOTELS
  HBMM Reset Threshold: 50000
    Up to any 2 of the nodes: QUEBEC,SWEDEN,FINLND
    Any 1 of the nodes: ALASKA,GRNLND
```

The command in this example displays a list of all named HBMM policies on the cluster.

14. `$ SHOW SHADOW DSA12 /OUTPUT=DSA12_OUTPUT.LIS`

The output of this **SHOW SHADOW** command is directed to the file named DSA12_OUTPUT.LIS.

15. `$ SHOW SHADOW DSA12 /FULL`

```
_DSA12:   VolumeLabel: DGA12
  Virtual Unit State:  Steady State
  Enhanced Shadowing Features in use:
      Host-Based Minimerge (HBMM)


  Cluster Virtual Unit Status: 0001 - normal
  Local Virtual Unit Status: 00000110 - Local Read,Policy Active

  Total Devices         2       VU_UCB          81435180
  Source Members        2       SCB LBN         003FEB5F
  Act Copy Target       0       Generation      00A2BA31
  Act Merge Target      0        Number         441DA800
  Last Read Index       0       Master Mbr Index       0
  Copy Hotblocks        0       Copy Collisions        0
  SCP Merge Repair Cnt  0       APP Merge Repair Cnt   0
  VU Timeout Value      300     VU Site Value          0
  Copy/Merge Priority   5000    Mini Merge       Enabled
  Served Path Delay     0

  HBMM Policy
    HBMM Reset Threshold: 50000
    HBMM Master lists:
      Up to any 2 of the nodes: XYZ100,XYZ200
    HBMM bitmaps are active on the following nodes: XYZ200,XYZ100
  Modified blocks since bitmap creation: 254

  Device $1$DGA12               Master Member
    Index 0 Status  000000A0    src,valid
    Ext. Member Status    00
    Read Cost             2     Site 0
    Member Timeout        180   UCB  81643840

  Device $1$DGA22
    Index 1 Status  000000A0    src,valid
    Ext. Member Status    00
    Read Cost             2     Site 0
    Member Timeout        180   UCB  81642840


  Device $1$DGA32
    Index 1 Status 000000A0     src,valid
    Ext. Member Status    00
    Read Cost             2     Site  0
    Member Timeout        180   UCB  81641840
```

This example shows a full information display for device DSA12.

16. `$ SHOW SHADOW DSA0 /FULL`

```
_DSA0:    Volume Label: TOPAZ_XA7V
```

```
  Virtual Unit State:   Steady State
  Cluster Virtual Unit Status: 0001 - normal
  Local Virtual Unit Status: 00000110 - enforce_local_read,
hbmm_eval_policy_enabled

  Enhanced Shadowing Features in use:
        Dissimilar Device Shadowing (DDS)
        Host-Based Minimerge (HBMM)

  Total Devices             2    VU_UCB          817AFD80
  Source Members            2    SCB LBN         001F5C50
  Act Copy Target           0    Generation      00A31141
  Act Merge Target          0     Number         C6925F29
  Last Read Index           0    Master Mbr Index       0
  Copy Hotblocks            0    Copy Collisions        0
  SCP Merge Repair Cnt      0    APP Merge Repair Cnt   0
  VU Timeout Value       3600    VU Site Value          0
  Copy/Merge Priority    5000    Mini Merge       Enabled
  Served Path Delay        30

  HBMM Policy
    HBMM Reset Threshold: 50000
    HBMM Master lists:
      Up to any 6 nodes in the cluster
    HBMM bitmaps are active on ABCD2,ABCD1
    Modified blocks since bitmap creation: 254

  Device $252$DKA0              Master Member
    Index 0 Status  000000A0    mbr_src,mbr_valid
    Ext. Member Status    00
    Read Cost              2    Site 0
    Member Timeout       120    UCB  8163D880

  Device $252$DKA100
    Index 1 Status  000000A0    mbr_src,mbr_valid
    Ext. Member Status    00
    Read Cost              2    Site 0
    Member Timeout       120    UCB  8163E880
```

This example shows sample output when both HBMM and Dissimilar Device Shadowing are in use.

SHOW STATUS

SHOW STATUS — Displays the current status of your process.

Format

SHOW STATUS

Parameters

None.

Description

The **SHOW STATUS** command verifies that your current process is running. The **SHOW STATUS** information can also indicate if the process is using an abnormal amount of CPU time, is opening too many files, or is accruing too many page faults. The information displayed includes the following:

- Current time and date
- Elapsed CPU time used by the current process
- Buffered I/O count
- Current working set size
- Open file count
- Direct I/O count
- Current amount of physical memory occupied
- Number of page faults

The information displayed by **SHOW STATUS** is similar to that displayed by the **Ctrl/T** function. (See the **SET CONTROL** command.)

Example

```
$ SHOW STATUS
Status on 14-DEC-2001 12:56:48.68    Elapsed CPU :   0 00:00:55.02
Buff. I/O :    5117    Cur. ws. :    300        Open files :     1
Dir. I/O :      458    Phys. Mem. :  162        Page Faults : 8323
```

The **SHOW STATUS** command in this example displays the current status of your process.

SHOW SYMBOL

SHOW SYMBOL — Displays the value of the specified symbol.

Format

SHOW SYMBOL [*symbol-name*]

Parameter

symbol-name

Specifies the name of the symbol whose value you want to display. You must specify a symbol name unless you use the **/ALL** qualifier. The asterisk (*) and the percent sign (%) wildcard characters are allowed in the *symbol-name* parameter.

Note

When you specify a wildcard character, the command searches only the local symbol table of the current command level and the global symbol table (as if you had specified **/LOCAL/GLOBAL**). You can search symbol tables of preceding command levels by symbol name, but not by wildcard.

Description

The **SHOW SYMBOL** command searches for the specified symbol name in several places. First, it looks in the local symbol table of the current command level. Next, it searches the local symbol tables of preceding command levels (unless a wildcard was specified). Last, it searches the global symbol table. The **SHOW SYMBOL** command displays the first match it finds.

The **/LOCAL** and **/GLOBAL** qualifiers override the search order.

Qualifiers

/ALL

Displays the current values of all symbols in the specified symbol table (**/LOCAL** or **/GLOBAL**). If you specify the **/ALL** qualifier and do not specify either the **/LOCAL** or the **/GLOBAL** qualifier, the **SHOW SYMBOL** command displays the contents of the local symbol table for the current command level.

/GLOBAL

Searches only the global symbol table for the specified symbol name. If you specify both the **/ALL** and **/GLOBAL** qualifiers, all names in the global symbol table are displayed.

/LOCAL

Searches only the local symbol table for the current command level for the specified symbol name. If you specify both the **/ALL** and **/LOCAL** qualifiers, all names in the local symbol table for the current command level are displayed.

/LOG (default)
/NOLOG

Controls whether the system generates an informational message if the symbol value has been truncated. The value is truncated if it exceeds 255 characters.

Examples

1.
```
$ SHOW SYMBOL PURGE
  PURGE = "PURGE/KEEP=2"
```

   The **SHOW SYMBOL** command in this example displays the current value of the symbol name PURGE. The command interpreter first searches the local symbol table for the current command level, then local symbol tables for preceding command levels, and finally the global symbol table. The single equal sign (=) following PURGE means it is a local symbol.

2.
```
$ SHOW SYMBOL/GLOBAL/ALL
    $FACILITY == "%X00000000"
    $IDENT == "%X00000000"
    $RESTART == "FALSE"
    $SEVERITY == "1"
    $STATUS == "%X10000001"
    LOG == "@LOG"
    TIME == "SHOW TIME"
```

The **SHOW SYMBOL** command in this example displays all the symbols defined in the global symbol table. Note that the symbols $FACILITY, $IDENT, $RESTART, $SEVERITY, and $STATUS, which are maintained by the system, are also displayed.

3.
```
$ SHOW SYMBOL/LOCAL TIME
%DCL-W-UNDSYM, undefined symbol
```

The **SHOW SYMBOL** command in this example searches only the local symbol table for the symbol TIME. The response indicates that TIME currently has no value.

SHOW SYSTEM

SHOW SYSTEM — Displays status information about current processes.

Format

SHOW SYSTEM

Parameters

None.

Description

The **SHOW SYSTEM** command displays the following information about processes on the system:

Time
Process name and identification
Processing state
Priority
Total process I/O count
Cumulative processor time used
Cumulative page faults
Amount of physical memory being used
Type of process

It also checks to see whether the machine is a multiprocessor. A machine is considered a multiprocessor if the following conditions are true:

- The system parameter MULTIPROCESSING is not equal to zero.
- The machine accommodates more than one CPU.

If the machine is a multiprocessor, the **SHOW SYSTEM** command includes information about the multiprocessing environment. Each process in the currently executing state (indicated by the CUR symbol) reveals which processor it is running on. It does this by including a number beside the CUR symbol.

The **SHOW SYSTEM** command examines the processes on the system without stopping activity on the system. This means that process information may change during the time that **SHOW SYSTEM** collects the data to be displayed. As a result, there may be minor inconsistencies in the **SHOW SYSTEM** display. For example, **SHOW SYSTEM** may display two processes that schedule state CUR on the same CPU.

Qualifiers

/BATCH (default)
/NOBATCH

Displays all batch jobs for the local system. When used with the **/CLUSTER** qualifier, the **/BATCH** qualifier displays all batch jobs in the OpenVMS Cluster environment. When you use the **/NOBATCH** qualifier, all batch jobs are excluded from the display.

/CLUSTER

Displays the specified processes on all nodes in an OpenVMS Cluster.

/EXACT

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify a search string that must match the search string exactly and must be enclosed with quotation marks (" ").

If you specify the **/EXACT** qualifier without the **/SEARCH** qualifier, exact search mode is enabled when you set the search string with the Find (**E1**) key.

/FULL

Displays the user identification code (UIC), the processor type, and the process size, in addition to the default information. The UIC is displayed below the process name.

/GRAND_TOTAL
/NOGRAND_TOTAL (default)

Displays a trailing line with the total of all processes matching the selection criteria and the total number of all processes on all systems displayed by the command. (See also the **/TOTAL** qualifier.)

/HEADING
/NOHEADING

Displays a heading line above the system output. When you use the **/NOHEADING** qualifier, the heading line is excluded from the display.

/IMAGE

Lists all processes with their current image. (Requires WORLD privilege.)

/HIGHLIGHT[=*keyword*]

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify the type of highlighting you want when a search string is found. When a string is found, the entire line is highlighted. You can use the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting.

/IDENT=*pid*

Displays all processes on the system that have the specified process identification (PID) code number.

/INTERACTIVE
/NOINTERACTIVE

Displays all interactive processes on the system. When you use the **/NOINTERACTIVE** qualifier, all interactive processes are excluded from the display.

/MULTITHREAD

Displays all multithreaded processes for the local system. When used with the **/FULL** qualifier, the **/MULTITHREAD** qualifier displays information about each kernel thread in a multithreaded process.

For more information on kernel threads, see the relevant section in the *VSI OpenVMS Programming Concepts Manual* [https://docs.vmssoftware.com/vsi-openvms-programming-concepts-manual-volume-i/#KTHREADS_STRUCTURE].

/NETWORK
/NONETWORK

Displays all network processes on the system. When you use the **/NONETWORK** qualifier, all network processes are excluded from the display.

/NODE[=(*name*,...)]

Displays all the processes on the specified node or nodes. If you enter the **/NODE** qualifier without a value, the qualifier displays all the processes on the local node of an OpenVMS Cluster environment.

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the command is sent. By default, the output of the **SHOW SYSTEM** command is sent to the current SYS$OUTPUT device (usually your terminal). To send the output to a file, use the **/OUTPUT** qualifier followed by a file specification.

The file specification may not include the asterisk (*) and the percent sign (%) wildcard characters. If you enter a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS is the default file type.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

/OWNER_UIC=*uic*

Displays all processes on the system that have the specified user identification code (UIC) number.

/PAGE[=*keyword*]
/NOPAGE (default)

Controls the display of system information on the screen.

You can use the following keywords with the **/PAGE** qualifier:

Keyword	Description
CLEAR_SCREEN	Clears the screen before each page is displayed.
SCROLL	Displays information one line at a time.
SAVE[=*n*]	Enables screen navigation of information, where *n* is the number of pages to store.

The **/PAGE=SAVE** qualifier allows you to navigate through screens of information. The **/PAGE=SAVE** qualifier stores up to 5 screens of up to 255 columns of information. When you use the **/PAGE=SAVE** qualifier, you can use the following keys to navigate through the information:

Key Sequence	Description
Up arrow key, **Ctrl/B**	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.
Right arrow key	Scroll right one column.
Find (**E1**)	Specify a string to find when the information is displayed.
Insert Here (**E2**)	Scroll right one half screen.
Remove (**E3**)	Scroll left one half screen.
Select (**E4**)	Toggle 80/132 column mode.
Prev Screen (**E5**)	Get the previous page of information.
Next Screen (**E6**), **Return**, **Enter**, **Space**	Get the next page of information.
F10, **Ctrl/Z**	Exit. (Some utilities define these differently.)
Help (**F15**)	Display utility help text.
Do (**F16**)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.

The **/PAGE** qualifier is not compatible with the **/OUTPUT** qualifier.

/PROCESS[=*name*] (default)
/NOPROCESS

Displays all specified processes on the system. If you do not specify a process name with the **/PROCESS** qualifier, all processes are displayed.

When you use the **/NOPROCESS** qualifier with the **/CLUSTER** qualifier, only the node or nodes available on the cluster are displayed with their respective uptimes and operating system version.

You can include the asterisk (*) and the percent sign (%) wildcard characters in the process name.

/SCHEDULING_CLASS[=*class_name*]
/NOSCHEDULING_CLASS

Displays processes that belong to a specific scheduling class (*class_name*). If the class name is not specified, all class scheduled processes are displayed along with the name of their scheduling class.

/SEARCH="*string*"

Use with the **/PAGE=SAVE** qualifier to specify a string that you want to find in the information being displayed. Quotation marks are required for the **/SEARCH** qualifier, if you include spaces in the text string.

You can also dynamically change the search string by pressing the Find key (**E1**) while the information is being displayed. Quotation marks are not required for a dynamic search.

/STATE=*process-state*

Displays all processes with the specified process state. You can include the asterisk (*) and the percent sign (%) wildcard characters in the process-state name. The following process states are available:

CEF (Common event flag wait)	COLPG (Collided page wait)
COM (Computable)	COMO (Computable but outswapped)
CUR (Current)	FPG (Free page wait)
HIB (Hibernating)	HIBO (Hibernating and outswapped)
LEF (Local event flag wait)	LEFO (Local event flag wait and outswapped)
MUTEX[1] (Miscellaneous wait)	PFW (Page fault wait)
PSXFR (POSIX fork wait)	RWAST (AST wait)
RWBRK (Waiting for BROADCAST to finish)	RWCAP (CPU capability required)
RWCLU (Cluster transition wait)	RWCSV (Cluster server)
RWIMG (Image activation lock)	RWLCK (Lock ID database)
RWMBX (Mailbox full)	RWMPB (Modified page writer busy)
RWMPE (Modified page list empty)	RWNPG (Nonpaged pool)
RWPAG (Paged pool)	RWPFF (Page file full)
RWQUO (Pooled quota)	RWSCS (SCS wait)
RWSWP (Swapfile space list empty)	SUSP (Suspended)
SUSPO (Suspended and outswapped)	

[1]MUTEX is output if one of the RW* states cannot be output.

/SUBPROCESS
/NOSUBPROCESS

Displays all subprocesses on the system. When you use the **/NOSUBPROCESS** qualifier, all subprocesses are excluded from the output.

/TOTAL
/NOTOTAL (default)

Displays a trailer after the system process list containing the total number of processes matching the selection criteria and the total number of processes on a system.

/WRAP
/NOWRAP (default)

Use with the **/PAGE=SAVE** qualifier to limit the number of columns to the width of the screen and to 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 and can be seen when you use the scrolling (left and right) features provided by the **/PAGE=SAVE** qualifier.

Examples

1.
```
$ SHOW SYSTEM
OpenVMS 7.3 on node KRYPTN 14-DEC-2001 17:45:47:78 Uptime 2 21:53:59
   Pid    Process Name State  Pri    I/O       CPU   Page flts Pages
 27400201 SWAPPER        HIB   16      0  0 00:29:52.05      0    0
 27401E03 DOCBUILD       LEF    4  37530  0 00:05:47.62  96421  601
 27402604 BATCH_789      LEF    4   3106  0 00:00:48.67   4909 2636 B
 27401C05 BATCH_60       LEF    6    248  0 00:00:06.83   1439 1556 B
 27400207 ERRFMT         HIB    8   6332  0 00:00:41.83     89  229
 27400208 CACHE_SERVER   HIB   16   2235  0 00:00:05.85     67  202
 27400209 CLUSTER_SERVER HIB    8   4625  0 00:22:13.28    157  448
 2740020C JOB_CONTROL    HIB   10 270920  0 01:07:47.88   5163 1384
 2740020D CONFIGURE      HIB    9    125  0 00:00:00.53    104  264
  .
  .
  .

 27400E8D Sir Lancelot   LEF    5    226  0 00:00:07.87   4560  697
 2740049A Guenevere      LEF    4    160  0 00:00:02.69    534  477
 27401EA0 BATCH_523      CUR 4  4  17470  0 03:25:49.67   8128 5616 B
 274026AF KING           CUR 6  4  14045  0 00:02:03.24  20032  397
 274016D5 DAVIS          LEF    6    427  0 00:00:09.28   5275 1384
 27401ED6 deGroot        HIB    5    935  0 00:00:10.17   3029 2204 S
 274012D7 BATCH_689      LEF    4  49216  0 00:14:18.36   7021 3470 B
 274032D9 DECW$MAIL      LEF    4   2626  0 00:00:51.19   4328 3087 B
 274018E3 SERVER_0021    LEF    6    519  0 00:00:07.07   1500  389 N
 274016E8 NMAIL_0008     HIB    4  10955  0 00:00:55.73   5652  151
 274034EA DOERING        LEF    4   2132  0 00:00:23.85   5318  452
 274022EB KERR           CUR 6  4    492  0 00:00:12.15   5181  459
 274018EF DUFRESNE       LEF    5 121386  0 00:28:00.97   7233 4094
 27401AF0 EMACS$RTA43    LEF    4  14727  0 00:03:56.54   8411 4224 S
 27400CF4 EAGAN          HIB    5  25104  0 00:06:07.76  37407 1923
 274020F5 TAN            LEF    7  14726  0 00:02:10.74  34262 1669
 27400CF6 KELLEY         LEF    9  40637  0 00:05:15.63  18454  463
The SHOW SYSTEM command in this example was issued from a VAX system and shows
all processes on the node EON.
```

The **SHOW SYSTEM** command in this example displays all processes on the system.

The information in this example includes the following:

- Process identification (PID) code – A 32-bit binary value that uniquely identifies a process.
- Process name – A 1 to 15-character string used to identify a process.
- Process state – The activity level of the process, such as COM (computing), HIB (hibernation), LEF (local event flag) wait, or CUR (if the process is current). If a multiprocessing environment exists, the display shows the CPU ID of the processor on which any current process is executing.

  Note that the **SHOW SYSTEM** command examines the processes on the system without stopping activity on the system. In this example, process information changed during the time that the **SHOW SYSTEM** command collected the data to be displayed. As a result, this display includes two processes, named GAWAIN and S. Whiplash, with the state CUR on the same CPU, CPU ID 6 in the example.
- Current priority[1] – The priority level assigned to the process (the higher the number, the higher the priority).

- Total process I/O count[1]– The number of I/O operations the process has used thus far. This consists of both the direct I/O count and the buffered I/O count.
- Charged CPU time[1] – The amount of CPU time that a process has used so far.
- Number of page faults[1] – The number of exceptions generated by references to pages that are not in the process's working set.
- Pages[1] – The number of CPU-specific pages in physical memory that the process is currently occupying.
- Process indicator – Letter B indicates a Batch Job; letter S indicates a Sub Process; letter N indicates a Network Process. If the letter M precedes B, S, or N, it indicates that it is a multithreaded process.
- User identification code (UIC) – An 8-digit octal number assigned to a process. This number is displayed only if the **/FULL** qualifier is specified.

2.
```
$ SHOW SYSTEM /CLUSTER
OpenVMS V7.3 on node ALPES 14-DEC-2001 09:09:58.61  Uptime 2 2:27:11
Pid       Process Name   State  Pri I/O       CPU     Page flts Pages
31E00041  SWAPPER        HIB    16    0  0 00:00:02.42       0     0
31E00047  CACHE_SERVER   HIB    16   58  0 00:00:00.26      80    36
31E00048  CLUSTER_SERVER CUR     9  156  0 00:00:58.15    1168    90
31E00049  OPCOM          HIB     7 8007  0 00:00:33.46    5506   305
31E0004A  AUDIT_SERVER   HIB     9  651  0 00:00:21.17    2267    22
31E0004B  JOB_CONTROL    HIB    10 1030  0 00:00:11.02     795   202
   .
   .
   .
```

The **SHOW SYSTEM** command in this example is taken from an Alpha system and shows all processes on all nodes of the cluster.

3.
```
$ SHOW SYSTEM /NODE=ALTOS
OpenVMS V7.3-2  on node ALTOS  15-SEP-2009 10:26:49.59  Uptime  73 13:53:35
  Pid    Process Name    State  Pri      I/O       CPU       Page flts  Pages
24200201 SWAPPER         HIB     16        0   0 00:02:41.09         0      0
24200207 CLUSTER_SERVER  CUR  7  13    19735   0 01:21:54.29        95    115
24200208 SHADOW_SERVER   HIB      6  5445809   0 00:04:53.18        68    122
24200209 CONFIGURE       HIB     10       23   0 00:04:20.35        42     30
2420020A USB$UCM_SERVER  HIB      6      165   0 00:00:02.72       164    225
2420020B LANACP          HIB     12       79   0 00:00:00.01       108    138
2420020D FASTPATH_SERVER HIB     10        9   0 00:00:00.01        76     96
2420020E IPCACP          HIB     10       11   0 00:00:06.88        35     51
2420020F ERRFMT          HIB      7   210219   0 00:00:49.99       113    138
24200210 CACHE_SERVER    HIB     16    18716   0 00:00:01.33        31     46
```

The **SHOW SYSTEM** command in this example is taken from an Alpha system and shows all processes on the node ALTOS.

4.
```
$ SHOW SYSTEM/NOPROCESS/CLUSTER/FULL
OpenVMS V7.3-2  on node ALTOS   6-AUG-2009 14:31:26.31  Uptime  33 17:58:12
        hp AlphaServer GS1280 7/1150

OpenVMS V7.3-2  on node ARUSHA   6-AUG-2009 14:31:26.32  Uptime  32 21:42:11
```

[1]This information is displayed only if the process is currently in the balance set; if the process is not in the balance set, these columns contain the following message:

```
-- swapped out --
```

```
        hp AlphaServer GS1280 7/1150

OpenVMS V8.3-1H1  on node BANFF   6-AUG-2009 14:31:26.33  Uptime  33 17:44:02
        HPE rx2600  (1.30GHz/3.0MB)

OpenVMS V7.3-2  on node CHOBE   6-AUG-2009 14:31:26.33  Uptime  32 21:14:00
        hp AlphaServer GS1280 7/1150
```

This example displays only the nodes available on the cluster with their respective uptimes and operating system versions because the **/NOPROCESS** qualifier suppresses the display of processes. The **/FULL** qualifier also displays the type of hardware on which the system is running.

5.
```
$ SHOW SYSTEM/PROCESS=S*
OpenVMS V7.3 on node AAAAAA 12-MAY-2001 10:30 AM  Uptime 3 12:43:55
  Pid    Process Name  State   Pri  I/O      CPU      Page flts Pages
20C00101 SWAPPER         HIB    16    0  0 00:00:00.05        0     0
20C0010E SHADOW_SERVER   HIB     6    7  0 00:00:00.03       32     4
20C0010F SECURITY_SERVER HIB    10   35  0 00:00:02.16      292   323
20C00110 SMISERVER       HIB     9   67  0 00:00:00.20      152   131
20C0011A STEFANELLI      CUR 0   4   63  0 00:00:00.29      538    55
```

This example displays only the processes on the system that begin with the letter `S`.

6.
```
$ SHOW SYSTEM/NOHEADING/OUTPUT=SYSPROC.DAT
20C00101 SWAPPER         HIB    16    0  0 00:00:00.05        0     0
20C00105 CONFIGURE       HIB    10   24  0 00:02:00.86       41    21
20C00107 IPCACP          HIB    10    9  0 00:00:00.04       33    43
20C00108 ERRFMT          HIB     7 3583  0 00:00:04.31       50    61
20C00109 CACHE_SERVER    HIB    16    3  0 00:00:00.02       27    35
20C0010A CLUSTER_SERVER  HIB    10   11  0 00:00:00.10       59    75
```

This example places the display of system processes into the `SYSPROC.DAT` file without the heading information using the **/OUTPUT** qualifier. In this way, you can create a data file for processing because the heading is suppressed using the **/NOHEADING** qualifier.

7.
```
$ SHOW SYSTEM/IDENT=20C0010D/FULL
OpenVMS V7.3 on node AAAAAA 12-MAY-2001 10:35 AM  Uptime 3 12:48:19
        DEC 4000 Model 610
  Pid    Process Name State  Pri   I/O      CPU      Page flts  Pages
20C0010D JOB_CONTROL   HIB    10    41  0 00:00:00.25       51     66
         [SYSTEM]                                              528Kb
```

This example displays a system process by PID number.

8.
```
$ SHOW SYSTEM/OWNER=TIMMER
OpenVMS V7.3 on node AAAAAA 12-MAY-2001 10:34 AM  Uptime 3 12:47:37
  Pid    Process Name State  Pri   I/O      CPU      Page flts  Pages
20C00119 DECW$TE_0119  LEF     6   334  0 00:00:01.00      975    657
20C0011A TIMMER        LEF     8    63  0 00:00:00.29      538     55
20C0011B _RTA1:        CUR 0   4   554  0 00:00:01.72     1771     70
```

This example displays the system processes belonging to user `TIMMER`.

9.
```
$ SHOW SYSTEM/STATE=L*
OpenVMS V7.3 on node AAAAAA 12-MAY-2001 10:36 AM  Uptime 3 12:49:46
  Pid    Process Name State  Pri   I/O      CPU      Page flts  Pages
20C00119 DECW$TE_0119  LEF     6   334  0 00:00:01.00      975    657
20C0011A STADLER       LEF     8    63  0 00:00:00.29      538     55
```

This example displays the system processes with process states that begin with the letter `L`.

10.
```
$ SHOW SYSTEM/MULTITHREAD
OpenVMS V7.3 on node ZIMBER 30-OCT-2001 11:34:11.51 Uptim  8 20:21:31
```

```
  Pid    Process Name    State Pri  I/O      CPU    Page flts  Pages
0000008C SECURITY_SERVER HIB    11  131  0 00:55:25.25   4370   194 M

$ SHOW SYSTEM/MULTITHREAD/FULL
OpenVMS V7.3 on node ZIMBER 30-OCT-2001 11:34:13.98 Uptime 8 20:21:34
        AlphaStation 400 4/233
  Pid    Process Name    State Pri  I/O      CPU    Page flts Pages
0000008C SECURITY_SERVER HIB    11  131  0 00:55:25.26   4370   194 M
0000010C                 HIB    11
         [SYSTEM]                                          1552Kb
```

These examples show the results of the **/MULTITHREAD** qualifier and the combination results of the **/MULTITHREAD** and **/FULL** qualifiers.

11.
```
$ SHOW SYSTEM/CLUSTER/TOTAL/SUBPROCESS/GRAND_TOTAL
OpenVMS V7.2-2  on node CHEER  27-JAN-2003 16:09:26.94  Uptime  3 23:20:43
  Pid    Process Name State  Pri     I/O      CPU       Page flts  Pages
262000AC DTGREET      LEF      4     757  0 00:00:06.76       476    493  S

Total of 1/31 total processes.


OpenVMS V7.3-2  on node GATOR  27-JAN-2003 16:09:26.95  Uptime  1 14:41:58
  Pid    Process Name State  Pri     I/O      CPU       Page flts  Pages
27000428 DTWM         LEF      4   26422  0 00:00:11.61       715    846  S
270004B9 SYSTEM_141   HIB      1     155  0 00:00:00.12       351    393  S

Total of 2/37 total processes.


OpenVMS V7.3-1  on node FISHER 27-JAN-2003 16:09:26.97  Uptime  5 02:16:57
  Pid    Process Name State  Pri     I/O       CPU       Page flts  Pages
25A00149 DTWM         LEF      4  395380  0 00:12:51.27       712    837  S
25A010AE SYSTEM_76    HIB      1     209  0 00:00:00.66       339    381  S

Total of 2/34 total processes.


OpenVMS V7.1-2  on node MONKEY 27-JAN-2003 16:09:27.05  Uptime  1 14:56:02
  Pid    Process Name State  Pri     I/O      CPU       Page flts  Pages
26C00130 DTWM         LEF      6    5868  0 00:00:11.84       642    777  S

Total of 1/31 total processes.


OpenVMS V7.3-2  on node QUERY  27-JAN-2003 16:09:27.10  Uptime  3 22:16:23
  Pid    Process Name State  Pri     I/O      CPU       Page flts  Pages
26601139 SYSTEM_203   HIB      2     157  0 00:00:00.68       346    387  S
2660013A DTWM         LEF      4  275709  0 00:10:24.99       728    852  S

Total of 2/37 total processes.

Grand total of 8/183 processes on 6 nodes.
```

The **SHOW SYSTEM** command in this example displays the total count of subprocesses and processes on each node of a cluster. The last line lists the grand total of all subprocesses in the cluster (8) followed by the grand total of all processes in the cluster (183).

SHOW TERMINAL

SHOW TERMINAL — Displays the current characteristics of a specific terminal. Each characteristic corresponds to an option of the **SET TERMINAL** command.

Format

SHOW TERMINAL [*device-name*[:]]

Parameter

***device-name*[:]**

Specifies the name of the terminal for which you want the characteristics displayed. The default is your terminal (SYS$COMMAND).

Description

The **SHOW TERMINAL** command displays the information about terminal settings for such things as terminal speed, width, number of lines, graphics, and device type.

Note that the **SHOW TERMINAL** command does not describe terminal fallback characteristics if any are activated. If the Terminal Fallback Facility (TFF) is enabled, you can invoke the Terminal Fallback Utility (TFU) and issue the subcommand **SHOW TERMINAL/FALLBACK**. For more information, see the *OpenVMS Terminal Fallback Utility Manual*.

Qualifiers

/BRIEF

Displays a subset of more commonly used terminal characteristics.

/FULL (default)

Displays all terminal characteristics.

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter the **/OUTPUT** qualifier without a file specification, the output is sent to the current process default output stream or device, identified by the logical name SYS$OUTPUT.

If you enter the **/OUTPUT** qualifier with a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS the default file type. If you enter a file specification, it cannot include the asterisk (*) and the percent sign (%) wildcard characters.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

/PERMANENT

Requires LOG_IO (logical I/O) or PHY_IO (physical I/O) privilege.

Displays the permanent characteristics of the terminal.

Examples

1.
```
$ SHOW TERMINAL/BRIEF
  Terminal: _RTA1:      Device_Type: VT300_Series  Owner: _RTA1:
```

```
                                                  Username: REHOR

Terminal Characteristics:
   Speed:   9600      Page:  62          Width: 80
   No Remote          Hostsync           TTsync
   No Modem           Broadcast          Wrap
   No Hangup          No Disconnect      Overstrike editing
   No Dialup          Application keypad Line Editing
```

In this example, the **SHOW TERMINAL** command displays a subset of the more commonly used terminal characteristics.

2.
```
$ SHOW TERMINAL/FULL
Terminal: _TTE4:       Device_Type: VT102         Owner: FRANKLIN
Physical Terminal: _LTA49
   Input:   9600      LFfill:  0      Width:  80      Parity: None
   Output:  9600      CRfill:  0      Page:   24
Terminal Characteristics:
 Interactive        Echo               Type_ahead    No Escape
 No Hostsync        TTsync             Lowercase     Tab
 Wrap               Scope              No Remote     Eightbit
 Broadcast          No Readsync        No Form       Fulldup
 No Modem           No Local_echo      No Autobaud   Hangup
 No Brdcstmbx       No DMA             No Altypeahd  Set_speed
 Line Editing       Overstrike editing No Fallback   No Dialup
 No Secure server   No Disconnect      No Pasthru    No Syspassword
 No SIXEL Graphics  Soft Characters    Printer port  Numeric Keypad
 ANSI_CRT           No Regis           No Block_mode Advanced_video
 Edit_mode          DEC_CRT            DEC_CRT2      No DEC_CRT3
```

In this example, the **SHOW TERMINAL** command displays the characteristics of this specific terminal. If you are displaying statistics about a terminal allocated to another user, the input, output, LFfill, CRfill, width, page, and parity statistics are not shown (see the next example).

3.
```
$ SHOW TERMINAL/FULL
Terminal: _RTA1:      Device_Type: VT300_Series  Owner: _RTA1:
                                                 Username: VILLA
Remote Port Info: NODE12::VILLA

   Input:    9600     LFfill:  0      Width:  80      Parity: None
   Output:   9600     CRfill:  0      Page:   62

Terminal Characteristics:
 Interactive        Echo             Type_ahead         No Escape
 Hostsync           TTsync           Lowercase          Tab
 Wrap               Scope            No Remote          Eightbit
 Broadcast          No Readsync      No Form            Fulldup
 No Modem           No Local_echo    No Autobaud        No Hangup
 No Brdcstmbx       No DMA           No Altypeahd       Set_speed
 No Commsync        Line Editing     Overstrike editing No Fallback
 No Dialup          No Secure server No Disconnect      No Pasthru
 No Syspassword     SIXEL Graphics   No Soft Characters Printer port
 Application keypad ANSI_CRT         Regis              No Block_mode
 Advanced_video     Edit_mode        DEC_CRT            DEC_CRT2
 DEC_CRT3           No DEC_CRT4      No DEC_CRT5        Ansi_Color
 VMS Style Input    <CTRL-H> Delete
```

In this example, the **SHOW TERMINAL** command displays the full terminal characteristics.

SHOW TIME

SHOW TIME — Displays the current date and time.

Format

SHOW [DAY]TIME

Parameters

None.

Example

```
$ SHOW TIME
   3-NOV-2001 00:02:00
```

The **SHOW TIME** command in this example displays the current date and time.

SHOW TRANSLATION

SHOW TRANSLATION — Displays the first translation found for the specified logical name. You can specify the tables that are searched. **Requires read (R) access to a logical name table to display information about any logical name cataloged in that table.**

Format

SHOW TRANSLATION *logical-name*

Parameter

logical-name

Specifies the logical name whose translation you want to display.

Description

The **SHOW TRANSLATION** command searches one or more logical name tables for a specified logical name and returns the equivalence name of the first match found. You can specify the tables you want to search. If you do not specify a table, **SHOW TRANSLATION** searches the tables defined by the logical name LNM$DCL_LOGICAL.

The logical name LNM$DCL_LOGICAL contains the list of logical name tables and the order in which they are searched. Unless LNM$DCL_LOGICAL has been redefined for your process, the process, job, group, and system logical name tables are searched, in that order. The first match found is displayed. (To see how LNM$DCL_LOGICAL is defined for your process, enter the command **SHOWLOGICAL /TABLE=LNM$DIRECTORIES LNM$DCL_LOGICAL**.)

If a table contains more than one entry with the same name, but each name has a different mode, then the translation for the name with the outermost (least privileged) mode is returned.

The **SHOW TRANSLATION** command is similar to the **SHOW LOGICAL** command; however, the **SHOW TRANSLATION** command is executed within the DCL command interpreter (the

SHOW LOGICAL command calls an image). Therefore, the **SHOW TRANSLATION** command does not cause the current image to exit and does not deassign user mode logical names. Also, the **SHOW TRANSLATION** command does not display iterative translations of a name.

Qualifier

/TABLE=*name*

Searches the specified table. The default value for the *name* parameter is LNM$DCL_LOGICAL.

If you specify the table name by using a logical name that translates to more than one table, then each table is searched in the order specified until a match is found.

Examples

1.
```
$ SHOW TRANSLATION PAYROLL
PAYROLL = DISK1:[ACCOUNTS.WORKING]FACTOR1.DAT;37 (LNM$PROCESS_TABLE)
```

   The **SHOW TRANSLATION** command in this example displays the translation for the logical name PAYROLL and also displays the name of the table where the logical name was found. In this example, PAYROLL was found in LNM$PROCESS_TABLE, the process logical name table.

2.
```
$ DEFINE DISK  DKA1:
$ DEFINE/GROUP DISK  DKA2:
$ SHOW TRANSLATION DISK
  DISK = DKA1:(LNM$PROCESS_TABLE)
```

   The **DEFINE** commands in this example place entries for the logical name DISK in both the process and group logical name tables. Then, the **SHOW TRANSLATION** command shows the translation associated with the logical name DISK. By default, the process, job, group, and system tables are searched (in that order). The first match found is displayed. The logical name DISK from the process table (LNM$PROCESS_TABLE) is displayed because it is found before the name DISK in the group table.

3.
```
$ RUN ORION
Ctrl/Y
$ SHOW TRANSLATION TERMINAL
  TERMINAL = _TTT3: (LNM$PROCESS_TABLE)
$ CONTINUE
```

   The **RUN** command in this example executes the image ORION.EXE. After the **Ctrl/Y** function interrupts the image, the **SHOW TRANSLATION** command displays a logical name assignment. The **CONTINUE** command resumes the execution of the image.

4.
```
$ SHOW TRANSLATION/TABLE=LNM$SYSTEM USER
  USER = "DKA2:"  (LNM$SYSTEM_TABLE)
```

   The **SHOW TRANSLATION** command in this example displays the translation for the logical name USER. Because a table name is specified, the **SHOW TRANSLATION** command does not use the default search order. Only the specified table, LNM$SYSTEM, is searched. LNM$SYSTEM is the system logical name table.

5.
```
$ DEFINE/TABLE=LNM$PROCESS_DIRECTORY MYPROC -
_$ TEST_TABLE, LNM$PROCESS
$ SHOW TRANSLATION/TABLE=MYPROC FILER
  FILER = "[SMITH.FILER]"   (TEST_TABLE)
```

In this example, MYPROC defines a list of logical name tables that you want searched. It asks the system to first search TEST_TABLE (a user-defined table) and then to search LNM$PROCESS (the process logical name table). MYPROC is stored in LNM$PROCESS_DIRECTORY, the process directory table. When you enter the **SHOW TRANSLATION** command to find FILER in the MYPROC table, the tables TEST_TABLE and LNM$PROCESS are searched, in that order. The first match found is displayed.

SHOW USERS

SHOW USERS — Displays the user name and node name (in an OpenVMS Cluster environment) of interactive, subprocess, and batch users on the system.

Format

SHOW USERS [*username*]

Parameter

username

Specifies the user about whom you want information. The asterisk (*) and the percent sign (%) wildcard characters are allowed. If you specify a string, all users whose user names begin with the string are displayed. For example, if you specify the string MAR, all user names that begin with MAR are displayed. If no user exists whose name matches the specified string, an informational message tells you that no processes were found.

If you omit the *username* parameter, a list of all interactive, subprocess, and batch users is displayed.

Qualifiers

/BATCH
/NOBATCH

Displays all batch users in the OpenVMS Cluster environment. To restrict the display to users on specific nodes, use the **/BATCH** qualifier with the **/NODE** qualifier. When you use the **/NOBATCH** qualifier, all batch users are excluded from the display.

/CLUSTER

Displays the specified user names on all nodes in an OpenVMS Cluster environment.

/EXACT

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify a search string that must match the search string exactly and must be enclosed with quotation marks (" ").

If you specify the **/EXACT** qualifier without the **/SEARCH** qualifier, exact search mode is enabled when you set the search string with the Find (**E1**) key.

/FULL

Displays the user name, the node name, the process name, the process identification (PID) code, terminal names (both virtual and physical), and port information of all interactive, subprocess, and batch users on the system.

/HEADING (default)
/NOHEADING

Displays a heading line above the system output. When you use the **/NOHEADING** qualifier, the heading line is excluded from the display.

/HIGHLIGHT[=*keyword*]

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify the type of highlighting you want when a search string is found. When a string is found, the entire line is highlighted. You can use the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting.

/INTERACTIVE
/NOINTERACTIVE

Displays all interactive processes held by users in the OpenVMS Cluster environment. To restrict the display to users on specific nodes, use the **/INTERACTIVE** qualifier with the **/NODE** qualifier. When you use the **/NOINTERACTIVE** qualifier, all interactive processes held by users in the OpenVMS Cluster environment are excluded from the display.

/NETWORK
/NONETWORK

Displays all network users in the OpenVMS Cluster environment. To restrict the display to users on specific nodes, use the **/NETWORK** qualifier with the **/NODE** qualifier. When you use the **/NONETWORK** qualifier, all network users in the OpenVMS Cluster environment are excluded from the display.

/NODE[=(*name*,...)]

Displays all interactive, subprocess, and batch users on the specified node or nodes. If you enter the **/NODE** qualifier without a value, the qualifier displays all the interactive, subprocess, and batch users on the local node.

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the command is sent. By default, the output of the **SHOW USERS** command is sent to the current SYS$OUTPUT device (usually your terminal). To send the output to a file, use the **/OUTPUT** qualifier followed by a file specification.

The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification. If you enter a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS is the default file type.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

/PAGE[=*keyword*]
/NOPAGE (default)

Controls the display of user information on the screen.

You can use the following keywords with the **/PAGE** qualifier:

Keyword	Description
CLEAR_SCREEN	Clears the screen before each page is displayed.
SCROLL	Displays information one line at a time.
SAVE[= *n*]	Enables screen navigation of information, where *n* is the number of pages to store.

The **/PAGE=SAVE** qualifier allows you to navigate through screens of information. The **/PAGE=SAVE** qualifier stores up to 5 screens of up to 255 columns of information. When you use the **/PAGE=SAVE** qualifier, you can use the following keys to navigate through the information:

Key Sequence	Description
Up arrow key, **Ctrl/B**	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.
Right arrow key	Scroll right one column.
Find (**E1**)	Specify a string to find when the information is displayed.
Insert Here (**E2**)	Scroll right one half screen.
Remove (**E3**)	Scroll left one half screen.
Select (**E4**)	Toggle 80/132 column mode.
Prev Screen (**E5**)	Get the previous page of information.
Next Screen (**E6**), **Return**, **Enter**, **Space**	Get the next page of information.
F10, **Ctrl/Z**	Exit. (Some utilities define these differently.)
Help (**F15**)	Display utility help text.
Do (**F16**)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.

The **/PAGE** qualifier is not compatible with the **/OUTPUT** qualifier.

/SEARCH="*string*"

Use with the **/PAGE=SAVE** qualifier to specify a string that you want to find in the information being displayed. Quotation marks are required for the /SEARCH qualifier, if you include spaces in the text string.

You can also dynamically change the search string by pressing the Find key (**E1**) while the information is being displayed. Quotation marks are no required for a dynamic search.

/SUBPROCESS
/NOSUBPROCESS

Displays all subprocess users in the OpenVMS Cluster environment. To restrict the display to users on specific nodes, use the **/SUBPROCESS** qualifier with the **/NODE** qualifier. When you use the **/NOSUBPROCESS** qualifier, all subprocess users in the OpenVMS Cluster environment are excluded from the output.

/WRAP
/NOWRAP (default)

Use with the **/PAGE=SAVE** qualifier to limit the number of columns to the width of the screen and to 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 and can be seen when you use the scrolling (left and right) features provided by the **/PAGE=SAVE** qualifier.

Examples

1.
```
$ SHOW USERS
       OpenVMS User Processes at 12-MAY-2001 10:37 AM
     Total number of users = 4,  number of processes = 14

 Username     Node     Interactive  Subprocess   Batch
 S_SKONETSKI BBBBBB            1
 WISNIEWSKI  XXXXXX            4         2
 HIBBITS     AAAAAA            1         4
 VAXMAN      AAAAAA            2
```

The **SHOW USERS** command in this example displays the user names and node names of all current interactive, subprocess, and batch users on the system.

2.
```
$ SHOW USERS/NOHEADING/OUTPUT=SYSUSERS.DAT
$ TYPE SYSUSERS.DAT
 S_SKONETSKI BBBBBB            1
 WISNIEWSKI  XXXXXX            4         2
 HIBBITS     AAAAAA            1         4
 VAXMAN      AAAAAA            2
```

The **SHOW USERS** command in this example displays the user names and node names of all current interactive, subprocess, and batch users on the system without the heading text using the **/NOHEADING** qualifier. Using the **/OUTPUT** qualifier, you can write the output to a file for processing or later review.

3.
```
$ SHOW USERS *LES*
       OpenVMS User Processes at 12-MAY-2001 10:41 AM
     Total number of users = 3,  number of processes = 10

 Username     Node     Interactive  Subprocess   Batch
 THALES      MILETS            -         -          1
 PRAXITELES  LESBOS            5         2
 PERICLES    ISLAND            1
```

The **SHOW USERS** command in this example displays the user name and node names of all users whose user names contain the string LES.

4.
```
$ SHOW USERS/FULL/NODE=AAAAAA
       OpenVMS User Processes at 9-JUN-2001 02:23 PM
     Total number of users = 3,  number of processes = 3

 Username    Node   Process Name    PID     Terminal
 DJONES     AAAAAA  Aaaaaa_fta2:  2180012D  FTA2:
 DYSLI      AAAAAA  DYSLI         2180011A  FTA1:
 SINDBY     AAAAAA  SINDBY        2180011E  RTA1:   (JJJJJ::SINDBY)
```

```
<tag>(login) B4B4    _WSA1         3100009F
```

The **SHOW USERS** command in this example displays the user name, the local node, the process name, the process identification (PID) number, the terminal names (both virtual and physical), and port information for all interactive, subprocess, and batch users on system AAAAAA. A user name of <LOGIN> indicates that someone is in the process of logging in.

5.
```
$ SHOW USERS /NODE=(AAAAAA,BBBBBB,MMMMMM)
      OpenVMS User Processes at 9-JUN-2001 02:23 PM
    Total number of users = 5,  number of processes = 7

 Username    Node   Process Name    PID     Terminal
 BRICKLEY   BBBBBB  BRICKLEY      21E0009E  RTA1:  (CBREEZ::BRICKLEY)
 DCLDCT     MMMMMM  Mmmmmm_rta1:  2020015D  RTA1:  (MMMMMM::DJOHNSON)
 SMITH      AAAAAA  Aaaaaa_fta2:  2180012D  FTA2:
 PETERSON   MMMMMM  Mmmmmm_fta1:  2020013E  FTA1:
 PRESTON    MMMMMM  Mmmmmm_rta3:  20200184  RTA3:  (DDRSND::PRESTON)
 PRUSS      AAAAAA  PRUSS         2180011E  RTA1:  (IVOK::PRUSS)
 RABAHY     AAAAAA  RABAHY        2180011A  FTA1:
```

The **SHOW USERS** command in this example displays the user name and node name for all interactive, batch, and subprocess users on the nodes AAAAAA, BBBBBB, and MMMMMM.

SHOW WORKING_SET

SHOW WORKING_SET — Displays the working set limit, quota, and extent (in pagelets and CPU-specific pages) assigned to the current process. This information also can be output in bytes.

Format

SHOW WORKING_SET

Parameters

None.

Qualifier

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter the **/OUTPUT** qualifier without a file specification, the output is sent to the current process default output stream or device, identified by the logical name SYS$OUTPUT.

If you enter the **/OUTPUT** qualifier with a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS the default file type. The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

Examples

1.
```
$ SHOW WORKING_SET
```

```
Working Set (pagelets)  /Limit= 2000  /Quota= 4000 /Extent= 6000
Adjustment enabled  Authorized Quota= 4000  Authorized Extent= 6000
Working Set (8Kb pages) /Limit= 125   /Quota= 250  /Extent= 375
                    Authorized Quota= 250  Authorized Extent= 375
```

This example shows the **SHOW WORKING_SET** command on Alpha. The process has a working set limit of 2000 pagelets (125 pages) and a quota of 4000 pagelets (250 pages), and the current quota is equal to the authorized limit (4000 pagelets; 250 pages). It also shows that the current process has a working set extent of 6000 pagelets (375 pages) and that the current extent is equal to the authorized limit (6000 pagelets; 375 pages). A pagelet is 512 bytes.

2.
```
$ SHOW WORKING_SET
  Working Set (pagelets)  /Limit=3632  /Quota=8192  /Extent=524288
  Adjustment enabled      Authorized Quota=8192  Authorized
 Extent=524288

  Working Set (8Kb pages) /Limit=227  /Quota=512  /Extent=32768
                          Authorized Quota=512  Authorized Extent=32768
$ SET PROCESS/UNITS=BYTES
$ SHOW WORKING_SET
  Working Set (bytes)     /Limit=1.77MB  /Quota=4MB  /Extent=256MB
  Adjustment enabled      Authorized Quota=4MB  Authorized Extent=256MB
```

This example shows the same display, first in bits and then in bytes.

3.
```
$ SHOW WORKING_SET
Working Set      /Limit= 180   /Quota= 350           /Extent=1200
Adjustment enabled  Authorized Quota= 350  Authorized Extent=1200
```

This example shows the **SHOW WORKING_SET** command. The process has a working set limit of 180 pages, a quota of 350 pages, and the quota is equal to the authorized limit (350 pages). It also shows that the current process has a working set extent of 1200 and that the current extent is equal to the authorized limit (1200).

SORT

SORT — Invokes the Sort/Merge utility, which reorders the records in a file into a defined sequence and creates either a new file of the reordered records or an address file by which the reordered records can be accessed.

Format

SORT *input-filespec*[,...] *output-filespec*

Description

For more information about the Sort/Merge utility, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#SORT_CH] or online help.

SPAWN

SPAWN — Creates a subprocess of the current process. Portions of the current process context are copied to the subprocess. **The RESOURCE_WAIT state is required to spawn a process. Requires TMPMBX (temporary mailbox) or PRMMBX (permanent mailbox) privilege. The SPAWN**

command does not manage terminal characteristics. The SPAWN and ATTACH commands cannot be used if your terminal has an associated mailbox.

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 the parent process. If both a command string and the **/INPUT** qualifier are specified, the specified command string executes before additional commands are obtained from the **/INPUT** qualifier.

If the command string contains apostrophe substitution, the apostrophe substitution is performed in the context of the parent command line interpreter (CLI); other phases of substitution are performed in the context of the subprocess CLI.

Description

The **SPAWN** command creates a subprocess of your current process with the following attributes copied from the parent process:

- All global symbols except $RESTART, $SEVERITY, $STATUS
- All local symbols defined for the current command procedure level
- Key definitions
- The current keypad state
- The current prompt string
- All process logical names and logical name tables except those explicitly marked CONFINE or those created in executive or kernel mode
- Default disk and directory
- Current **SET MESSAGE** settings
- Current process privileges
- Control and verification states

Note that some attributes, such as the process's current command tables, are *not* copied.

When the subprocess is created, the process-permanent open files and any image or procedure context are *not* copied from the parent process. The subprocess is set to command level 0 (DCL level with the current prompt).

If you do not specify the **/PROCESS** qualifier, the name of this subprocess is composed of a portion of the parent process name and a unique number. For example, if the parent process name is SKONETSKI, the subprocess names can be SKONETSKI_123, SKONETSKI_4732, and so on.

The LOGIN.COM file of the parent process is not executed for the subprocess because the context is copied separately, allowing quicker initialization of the subprocess. When the **/WAIT** qualifier is in effect, the parent process remains in hibernation until the subprocess terminates or returns control to the parent by way of the **ATTACH** command.

More than one process simultaneously attempts to use the same input or output stream in the following circumstances:

- When several processes share that stream and you terminate a subprocess to which you are not currently attached
- When you terminate a process that is not spawned from the process to which you are currently attached

You should use the **LOGOUT** command to terminate the subprocess and return to the parent process. You can also use the **ATTACH** command to transfer control of the terminal to another process in the subprocess tree, including the parent process. (The **SHOW PROCESS/SUBPROCESSES** command displays the processes in the subprocess tree and points to the current process.)

Note

Because a tree of subprocesses can be established using the **SPAWN** command, you must be careful when terminating any process in the tree. When a process is terminated, all subprocesses below that point in the tree are terminated automatically.

Qualifiers used with the **SPAWN** command must directly follow the command verb. The command string parameter begins after the last qualifier and continues to the end of the command line.

Qualifiers

/CARRIAGE_CONTROL
/NOCARRIAGE_CONTROL

Determines whether carriage-return and line-feed characters are prefixed to the subprocess's prompt string. By default, **SPAWN** copies the current setting of the parent process.

/CLI=*cli-filespec*
/NOCLI

Specifies the name of a command language interpreter (CLI) to be used by the subprocess. The default CLI is the same as the parent process (defined in SYSUAF). If you specify the **/CLI** qualifier, the attributes of the parent process are copied to the subprocess.

The CLI you specify must be located in SYS$SYSTEM and have the file type .EXE.

/INPUT=*filespec*

Specifies an input file containing one or more DCL commands to be executed by the spawned subprocess. The file type defaults to .COM. The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification. Once processing of the input file is complete, the subprocess is terminated. If both a command string and the **/INPUT** qualifier are specified, the specified command string executes before additional commands are obtained from the **/INPUT** qualifier. If neither is specified, SYS$INPUT is assumed (in which case a **SPAWN/NOWAIT** command is aborted if **Ctrl/Y** is pressed to abort something running in your parent process).

You cannot explicitly specify non-record-oriented, process-permanent files (NRO PPFs) with the **/INPUT** qualifier. The system displays an error message when it encounters such a file as the value for the **/INPUT** qualifier.

Note that when NRO PPFs are used as implicit input (that is, the **/INPUT** qualifier is not specified and SYS$INPUT is a NRO PPF), the **SPAWN** command can succeed, as shown in the following table:

Process Type	SYS$INPUT	Implicit Input
Interactive	NRO PPF	SYS$COMMAND
Noninteractive	NRO PPF	Null device
Any	Any other	SYS$INPUT

If SYS$INPUT is a terminal, it cannot have an associated terminal mailbox.

/KERNEL_THREAD_LIMIT=*n*

Sets the limit of the number of kernel threads that can be created in the new process. If the value specified is higher than the SYSGEN parameter MULTITHREAD, an error message is returned. Specifying the value 0 makes the process use the systemwide limit set by the MULTITHREAD parameter.

/KEYPAD (default)
/NOKEYPAD

Copies keypad key definitions and the current keypad state from the parent process. By default, if you have established key definitions or states with the **DEFINE/KEY** or the **SET KEY** command, these settings are copied to the subprocess. Use the **/NOKEYPAD** qualifier if you do not want the key settings to be copied.

/LOG (default)
/NOLOG

Displays the assigned subprocess name and any messages indicating transfer of control between processes.

/LOGICAL_NAMES (default)
/NOLOGICAL_NAMES

Copies process logical names and logical name tables to the subprocess. By default, all process logical names and logical name tables are copied to the subprocess except those explicitly marked CONFINE or created in executive or kernel mode.

/NOTIFY
/NONOTIFY (default)

Controls whether a message is broadcast to your terminal notifying you that your subprocess has completed or aborted. This qualifier should not be used unless you specify the **/NOWAIT** qualifier. The **/NOTIFY** qualifier cannot be specified when the **SPAWN** command is executed from within a noninteractive process.

Note that messages broadcast as a result of using the **/NOTIFY** qualifier are considered to be DCL messages. Therefore, if **SET BROADCAST=NODCL** is in effect, all such notification messages are suppressed.

/OUTPUT=*filespec*

Specifies the output file to which the results of the **SPAWN** operation are written. The asterisk (*) and the percent sign (%) wildcard characters cannot be used in the file specification. (Do not specify SYS$COMMAND as a file specification for the **/OUTPUT** qualifier when using the **/NOWAIT** qualifier; both parent and subprocess output will be displayed simultaneously on your terminal.)

Caution

If you include a version number in the filespec argument of the **/OUTPUT** qualifier, and that version of the specified file already exists, the existing file is replaced with the new output.

You cannot explicitly specify non-record-oriented, process-permanent files (NRO PPFs) with the **/OUTPUT** qualifier. The system displays an error message when it encounters such a file as the value for the **/OUTPUT** qualifier.

Note that when NRO PPFs are used as implicit output, the **SPAWN** command can succeed, as shown in the following table:

Process Type	SYS$OUTPUT	Implicit Output
Any	NRO PPF	Mailbox transmitting records for parent to write to its current SYS$OUTPUT device
Any	Any other	SYS$OUTPUT

If you omit the **/OUTPUT** qualifier, output is written to the current SYS$OUTPUT device.

/PRIVILEGES=CURRENT|AUTHORIZED

Determines whether the spawned process inherits the current process' current or authorized privileges as its authorized privileges. By default, the authorized privilege mask for the subprocess is taken from the current privileges of its creator. (This corresponds to **/PRIVILEGES=CURRENT**.) If the **/PRIVILEGES=AUTHORIZED** qualifier is specified, the subprocess' authorized privileges will be taken from the caller's authorized privileges.

/PROCESS=*subprocess-name*

Specifies the name of the subprocess to be created. If you omit the **/PROCESS** qualifier, a unique process name is assigned with the same base name as the parent process and a unique number.

The default subprocess name format is *username_n*. If you specify a process name that already exists, an error message is displayed. If the **/LOG** qualifier has been specified, the assigned name of the subprocess is displayed.

Note

Prior to OpenVMS Version 7.3-1, if no process name was supplied, the system constructed a name by appending *_n* to the user name, where *n* was the next available nonduplicate integer for any process currently in the system. For example, the first spawned process from user SYSTEM would be called SYSTEM_1, the second, SYSTEM_2, and so on. The next available number was chosen as soon as a gap was found.

Starting with OpenVMS Alpha Version 7.3-1, a random number is chosen to append to the user name. In addition, starting with OpenVMS Version 8.3, the chosen number has, by default,

a maximum value of 65535 instead of the former maximum value of 255. Therefore, the first processes that are spawned from user SYSTEM might be SYSTEM_7376, SYSTEM_9, SYSTEM_47124, and so on.

Some applications might rely on the prior methods of assigning subprocess names. The DCL_CTLFLAGS system parameter allows you to configure the system as necessary.

The rationale for changing from sequential to random number generation is that determining the next available open sequential number slot is very expensive in terms of performance. The mechanism attempts to create the process by incrementing names until it finds one that is unique. When several subprocesses already exist, the cost of creating the subprocess in the same OpenVMS group, the cost multiplies because process names must be unique throughout the group.

The use of random number generation results in a very high probability of finding a unique name on the first try, because it is unlikely that the same number is already in use. This greatly reduces the cost of process creation, and applications that rely on spawned subprocesses might see a dramatic performance improvement as a result of this change.

For more information, see the relevant section in the *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/#d0e36924].

/PROMPT[=*string*]

Specifies the prompt string for DCL to use in the subprocess. The default is the prompt of the parent process.

The string can consist of more than one character. All valid ASCII characters can be used in the string. The string must be enclosed in quotation marks (" ") if it contains spaces, special characters, or lowercase characters. Otherwise, letters are automatically converted to uppercase, and leading and trailing spaces are removed.

If no string is specified, the dollar sign ($) DCL default prompt string is used for the subprocess.

/SYMBOLS (default)
/NOSYMBOLS

Determines whether global and local symbols (except $RESTART, $SEVERITY, and $STATUS) are passed to the subprocess. $RESTART, $SEVERITY, and $STATUS symbols are *never* passed to the subprocess.

/TABLE=*command-table*

Specifies the name of an alternate command table to be used by the subprocess.

/TRUSTED
/NOTRUSTED

Indicates that the **SPAWN** command input originates in a trusted command procedure. **SPAWN** commands are not allowed in CAPTIVE accounts. The **/TRUSTED** qualifier provides a way for properly written captive command procedures to perform **SPAWN** operations when the command input originates in the captive command procedure where it can be trusted. For more information about trusted command procedures, see the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/].

/WAIT (default)
/NOWAIT

Requires that you wait for the subprocess to terminate before you enter another DCL command. The **/NOWAIT** qualifier allows you to enter new commands while the subprocess is running. (Use the **/OUTPUT** qualifier with the **/NOWAIT** qualifier to avoid displaying both parent and subprocess output on the terminal simultaneously.)

Note that specifying the **/NOWAIT** qualifier causes both input and output to be shared with the parent process. If the input device is a terminal, control characters, such as **Ctrl/T** or **Ctrl/Y**, also affect all subprocesses sharing the input device. The **Ctrl/Y** control character, for example, interrupts all such subprocesses.

This problem may be avoided by specifying **/INPUT=NL**:

Example

```
$ RUN MYPROG
  .
  .
  .

$ Ctrl/Y
$ SPAWN MAIL
%DCL-S-SPAWNED, process SKONETSKI_272 spawned
%DCL-S-ATTACHED, terminal now attached to process SKONETSKI_272
MAIL> READ
  .
  .
  .
MAIL> EXIT
%DCL-S-RETURNED, control returned to process SKONETSKI
$ CONTINUE
```

The **SPAWN** command in this example allows you to enter the OpenVMS Mail utility (**MAIL**) without terminating the currently running program. After you exit from Mail, control is returned to the parent process.

START/CPU

START/CPU — Starts the specified secondary processor or processors (and any associated vector processors). The **/CPU** qualifier is required. **Applies only to OpenVMS multiprocessing systems. Requires CMKRNL(change mode to kernel) privilege.**

Format

START/CPU [*cpu-id*[,...]]

Parameter

***cpu-id*[,...]**

Specifies a decimal value representing the identity of a processor in an OpenVMS multiprocessing system. On an Alpha 7000 system, the CPU ID is the backplane slot number of the processor. If you do

not specify a CPU ID and do not include the **/ALL** qualifier, the **START/CPU** command selects a single available processor to join the multiprocessing system.

Description

The **START/CPU** command starts a secondary processor in a OpenVMS multiprocessing system.

You can issue a **START/CPU** command only for processors in the STOPPED or TIMOUT state, as represented by the **SHOW CPU** command; otherwise, the **START/CPU** command has no effect.

Qualifiers

/ALL

Selects all remaining processors in the system's available set to join the multiprocessing system.

/DEFAULT_CAPABILITIES

Eliminates all previous capability (user and system) modifications for the specified CPU and reinitializes them with the values in the global initialization variable SCH$GL_DEFAULT_CPU_CAP.

Normally, user capabilities survive CPU shutdowns and restarts (not reboots), making the downtime as transparent to the user as possible. The CPU user capability bits are only initialized from SCH$GL_DEFAULT_CPU_CAP at the first boot of the CPU. (The system capability bits, however, are reinitialized to their defaults taken from SCH$GL_DEFAULT_CPU_CAP.)

However, there may be times when the CPU needs to be returned to a known, consistent state. The **/DEFAULT_CAPABILITIES** qualifier mimics the behavior of the initial bootstrap of the CPU.

/POWER[=ON] (Alpha/IA-64 only)

Powers on the CPU prior to bringing the CPU into the active set.

Supported only on AlphaServer GS series systems.

Examples

1. `$ START/CPU`

   The **START/CPU** command in this example selects a single inactive processor from the set of those processors that are currently available but inactive. When it completes its initialization, the selected processor becomes part of the system's active set and is capable of scheduling and executing processes.

2. `$ START/CPU 4,7`

   The **START/CPU** command in this example selects the processors with CPU IDs `4` and `7`, if they are currently available and inactive. When they complete initialization, these processors become part of the system's active set and are capable of scheduling and executing processes.

3. `$ START/CPU/ALL`

   The **START/CPU/ALL** command in this example selects all remaining inactive and available processors. When they complete initialization, these processors become part of the system's active set and are capable of scheduling and executing processes.

START/NETWORK

START/NETWORK — Starts or restarts the specified network service on the local node. The **/NETWORK** qualifier is required.

Format

START/NETWORK *network-service*

Parameter

network-service

Specifies the name of the network service to be started or restarted.

Description

The **START/NETWORK** command causes a network service to begin executing on the local node.

Use the **STOP/NETWORK** command to pause the network.

Example

```
$ START/NETWORK  DECnet
```

This command starts the `DECnet` network service.

START/QUEUE

START/QUEUE — Starts or restarts the specified queue after it has been initialized. You also can use this command to change the options of the specified queue. The **/QUEUE** qualifier is required. **Requires manage (M) access to the queue.**

Format

START/QUEUE *queue-name*[:]

Parameter

***queue-name*[:]**

Specifies the name of the queue to be started or restarted.

Description

The **START/QUEUE** command restarts a queue that has been stopped or paused. You must create the queue with the **INITIALIZE/QUEUE** command before you can use the **START/QUEUE** command. To create and start the queue at the same time, use the **INITIALIZE/QUEUE/START** command.

If the specified queue is an autostart queue, the **START/QUEUE** command activates the queue for autostart. The queue begins processing jobs when autostart is enabled with the **ENABLE AUTOSTART/QUEUES** command for any node on which the queue can run.

You can specify with the **START/QUEUE** command most qualifiers that can be used with the **INITIALIZE/QUEUE** command. The defaults for the **START/QUEUE** qualifiers depend on the qualifiers specified when the queue was initialized. For example, the default for **/JOB_LIMIT** with **INITIALIZE/QUEUE** is 1. However, if the queue you are altering was initialized with a job limit of 3, and if you do not specify the **/JOB_LIMIT** qualifier with the **START/QUEUE** command, the job limit remains at 3 for that queue.

Values or options specified with qualifiers on the **START/QUEUE** command override the values or options of those same qualifiers that were specified when the queue was initialized.

If the specified queue is running when you enter the **START/QUEUE** command, the system returns an error message. To change the options of a running queue, use the **SET QUEUE** command. To change queue options that cannot be altered with the **SET QUEUE** command, use the following procedure:

1. Stop the queue with the **STOP/QUEUE/NEXT** command.
2. Restart the queue with the **START/QUEUE** or **INITIALIZE/QUEUE/START** command, specifying the appropriate qualifiers for the options you desire.

The **/TOP_OF_FILE**, **/BACKWARD**, **/FORWARD**, **/SEARCH**, **/ALIGN**, and **/NEXT** qualifiers are effective only when restarting an output execution queue from a paused state. They do not apply to queues that are stopped or to batch queues. When you specify more than one of the first five of these qualifiers, the print symbiont processes them in the following order:

1. **/TOP_OF_FILE**
2. **/BACKWARD**
3. **/FORWARD**
4. **/SEARCH**
5. **/ALIGN**

Qualifiers

/ALIGN[=(*option*[,...])]

Prints alignment pages to aid in aligning printer forms. Use this qualifier only when restarting an output execution queue from a paused state.

After the alignment is complete, the queue enters a paused state until you restart it by reentering the **START/QUEUE** command. Printing resumes from the point where alignment data started; that is, the task is backspaced over the pages printed for alignment.

Possible options are as follows:

Option	Description
MASK	Specifies that input data is masked by replacing alphabetic characters with x's and numbers with 9s; nonalphanumeric characters are not masked. Mask characters allow you to prevent the printing of sensitive information. If you omit the MASK option, data is printed unaltered.
n	Specifies the number of alignment pages to print. The value of *n* can be from 1 to 20. By default, one page of alignment data is printed.

/AUTOSTART_ON=(*node*::[*device*][,...])

Designates the queue as an autostart execution queue and specifies the node, or node and device, on which the queue can be located. Both *node* and *device* must be specified for output queues. For batch queues, only *node* is applicable.

In an OpenVMS Cluster, you can specify more than one node (or node and device) on which a queue can run, in the preferred order in which nodes should claim the queue. This allows the queue to fail over to another node if the node on which the queue is running leaves the cluster.

For autostart queues, the **START/QUEUE** command activates the queue for autostart. The queue will begin processing jobs when the **ENABLE AUTOSTART/QUEUES** command is entered for a node on which the queue can run.

This qualifier cannot be used with the **/ON** or **/GENERIC** qualifier; however, you can specify the **/AUTOSTART_ON** qualifier for a queue previously created or started with the **/ON** qualifier. Doing so overrides the **/ON** option and makes the queue an autostart queue.

For more information about autostart queues, see the chapter about managing queues in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#_6017QUEUE_MGR].

/BACKWARD=*n*

Restarts a print queue *n* pages before the current page; *n* defaults to 1. If you omit the page value, printing resumes at the top of the current page. Use this qualifier only when restarting an output execution queue from a paused state.

Note

Using the **START/QUEUE/BACKWARD=*n*** command to restart a print job that uses Fortran carriage control and that was printed with the **/NOFEED** qualifier can have unexpected results, in particular:

- The page positioning in the restarted job may not be correct: the output may not begin at the top of the page specified by *n*.
- The output from the print job may be preceded by extra meaningless information.

/BASE_PRIORITY=*n*

Specifies the base process priority at which jobs are initiated from a batch execution queue. By default, if you omit the qualifier, jobs are initiated at the same priority as the base priority established by DEFPRI at system generation (usually 4). The base priority specifier can be any decimal value from 0 to 15.

/BLOCK_LIMIT=([*lowlim*,]*uplim*)
/NOBLOCK_LIMIT

Limits the size of print jobs that can be processed on an output execution queue. This qualifier allows you to reserve certain printers for certain size jobs. You must specify at least one of the parameters.

The *lowlim* parameter is a decimal number referring to the minimum number of blocks that are accepted by the queue for a print job. If a print job is submitted that contains fewer blocks than the

lowlim value, the job remains pending until the block limit for the queue is changed. After the block limit for the queue is decreased sufficiently, the job is processed.

The *uplim* parameter is a decimal number referring to the maximum number of blocks that are accepted by the queue for a print job. If a print job is submitted that exceeds this value, the job remains pending until the block limit for the queue is changed. After the block limit for the queue is increased sufficiently, the job is processed.

If you specify only an upper limit for jobs, you can omit the parentheses. For example, **/BLOCK_LIMIT=1000** means that only jobs with 1000 blocks or less are processed in the queue. To specify only a lower job limit, you must use a null string ("") to indicate the upper specifier. For example, **/BLOCK_LIMIT=(500,"")** means any job with 500 or more blocks is processed in the queue. You can specify both a lower and upper limit. For example, **/BLOCK_LIMIT=(200,2000)** means that jobs with less than 200 blocks or more than 2000 blocks are not processed in the queue.

The **/NOBLOCK_LIMIT** qualifier cancels the previous setting established by the **/BLOCK_LIMIT** qualifier for the queue.

/CHARACTERISTICS=(*characteristic*[,...])
/NOCHARACTERISTICS

Specifies one or more characteristics for processing jobs on an execution queue. If a queue does not have all the characteristics that have been specified for a job, the job remains pending. If you specify only one characteristic, you can omit the parentheses. Each time you specify the **/CHARACTERISTICS** qualifier, all previously set characteristics are canceled. Only the characteristics specified with the qualifier are established for the queue.

Queue characteristics are installation specific. The characteristic parameter can be either a value from 0 to 127 or a characteristic name that has been defined by the **DEFINE/CHARACTERISTIC** command.

The **/NOCHARACTERISTICS** qualifier cancels any settings previously established by the **/CHARACTERISTICS** qualifier for the queue.

/CLOSE

Prevents jobs from being entered in the queue through **PRINT** or **SUBMIT** commands or as a result of requeue operations. To allow jobs to be entered, use the **/OPEN** qualifier. Whether a queue accepts or rejects new job entries is independent of the queue's state (such as paused, stopped, or stalled). When a queue is marked closed, jobs executing continue to execute. Jobs already pending in the queue continue to be candidates for execution.

/CPUDEFAULT=*time*

Defines the default CPU time limit for jobs in this batch execution queue. You can specify time as delta time, 0, INFINITE, or NONE. You can specify up to 497 days of delta time.

If the queue does not have a specified CPUMAXIMUM time limit and the value established in the user authorization file (UAF) has a specified CPU time limit of NONE, either the value 0 or the keyword INFINITE allows unlimited CPU time. If you specify NONE, the CPU time value defaults to the value specified either in the UAF or by the **SUBMIT** command (if included). CPU time values must be greater than or equal to the number specified by the system parameter PQL_MCPULM.

For information on specifying delta times, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT]

or the online help topic Date. For more information on specifying CPU time limits, see the "CPU Time Limit Specifications and Actions" table for the **INITIALIZE/QUEUE** command in the *VSI OpenVMS DCL Dictionary: A–M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_119].

/CPUMAXIMUM=*time*

Defines the default CPU time limit for all jobs in this batch execution queue. You can specify time as delta time, 0, INFINITE, or NONE. You can specify up to 497 days of delta time.

If the queue does not have a specified CPUMAXIMUM time limit and the value established in the UAF has a specified CPU time limit of NONE, either the value 0 or the keyword INFINITE allows unlimited CPU time. If you specify NONE, the CPU time value defaults to the value specified either in the UAF or by the **SUBMIT** command (if included). CPU time values must be greater than or equal to the number specified by the system parameter PQL_MCPULM. The time cannot exceed the CPU time limit set by the **/CPUMAXIMUM** qualifier.

For information on specifying delta times, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date. For more information on specifying CPU time limits, see the "CPU Time Limit Specifications and Actions" table for the **INITIALIZE/QUEUE** command in the *VSI OpenVMS DCL Dictionary: A–M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_119].

/DEFAULT=(*option*[,...])
/NODEFAULT

Establishes defaults for certain options of the **PRINT** command. Defaults are specified by the list of options. If you specify only one option, you can omit the parentheses. After you set an option for the queue with the **/DEFAULT** qualifier, you do not have to specify that option in your **PRINT** commands. If you do specify these options in your **PRINT** command, the values specified with the PRINT command override the values established for the queue with the **/DEFAULT** qualifier.

You cannot use the **/DEFAULT** qualifier with the **/GENERIC** qualifier.

Possible options are as follows:

Option	Description
[NO]BURST[=*keyword*]	Controls whether two file flag pages with a burst bar between them are printed preceding output. If you specify the value ALL (default), these flag pages are printed before each file in the job. If you specify the value ONE, these flag pages are printed once before the first file in the job.
[NO]FEED	Specifies whether a form feed is inserted automatically at the end of a page.
[NO]FLAG[=*keyword*]	Controls whether a file flag page is printed preceding output. If you specify the value ALL (default), a flag page is printed before each file in the job. If you specify the value ONE, a flag page is printed once before the first file in the job.
FORM=*type*	Specifies the default form for an output execution queue. If a job is submitted without an explicit form definition, this form is used to process the job. If no form type is explicitly specified with the

Option	Description
	FORM keyword, the system assigns the form "DEFAULT" to the queue. See also the description of the **/FORM_MOUNTED** qualifier.
[NO]TRAILER[=*keyword*]	Controls whether a file trailer page is printed following output. If you specify the value ALL (default), a trailer page is printed after each file in the job. If you specify the value ONE, a trailer page is printed once after the last file in the job.

When you specify the BURST option for a file, the [NO]FLAG option does not add or subtract a flag page from the two flag pages that are printed preceding the file. For information on establishing mandatory queue options, see the description of the **/SEPARATE** qualifier. For more information on specifying default queue options, see the relevant section in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#_6017QUEUE_MGR].

/DESCRIPTION=*string*
/NODESCRIPTION

Specifies a string of up to 255 characters that is used to provide operator-supplied information about the queue.

Enclose strings containing lowercase letters, blanks, or other non alphanumeric characters (including spaces) in quotation marks (" ").

The **/NODESCRIPTION** qualifier removes any descriptive text that may be associated with the queue.

/DISABLE_SWAPPING
/NODISABLE_SWAPPING

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

/ENABLE_GENERIC
/NOENABLE_GENERIC

Specifies whether files queued to a generic queue that does not specify explicit queue names with the **/GENERIC** qualifier can be placed in this execution queue for processing. For more information, see the description of the **/GENERIC** qualifier.

/FORM_MOUNTED=*type*

Specifies the mounted form for an output execution queue.

If no form type is explicitly specified, the system assigns the form "DEFAULT" to the queue.

If the stock of the mounted form does not match the stock of the default form, as indicated by the **/DEFAULT=FORM** qualifier, all jobs submitted to this queue without an explicit form definition enter a pending state and remain pending until the stock of the mounted form of the queue is identical to the stock of the form associated with the job.

If a job is submitted with an explicit form and the stock of the explicit form is not identical to the stock of the mounted form, the job enters a pending state and remains pending until the stock of the mounted form of the queue is identical to the stock of the form associated with the job.

To specify the form type, use either a numeric value or a form name that has been defined by the **DEFINE/FORM** command. Form types are installation-specific. You cannot use the **/FORM_MOUNTED** qualifier with the **/GENERIC** qualifier.

/FORWARD=*n*

Advances the specified number of pages before resuming printing the current file in the current job; the default is 1. If you omit the page value, printing resumes at the top of the next page. Use this qualifier only when restarting an output execution queue from a paused state.

/GENERIC[=(*queue-name*[,...])]
/NOGENERIC

Specifies a generic queue. Also specifies that jobs placed in this queue can be moved for processing to compatible execution queues. The **/GENERIC** qualifier optionally accepts a list of target execution queues that have been previously defined. For a generic batch queue, these target queues must be batch execution queues. For a generic output queue, these target queues must be output execution queues, but can be of any type (printer, server, or terminal). For example, a generic printer queue can feed a mixture of printer and terminal execution queues.

Use the **/GENERIC** qualifier to change the list of target nodes for a generic queue. The queue must have been initialized as a generic queue with the **INITIALIZE/QUEUE/GENERIC** command.

If you do not specify any target execution queues with the **/GENERIC** qualifier, jobs can be moved to any execution queue that (1) is initialized with the **/ENABLE_GENERIC** qualifier, and (2) is the same type (batch or output) as the generic queue.

To define the queue as a generic batch or output queue, you use the **/GENERIC** qualifier with either the **/BATCH** or the **/DEVICE** qualifier. If you specify neither the **/BATCH** nor the **/DEVICE** qualifier on creation of a generic queue, by default the queue becomes a generic printer queue.

/JOB_LIMIT=*n*

Specifies the number of batch jobs that can be executed concurrently from the queue. Specify a number in the range 0 to 255.

/LIBRARY=*filename*
/NOLIBRARY

Specifies the file name for the device control library. When you initialize an output execution queue, you can use the **/LIBRARY** qualifier to specify an alternate device control library. You can use only a file name as the parameter of the **/LIBRARY** qualifier. The system always assumes that the file is located in SYS$LIBRARY and that the file type is .TLB.

/NEXT

Aborts the currently suspended print job and begins processing of the first pending job in the queue. Use this qualifier only when restarting an output execution queue from a paused state.

/NO_INITIAL_FF
/NONO_INITIAL_FF (default)

Specifies whether a form feed should be sent to a printer device when a queue starts. To suppress the initial form feed, use the **/NO_INITIAL_FF** qualifier.

The **/NONO_INITIAL_FF** qualifier sends a form feed to the output device to ensure that the paper is at the top of a page before printing begins.

/ON=[*node*::]*device*[:] (printer, terminal, server queue)
/ON=*node*:: (batch queue)

Specifies the node or device, or both, on which this execution queue is located. For batch execution queues, you can specify only the node name. For output execution queues, you can include both the node name and the device name.

The node name is used only in VAXcluster systems; it must match the node name specified by the system parameter SCSNODE for the VAX computer on which the queue executes.

You cannot use the **/ON** qualifier with the **/AUTOSTART_ON** or **/GENERIC** qualifier; however, you can specify the **/ON** qualifier for a queue previously created or started with the **/AUTOSTART_ON** qualifier. Doing so overrides the **/AUTOSTART_ON** qualifier and makes the queue a nonautostart queue.

/OPEN

Allows jobs to be entered in the queue through **PRINT** or **SUBMIT** commands or as the result of requeue operations. To prevent jobs from being entered in the queue, use the **/CLOSE** qualifier. Whether a queue accepts or rejects new job entries is independent of the queue's state (such as paused, stopped, or stalled).

/OWNER_UIC=*uic*

Requires manage (M) access to the queue.

Enables you to change the user identification code (UIC) of the queue. Specify the UIC by using standard format as described in the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#UIC_SECTION].

/PROCESSOR=*filename*
/NOPROCESSOR

Requires OPER (operator) privilege to change the file name from the one with which the queue was initialized.

Allows you to specify your own print symbiont for an output execution queue. You can use any valid file name as a parameter of the **/PROCESSOR** qualifier. The system supplies the device and directory name SYS$SYSTEM and the file type .EXE. If you use this qualifier for an output queue, it specifies that the symbiont image to be executed is SYS$SYSTEM:*filename*.EXE.

By default, SYS$SYSTEM:PRTSMB.EXE is the symbiont image associated with an output execution queue.

The **/NOPROCESSOR** qualifier cancels any previous setting established by the **/PROCESSOR** qualifier, and causes SYS$SYSTEM:PRTSMB.EXE to be used.

/PROTECTION= (*ownership*[:*access*],...)

Requires OPER (operator) privilege, or control (C) and execute (E) access to the queue.

Specifies the protection of the queue.

- Specify the *ownership* parameter as system (S), owner (O), group (G), or world (W).

- Specify the *access* parameter as read (R), submit (S), manage (M), or delete (D). A null access specification means no access.

If you include only one protection code, you can omit the parentheses.

For more information on specifying protection codes, see the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#PROTECT_CODE_DETAILS]. For more information on controlling queue operations through UIC-based protection, see the relevant section in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#SECURITY].

/RAD=*n*

Specifies the RAD number on which to run batch jobs assigned to the queue. The RAD value is validated as a positive integer between 0 and the value returned by the $GETSYI item code, SYI$_RAD_MAX_RADS.

RAD is supported on AlphaServer GS series systems and, starting from OpenVMS Version 8.4, support is extended to NUMA-capable IA-64 systems.

/RECORD_BLOCKING
/NORECORD_BLOCKING

Determines whether the symbiont can concatenate (or block together) output records for transmission to the output device. If you specify the **/NORECORD_BLOCKING** qualifier, the symbiont sends each formatted record in a separate I/O request to the output device. For the standard OpenVMS print symbiont, record blocking can have a significant performance advantage over single-record mode.

/RETAIN[=*option*]
/NORETAIN

Holds jobs in the queue in a retained status after they have executed. The **/NORETAIN** qualifier enables you to reset the queue to the default. Possible options are as follows:

Option	Description
ALL	Holds all jobs in the queue after execution.
ERROR	Holds in the queue only jobs that fail to complete.

A user can request a job retention option for a job by specifying the **/RETAIN** qualifier with the **PRINT**, **SUBMIT**, or **SET ENTRY** command. However, the job retention option you specify for a queue overrides any job retention option requested by a user for a job in that queue.

/SCHEDULE=[NO]SIZE

Specifies whether pending jobs in an output queue are scheduled for printing based on the size of the job. When the **/SCHEDULE=SIZE** qualifier is in effect, shorter jobs are printed before longer ones. When the **/SCHEDULE=NOSIZE** qualifier is in effect, jobs are printed in the order they were submitted, regardless of size.

If you enter this command while there are pending jobs in any queue, its effect on future jobs is unpredictable.

/SEARCH="search-string"

Specifies that printing is to resume at the page containing the specified string. The search for the string moves forward, beginning on the page following the current page. During the search, consecutive tabs and spaces are treated as a single space, and character case is ignored. The string can be from 1 to 63 characters and must be enclosed in quotation marks (" "). Use this qualifier only when restarting an output execution queue from a paused state.

/SEPARATE=(*option*[,...])
/NOSEPARATE

Specifies the mandatory queue options, or job separation options, for an output execution queue. Job separation options cannot be overridden by the **PRINT** command.

You cannot use the **/SEPARATE** qualifier with the **/GENERIC** qualifier.

The job separation options are as follows:

Option	Description
[NO]BURST	Specifies whether two job flag pages with a burst bar between them are printed at the beginning of each job.
[NO]FLAG	Specifies whether a job flag page is printed at the beginning of each job.
[NO]TRAILER	Specifies whether a job trailer page is printed at the end of each job.
[NO]RESET=(*module*[,...])	Specifies one or more device control library modules that contain the job reset sequence for the queue. The specified modules from the queue's device control library (by default SYS$LIBRARY:SYSDEVCTL) are used to reset the device each time a job reset occurs. The RESET sequence occurs after any file trailer and before any job trailer. Thus, all job separation pages are printed when the device is in its RESET state.

When you specify **/SEPARATE=BURST**, the [NO]FLAG separation option does not add or subtract a flag page from the two flag pages that are printed preceding the job.

For information on establishing queue options that can be overridden, see the description of the **/DEFAULT** qualifier.

For more information on specifying mandatory queue options, see the relevant section in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#_6017QUEUE_MGR].

/TOP_OF_FILE

Resumes printing at the beginning of the file that was current when the output execution queue paused. Use this qualifier only when restarting an output execution queue from a paused state.

/WSDEFAULT=*n*

Defines for a batch job a working set default, the default number of physical pages that the job can use. The value set by this qualifier overrides the value defined in the user authorization file (UAF) of any user submitting a job to the queue.

You also can specify this qualifier for an output execution queue. Used in this context, the **/WSDEFAULT** qualifier establishes the working set default of the symbiont process for an execution queue when the symbiont process is created.

Specify the value of *n* as a number of 512-byte pagelets on Alpha. Note that the operating systems rounds up this value to the nearest CPU-specific page so that the actual amount of physical memory allowed may be larger than the specified amount on Alpha.

If you specify the value 0 or NONE, the working set default value defaults to the value specified in the UAF or by the **SUBMIT** command (if included).

For more information about the way a working set default affects batch jobs, see *Table 9, "Working Set Default, Extent, and Quota Decision"*.

/WSEXTENT=*n*

Defines for the batch job a working set extent, the maximum amount of physical memory that the job can use. The job uses the maximum amount of physical memory only when the system has excess free pages. The value set by this qualifier overrides the value defined in the user authorization file (UAF) of any user submitting a job to the queue.

You also can specify this qualifier for an output execution queue. Used in this context, the **/WSEXTENT** qualifier establishes the working set extent of the symbiont process for an output execution queue when the symbiont process is created.

Specify the value of *n* as a number of 512-byte pagelets on Alpha. Note that the operating system rounds up this value to the nearest CPU-specific page so that the actual amount of physical memory allowed may be larger than the specified amount on Alpha.

If you specify the value 0 or NONE, the working set extent value defaults to the value specified in the UAF or by the **SUBMIT** command (if included).

For more information about the way a working set extent affects batch jobs, see *Table 9, "Working Set Default, Extent, and Quota Decision"*.

/WSQUOTA=*n*

Defines for a batch job a working set quota, the amount of physical memory that is guaranteed to the job. The value set by this qualifier overrides the value defined in the user authorization file (UAF) of any user submitting a job to the queue.

You also can specify this qualifier for an output execution queue. Used in this context, the **/WSQUOTA** qualifier establishes the working set quota of the symbiont process for an output execution queue when the symbiont process is created.

Specify the value of *n* as a number of 512-byte pagelets on Alpha. Note that the operating system rounds up this value to the nearest CPU-specific page so that the actual amount of physical memory allowed may be larger than the specified amount on Alpha.

If you specify the value 0 or NONE, the working set quota value defaults to the value specified in the UAF or by the **SUBMIT** command (if included).

Working set default, working set quota, and working set extent values are included in each user record in the system UAF. You can specify working set values for individual jobs or for all jobs in a given queue. The decision table (*Table 9, "Working Set Default, Extent, and Quota Decision"*)

shows the action taken for different combinations of specifications that involve working set size and working set quota values.

Table 9. Working Set Default, Extent, and Quota Decision

Value Specified by the SUBMIT Command?	Value Specified for the Queue?	Action Taken
No	No	Use the UAF value.
No	Yes	Use value for the queue.
Yes	Yes	Use smaller of the two values.
Yes	No	Compare specified value with UAF value; use the smaller.

Examples

1.
```
$ STOP/QUEUE  LPA0
$ START/QUEUE/TOP_OF_FILE  LPA0
```

   The **STOP/QUEUE** command in this example suspends the job that is currently executing on the printer queue LPA0 and places that queue in the paused state. The **START/QUEUE** command releases the queue from the paused state. The **/TOP_OF_FILE** qualifier causes the job that was suspended to resume printing at the beginning of the file rather than at where it was interrupted.

2.
```
$ INITIALIZE/QUEUE  LPA0
   .
   .
   .
$ START/QUEUE/DEFAULT=FLAG  LPA0
```

   The **INITIALIZE/QUEUE** command in this example initializes the queue named LPA0. Later, the **START/QUEUE** command starts the queue. The **/DEFAULT** qualifier requests that a flag page precede each file in each job.

3.
```
$ START/QUEUE/DEFAULT=FORM=LN01_PORTRAIT LN01_PRINT
```

   The **START/QUEUE** command in this example restarts the LN01_PRINT queue with the default form LN01_PORTRAIT.

4.
```
$ INITIALIZE/QUEUE/START/GENERIC=(A,B)  MYQUEUE
.
.  [new printers X and Y are brought in at a later date]
.
$ STOP/QUEUE/NEXT MYQUEUE
$ START/QUEUE/GENERIC=(X,Y) MYQUEUE
```

   This example changes the list of target nodes for a generic queue. Note that the queue was previously initialized as a generic queue.

5.
```
$ START/QUEUE/RAD=1 BATCHQ1

$ SHOW QUEUE/FULL BATCHQ1
Batch queue BATCHQ1, idle, on QUEBID::
 /BASE_PRIORITY=4 /JOB_LIMIT=3 /OWNER=[SYSTEM]
 /PROTECTION=(S:M,O:D,G:R,W:S) /RAD=1
```

This example modifies BATCHQ1 to run all assigned jobs on RAD 1 of QUEBID, and readies the queue to accept jobs for processing.

START/QUEUE/MANAGER

START/QUEUE/MANAGER — Starts the clusterwide queue manager for the queuing system and opens that queue manager's queue database files. The **/QUEUE** qualifier is optional, but the **/MANAGER** qualifier is required. By default, the command affects the default queue manager, SYS$QUEUE_MANAGER. Specify the **/NAME_OF_MANAGER** qualifier to start a queue manager other than the default. For more information, see the chapter on the queue manager in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#_6017QUEUE_MGR]. **Requires OPER (operator) and SYSNAM (system logical name) privileges.**

Format

START/QUEUE/MANAGER [*dirspec*]

Parameter

dirspec

Specifies the directory location to contain the system queue and journal files of the queue database. The queue file has a file type of QMAN$QUEUES and contains queue definitions. The journal file has a file type of QMAN$JOURNAL and contains job and other information that lets the queue manager to return to its last known state should a system be stopped unexpectedly. These files must reside in the same directory.

The default location of the queue and journal files is SYS$COMMON:[SYSEXE]. The optional *dirspec* parameter is used only for specifying an alternate location for the queue and journal files. The specification must include at least the device and directory name. The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the directory specification.

The directory you specify must be available to all nodes that can run the queue manager. If the directory specification is a concealed logical name, it must be identically defined on all nodes in the cluster.

The location of the queue and journal files is stored in the master file of the queue database. You do not have to respecify the directory location with subsequent **START/QUEUE/MANAGER** commands.

For information about changing the location of any of the queue database files, see the chapter on the queue manager in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#_6017QUEUE_MGR].

Description

The **START/QUEUE/MANAGER** command has the following uses:

- Enter the command **START/QUEUE/MANAGER/NEW_VERSION** to create the queue database and initially start a queue manager. See the description of the **/NEW_VERSION** qualifier for more information. Once the queue manager has been started, it will remain running unless it is explicitly stopped with the **STOP/QUEUE/MANAGER/CLUSTER** command.

- If the **STOP/QUEUE/MANAGER/CLUSTER** command has been executed, enter the **START/QUEUE/MANAGER** command to restart a queue manager.
- In an OpenVMS Cluster, enter the **START/QUEUE/MANAGER** command with the **/ON** qualifier to modify the list of preferred nodes on which a queue manager can run. See the description of the **/ON** qualifier for more information.
- In an OpenVMS Cluster, enter the **START/QUEUE/MANAGER** command to ensure that a queue manager process is executing on the most preferred, available node. If the queue manager is not running on the most preferred, available node, the queue manager will be moved to that node without interruption of service. If you are using the default node list (*), the queue manager will not move. For more information, see the description of the **/ON** qualifier.

If the queue manager is in a location other than the default, and in OpenVMS Cluster environments with multiple system disks, you must define the logical name QMAN$MASTER. For instructions, see the chapter about the queue manager and queue database in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#_6017QUEUE_MGR].

If a queue manager does not start when you enter the **START/QUEUE/MANAGER** command, you will receive the following message:

```
%JBC-E-QMANNOTSTARTED, queue manager could not be started
```

If you see this message, search the operator log file SYS$MANAGER:OPERATOR.LOG (or look on the operator console) for messages from the facilities QUEUE_MANAGE and JOB_CONTROL for information about the problem, as follows:

```
$  SEARCH SYS$MANAGER:OPERATOR.LOG /WINDOW=5 QUEUE_MANAGE,JOB_CONTROL
```

Qualifies

/ADD

Creates an additional queue manager in the existing queue database. If the named queue manager already exists, the request will be rejected.

/NAME_OF_MANAGER=*name*

Creates a non-default queue manager. The required name value may be up to 31 characters long and may be a logical. The name will serve as the identifier for the queue manager process and the portion of the database that it is managing.

/NEW_VERSION
/NONEW_VERSION (default)

Specifies that a new (empty) version of the queue database is to be created. This qualifier is required when initially creating and starting the queuing system.

If you specify this qualifier and a queue database already exists, the new master and queue files of the queue database supersede existing versions of those files; however, the journal file of the existing queue database is deleted. Jobs and other information are lost.

/ON=(*node*[,...])

In an OpenVMS Cluster, specifies the nodes on which a clusterwide queue manager can run. The default value for the node list is the asterisk (*) wildcard character, meaning that all nodes in the

cluster are eligible to run the queue manager. If the node on which the queue manager is running leaves the cluster, the queue manager can automatically fail over to any available node in the cluster. However, to specify a preferred order in which the nodes should claim the queue manager, or to limit the nodes which can run it, you must specify the **/ON** qualifier.

The node list you specify is stored in the queue database. Anytime the **START/QUEUE/MANAGER** command is entered and neither the **/NEW_VERSION** nor **/ON** qualifier is specified, the **/ON** list stored in the queue database remains unchanged.

For highest availability, specify the asterisk (*) wildcard character as the last node in the node list to indicate that any remaining unlisted node can claim the queue manager, with no preferred order. If you do not specify the asterisk (*) wildcard character last in the node list, the queue manager can only fail over if one of the nodes in the list is available; however, if you want to exclude certain nodes from being eligible to run the queue manager, you cannot use the asterisk (*) wildcard character. You cannot specify the asterisk (*) wildcard character as part of a node name.

Anytime the **START/QUEUE/MANAGER** command is entered (with or without the **/ON** qualifier), the job controller will check to see if one or more preferred queue manager nodes was currently or previously specified with the **/ON** qualifier. If one or more preferred nodes was specified, and the queue manager is running on a node other than the first available node of those specified, the queue manager process is moved from its current node and restarted on the first available preferred node. Despite the transition, queues on the running nodes are not stopped. All requests to the queuing system, for example, **PRINT**, **SUBMIT**, and **SHOW ENTRY** requests, will complete as expected.

Examples

1.
```
$ START/QUEUE/MANAGER/NEW_VERSION
$ SHOW QUEUE
%JBC-E-NOSUCHQUE, no such queue
```

   The **START/QUEUE/MANAGER** command in this example starts the queue manager and creates the queue and journal files in the default location, SYS$COMMON:[SYSEXE]. Because the asterisk (*) wildcard character is used by default as the value for the list of nodes on which the queue manager can run, the queue manager can fail over to any available node in the cluster.

   This command starts the default queue manager SYS$QUEUE_MANAGER because the **/NAME_OF_MANAGER** qualifier is not specified.

   Both the SYS$COMMON:[SYSEXE] location and the value for the **/ON** qualifier (which is * by default in this example) are stored in the queue database for future reference. The newly created queue database contains no queues or jobs. The **SHOW QUEUE** command shows that no queues are defined on this cluster.

2.
```
$ START/QUEUE/MANAGER/NEW_VERSION -
_$ /ON=(SATURN,VENUS,NEPTUN,*) DUA5:[SYSQUE]
```

   The **START/QUEUE/MANAGER** command in this example creates the queue and journal files on the cluster-accessible disk volume DUA5, in directory SYSQUE. You must mount the disk before you enter the **START/QUEUE/MANAGER** command.

   The **/ON** qualifier specifies that the queue manager should run first on node SATURN. If SATURN leaves the cluster, the queue manager will attempt to fail over to VENUS. If VENUS is not available, the queue manager will attempt to fail over to NEPTUN. If NEPTUN is not available, the queue manager will fail over to any other available node in the cluster.

3.
```
$ START/QUEUE/MANAGER/NEW_VERSION -
_$ /ON=(SATURN,VENUS,NEPTUN,*) DUA5:[SYSQUE])
   .
   .
   .
$ START/QUEUE/MANAGER
```

   The **START/QUEUE/MANAGER** command in this example creates the queue database as shown in the previous example. Suppose the queue manager started on node SATURN.

   Later, SATURN is removed from the cluster, and the queue manager fails over to node VENUS. When SATURN rejoins the cluster, the second **START/QUEUE/MANAGER** command in the example is entered to move the queue manager back to node SATURN.

   The second **START/QUEUE/MANAGER** command does not specify the DUA5:[SYSQUE] parameter value or the **/ON** qualifier and its node list because those previously supplied pieces of information are stored in the queue database. The queue manager continues to use the queue and journal files found at the location stored in its database. The **/ON** list, stored as a result of the previous **START/QUEUE/MANAGER** command, also remains unchanged.

4.
```
$ START/QUEUE/MANAGER DUA4:[SYSQUE]
%JBC-E-QMANNOTSTARTED, queue manager could not be started
$ SEARCH SYS$MANAGER:OPERATOR.LOG /WINDOW=5 QUEUE_MANAGE,JOB_CONTROL
%%%%%%%%%%  OPCOM  14-DEC-2001 18:55:18.23  %%%%%%%%%%
Message from user QUEUE_MANAGE on QMUNGR
%QMAN-E-OPENERR, error opening DUA4:[SYSQUE]SYS$QUEUE_MANAGER.QMAN
$QUEUES;

%%%%%%%%%%  OPCOM  14-DEC-2001 18:55:18.29  %%%%%%%%%%
Message from user QUEUE_MANAGE on QMUNGR
-RMS-F-DEV, error in device name or inappropriate device type for
 operation

%%%%%%%%%%  OPCOM  14-DEC-2001 18:55:18.31  %%%%%%%%%%
Message from user QUEUE_MANAGE on QMUNGR
-SYSTEM-W-NOSUCHDEV, no such device available
$ START/QUEUE/MANAGER DUA5:[SYSQUE]
```

   In this example, the first **START/QUEUE/MANAGER** command specifies device DUA4: as the location of the queue and journal files. The error message indicates that the queue manager does not start. The **SEARCH** command searches the operator log file for relevant messages, and reveals that device DUA4: does not exist. The second **START/QUEUE/MANAGER** command specifies the correct device name, DUA5:.

STOP

STOP — Terminates execution of a command, an image, a command procedure, a command procedure that was interrupted by a **Ctrl/Y** function, or a detached process or subprocess. **Requires GROUP privilege to stop other processes in the same group. Requires WORLD privilege to stop processes outside your group.**

Format

STOP [*process-name*]

Parameter

process-name

Requires that the process be in your group.

Specifies the name of the process to be deleted. The process name can have from 1 to 15 alphanumeric characters. If the *process-name* includes spaces or lowercase letters, enclose the name in quotation marks (" ") to preserve the correct spelling.

The specified process must have the same group number in its user identification code (UIC) as the current process; you cannot use the *process-name* parameter to stop a process outside of your group. To stop a process outside of your group, you must use the qualifier **/IDENTIFICATION=*pid***.

The process name is incompatible with the **/IDENTIFICATION** qualifier; if you use the **/IDENTIFICATION** qualifier, the process name is ignored. If you include neither the *process-name* parameter nor the **/IDENTIFICATION** qualifier with the **STOP** command, the image executing in the current process is terminated.

Description

The **STOP** command causes an abnormal termination of the image that is currently executing. If the image has declared user-mode exit-handling routines using calls to the $DCLEXH system service, by default these exit handlers are not invoked. If execution of exit-handling routines is required, use the **EXIT** command or the **STOP/EXIT** command to terminate the image so that the mode-specific exit handlers are invoked.

If the **STOP** command is executed from a noninteractive process (such as a batch job), the process terminates.

Note that when an image has been interrupted by a **Ctrl/Y** function and subsequently the DCL **RUN** command or a non-CLI-based DCL verb is entered, the interrupted image is terminated. However, in this case, exit-handling routines execute before the next image is run. For more information about process and image rundown processing, see the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/] or the *VSI OpenVMS Programming Concepts Manual, Volume I* [https://docs.vmssoftware.com/vsi-openvms-programming-concepts-manual-volume-i/] and *Volume II* [https://docs.vmssoftware.com/vsi-openvms-programming-concepts-manual-volume-ii/].

If you press **Ctrl/Y** to interrupt a command procedure and then enter the **STOP** command, or if the **STOP** command is executed in a command procedure, all command levels are unstacked and control returns to command level 0 (DCL level with the $ prompt).

If you specify a process name or process identification (PID) code, the **STOP** command terminates the image currently executing in the specified process and deletes the process. If the process is noninteractive, no notification of the deletion occurs and the log file for the job is not printed.

Qualifiers

/IDENTIFICATION=*pid*

Specifies the system-assigned process identification (PID) code. When you create a process with the **RUN** command, the **RUN** command displays the PID code of the newly created process. The **/IDENTIFICATION** qualifier can be used in place of the process name parameter.

You can omit any leading zeros in specifying the PID code.

/IMAGE [/IDENTIFICATION=*pid*] [*process-name*]

Calls the $FORCEX system service to stop the image of the target process specified in the process id or process name that is currently executing. The target process is not deleted.

If you omit the **/IDENTIFICATION** qualifier and the process name, the **STOP/IMAGE** command is identical to the **STOP** command.

/EXIT[=*access-mode*] (default)
/NOEXIT

Specifies an option to call exit handlers prior to deletion of the process.

The following table describes the access mode options:

Mode	Description
EXECUTIVE_MODE	Execute executive and more privileged mode exit handlers (default, if no access mode specified).
KERNEL_MODE	Execute kernel mode exit handlers.
SUPERVISOR_MODE	Execute supervisor and more privileged mode exit handlers.
USER_MODE	Execute user and more privileged mode exit handlers.

Examples

1.
```
$ RUN MYPROG
   .
   .
   .
Ctrl/Y
Interrupt
$ STOP
```

The **RUN** command in this example begins executing the image MYPROG. Subsequently, the **Ctrl/Y** function interrupts the execution. The **STOP** command then terminates the image.

2.
```
$ @TESTALL
   .
   .
   .
Ctrl/Y
Interrupt
$ STOP
```

The **@** (execute procedure) command in this example executes the procedure TESTALL.COM. Subsequently, the **Ctrl/Y** function interrupts the procedure. The **STOP** command then returns control to the DCL command interpreter.

3.
```
$ RUN/PROCESS_NAME=LIBRA  LIBRA
%RUN-S-PROC_ID, identification of created process is 0013340D
   .
   .
   .
$ STOP LIBRA
```

The **RUN** command in this example creates a subprocess named LIBRA to execute the image LIBRA.EXE. Subsequently, the **STOP** command causes the image to exit and deletes the process.

4.
```
$  ON ERROR THEN STOP
   .
   .
   .
```

In a command procedure, the **ON** command establishes a default action when any error occurs in the execution of a command or program. The **STOP** command stops all command levels. If this **ON** command is executed in a command procedure, which in turn is executed from within another procedure, control does not return to the outer procedure, but to DCL command level 0.

5.
```
$ STOP/EXIT
```

Terminates the process and runs exit handlers beginning at executive mode.

6.
```
$ STOP/IMAGE/ID=12345678
```

Terminates the current user image being executed by process 12345678.

STOP/CPU

STOP/CPU — Stops the specified secondary processor or processors (and any associated vector processors). The **/CPU** qualifier is required. **Applies only to OpenVMS multiprocessing systems. Requires CMKRNL(change mode to kernel) privilege.**

Format

STOP/CPU [*cpu-id*[,...]]

Parameter

***cpu-id*[,...]**

Specifies a decimal value representing the identity of a processor in an OpenVMS multiprocessing system. On an Alpha 7000 system, the CPU ID is the backplane slot number of the processor. If you do not specify a CPU ID, the **STOP/CPU** command selects a processor in the current active set to stop.

Description

The **STOP/CPU** command removes a secondary processor from the active set in an OpenVMS multiprocessing system. If the secondary processor is not executing a process when the **STOP/CPU** command is issued, it enters the STOPPED state. If the secondary is executing a process at the time, it continues to execute the current process until it attempts to schedule another process. When this occurs, the secondary enters the STOPPED state.

The OpenVMS operating system subjects a processor to a set of checks when it is the object of a **STOP/CPU** command. As a result, you may not be permitted to stop certain processors that are vital to the functioning of the system. In these cases, there is usually a process in the system that can execute only on the processor you intend to stop. You can determine this by issuing a **SHOW CPU/FULL** command. In unusual circumstances, you can bypass the checking mechanism by using the **/OVERRIDE_CHECKS** qualifier in the command.

The **STOP/CPU** command has no effect if its object processor is already in the STOPPED state when it is issued.

Qualifiers

/ALL

Stops all eligible secondary processors in the system's active set.

/ASSIGN=*option* (Alpha/IA-64 only)

Assigns specified processors to the hard partition node after they are stopped.

Option	Description
$$HARD_PARTITION	The configuration tree hard partition node. All instances running in the hard partition defined by this node have visibility and access to CPUs owned at this level.

Supported only on AlphaServer systems that support partitioning.

/MIGRATE (Alpha/IA-64 only)

Transfers ownership of the CPU from the current instance to another soft partition.

Option	Description
instance_name	The name of any valid running instance in the current hard partition.
partitionID	The numeric ID of any partition(reflected in the configuration tree) in the current hard partition. An operating system instance is not required to be running with this identifier.

Supported only on AlphaServer systems that support partitioning.

/OVERRIDE_CHECKS

Directs the **STOP/CPU** command to bypass a series of OpenVMS scheduling checks that determine whether the specified processor is eligible for removal from the active set.

Note that this is not an unconditional operation; other CPU load or configuration constraints may prevent the specified processor from being stopped.

/POWER=OFF (Alpha/IA-64 only)

Powers down the CPU after it is removed from the active set. The CPU will be powered down while still owned by the instance, prior to any assignments.

The **/POWER** qualifier cannot be used in conjunction with the **/MIGRATE** qualifier.

Supported only on AlphaServer GS series systems.

Examples

1. `$ STOP/CPU`

The **STOP/CPU** command in this example selects a processor and removes it from the multiprocessing system's active set.

2. `$ STOP/CPU 4,7`

   The **STOP/CPU** command in this example selects the processors with CPU IDs `4` and `7` and removes them from the multiprocessing system's active set.

3. `$ STOP/CPU/OVERRIDE_CHECKS 8`

   The **STOP/CPU/OVERRIDE_CHECKS** command in this example overrides some OpenVMS scheduling states that ordinarily prevent the operation and stops the processor with the CPU ID of `8`. Then it is removed from active participation in the multiprocessing system.

4. `$ STOP/CPU/ALL`

   The **STOP/CPU/ALL** command in this example stops all eligible secondary processors in the active set and removes them from the multiprocessing system.

5. `$ STOP/CPU/MIGRATE=WFGLXE 5`

   The **STOP/CPU/MIGRATE** command in this example removes CPU `5` from the current instance's active set and transfers ownership to instance `WFGLXE` in the current hard partition.

6. `$ STOP/CPU/ASSIGN=$$HARD 6`

   The **STOP/CPU/MIGRATE** command in this example removes CPU `6` from the current instance's active set and transfers ownership to the hard partition node in the configuration tree. The CPU is immediately available for assignment for any instance within the hard partition defined by that node.

STOP/NETWORK

STOP/NETWORK — Stops the specified network service on the local node. The **/NETWORK** qualifier is required.

Format

STOP/NETWORK *network-service*

Parameter

network-service

Specifies the name of the network service that you want to halt.

Description

The **STOP/NETWORK** command causes the specified network to stop executing on the local node.

Use the **START/NETWORK** command to release the network from the paused state.

Example

```
$ STOP/NETWORK DECnet
```

The **STOP/NETWORK** command in this example stops the current network service.

STOP/QUEUE

STOP/QUEUE — The **STOP/QUEUE** command causes the specified execution queue to pause. All jobs currently executing in the queue are suspended (until the queue is restarted with the **START/QUEUE** command), and no new jobs are initiated. The **/QUEUE** qualifier is required. **Requires manage (M) access to the queue.**

Format

STOP/QUEUE *queue-name*[:]

Parameter

***queue-name*[:]**

Specifies the name of the queue that you want to pause.

Description

The **STOP/QUEUE** command causes the specified queue to pause. All jobs currently executing in the queue are suspended. No new jobs are initiated.

Use the **START/QUEUE** command to release the queue from the paused state. When you restart the queue, any jobs that were executing resume executing at the point where they left off, unless you use the **/BACKWARD**, the **/FORWARD**, the **/SEARCH**, or the **/TOP_OF_FILE** qualifier to restart a print job at a different place.

For information about how to stop a queue, see the following commands:

STOP/QUEUE/ABORT
STOP/QUEUE/ENTRY
STOP/QUEUE/MANAGER/CLUSTER
STOP/QUEUE/NEXT
STOP/QUEUE/REQUEUE
STOP/QUEUE/RESET
STOP/QUEUES/ON_NODE

Examples

1. ```
   $ STOP/QUEUE LPA0
   ```

   The **STOP/QUEUE** command in this example suspends the current print job in the queue LPA0 and places that queue in the paused state.

2. ```
   $ STOP/QUEUE JADE_PRINT
   $ START/QUEUE/TOP_OF_FILE JADE_PRINT
   ```

   The **STOP/QUEUE** command in this example suspends the job that is currently printing on the printer queue JADE_PRINT and places that queue in the paused state. The **START/QUEUE** command releases the queue from the paused state. The **/TOP_OF_FILE** qualifier causes the job that was suspended to resume printing at the beginning of the file rather than at where it was interrupted.

STOP/QUEUE/ABORT

STOP/QUEUE/ABORT — Aborts a job that is printing or processing on an output queue, deletes it from the queue, and begins processing the first pending job in the queue. The **/QUEUE** qualifier is optional, but the **/ABORT** qualifier is required. **Requires delete (D) access to the current job.**

Format

STOP/QUEUE/ABORT *queue-name*[:]

Parameter

***queue-name*[:]**

Specifies the name of the queue containing the job you want to abort.

Description

When you abort a print job, the print job completes the page that is currently printing. Then the job is removed from the queue. If the output queue has been set up to put trailing pages at the end of jobs, a trailer page is printed after the current page is completed. For information on how a batch job is aborted, see the description of the **STOP/QUEUE/ENTRY** command.

Use the **STOP/QUEUE/ABORT** command to abort the current print job and to delete it from the queue. You do not specify a job entry number with the **/ABORT** qualifier, because output queues can have only one current job at a time.

Use the **STOP/QUEUE/ENTRY** command to abort one or more batch jobs that are executing currently on a queue and to delete them from the queue. To stop a batch job, you must specify an entry number because batch queues, unlike print queues, can have more than one job executing at the same time. (You also can use the **STOP/QUEUE/ENTRY** command to abort a print job that is printing or processing currently on a queue, and to delete it from the queue.)

Use the **STOP/QUEUE/REQUEUE** command to stop batch or print jobs and to requeue them. Use the **DELETE/ENTRY** command to delete an entry that is queued and awaiting execution.

Note

If you enter the **STOP/QUEUE/ABORT** command accidentally for a malfunctioning queue, enter the **STOP/QUEUE/RESET** command to stop the queue in an orderly fashion.

Examples

```
$ STOP/QUEUE/ABORT LPA0
```

This example aborts the current print job on the queue `LPA0`. The print symbiont begins to process the first pending job in the queue. Assuming there is no problem with the printer, the current page of the file completes printing. If the printer queue has been set up to put trailer pages at the end of jobs, a trailer page is printed after the current page is completed.

For batch queues an entry number must be provided. To abort a batch job, use the **STOP/QUEUE/ENTRY** command.

STOP/QUEUE/ENTRY

STOP/QUEUE/ENTRY — Aborts one or more jobs that are executing on a batch queue or printing on an output queue, deletes them from the queue, and begins processing the first pending job in the queue. The **/QUEUE** qualifier is optional, but the **/ENTRY** qualifier is required. **Requires delete (D) access to the specified job.**

Format

STOP/QUEUE/ENTRY=(*entry-number*[,...]) [*queue-name*[:]]

Parameters

***entry-number*[,...]**

Specifies the entry number (or a list of entry numbers) of jobs to be deleted. If you specify only one entry number, you can omit the parentheses. If you do not specify a queue name, you can delete entries from multiple queues.

The system assigns a unique entry number to each queued print or batch job in the system. By default, the **PRINT** and **SUBMIT** commands display the entry number when they successfully queue a job for processing. These commands also create or update the local symbol $ENTRY to reflect the entry number of the most recently queued job. To find a job's entry number, enter the **SHOW ENTRY** or the **SHOW QUEUE** command.

***queue-name*[:]**

Specifies the name of the queue that contains the jobs that you want to abort. The queue name can refer either to the queue to which the job was submitted or to the queue where the job is executing. The *queue-name* parameter is optional syntax; however, when you specify a queue name, the OpenVMS system uses it to verify an entry in the specific queue before stopping and deleting the entry.

Description

When you abort a batch job, the system tries to stop the job in an orderly fashion by closing any open files and sending a message to the log file. For information on how a print job is aborted, see the description of the **STOP/QUEUE/ABORT** command.

Use the **STOP/QUEUE/ENTRY** command to abort one or more batch jobs that are executing currently on a queue and to delete them from the queue. To stop a batch job, you must specify an entry number because batch queues, unlike print queues, can have more than one job executing at the same time. (You also can use the **STOP/QUEUE/ENTRY** command to abort a print job that is currently printing or processing on a queue and to delete it from the queue.)

Use the **STOP/QUEUE/REQUEUE** command to stop batch or print jobs and to requeue them. Use the **DELETE/ENTRY** command to delete an entry that is queued and is awaiting execution.

Note

If you enter the **STOP/QUEUE/ENTRY** command accidentally for a malfunctioning queue, enter the **STOP/QUEUE/RESET** command to stop the queue in an orderly fashion.

Example

```
$ STOP/QUEUE/ENTRY=365 SYS$BATCH
```

The **STOP/QUEUE/ENTRY** command in this example aborts batch job number 365 currently executing on the SYS$BATCH queue and begins the first pending job in the queue.

STOP/QUEUE/MANAGER/CLUSTER

STOP/QUEUE/MANAGER/CLUSTER — Shuts down the queue manager on a standalone node or an OpenVMS Cluster. The **/QUEUE** qualifier is optional, but the **/MANAGER** and **/CLUSTER** qualifiers are required on both standalone and clustered systems. By default, the command affects the default queue manager, SYS$QUEUE_MANAGER. Specify the **/NAME_OF_MANAGER** qualifier to shut down a queue manager other than the default. **Requires OPER (operator) and SYSNAM (system logical name) privileges.**

Format

STOP/QUEUE/MANAGER/CLUSTER

Parameters

None.

Description

The **STOP/QUEUE/MANAGER/CLUSTER** command notifies the queue manager to perform the following on the standalone node or on all nodes in the cluster:

- Abort all of the current jobs that cannot be restarted, and requeue all of the current restartable jobs on the queue manager
- Stop all of the execution queues on the queue manager
- Disable the autostart feature for all of the queue manager's autostart queues
- Close the queue manager's queue and journal files

Once the **STOP/QUEUE/MANAGER/CLUSTER** command is entered, the affected queue manager process remains stopped, and requests to the queuing system are denied until the DCL command **START/QUEUE/MANAGER** is entered.

Qualifier

/NAME_OF_MANAGER=*name*

Specifies the name of the queue manager process to be stopped. If the **/NAME_OF_MANAGER** qualifier is omitted, the default queue manager name SYS$QUEUE_MANAGER is used. For more information, see the chapter about the queue manager in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#_6017QUEUE_MGR].

Example

```
$ STOP/QUEUE/MANAGER/CLUSTER
```

The **STOP/QUEUE/MANAGER/CLUSTER** command in this example stops the queue manager process, SYS$QUEUE_MANAGER. The process remains stopped and requests to SYS$QUEUE_MANAGER will be denied until the **START/QUEUE/MANAGER** command is entered.

This command stops the default queue manager SYS$QUEUE_MANAGER because the **/NAME_OF_MANAGER** qualifier is not specified.

STOP/QUEUE/NEXT

STOP/QUEUE/NEXT — Stops the specified queue after all executing jobs have completed processing. No new jobs are initiated. The **/QUEUE** qualifier is optional, but you must specify the **/NEXT** qualifier. **Requires manage (M) access to the queue.**

Format

STOP/QUEUE/NEXT *queue-name*[:]

Parameter

***queue-name*[:]**

Specifies the name of the queue that you want to stop.

Description

The **STOP/QUEUE/NEXT** command stops the queue after it allows any current jobs to complete execution. No new jobs are initiated.

You should use the **STOP/QUEUE/NEXT** command before deassigning, deleting, or merging a queue, as this command allows all currently executing jobs to complete processing before changes are made to the queue. You might also use this command to stop a queue if you need to perform planned service on a printing device. Use the **START/QUEUE** command to restart the queue.

For autostart queues, the **STOP/QUEUE/NEXT** command stops the queue and makes it inactive for autostart, so it will not be automatically started. For this reason, you might use this command to prevent an autostart queue from being accidentally restarted when a printing device needs to undergo planned service. Use the **START/QUEUE** command to reactivate the queue for autostart.

Examples

1.
```
$ STOP/QUEUE/NEXT LASER_PRINT
```

   In this example, the **STOP/QUEUE/NEXT** command prepares to stop the queue LASER_PRINT. The currently printing job is allowed to complete, but no new job is allowed to initiate. Once the current job has finished, the queue is stopped.

   If LASER_PRINT is an autostart queue, it will not be automatically started until you enter the **START/QUEUE** command.

2.
```
$ STOP/QUEUE/NEXT BATCH_1
$ SHOW QUEUE/ALL BATCH_1
Batch queue BATCH_1, stopped, on JADE::
$ DELETE/QUEUE BATCH_1
```

   This example shows how to delete the batch queue BATCH_1. First, the **STOP/QUEUE/NEXT** command is entered, which stops the queue after jobs currently executing on the queue are completed. Second, the **SHOW QUEUE/ALL** command is entered to ensure that no jobs are pending in the queue. The screen display shows that no jobs are pending. Finally, the **DELETE/QUEUE** command is entered to delete the queue BATCH_1.

STOP/QUEUE/REQUEUE

STOP/QUEUE/REQUEUE — Stops the current jobs on the specified queue and requeues them for later processing. The queue does not stop; processing of the first pending job in the queue begins. The **/QUEUE** qualifier is optional, but the **/REQUEUE** qualifier is required. The **/ENTRY** qualifier is required to requeue batch jobs. **Requires delete (D) access to the current or specified job.**

Format

STOP/QUEUE/REQUEUE[=*queue-name*] *queue-name*[:][=*queue-name*] /ENTRY=(*entry-number*[,...])*queue-name*[:]

Parameters

***queue-name*[:]**

Specifies the name of the queue that contains the jobs that you want to stop. When you specify a queue name as a parameter for the **/REQUEUE** qualifier, the jobs are requeued to that queue; otherwise, the jobs are requeued in the current queue.

***entry-number*[,...]**

Specifies the entry number (or a list of entry numbers) of the jobs you want to requeue. If you specify only one entry number, you can omit the parentheses.

The system assigns a unique entry number to each queued print or batch job in the system. By default, the **PRINT** and **SUBMIT** commands display the entry number when they successfully queue a job for processing. These commands also create or update the local symbol $ENTRY to reflect the entry number of the most recently queued job. To find a job's entry number, enter the **SHOW ENTRY** or **SHOW QUEUE** command.

Description

The **STOP/QUEUE/REQUEUE** command causes the system to requeue the jobs for later processing or for execution in the queue. If the print symbiont has sent checkpoint information about a print job to the job controller, printing resumes at the last checkpoint recorded, unless you enter the **SET ENTRY/NOCHECKPOINT** command before the job is reinitiated. Batch jobs generally restart at the beginning; however, batch jobs that contain **SET RESTART_VALUE** commands run only those portions of the job that have not completed successfully.

Use the **STOP/QUEUE/ABORT** command to stop a current print job without requeueing it. Use the **STOP/QUEUE/ENTRY** command to stop a current batch job without requeueing it. You also

can use the **STOP/QUEUE/ENTRY** command to stop a print job without requeueing it. Use the **DELETE/ENTRY** command to delete an entry that is queued and is waiting to be executed or processed.

Note

If you enter the **STOP/QUEUE/REQUEUE** command accidentally for a malfunctioning queue, enter the **STOP/QUEUE/RESET** command to stop the queue in an orderly fashion.

Qualifiers

/ENTRY=(*entry-number*[,...])

Specifies the entry number of one or more jobs you want to abort. If you specify only one entry number, you can omit the parentheses.

The system assigns a unique entry number to each queued print or batch job in the system. By default, the **PRINT** and **SUBMIT** commands display the entry number when they successfully queue a job for processing. These commands also create or update the local symbol $ENTRY to reflect the entry number of the most recently queued job. To find a job's entry number, enter the **SHOW ENTRY** or the **SHOW QUEUE** command.

You must use the **/ENTRY** qualifier when you enter the **STOP/QUEUE/REQUEUE** command for a batch queue. Entry numbers specified must match entry numbers of executing jobs.

/HOLD

Places the aborted job or jobs in a hold state for later release with the **SET ENTRY/RELEASE** or **SET ENTRY/NOHOLD** command.

/PRIORITY=*n*

Requires OPER (operator) or ALTPRI (alter privilege) privilege to raise the priority value above the value of the system parameter MAXQUEPRI.

Changes the priority of the requeued job or jobs. The parameter *n* can be from 0 to 255; the default value of the *n* parameter is the same as the priority value that the job or jobs had when they were stopped.

Generally, the **/PRIORITY** qualifier is used to lower the priority of a job or jobs, which ensures that the job or jobs runs when the queue contains no other jobs. No privilege is needed to set the priority lower than the MAXQUEPRI value.

/REQUEUE=[*queue-name*]

Specifies a queue where current jobs are to be requeued. If you do not specify the *queue-name* parameter, the jobs are requeued in the current queue.

Examples

1. `$ STOP/QUEUE/REQUEUE=LPB0 LPA0`

   In this example, the current print job on queue `LPA0` is stopped and requeued to queue `LPB0`. If the print symbiont sent checkpoint information about the print job to the job controller, printing resumes on `LPB0` at the last checkpoint recorded.

2. (wide)
```
$ STOP/QUEUE/REQUEUE/HOLD LPA0
   .
   .
   .
$ SET ENTRY 254/RELEASE
```

   In this example, the current print job on LPA0 is suspended and placed in the hold state. Later, when you enter the **SET ENTRY** command with the **/RELEASE** qualifier, the job is released from the hold state and is rescheduled to print on queue LPA0. If the print symbiont sent checkpoint information about the print job to the job controller, printing resumes at the last checkpoint.

3. `$ STOP/QUEUE/REQUEUE/ENTRY=758 SYS$BATCH`

   In this example, batch job number 758 is stopped and requeued for later processing on SYS $BATCH. If the batch job has been programmed with appropriate **SET RESTART_VALUE** commands, those portions of the job that have completed successfully are not rerun.

STOP/QUEUE/RESET

STOP/QUEUE/RESET — Abruptly stops the queue and returns control to the system. Any jobs that cannot be restarted are aborted immediately. The **/QUEUE** qualifier is optional, but you must specify the **/RESET** qualifier. **Requires manage (M) access to the queue.**

Format

STOP/QUEUE/RESET *queue-name*[:]

Parameter

***queue-name*[:]**

Specifies the name of the queue you want to reset.

Description

The **STOP/QUEUE/RESET** command stops the queue as soon as the system receives the command. The queue manager requests termination for all executing jobs, but aborts or requeues executing jobs without waiting for termination status to be received. Current jobs that can be restarted are requeued for processing. Current jobs that cannot be restarted are aborted and must be resubmitted for processing. (Print jobs are restartable by default. Batch jobs are restartable if submitted with the **SUBMIT/RESTART** command.) Use the **START/QUEUE** command to restart the queue.

For autostart queues, the **STOP/QUEUE/RESET** command stops the queue and makes it inactive for autostart, so it will not be automatically started. For this reason, you might use this command to prevent an autostart queue from being automatically restarted when a printing device fails and needs to be serviced. Use the **START/QUEUE** command to reactivate the queue for autostart.

Examples

1.
```
$ INITIALIZE/QUEUE/AUTOSTART_ON=MYNODE::LPAO/START LPA0
$ ENABLE AUTOSTART/QUEUES/ON_NODE=MYNODE
```

```
   .
   .
   .
$ STOP/QUEUE/RESET LPA0
$ START/QUEUE LPA0
```

The **INITIALIZE/QUEUE** command in this example creates an autostart output queue named LPA0. The **/START** qualifier activates the queue for autostart, and the **ENABLE AUTOSTART/QUEUES** command causes the queue to begin executing.

Suppose the printer LPA0**::** runs out of paper. The **STOP/QUEUE/RESET** command abruptly stops the queue LPA0. The current print job stops immediately and is requeued. The **STOP/QUEUE/RESET** command also makes the queue inactive for autostart, so it will not be restarted while the printer's paper is being resupplied.

After the printer is loaded with paper, the **START/QUEUE** command makes the queue active for autostart and allows the queue manager to automatically start it again.

2. `$ STOP/QUEUE/RESET TEXBATCH`

   The **STOP/QUEUE/RESET** command in this example stops the batch queue TEXBATCH. Any current jobs that were submitted with the **/RESTART** qualifier are requeued for processing when the queue is restarted. Users might want to resubmit current jobs that were not originally submitted with the **/RESTART** qualifier.

STOP/QUEUES/ON_NODE

STOP/QUEUES/ON_NODE — Stops all queues on the specified node. This command is useful if you want to stop a set of queues on a node with a single command (for example, before shutting down a node), but do not want to stop a queue manager throughout an OpenVMS Cluster. The **/QUEUES** qualifier is optional, but the **/ON_NODE** qualifier is required. **Requires OPER (operator) privilege.**

Format

STOP/QUEUES/ON_NODE [=*node*]

Parameters

None.

Description

The **STOP/QUEUES/ON_NODE** command notifies the queue manager to perform the following tasks:

- Abort all current jobs that cannot be restarted (in all of that queue manager's execution queues on the appropriate node), and requeue all current restartable jobs.
- Stop all of that queue manager's execution queues on the appropriate node. Force any of that queue manager's autostart queues on the node to fail over to the next available node in the queue's failover list (if any) on which autostart is enabled.
- Prevent any of that queue manager's autostart queues from failing over to the node.

By default, this command affects the node from which it is entered. Specify a nodename with the **/ON_NODE** qualifier to stop queues on a different node.

By default, the command affects autostart queues managed by the default queue manager, SYS$QUEUE_MANAGER. Specify the **/NAME_OF_MANAGER** qualifier to disable autostart of a different queue manager's autostart queues (on the node).

The **STOP/QUEUES/ON_NODE** command affects all autostart and nonautostart execution queues on the appropriate node that are managed by the queue manager. Autostart queues stopped as a result of this command remain active for autostart and will be restarted when the **ENABLE AUTOSTART** command is entered for the affected node or a node to which the queue can fail over. Nonautostart queues stopped as a result of this command must each be started with a **START/QUEUE** command specifying the queue name.

If you are using autostart queues, you might want to enter the **DISABLE AUTOSTART/QUEUES** command. For more information on the relationship between **DISABLE AUTOSTART/QUEUES** and **STOP/QUEUES/ON_NODE**, see the relevant section in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#STOPQ_SHUTDOWN].

The **STOP/QUEUES/ON_NODE** command is included in the shutdown command procedure SYS$SYSTEM:SHUTDOWN.COM. If you shut down a node without using SHUTDOWN.COM, you might want to enter the **STOP/QUEUES/ON_NODE** command first. For more information on stopping queues before shutting down a node, see the relevant section in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#_6017QUEUES].

Qualifier

/NAME_OF_MANAGER=*name*

Specifies the name of the queue manager controlling the queues you want to stop.

If the **/NAME_OF_MANAGER** qualifier is omitted, then the default queue manager name SYS$QUEUE_MANAGER is used. For more information on multiple queue managers, see the relevant section in the *VSI OpenVMS System Manager's Manual, Volume 1: Essentials* [https://docs.vmssoftware.com/vsi-openvms-system-manager-s-manual-volume-1-essentials/#_6017QUEUES].

Examples

1.
```
$ STOP/QUEUES/ON_NODE
```

   The **STOP/QUEUES/ON_NODE** command in this example stops all queues on the node from which it is entered. The autostart feature is disabled on this node for all autostart queues.

   However, the queue manager process continues to run and schedules jobs as requested for execution on its unstopped queues on other nodes in the OpenVMS Cluster.

   This command only affects queues managed by the default queue manager SYS$QUEUE_MANAGER because the **/NAME_OF_MANAGER** qualifier is not specified.

2.
```
$ INITIALIZE/QUEUE -
_$ /AUTOSTART_ON=(JADE::,RUBY::,OPAL::)/BATCH/START BATCH_1
$ENABLE AUTOSTART/QUEUES/ON_NODE=JADE
```

```
$ENABLE AUTOSTART/QUEUES/ON_NODE=RUBY
  .
  .
  .
$ STOP/QUEUES/ON_NODE=JADE
$ SHOW QUEUE BATCH_1
Batch queue BATCH_1, idle, on RUBY::
```

The **INITIALIZE/QUEUE** command in this example creates the autostart queue BATCH_1 capable of running on node JADE, RUBY, or OPAL. The **/START** qualifier activates the queue for autostart. The first **ENABLE AUTOSTART/QUEUES** command enables autostart of all autostart queues on node JADE, causing the queue BATCH_1 to begin processing on that node. The second **ENABLE AUTOSTART/QUEUES** command enables autostart for all autostart queues on node RUBY.

Later, suppose node JADE needs to be removed from the cluster. The **STOP/QUEUES/ON_NODE** command in the example stops all queues on node JADE without stopping the clusterwide queue manager, and causes the autostart queue BATCH_1 to failover to node RUBY, the next available node in its failover list. Because the **STOP QUEUES/ON_NODE** command disables autostart, the **ENABLE AUTOSTART/QUEUES** command must be executed on node JADE when it reboots for autostart queues to run on that node in the future.

This command only affects queues managed by the default queue manager SYS$QUEUE_MANAGER because the **/NAME_OF_MANAGER** qualifier is not specified.

SUBMIT

SUBMIT — Queues one or more files containing command procedures to a batch queue. **Requires submit (S) access to the queue.**

Format

SUBMIT *filespec*[,...]

Parameter

***filespec*[,...]**

Specifies one or more files containing command procedures. The asterisk (*) and the percent sign (%) wildcard characters are allowed in the directory specification, file name, file type, and version number fields. The default file type is that of the preceding file. If no previous file specification contains an explicit file type, the default file type is .COM.

If you specify more than one file, separate the file specifications with either commas (,) or plus signs (+).

If you specify a node name, you must use the **/REMOTE** qualifier.

Description

The **SUBMIT** command places the specified files in a batch queue for subsequent execution. By default, this queue is SYS$BATCH. All files queued by a single **SUBMIT** command are processed serially as one job. By default, the name of the batch job is the name of the first file specified in the **SUBMIT** command.

The system assigns a unique entry number to each batch job in the queue. When you enter the **SUBMIT** command, by default, the system displays the job name, the queue name, the entry number, and the job status.

The system automatically creates or updates the local symbol $ENTRY when a **SUBMIT** command completes successfully. The value of $ENTRY is a string that identifies the entry number of the most recently queued job. If you want to refer to a job's entry number later, store the value of $ENTRY in another symbol.

After you queue a batch job, the version of the file submitted is processed, even if a newer version of the file is created before the batch job runs. Also, another file with the same name and version number as the file queued cannot be substituted for the file that was submitted.

When the system executes a command procedure submitted to a batch queue, it creates a detached process to execute the commands. This process receives the owner's disk and directory defaults and the same resource quotas and privileges that were given to the owner's interactive process when you logged in.

If multiple procedures are submitted, the job terminates as soon as any procedure exits with an error or severe (fatal) error status.

In an OpenVMS Cluster environment, you should periodically keep the cluster in time synchronization using the **SET TIME/CLUSTER** command because the differences in system times between nodes in an OpenVMS Cluster may vary. This variance may cause a job to begin executing a little before or after the specified time because the redundancy built into the batch and print system allows more than one job controller in the cluster to receive a timer asynchronous system trap (AST) for the job and, thus, to schedule it for execution.

Batch Job Output

By default, all output from each command procedure in a batch job is written to a log file. This log file is queued for printing on SYS$PRINT when the batch job finishes. By default, this log file is deleted after it is printed. When you stop a batch job by using the **STOP/QUEUE/ENTRY**, the **STOP/IDENTIFICATION**, or the **DELETE/ENTRY** command, the log file is not queued for printing.

For information on how to change the system's handling of the log file, see the descriptions of the **/LOG_FILE**, **/PRINTER**, and **/KEEP** qualifiers.

Qualifiers

/AFTER=*time*
/NOAFTER

Requests that the job be held until after a specific time. If the specified time has passed already, the job is processed immediately.

You can specify time as either absolute time or as a combination of absolute and delta times. For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

In an OpenVMS Cluster, a batch job submitted to execute at a specific time may begin execution a little before or after the requested time. This occurs when the clocks of the member systems in the OpenVMS Cluster are not synchronized. For example, a job submitted using the DCL command **SUBMIT/AFTER=TOMORROW** may execute at 11:58 P.M. relative to the host system's clock.

This problem can occur in a cluster even if a job is run on the same machine from which it was submitted, because the redundancy built into the batch/print system allows more than one job controller in the cluster to receive a timer asynchronous system trap (AST) for the job and, thus, to schedule it for execution. Moreover, this behavior is exacerbated if the batch job immediately resubmits itself to run the next day using the same **SUBMIT** command. This can result in having multiple instances of the job executing simultaneously because TOMORROW (after midnight) might be only a minute or two in the future.

A solution to this problem is to place the **SUBMIT** command in a command procedure that begins with a **WAIT** command, where the delta-time specified in the **WAIT** command is greater than the maximum difference in time between any two systems in the cluster. Use the **SHOW TIME** command on each system to determine this difference in time. Use the SYSMAN command **CONFIGURATION SET TIME** to synchronize clocks on the cluster. For complete information on the SYSMAN command **CONFIGURATION SET TIME**, see the relevant section in the *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/#d0e30791].

/BACKUP
/NOBACKUP

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/BACKUP** qualifier selects files according to the dates of their most recent backups. This qualifier is incompatible with the **/CREATED**, **/EXPIRED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/BEFORE[=*time*]
/NOBEFORE

Selects only those files dated prior to the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the **/BEFORE** qualifier to indicate the time attribute to be used as the basis for selection: **/BACKUP**, **/CREATED** (default), **/EXPIRED**, or **/MODIFIED**.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/BY_OWNER[=*uic*]
/NOBY_OWNER

Selects only those files whose owner user identification code (UIC) matches the specified owner UIC. The default UIC is that of the current process.

Specify the UIC by using standard UIC format as described in the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#UIC_SECTION].

/CHARACTERISTICS=(*characteristic*[,...])

Specifies the name or numbers of one or more characteristics to be associated with the job. Characteristics can refer to such things as color of ink. If you specify only one characteristic, you can omit the parentheses.

A characteristic's number must range from 0 to 127. To see which characteristics have been defined for your system, use the **SHOW QUEUE/CHARACTERISTICS** command. To see which characteristics are associated with a particular queue, use the **SHOW QUEUE/FULL** command.

A job can be executed on a queue only if none, some, or all of the characteristics associated with the queue also are associated with the job, that is, the job's characteristics must be a subset of the queue's characteristics. However, if any of the characteristics associated with the job are not associated with the queue, the job remains pending until one or more of the following occurs:

- The characteristics specified with the queue are changed to make the job's characteristics a subset of the queue's characteristics (using, for example, the **SET QUEUE/CHARACTERISTICS** command).
- The characteristics specified with the job are changed to make the job's characteristics a subset of the queue's characteristics (using, for example, the **SET ENTRY/CHARACTERISTICS** command).
- The job is moved to a queue on which all the job's characteristics have been specified (using, for example, the **SET ENTRY/REQUEUE** command).
- The job is deleted (using, for example, the **DELETE/ENTRY** command).

/CLI=*filename*

Specifies the command language interpreter (CLI) to be used to process the job. The file specification assumes the device name SYS$SYSTEM: and the file type .EXE (SYS$SYSTEM:*filename*.EXE). The default CLI is that defined in the user authorization file (UAF).

/CONFIRM
/NOCONFIRM (default)

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

YES	NO	QUIT
TRUE	FALSE	**Ctrl/Z**
1	0	ALL
	Return	

You can use any combination of uppercase and lowercase letters for word responses. You can abbreviate word responses to one or more letters (for example, T, TR, or TRU for TRUE), but these abbreviations must be unique. Affirmative answers are YES, TRUE, and 1. Negative answers include: NO, FALSE, 0, and pressing **Return**. Entering QUIT or pressing **Ctrl/Z** indicates that you want to stop processing the command at that point. When you respond by entering ALL, the command continues to process, but no further prompts are given. If you type a response other than one of those in the list, DCL issues an error message and redisplays the prompt.

/CPUTIME=*time*

Defines a CPU time limit for the batch job. You can specify time as delta time, 0, INFINITE, or NONE. If the queue on which the job executes has a defined CPUMAXIMUM value, the smaller of the **SUBMIT** command and queue values is used. If the queue on which the job executes does

not have a specified maximum CPU time limit, the smaller of the **SUBMIT** command and user authorization file (UAF) values is used. If the queue on which the job executes does not have a specified maximum CPU time limit and the UAF has a specified CPU time limit of NONE, either the value 0 or the keyword INFINITE allows unlimited CPU time. If you specify the keyword NONE, the specified queue or UAF value is used. CPU time values must be greater than or equal to the number specified by the system parameter PQL_MCPULM. For more information on specifying CPU time limits, see the "CPU Time Limit Specifications and Actions" table for the **INITIALIZE/QUEUE** command in the *VSI OpenVMS DCL Dictionary: A-M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_119].

/CREATED (default)
/NOCREATED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/CREATED** qualifier selects files based on their dates of creation. This qualifier is incompatible with the **/BACKUP**, **/EXPIRED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/DELETE
/NODELETE (default)

Positional qualifier.

Controls whether files are deleted after processing. If you specify the **/DELETE** qualifier after the **SUBMIT** command name, all files in the job are deleted after processing. If you specify the **/DELETE** qualifier after a file specification, only that file is deleted after it is processed.

For the **/DELETE** qualifier to work, the protection code on the input files must allow delete (D) access to the user identification code (UIC) of the user submitting the job. If delete (D) access is not allowed, or if any file in the command parameter list cannot be opened as input, command processing stops and no batch job is created.

/EXCLUDE=(*filespec*[,...])
/NOEXCLUDE

Excludes the specified files from the submit operation. You can include a directory but not a device in the file specification. The asterisk (*) and the percent sign (%) wildcard characters are allowed in the file specification. If you specify only one file, you can omit the parentheses.

/EXPIRED
/NOEXPIRED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/EXPIRED** qualifier selects files according to their expiration dates. (The expiration date is set with the **SET FILE/EXPIRATION_DATE** command.) The **/EXPIRED** qualifier is incompatible with the **/BACKUP**, **/CREATED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/HOLD
/NOHOLD (default)

Controls whether the job is made available for immediate processing. The **/HOLD** qualifier holds the job until it is released by the **SET ENTRY/RELEASE** or the **SET ENTRY/NOHOLD** command.

/IDENTIFY (default)
/NOIDENTIFY

Displays the job name, the queue name, the entry number, and the status of the job when it is queued.

/KEEP
/NOKEEP

Controls whether the log file is deleted after it is printed; the **/NOKEEP** qualifier is the default unless the **/NOPRINTER** qualifier is specified.

/LOG_FILE[=*filespec*]
/NOLOG_FILE

Names the log file. The asterisk (*) and the percent sign (%) wildcard characters are not allowed in the file specification.

When you use the **/LOG_FILE** qualifier, the system writes the batch job's output to the file you specify. If you use the **/NOLOG_FILE** qualifier, no log file is created. By default, a log file is created, is written to the directory defined by the logical name SYS$LOGIN in the UAF, and is given the batch job's name as its file name with a file type of .LOG.

You can use the **/LOG_FILE** qualifier to write the log file to a different device. Logical names in the file specification are translated in the context of the process that submits the job. The process executing the batch job must have access to the device on which the log file will reside.

If you omit the **/LOG_FILE** qualifier and specify the **/NAME** qualifier, the log file is written to a file having the same file name as that specified by the **/NAME** qualifier with the file type .LOG. When you omit the **/LOG_FILE** qualifier, the job-name value used with the **/NAME** qualifier must be a valid file name.

/MODIFIED
/NOMODIFIED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/MODIFIED** qualifier selects files according to the dates on which they were last modified. This qualifier is incompatible with the **/BACKUP**, **/CREATED**, and **/EXPIRED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time modifiers, the default is the **/CREATED** qualifier.

/NAME=*job-name*

Names the job (and possibly the batch job log file). The job name must be 1 to 39 alphanumeric characters. If characters other than alphanumeric characters, underscores (_), or dollar signs ($) are used in the name, enclose the name in quotation marks (" "). The default job name is the name of the first file in the job.

If you omit the **/LOG_FILE** qualifier, the job-name value must be a valid file name. The **SHOW ENTRY** and **SHOW QUEUE/FULL** commands display the job name.

/NOTE=*text*

Specifies a message string of up to 255 characters. This message string appears as part of the display on a **SHOW QUEUE/FULL** command and can thus be used to convey information concerning a job.

The message specified with the **/NOTE** qualifier is also printed on the flag page of the log file (if printed).

/NOTIFY
/NONOTIFY (default)

Controls whether a message is broadcast to your terminal session when the job is completed or aborted.

/PARAMETERS=(*parameter*[,...])

Provides the values of up to eight optional parameters (equated to the symbols P1 to P8, respectively, for each command procedure in the job). The symbols are local to the specified command procedure.

Commas (,) separate individual parameters. If you specify only one parameter, you can omit the parentheses.

If the parameter contains spaces, special characters, or lowercase characters, enclose it in quotation marks. The size of the parameter can be from 1 to 255 characters.

/PRINTER[=*queue-name*] (default)
/NOPRINTER

Queues the job log file for printing when your job is completed. The **/PRINTER** qualifier allows you to specify a particular print queue; the default print queue is SYS$PRINT. If you specify the **/NOPRINTER** qualifier, the **/KEEP** qualifier is assumed.

When batch and output queues are managed by the same queue manager, the queue manager checks to ensure that the queue specified on the **SUBMIT/PRINTER** command is an output queue and that the user has access to the output queue. These checks are not made if the batch queue specified by the **SUBMIT** command and the output queue specified by the **/PRINTER** qualifier are managed by different queue managers.

If you explicitly specify an output queue for the log file when submitting a batch job, be sure the queue you specify with the **/PRINTER** qualifier is an output queue and not a batch queue. Also, be sure that you have access to the output queue.

/PRIORITY=*n*

Requires OPER (operator) or ALTPRI (alter privilege) privilege to specify a priority greater than the queue's maximum scheduling priority.

Specifies the job-scheduling priority for the batch job with respect to other jobs in the same queue. The value of *n* is an integer in the range of 0 to 255, where 0 is the lowest priority and 255 is the highest.

The default value is the value of the system parameter DEFQUEPRI. No privilege is needed to set the priority lower than the queue's maximum scheduling priority, which is the value of the system parameter MAXQUEPRI.

The **/PRIORITY** qualifier has no effect on the job's process execution priority. The job's process execution priority is determined by the base priority attribute of the **INITIALIZE/QUEUE/BASE_PRIORITY** command.

/QUEUE=*queue-name*[:]

Identifies the batch queue on which the job is entered. The default queue is SYS$BATCH.

/RAD=*n*

Specifies the RAD number on which the submitted batch job is to execute.

When a job is submitted to a batch queue that does not have a RAD setting, the job will execute using the RAD specified. If the batch queue has a RAD setting, the job will execute using the RAD specified on the queue.

When you specify a RAD on a batch job, the job controller creates the process with the $CREPRC *home_rad* argument set to the RAD value on the job. If the RAD specified on the job is invalid on the target system, the job fails to execute and the job controller outputs a message to the operator console indicating that a bad RAD was specified. If the bad RAD value matches the RAD setting on the batch queue, the batch queue is stopped and the job remains in the queue.

RAD is supported on AlphaServer GS series systems and, starting from OpenVMS Version 8.4, support is extended to NUMA-capable IA-64 systems.

/REMOTE

Queues the job to SYS$BATCH on the remote node specified. When you use the **/REMOTE** qualifier, you *must* include the node name in the file specification.

Note that, unlike submitting a job on a local node, multiple command procedures queued by a single **SUBMIT/REMOTE** command are considered separate jobs.

You can specify only the following qualifiers with the **/REMOTE** qualifier:

/BACKUP	**/BEFORE**	**/BY_OWNER**	**/CONFIRM**
/CREATED	**/EXCLUDE**	**/EXPIRED**	**/MODIFIED**
/SINCE			

/RESTART
/NORESTART (default)

Indicates whether the job restarts after a system failure or after a **STOP/QUEUE/REQUEUE** command.

/RETAIN=*option*

Specifies the circumstances under which you want your jobs to be retained in a queue. When a job is retained in the queue, you can issue the **SHOW QUEUE** command after the job completes to see the status of the job. Without job retention, no record of a job is left in a queue after a job completes.

Use the following options to specify job retention:

- ALWAYS – Holds the job in the queue regardless of the job's completion status.
- DEFAULT – Holds the job in the queue as specified by the queue's retention policy.
- ERROR – Holds the job in the queue only if the job completes unsuccessfully.

- UNTIL=*time-value*– Holds the job in the queue for the specified length of time, regardless of the job's completion status.

Note

You cannot specify a **/NORETAIN** qualifier with the **SUBMIT** command (as system managers can with the commands **INITIALIZE/QUEUE**, **START/QUEUE**, and **SET QUEUE**); however, you can specify **/RETAIN=DEFAULT** with the **SUBMIT** command. The default option holds the job in the queue as specified by the queue's retention policy. If the system manager has not specified retention for the queue, the job is not retained.

How Job Retention Is Determined

Although you can specify job retention options for your own jobs, the job retention option you specify may be overridden by the job retention option of the queue on which your job executed. If you submit or print a job to a generic queue, the generic queue's job retention setting may also override the job retention option you specify. This section describes how job retention is determined.

An execution queue's job retention setting takes precedence over a generic queue's job retention setting; however, if the job's completion status does not match the job retention setting (if any) on the execution queue, then the generic queue's job retention setting attempts to control job retention. If the job's completion status does not match the job retention setting (if any) on the generic queue, then the user-specified job retention setting is used. Jobs submitted directly to execution queues are not affected by job retention settings on generic queues.

If the execution queue's retention setting applies, the job is retained on the execution queue. Likewise, if the generic queue's retention setting applies, the job is retained on the generic queue. If the user-specified setting applies, the job is retained in the queue to which it was submitted.

The following example describes how the queue manager determines job retention:

Suppose you submit a job to a generic queue and specify **/RETAIN=ALWAYS**, and the job completes successfully.

First, the queue manager compares the job's completion status to the execution queue's retention setting. Suppose the queue is set with **/RETAIN=ERROR** (retains only jobs that complete unsuccessfully). The job is not retained in the execution queue because the error condition was not met.

The queue manager then compares the job's completion status to the generic queue's retention setting. Suppose the generic queue has no retention setting. The queue manager's comparison again fails to retain the job.

Finally, the queue manager compares the job's completion status to the retention setting you specified for the job. This comparison reveals that the job should be retained. Because the user-specified setting leads the queue manager to retain the job, the job is held in the queue to which the job was submitted – in this case, the generic queue.

For more information on types of queues, see the **INITIALIZE/QUEUE** command in the *VSI OpenVMS DCL Dictionary: A-M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_119]. For more information on setting retention options for queues, see the **START/QUEUE** or **SET QUEUE** command, or the **INITIALIZE/QUEUE** command in the *VSI OpenVMS DCL Dictionary: A-M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_119].

Timed Retention

Timed retention, which you specify using the UNTIL=*time-value* option, allows you to retain a job in the queue only as long as you need it. This eliminates the need to delete the job from the queue later.

For example, the following command retains the job MYFILE in the queue until 7:31 on December 14, when the job will automatically be deleted from the queue:

```
$ SUBMIT/RETAIN=UNTIL=14-DEC-2001:07:31:0.0 MYFILE.COM
```

However, depending on the queue's job retention policy, the job might be retained indefinitely. The job retention policy set on the queue takes precedence over the user-specified job retention setting. Because system managers cannot specify timed job retention for a queue, any jobs retained as a result of a queue's setting are retained indefinitely.

If you specify the **/RETAIN=UNTIL=*time-value*** option, you must supply a time value. The time value is first interpreted as a delta time, then as a combination time, and finally as an absolute time. If you specify a delta time, the delta begins when the job completes. For example, if you specify **SUBMIT/RETAIN=UNTIL="3:00"**, the job will be retained for three hours after the job completes. For information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT].

/SINCE[=*time*]
/NOSINCE

Selects only those files dated on or after the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, JOB_LOGIN, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the **/SINCE** qualifier to indicate the time attribute to be used as the basis for selection: **/BACKUP**, **/CREATED** (default), **/EXPIRED**, or **/MODIFIED**.

For complete information about specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/STYLE=*keyword*

Specifies the file name format for display purposes.

The valid keywords for this qualifier are CONDENSED and EXPANDED. Descriptions are as follows:

Keyword	Description
CONDENSED (default)	Displays the file name representation of what is generated to fit into a 255-length character string. This file name may contain a DID or FID abbreviation in the file specification.
EXPANDED	Displays the file name representation of what is stored on disk. This file name does not contain any DID or FID abbreviations.

The keywords CONDENSED and EXPANDED are mutually exclusive. This qualifier specifies which file name format is displayed in the output message, along with the confirmation if requested.

File errors are displayed with the CONDENSED file specification unless the EXPANDED keyword is specified.

See the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#FILE_ACCESS] for more information.

/SYMLINK=*keyword*

The valid keywords for this qualifier are [NO]WILDCARD and [NO]ELLIPSIS. Descriptions are as follows:

Keyword	Description
WILDCARD	Indicates that symlinks are enabled during wildcard searches.
NOWILDCARD	Indicates that symlinks are disabled during directory wildcard searches.
ELLIPSIS	Equivalent to WILDCARD (included for command symmetry).
NOELLIPSIS	Indicates that symlinks are matched for all wildcard fields except for ellipsis.

If the file named in the **SUBMIT** command is a symlink, the command operates on the symlink target.

/USER=*username*

Requires CMKRNL (change mode to kernel) privilege and read (R) and write (W) access to the user authorization file (UAF).

Allows you to submit a job on behalf of another user. The job runs exactly as if that user had submitted it. The job runs under that user's user name and user identification code (UIC), and accounting information is logged to that user's account. By default, the user identification comes from the requesting process. The username qualifier value can be any user name that is validated on your system. The name must be 1 to 12 alphanumeric characters.

/WSDEFAULT=*n*

Defines for a batch job a working set default, the default number of physical pages that the job can use. If the queue on which the job executes has a nonzero default working set, the smaller of the specified job and queue values is used. If the queue on which the job executes has a working set default of 0, the smaller of the specified job and UAF values (value established in the user authorization file) is used.

Specify the value of *n* as a number of 512-byte pagelets on Alpha. Note that OpenVMS systems round up this value to the nearest CPU-specific page so that the actual amount of physical memory allowed may be larger than the specified amount on Alpha. If you specify the value 0 or NONE, the specified queue or UAF value is used. Working set default values must range between the numbers specified by the system parameters PQL_MWSDEFAULT and WSMAX.

/WSEXTENT=*n*

Defines for the batch job a working set extent, the maximum amount of physical memory that the job can use. The job uses the maximum amount of physical memory only when the system has excess free pages. If the queue on which the job executes has a nonzero working set extent, the smaller of the specified job and queue values is used. If the queue on which the job executes

has a working set extent of 0, the smaller of the specified job and the value established in the user authorization file (UAF) is used.

Specify the value of *n* as a number of 512-byte pagelets on Alpha. Note that OpenVMS systems round up this value to the nearest CPU-specific page so that the actual amount of physical memory allowed may be larger than the specified amount on Alpha. If you specify the value 0 or NONE, the specified queue or UAF value is used. Working set extent values must range between the numbers specified by the system parameters PQL_MWSEXTENT and WSMAX.

/WSQUOTA=*n*

Defines for the batch job a working set quota, the amount of physical memory that the job is guaranteed. If the queue on which the job executes has a nonzero working set quota, the smaller of the specified job and queue values is used. If the queue on which the job executes has a working set quota of 0, the smaller of the specified job or the value established in the user authorization file (UAF) is used.

Specify the value of *n* as a number of 512-byte pagelets on Alpha. Note that OpenVMS systems round up this value to the nearest CPU-specific page so that the actual amount of physical memory allowed may be larger than the specified amount on Alpha. If you specify the value 0 or NONE, the specified queue or UAF value is used. Working set quota values must range between the numbers specified by the system parameters PQL_MWSQUOTA and WSMAX.

Examples

1.
```
$ SUBMIT/AFTER=16:30 TRANSLATE
Job TRANSLATE (queue SYS$BATCH, entry 1401) holding until 14-DEC-2001
 16:30
```

   In this example, the command procedure `TRANSLATE.COM` is submitted to `SYS$BATCH`. The procedure is executed at 4:30 P.M. or later. When the batch job completes, the log file `TRANSLATE.LOG` is queued to SYS$PRINT for printing and is deleted.

2.
```
$ SUBMIT /PARAMETERS=(TXT,DOC,MEM) BACKUP, AVERAGE, RUNMASTER
  Job BACKUP (queue SYS$BATCH, entry 416) pending
```

   In this example, the **SUBMIT** command enters three command procedures in a single job. The job is given three parameters: `P1` is equated to the string `TXT`, `P2` to the string `DOC`, and `P3` to the string `MEM`. After the procedure `BACKUP.COM` is executed, the procedures `AVERAGE.COM` and `RUNMASTER.COM` are executed.

3.
```
$ SUBMIT/NAME=BATCH24/HOLD TESTALL
  Job BATCH24 (queue SYS$BATCH, entry 467) holding
```

   In this example, the **SUBMIT** command enters the procedure `TESTALL.COM` as a batch job and specifies that the job be held for later processing. The job is not released until the **SET ENTRY/RELEASE** command is entered. The **/NAME** qualifier requests that the batch job be identified as `BATCH24`.

4.
```
$ SUBMIT TEST.COM
  Job TEST (queue SYS$BATCH, entry 493) pending
$ BATCH_JOB = $ENTRY
  .
  .
  .
$ DELETE/ENTRY='BATCH_JOB'
```

In this command sequence, the $ENTRY symbol is used to refer to a particular job in the queue file. The **SUBMIT** command creates a batch job to run the command procedure TEST.COM. In addition, the **SUBMIT** command defines $ENTRY as a local symbol whose value is the entry number of the job (493 in this example). The second statement assigns the value $ENTRY to the local symbol BATCH_JOB. The last line of the example deletes the job by using the symbol BATCH_JOB to identify it.

5.
```
$ DEFINE JUNE WORKZ:[SCHREURS]ANNUAL_REPORT.COM
$ SUBMIT JUNE
Job ANNUAL_REPORT (queue SYS$BATCH, entry 229) started on ZOO_BATCH
```

In this example, the logical name JUNE is created and equated to ANNUAL_REPORT.COM with the **DEFINE** command. Using the logical name JUNE, the user submits ANNUAL_REPORT.COM to the batch queue. Note that the system translates the logical name JUNE to ANNUAL_REPORT.COM before ANNUAL_REPORT.COM is submitted to the batch queue. Also, the log file produced is named ANNUAL_REPORT.COM rather than JUNE.COM.

Note also that the job is submitted to the generic queue SYS$BATCH, but runs on the execution queue ZOO_BATCH.

6.
```
$ SUBMIT/HOLD/QUEUE=ANYRADQ /RAD=1  TEST.COM
Job TEST (queue ANYRADQ, entry 23) holding

$ SHOW ENTRY/FULL 23
 Entry  Jobname      Username     Blocks  Status
 -----  -------      --------     ------  ------
    23  TEST         SYSTEM               Holding
  On idle batch queue ANYRADQ
  Submitted 24-JUL-2001 14:19:37.44 /KEEP /NOPRINT /PRIORITY=100 /RAD=0
  File: _$1$DKB200:[SWEENEY.CLIUTL]TEST.COM;1
```

In this example, TEST.COM is submitted to the queue ANYRADQ. There is no RAD setting on the ANYRADQ queue.

7.
```
$ SUBMIT/HOLD/QUEUE=BATCHQ1 /RAD=1  TEST.COM
Job TEST (queue BATCHQ1, entry 24) holding

$ SHOW ENTRY 24/FULL
 Entry  Jobname      Username     Blocks  Status
 -----  -------      --------     ------  ------
    24  TEST         SYSTEM               Holding
  On idle batch queue BATCHQ1
  Submitted 24-JUL-2001 14:23:10.37 /KEEP /NOPRINT /PRIORITY=100 /RAD=0
  File: _$1$DKB200:[SWEENEY.CLIUTL]TEST.COM;2
```

In this example, the queue, BATCHQ1, is defined with **/RAD=0**. Therefore, use of the **SUBMIT** command creates a job that runs on RAD 0, even though RAD 1 was specified.

SUBROUTINE

SUBROUTINE — Defines the beginning of a subroutine in a command procedure. The **SUBROUTINE** command must be the first executable statement in a subroutine. For more information about the **SUBROUTINE** command, see the description of the **CALL** command in the *VSI OpenVMS DCL Dictionary: A–M* [https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/#BLUE_22].

Format

SUBROUTINE

SYNCHRONIZE

SYNCHRONIZE — Holds the process issuing the command until the specified job completes execution. **Requires delete (D) access to the specified job.**

Format

SYNCHRONIZE [*job-name*]

Parameter

job-name

Specifies the name of the job as defined when the job was submitted. You can specify only job names that are associated with your user name. (A job is associated with the user name of the process that submits it.)

To specify a job that does not have a unique name, use the **/ENTRY** qualifier to specify the entry number. If you specify both the *job-name* parameter and the **/ENTRY** qualifier, the *job-name* parameter is ignored.

If you specify the *job-name* parameter, the command searches for a job with the specified name in the queue SYS$BATCH. To specify a job in a different queue, you must specify the name of the queue using the **/QUEUE** qualifier.

Description

The **SYNCHRONIZE** command provides job synchronization by placing a process in a wait state until the specified job completes. If the specified job is not a current job in the system, the **SYNCHRONIZE** command displays an error message.

When a job specified in a **SYNCHRONIZE** command completes, the process is released from the wait state. The completion status for the **SYNCHRONIZE** command is the same as the completion status of the last command executed in the job. To make sure that you can determine the completion status when the **SYNCHRONIZE** command is part of a command procedure, include **SET NOON** before the **SYNCHRONIZE** command and **SHOW SYMBOL $STATUS** immediately after the **SYNCHRONIZE** command.

You must specify either the *job-name* parameter or the **/ENTRY** qualifier.

Qualifiers

/ENTRY=*entry-number*

Identifies the job by the system-assigned entry number.

The system assigns a unique entry number to each queued print or batch job in the system. By default, the **PRINT** and **SUBMIT** commands display the entry number when they successfully

queue a job for processing. These commands also create or update the local symbol $ENTRY to reflect the entry number of the most recently queued job. To find a job's entry number, enter the **SHOW ENTRY** or **SHOW QUEUE** command. If you specify both the *job-name* parameter and the **/ENTRY** qualifier, the job name is ignored.

/QUEUE=*queue-name*[:]

Names the queue containing the job. If you use the **/QUEUE** qualifier, you must specify either the *job-name* parameter or the **/ENTRY** qualifier. If you specify the *job-name* parameter, the default queue is SYS$BATCH. If you specify the **/ENTRY** qualifier, there is no default queue. If you specify a generic queue, the command looks for the job in the generic queue and in the generic queue's target execution queues.

/TIME_OUT=*n*

Allows the user to specify the number of seconds to wait before terminating the **SYNCHRONIZE** command when the target job is running.

Examples

1. ```
   $ SUBMIT/NAME=PREP FORMAT/PARAMETERS=(SORT,PURGE)
     Job PREP (queue SYS$BATCH, entry 219) started on queue SYS$BATCH
   $ SUBMIT PHASER
     Job PHASER (queue SYS$BATCH, entry 223) started on queue SYS$BATCH
   ```

   In this example, the first **SUBMIT** command submits the command procedure FORMAT.COM for execution and names the job PREP. The second **SUBMIT** command queues the procedure PHASER.COM. The procedure PHASER.COM contains the following line:

   ```
   $ SYNCHRONIZE PREP
   ```

   When this line is processed, the system verifies whether the job named PREP exists in the default queue SYS$BATCH. The procedure PHASER is forced to wait until the job PREP completes execution.

2. ```
   $ SUBMIT/NAME=TIMER       COMP.COM
     Job TIMER (queue SYS$BATCH, entry 214) started on queue SYS$BATCH
   $ SYNCHRONIZE /ENTRY=214
   ```

   In this example, a batch job named TIMER is submitted. Then the **SYNCHRONIZE** command is entered interactively. This command places the interactive process in a wait state until entry number 214 (TIMER) completes. You cannot enter subsequent commands from your terminal session until the **SYNCHRONIZE** command completes and your process is released from the wait state.

3. ```
   $ SUBMIT/NAME=TIMER COMP.COM/QUEUE=BUILD_QUEUE
   $ SYNCHRONIZE/QUEUE=BUILD_QUEUE TIMER
   ```

   The batch job name TIMER is submitted to the queue BUILD_QUEUE in this example. Then the **SYNCHRONIZE** command is entered interactively. Because the job named TIMER is not in the default queue SYS$BATCH, the **SYNCHRONIZE** command includes the **/QUEUE** qualifier to specify the queue in which the job is located.

TYPE

TYPE — Displays the contents of a file or group of files on the current output device.
   ```

Format

TYPE ***filespec*****[,...]**

Parameter

filespec**[,...]**

Specifies one or more files to be displayed. If you specify a file name and not a file type, the file type defaults to .LIS; however, you can specify a different default file type with the TYPE$DEFAULT_FILESPEC logical name. The **TYPE** command displays all files that satisfy the file description.

The asterisk (*) and the percent sign (%) wildcard characters are allowed in place of the directory name, file name, file type, or file version number field. If you specify more than one file, separate the file specifications with commas (,) or plus signs (+). The files are displayed in the order listed.

Description

When the **TYPE** command displays output, you can control the display in the following ways:

- Press **Ctrl/C** to stop the **TYPE** command for the current file that is being displayed. If you specified only one file with your **TYPE** command, or if the **TYPE** command is displaying the last file in the list, pressing **Ctrl/C** cancels the **TYPE** command. If you have specified more than one file with the **TYPE** command, pressing **Ctrl/C** causes the **TYPE** command to display the next file in the list.
- Press **Ctrl/S** to temporarily suspend the output. Press **Ctrl/Q** to resume the output display at the point of interruption.
- Press **Ctrl/O** to suppress the display but not to suspend the command processing. If you press **Ctrl/O** again before the **TYPE** command terminates, output resumes at the current point in command processing. However, if you press **Ctrl/O** when the **TYPE** command is displaying files in a list, the **TYPE** command suppresses typing the current file and begins typing the next file in the list. This behavior is an exception to normal **Ctrl/O** processing.
- Press **Ctrl/Y** to interrupt the command execution. You can enter the **CONTINUE** command after pressing **Ctrl/Y** to resume displaying the files where the interruption took place, provided you have not entered an intervening command that calls up a new image. If you press **Ctrl/Y** to stop command execution entirely, you can enter the **EXIT** command (or any other DCL command that activates an image) to run down the image.

In addition, the **/PAGE** qualifier may be used to display text one screen at a time.

The **TYPE** command opens the specified file with shared read (R) and write (W) access. Therefore, any file that has its attributes set to shared write is displayed, even if it is currently opened by another user.

You also can use the **TYPE** command to execute a command procedure on a remote node. This is useful on OpenVMS Cluster systems, for example, when you want to display the status of clusterwide services, such as queues, or when you want to display the users logged in to other nodes on the cluster. A sample command procedure follows:

```
$ @ SHOWUSERS.COM
$ IF F$MODE() .EQS. "NETWORK" THEN DEFINE/USER SYS$OUTPUT SYS$NET
$ SHOW USERS
```

This command procedure can be used with the **TYPE** command to display at the user's local node the users logged on to the remote node where the command procedure resides.

Specify the command procedure as a parameter to the **TYPE** command as follows:

```
$ TYPE node_name::"TASK=command_procedure"
```

where:

node_name	Specifies the name of the remote node on which the command procedure resides.
command_procedure	Specifies the file name of the command procedure to be run.

This form of the command finds the command procedure in the default DECnet account of the remote node.

To execute a command procedure in the SYS$LOGIN directory of a particular account, use an access control string in the command, as follows:

```
$ TYPE node_name"user_name password"::"TASK=command_procedure"
```

where:

user_name	Specifies the user name of the account on the remote node.
password	Specifies the password of the account on the remote node.

Qualifiers

/BACKUP

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/BACKUP** qualifier selects files according to the dates of their most recent backups. This qualifier is incompatible with the **/CREATED**, **/EXPIRED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/BEFORE[=*time*]

Selects only those files dated prior to the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the **/BEFORE** qualifier to indicate the time attribute to be used as the basis for selection: **/BACKUP**, **/CREATED** (default), **/EXPIRED**, or **/MODIFIED**.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/BY_OWNER[=*uic*]

Selects only those files whose owner user identification code (UIC) matches the specified owner UIC. The default UIC is that of the current process.

Specify the UIC by using standard UIC format as described in the relevant section in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#UIC_SECTION].

/CONFIRM
/NOCONFIRM (default)

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

YES	NO	QUIT
TRUE	FALSE	**Ctrl/Z**
1	0	ALL
	Return	

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), but these abbreviations must be unique. Affirmative answers are YES, TRUE, and 1. Negative answers include: NO, FALSE, 0, and pressing **Return**. Entering QUIT or pressing **Ctrl/Z** indicates that you want to stop processing the command at that point. When you respond by entering ALL, the command continues to process, but no further prompts are given. If you type a response other than one of those in the list, DCL issues an error message and redisplays the prompt.

/CONTINUOUS/INTERVAL=*n*

Upon reaching the end of a file, **`TYPE`** waits a specified number of seconds (the default is 30) then displays any records which have been added to the file and returns to the wait state. **`TYPE`** continues until either **Ctrl/C** or **Ctrl/Y** is entered by the user.

/CREATED (default)

Modifies the time value specified with the **`/BEFORE`** or the **`/SINCE`** qualifier. The **`/CREATED`** qualifier selects files based on their dates of creation. This qualifier is incompatible with the **`/BACKUP`**, **`/EXPIRED`**, and **`/MODIFIED`** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **`/CREATED`** qualifier.

/EXACT

Use with the **`/PAGE=SAVE`** and **`/SEARCH`** qualifiers to specify a search string that must match the search string exactly and must be enclosed with quotation marks (" ").

If you specify the **`/EXACT`** qualifier without the **`/SEARCH`** qualifier, exact search mode is enabled when you set the search string with the Find (**E1**) key.

/EXCLUDE=(*filespec*[,...])

Excludes the specified files from the type operation. You can include a directory but not a device in the file specification. The asterisk (*) and the percent sign (%) wildcard characters are allowed in the file specification; however, you cannot use relative version numbers to exclude a specific version. If you specify only one file, you can omit the parentheses.

/EXPIRED

Modifies the time value specified with the **`/BEFORE`** or the **`/SINCE`** qualifier. The **`/EXPIRED`** qualifier selects files according to their expiration dates. (The expiration date is set with the **`SET FILE/EXPIRATION_DATE`** command.) The **`/EXPIRED`** qualifier is incompatible with the

/BACKUP, **/CREATED**, and **/MODIFIED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time qualifiers, the default is the **/CREATED** qualifier.

/HEADER
/NOHEADER

Specifies whether a header line is placed at the top of each file that is displayed with the **TYPE /PAGE=SAVE** command.

/HIGHLIGHT[=*keyword*]

Use with the **/PAGE=SAVE** and **/SEARCH** qualifiers to specify the type of highlighting you want when a search string is found. When a string is found, the entire line is highlighted. You can use the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting.

/MODIFIED

Modifies the time value specified with the **/BEFORE** or the **/SINCE** qualifier. The **/MODIFIED** qualifier selects files according to the dates on which they were last modified. This qualifier is incompatible with the **/BACKUP**, **/CREATED**, and **/EXPIRED** qualifiers, which also allow you to select files according to time attributes. If you specify none of these four time modifiers, the default is the **/CREATED** qualifier.

/OUTPUT[=*filespec*]
/NOOUTPUT

Controls where the output of the command is sent. If you specify the **/OUTPUT=*filespec*** qualifier, the output is sent to the specified file, rather than to the current output device, SYS$OUTPUT. If you do not enter the qualifier, or if you enter the **/OUTPUT** qualifier without a file specification, the output is sent to SYS$OUTPUT.

If you enter the **/OUTPUT** qualifier with a partial file specification (for example, **/OUTPUT=[VAKHRAMEEV]**), TYPE is the default file name and .LIS the default file type. The file specification cannot include the asterisk (*) and the percent sign (%) wildcard characters.

If you enter the **/NOOUTPUT** qualifier, output is suppressed.

The **/OUTPUT** qualifier is incompatible with the **/PAGE** qualifier.

/PAGE[=*keyword*]
/NOPAGE (default)

Controls the display of information on the screen.

You can use the following keywords with the **/PAGE** qualifier:

Keyword	**Description**
CLEAR_SCREEN	Clears the screen before each page is displayed.
SCROLL	Displays information one line at a time.
SAVE[=*n*]	Enables screen navigation of information, where *n* is the number of pages to store.

The **/PAGE=SAVE** qualifier allows you to navigate through screens of information. The **/PAGE=SAVE** qualifier stores up to 5 screens of up to 255 columns of information. When you use the /PAGE=SAVE qualifier, you can use the following keys to navigate through the information:

Key Sequence	Description
Up arrow key, **Ctrl/B**	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.
Right arrow key	Scroll right one column.
Find (**E1**)	Specify a string to find when the information is displayed.
Insert Here (**E2**)	Scroll right one half screen.
Remove (**E3**)	Scroll left one half screen.
Select (**E4**)	Toggle 80/132 column mode.
Prev Screen (**E5**)	Get the previous page of information.
Next Screen (**E6**), **Return**, **Enter**, **Space**	Get the next page of information.
F10, **Ctrl/Z**	Exit. (Some utilities define these differently.)
Help (**F15**)	Display utility help text.
Do (**F16**)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.
N	Display the next file (when more than one file is specified with the **TYPE** command).
Q	Quit displaying output from the **TYPE** command.

The **/PAGE** qualifier is not compatible with the **/OUTPUT** qualifier.

/SEARCH="*string*"

Use with the **/PAGE=SAVE** qualifier to specify a string that you want to find in the information being displayed. Quotation marks are required for the **/SEARCH** qualifier, if you include spaces in the text string.

You can also dynamically change the search string by pressing the Find key (**E1**) while the information is being displayed. Quotation marks are not required for a dynamic search.

/SINCE[=*time*]

Selects only those files dated on or after the specified time. You can specify time as absolute time, as a combination of absolute and delta times, or as one of the following keywords: BOOT, JOB_LOGIN, LOGIN, TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the **/SINCE** qualifier to indicate the time attribute to be used as the basis for selection: **/BACKUP**, **/CREATED** (default), **/EXPIRED**, or **/MODIFIED**.

For complete information on specifying time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

/STYLE=*keyword*

Specifies the file name format for display purposes while typing files.

The valid keywords for this qualifier are CONDENSED and EXPANDED. Descriptions are as follows:

Keyword	Description
CONDENSED (default)	Displays the file name representation of what is generated to fit into a 255-length character string. This file name may contain a DID or FID abbreviation in the file specification.
EXPANDED	Displays the file name representation of what is stored on disk. This file name does not contain any DID or FID abbreviations.

The keywords CONDENSED and EXPANDED are mutually exclusive. This qualifier specifies which file name format is displayed in the output message, along with the confirmation if requested.

File errors are displayed with the CONDENSED file specification unless the EXPANDED keyword is specified.

See the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#FILE_ACCESS] for more information.

/SYMLINK=*keyword*

The valid keywords for this qualifier are [NO]WILDCARD and [NO]ELLIPSIS. Descriptions are as follows:

Keyword	Description
WILDCARD	Indicates that symlinks are enabled during wildcard searches.
NOWILDCARD	Indicates that symlinks are disabled during directory wildcard searches.
ELLIPSIS	Equivalent to WILDCARD (included for command symmetry).
NOELLIPSIS	Indicates that symlinks are matched for all wildcard fields except for ellipsis.

If the file named in the **`TYPE`** command is a symlink, the command operates on the symlink target.

/TAIL[=*n*]

Displays the last several lines of a log file. The value, *n*, defaults to *p*-2 where *p* is the current terminal page length. You can use **`TYPE/TAIL`** only if all of the following criteria are true:

- File organization is sequential.
- The longest record is less than 512 bytes.
- The record format is either VAR, VFC, STM, STMCR, or STMLF (for more information, see the description of FAB$B_RFM in the *VSI OpenVMS Record Management Services Reference Manual* [https://docs.vmssoftware.com/vsi-openvms-record-management-services-reference-manual/#RMS_RECORD_FORMAT_FIELD]).

- The file being typed is on a device that supports random access. The **TYPE/TAIL** command does not work on magnetic tape drives.

Even with this criteria, some file conditions cannot be anticipated and may not allow display of the last several lines of a log file, resulting in the following error message:

```
%TYPE-W-READERR, error reading DEVICE:[DIRECTORY]FILE.EXT;1
-SYSTEM-E-UNSUPPORTED, unsupported operation or function
```

/WRAP
/NOWRAP (default)

Use with the **/PAGE=SAVE** qualifier to limit the number of columns to the width of the screen and to 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 and can be seen when you use the scrolling (left and right) features provided by the **/PAGE=SAVE** qualifier.

Examples

1.
```
$ TYPE COMMON.DAT
```

In this example, the **TYPE** command requests that the file COMMON.DAT be displayed at the terminal.

2.
```
$ TYPE *.DAT
This is the first line in the file AA.DAT.
  .
  .
  .
Ctrl/O
This is the first line in the file BB.DAT.
  .
  .
  .
Ctrl/Y
Interrupt
$ STOP
```

In this example, the **TYPE** command contains an asterisk (*) wildcard character in place of the file name. All files with file types of .DAT are scheduled for display. When **Ctrl/O** is pressed, output of the current file stops and the **TYPE** command begins displaying the next file. The **Ctrl/Y** function interrupts the command; the **STOP** command terminates the **TYPE** command.

3.
```
$ TYPE LETTER*.MEM
December 14, 2002
  .
  .
  .
Ctrl/Y
Interrupt
$ SHOW TIME
 14-DEC-2002 15:48:07
$ CONTINUE
Sincerely yours,
  .
```

.
.

In this example, the **TYPE** command displays all files whose names begin with the word LETTER and have the file type .MEM. While the files are being displayed, the user presses **Ctrl/Y** to interrupt the **TYPE** operation and to display the time. After entering the **SHOW TIME** command, the user enters the **CONTINUE** command to resume the **TYPE** operation.

4. `$ TYPE/OUTPUT=SAVETEXT.TXT *.TXT`

   In this example, the **TYPE** command writes all .TXT files in your default directory to a file called SAVETEXT.TXT (also in your default directory).

5. `$ TYPE MEXICO::NOTICE.TEXT/OUTPUT=TEMP.TEXT`

   In this example, the **TYPE** command requests that the file NOTICE.TEXT at remote node MEXICO be written to the output file TEMP.TEXT on the local node, rather than to SYS$OUTPUT.

6. `$ TYPE SECSYS"FILES OFFICEFIL"::"TASK=SHOWUSERS"`

```
        VAX/VMS Interactive Users
           14-DEC-2002 17:20:13.30
   Total number of interactive users = 5
Username     Process Name      PID     Terminal
MIRANDA      Sec1           00536278   TXA1:
JESSICA      Sec2           00892674   VTA2:
EMILY        Sec3           00847326   TXA3:
ANDREW       Sec4           02643859   RTA1:
BRIAN        System Mangr   00007362   VTA1:
```

   In this example, the **TYPE** command executes the command procedure SHOWUSERS.COM found in the SYS$LOGIN directory of user FILES on remote node SECSSYS. The output of the **TYPE** command then is displayed at the local node.

VIEW

VIEW — Invokes the CDA Viewer, which has a character-cell interface for viewing text files and an DECwindows Motif for OpenVMS interface for viewing compound documents and files containing graphics, images, and tabular data.

Format

VIEW [*input-file*]

Description

For more information about the **VIEW** command, see the online help.

Note

The DEC CDA Base Services for DECwindows Motif for OpenVMS must be installed to use new versions of the CDA Viewer and DEC CDA Base Services converters.

WAIT

WAIT — Places a process into a wait state for the specified amount of time. The WAIT command is used in a command procedure to delay processing of either the procedure itself or a set of commands in the procedure.

Format

WAIT *delta-time*

Parameter

delta-time

Specifies a delta time interval in the following format. (A delta time is an offset from the current time to a time in the future.)

hour:*minute*:*second*.*hundredth*

The fields on the format line indicate the following:

hour	Specifies an integer in the range 0 to 23.
minute	Specifies an integer in the range 0 to 59.
second	Specifies an integer in the range 0 to 59.
hundredth	Specifies an integer in the range 0 to 99.

The colons (:) and period (.) are required delimiters; also, the delta time must begin with the number of hours and not a colon. Note that the days field, usually included in the delta time format, must be omitted here.

For more information on specifying delta time values, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#DATE_AND_TIME_SECT] or the online help topic Date.

Note that if you enter the **WAIT** command interactively, you are not prompted for a time value; however, in order for the command to have any effect, you must supply a time value.

Description

If you enter the **WAIT** command interactively, your current process is placed in a wait state and you cannot enter any more commands until the waiting period is over. (You can, however, receive unsolicited messages from other processes.) Press **Ctrl/C** or **Ctrl/Y** to restore normal terminal interaction.

Example

```
$ LOOP:
$ RUN KUDOS
$ WAIT 00:10
$ GOTO LOOP
```

In this example, the command procedure executes the program image KUDOS. After the **RUN** command executes the program, the **WAIT** command delays execution of the **GOTO** command for 10 minutes. Note that 00 is specified for the number of hours, because the time specification cannot begin with a

colon. After `10` minutes, the **GOTO** command executes, and the procedure transfers control to the label `LOOP` and executes the program `KUDOS` again. The procedure loops until it is interrupted or terminated.

If the procedure is executed interactively, terminate it by pressing **Ctrl/C** or **Ctrl/Y** and by entering the **STOP** command or another DCL command that runs a new image in the process. If the procedure is executed in a batch job, enter the **DELETE/ENTRY** command to terminate it.

WRITE

WRITE — Writes the specified data as one record to an open file specified by a logical name. All qualifiers must precede all data item expressions.

Format

WRITE *logical-name expression*[,...]

Parameters

logical-name

Specifies the logical name assigned to the output file. Use the logical name assigned by the **OPEN** command. In interactive mode, specify the process-permanent files identified by the logical names SYS$INPUT, SYS$OUTPUT, SYS$ERROR, and SYS$COMMAND. (The **OPEN** command assigns a logical name to a file and places the name in the process logical name table.)

***expression*[,...]**

Specifies data to be written as a single record to the output file. You can specify data items using character string expressions, which may be symbol names, character strings in quotation marks (" "), literal numeric values, or a lexical function. For more information on string expressions, see the relevant section in the *VSI OpenVMS User's Manual* [https://docs.vmssoftware.com/vsi-openvms-user-s-manual/#SYMB_BLK24].

You can specify a list of expressions separated by commas (,); the command interpreter concatenates the items into one record and writes the record to the output file.

The maximum size of any record that can be written is less than 1024 bytes, and the value of any symbol that is specified as part of a record cannot exceed 255 characters; however, if you specify the **/SYMBOL** qualifier, the maximum record size is 2048 bytes and the value of a symbol can exceed 255 characters.

Description

The **WRITE** command can write records to sequential, relative, or indexed files that have been opened for writing. When the **WRITE** command writes a record, it always positions the record pointer after the record just written.

To write to a file, the file must be opened by using either the **/WRITE** or the **/APPEND** qualifier with the **OPEN** command; however, the process-permanent files identified by the logical names SYS$INPUT, SYS$OUTPUT, SYS$ERROR, and SYS$COMMAND do not have to be opened explicitly to be written to.

If you do not specify the **/SYMBOL** qualifier, DCL places the command and the complete string expression (expanded if it was specified as one or more symbols) in a 1024-byte buffer. If you specify the **/SYMBOL** qualifier, DCL interprets the symbol or symbols and places the expanded string in a

separate 2048-byte buffer, and then performs the write operation. For this reason, use the **/SYMBOL** qualifier where the record contains approximately 1000 bytes or more.

Qualifiers

/ERROR=*label*

Transfers control on an I/O error to the location specified by *label* (in a command procedure). If no error routine is specified and an error occurs during the writing of the file, the current ON condition action is taken. The **/ERROR** qualifier overrides any ON condition action specified. If an error occurs and control passes successfully to the target label, the reserved global symbol $STATUS retains the error code.

/SYMBOL

Causes the expression to be interpreted and its expanded value placed in a 2048-byte (instead of a 1024-byte) buffer before the write operation is performed. If you specify multiple expressions, their values are concatenated and placed in the 2048-byte buffer. Use the **/SYMBOL** qualifier to write a very large record.

If you do not use the **/SYMBOL** qualifier, the entire command, including the expression or expressions, is placed in a 1024-byte buffer.

/UPDATE

Replaces the last record read with the record specified with the expression parameter. You must be able to read and write to a file to use the **/UPDATE** qualifier. Use the **WRITE/UPDATE** command only after a **READ** command. The **WRITE/UPDATE** command modifies the last record you have read.

With sequential files, you must replace a record with another record of the same size when you use the **WRITE/UPDATE** command.

/WAIT (default)
/NOWAIT

If you specify **/NOWAIT**, the Put service to a mailbox device uses the IO$M_NOW modifier, which causes the operation to complete immediately instead of synchronizing with another reader of the mailbox.

Currently this qualifier is supported on Alpha and IA-64 only.

Examples

1. ```
   $ WRITE SYS$OUTPUT "Beginning second phase of tests"
   ```

   The **WRITE** command writes a single line of text to the current output device.

2. ```
   $ OPEN/APPEND OUTPUT_FILE TRNTO::DKA1:[PGM]PLAN.DAT
   $ WRITE OUTPUT_FILE "BEGINNING PHASE 3"
   ```

   In this example, the **OPEN/APPEND** command opens the file `PLAN.DAT` at the remote node `TRNTO` and positions the pointer at the end of the file. The **WRITE** command writes a record to the end of the file `PLAN.DAT`.

3. ```
   $ OPEN/WRITE OUTPUT_FILE TESTFILE.DAT
   $ INQUIRE ID "Assign Test-id Number"
   ```
   ```

```
$ WRITE/ERROR=WRITE_ERROR  OUTPUT_FILE  "Test-id is ",ID
$ WRITE/ERROR=WRITE_ERROR  OUTPUT_FILE  ""
$ !
$ WRITE_LOOP:
   .
   .
   .
$ GOTO WRITE_LOOP
$ END_LOOP:
$ !
$ CLOSE OUTPUT_FILE
$ PRINT TESTFILE.DAT
$ EXIT
$ !
$ WRITE_ERROR:
$ WRITE SYS$OUTPUT "There was a WRITE error."
$ CLOSE OUTPUT_FILE
$ EXIT
```

In this example, the **OPEN** command opens the file TESTFILE.DAT; the **INQUIRE** command requests an identification number to be assigned to a particular run of the procedure. The number entered is equated to the symbol ID. The **WRITE** commands write a text line concatenated with the symbol name ID and a blank line.

The lines between the label WRITE_LOOP and END_LOOP process information and write additional data to the file. When the processing is finished, control is transferred to the label END_LOOP. The **CLOSE** and **PRINT** commands at this label close the output file and queue a copy of the file to the system printer.

The label WRITE_ERROR is used as the target of the **/ERROR** qualifier to the **WRITE** command; if an error occurs when a record is being written, control is transferred to the label WRITE_ERROR.

4.
```
$ OPEN/APPEND MYFILE [LAMPERT]TESTING.DAT
$ WRITE/SYMBOL MYFILE A,B,C
```

This example assumes that the symbols A, B, and C have already been defined. The **OPEN/APPEND** command opens the file [LAMPERT]TESTING.DAT and positions the pointer at the end of the file. The **WRITE/SYMBOL** command concatenates the values of the symbols A, B, and C and writes this data to a new record at the end of the file.

XAUTH (Alpha/IA-64 only)

XAUTH (Alpha/IA-64 only) — Invokes the X Authority utility (xauth) that you can use to manage the contents of one or more X authority files. The X authority file contains information used to authorize client connections to the X display server.

Format

XAUTH [*-f authfile*] [*-options...*] [*command*]

Description

This utility is typically used to extract authorization records from one system and combine them with the records on another system, such as when granting access to additional users or enabling remote logins. The actual record entries vary depending on the authorization scheme currently in use.

In contrast to other X Window System utilities that are available with DECwindows Motif, xauth is included as a part of OpenVMS Alpha and OpenVMS IA-64 operating systems. The xauth commands are case-insensitive and available directly from the DCL command line, xauth command line, or from a batch file.

For a complete description of xauth, see the DECwindows Motif documentation or the xauth online help.

Appendix A. Obsolete Commands

Table A.1, "Obsolete Commands" lists the obsolete DCL commands and the current services that replace them.

Table A.1. Obsolete Commands

Obsolete Command	Replaced by
`SET ACL`	`SET SECURITY/ACL`
`SET PROTECTION`	`SET SECURITY/PROTECTION`
`SET PROTECTION/DEVICE`	`SET SECURITY/PROTECTION/CLASS=DEVICE`
`SET UIC`	Not replaced.
`SHOW ACL`	`SHOW SECURITY`
`UNLOCK`	`SET FILE/UNLOCK`

Appendix B. Handling Numeric Values Outside the Range of DCL Integer Representation

B.1. Preface

The OpenVMS Version 8.4 release introduces support for storage devices up to 1.98 TB in size. For customers and products that will use devices over 1 TB, some changes to DCL procedures are necessary to keep them functioning properly. If you do not plan to use storage devices over 1 TB in size, there is likely no need to change any of your DCL code related to reading or changing disk values.

The document is mainly intended to address issues with large disks but the methods used are not specific to the large disk support project.

B.2. Introduction

Herein are some DCL examples handling retrieval and arithmetic on integer values that lie outside the DCL symbol integer limits. The document concentrates on processing values returned from the DCL $GETDVI lexical function. The examples can be used as guidelines for your specific needs. Examples are not warranted or guaranteed to meet any specific requirement. Examples center on handling disk storage size and free storage measurement but methodologies can be more generally applied.

The example procedures are designed to clearly demonstrate the methodology. The specific operations may be more efficiently performed by complex single statement DCL commands. The document chooses ease of reading over efficiency in the examples.

B.2.1. Problem Statement

OpenVMS system configuration and management procedures can update or monitor almost all possible system resource values with signed 32-bit longword integers. Over a period of time, a number of resource value maximums moved beyond the signed longword limit. For example, the number of virtual memory pages available to a process on OpenVMS Alpha and IA-64 systems require 64-bit integer storage for the maximum value. Unfortunately, the DCL command language stores and performs arithmetic only on 32-bit signed longwords. From the "OpenVMS User's Manual Chapter 12, Defining Symbols, Commands and Expressions":

"12.7.2 Internal Storage of Numbers

Numbers are stored internally as signed 4-byte integers, called longwords; positive numbers have values of 0 to 2147483647 and negative numbers have values of 4294967296 minus the absolute value of the number. The number -15237, for example, is stored as 4294952059. Negative numbers are converted back to minus-sign format for ASCII or decimal displays; however, they are not converted back for hexadecimal and octal displays. For example, the number-15237 appears in displays as hexadecimal FFFFC47B (decimal 4294952059) rather than hexadecimal –00003B85. "

This DCL behavior poses problems for scripts monitoring and modifying large valued resource settings on OpenVMS systems. OpenVMS version V8.4 will introduce support for large disks, up to 1.98 terabytes. We expect many DCL procedures will be adversely affected as disk block counts will now exceed the signed maximum value and appear as negative numbers in DCL integer symbols. The

F$GETDVI lexical function items MAXBLOCK, FREEBLOCKS, EXPSIZE and VOLSIZE are typically used to return disk information. All of these F$GETDVI items may return negative numbers on V8.4.

B.2.2. Prerequisites

A moderate knowledge of DCL programming is required. Availability of a disk sized over 1 TB.

B.3. DCL Examples

Some examples and procedures are included in this section to demonstrate ways to perform range checks and percentage calculations in Version 8.4 on systems with large disks.

B.3.1. DCL code checking a value in range 0 to 2147483647

A simple method to check for a minimum value or a range with a maximum value less than 2147483647 is demonstrated here.

change:

```
  IF F$GETDVI(device,"FREEBLOCKS") .GT. 100000
 THEN
   ...
```

to:

```
 IF F$GETDVI(device,"FREEBLOCKS") .LT. 0 .OR. -
           F$GETDVI(device,"FREEBLOCKS") .GT. 100000
        THEN
        ...
```

B.3.2. DCL code checking a value in range 0 to 4294967295

One method to check a range or size value exceeding the signed longword maximum is to convert the values to zero filled ASCII strings. After the conversion the DCL string logical operators can be used to perform the comparison.

```
$! Begin Example
$!
$ disk = P1
$ if disk .EQS. "" then disk = "SYS$LOGIN_DEVICE"
$ length        = f$length("4294967296")  ! string length of 2TiB in blocks
$ maxblocks     = f$fao("!''length'ZL",F$GETDVI(disk,"MAXBLOCK"))
$ freeblocks    = f$fao("!''length'ZL",F$GETDVI(disk,"FREEBLOCKS"))
$ expsize       = f$fao("!''length'ZL",F$GETDVI(disk,"EXPSIZE"))
$ volsize       = f$fao("!''length'ZL",F$GETDVI(disk,"VOLSIZE"))
$ MinBlocks     = f$fao("!''length'ZL",500000)
$!
$ write sys$output "Results for Freeblocks=''freeblocks' .comparison.
 MinBlocks=''MinBlocks'"
$ write sys$output  ".eqs. "  + F$string((freeblocks .eqs. MinBlocks))
$ write sys$output  ".nes. "  + F$string((freeblocks .nes. MinBlocks))
$ write sys$output  ".les. "  + F$string((freeblocks .les. MinBlocks))
```

```
$ write sys$output  ".ges. "  + F$string((freeblocks .ges. MinBlocks))
$ write sys$output  ".lts. "  + F$string((freeblocks .lts. MinBlocks))
$ write sys$output  ".gts. "  + F$string((freeblocks .gts. MinBlocks))
$ write sys$output ""
$ write sys$output "Results for maxblocks=''maxblocks' .comparison.
 volsize=''volsize'"
$ write sys$output  ".eqs. "  + F$string((maxblocks .eqs. volsize))
$ write sys$output  ".nes. "  + F$string((maxblocks .nes. volsize))
$ write sys$output  ".les. "  + F$string((maxblocks .les. volsize))
$ write sys$output  ".ges. "  + F$string((maxblocks .ges. volsize))
$ write sys$output  ".lts. "  + F$string((maxblocks .lts. volsize))
$ write sys$output  ".gts. "  + F$string((maxblocks .gts. volsize))
$!
$! End example
```

B.3.3. Checking the percentage of free blocks on a disk

Common disk management operations include calculating the unused percentage of a disk or determining if there is sufficient free space on the target disk for a copy or backup. This section provides examples on handling these particular functions on large disks.

One approach is to assume that any negative value returned on a FREEBLOCKS check means that there are sufficient free blocks to perform whatever storage operation you require.

```
 IF F$GETDVI(device,"FREEBLOCKS") .LT. 0 THEN GOTO SUCCESS
```

This example demonstrates how to check if the system device has at least 5% free blocks. It calls a subroutine PERCENT_FREE to perform the check. The subroutine scales the disk values so signed arithmetic can be performed.

```
$! begin Example
$!
$ percent_free == -1
$!
$ CALL percent_free "sys$sysdevice"
$ IF $status .AND. percent_free .gt. 0
$ then
$  if percent_free .ge. 5
$  then
$        write sys$output "Sufficient freeblocks on disk"
$  else
$        write sys$output "Warning Insufficient freeblocks on disk"
$ endif
$ else
$ write sys$output "Problem obtaining information for disk"
$ endif
$!
$ EXIT
$
$!
$   percent_free: subroutine
$ !
$ !   This subroutine will return, as an integer, the percentage of free
$ !   disk blocks for a given volume.  The percentage free is given by:
$ !
$ !                          (number of free blocks on the volume)
$ !     Percentage Free = ------------------------------------ * 100
$ !                                (volume size in blocks)
```

```
$ !
$ !   The volume name is passed in P1 and the result is returned in the
$ !   global symbol PERCENT_FREE.  For example:
$ !
$ !     $ call percent_free "disk$test"
$ !     $ write sys$output "Free space is ", percent_free, "%"
$ !
$ !   Since DCL does signed 32 bit arithmetic, all calculations and results
$ !   are done with the sign bit (bit 31) clear.  So, for values of free
$ !   blocks and volume size larger than %x7FFFFFFF, first clear the sign
$ !   bit, divide by two (shift right one bit) and then turn on bit 30.
$ !
$     freeblocks = f$getdvi(p1,"freeblocks")
$     volsize = f$getdvi(p1,"volsize")
$ !
$     if (freeblocks .lt. 0)
$     then pipe freeblocks[31,1] = 0 ; freeblocks = freeblocks / 2 ;
 freeblocks[30,1] = 1
$     else freeblocks = freeblocks / 2
$     endif
$ !
$     if (volsize .lt. 0)
$     then pipe volsize[31,1] = 0 ; volsize = volsize / 2 ; volsize[30,1] =
 1
$     else volsize = volsize / 2
$     endif
$ !
$     if ((volsize / 100) .gt. 0)
$     then percent_free == freeblocks / (volsize / 100)
$     else percent_free == 0
$     endif
$ !
$   endsubroutine
```

B.3.4. F$CUNITS() lexical function description

The pre V8.4 F$CUNITS() lexical function allowed conversion of a value of blocks to the corresponding value of bytes. The output of the lexical was automatically scaled to KB, MB, and so on depending upon the size of the resultant value. In Version 8.4 the lexical is enhanced to support bytes to blocks conversions. Additional units keyword values are introduced to allow the user to specify the resultant value scaling. The format of a F$CUNITS lexical call is:

```
 result = F$CUNITS(value, from-units, to-Units)
```

where:

value: a value in the range 1 to 4294967295

Pre V8.4:

```
 from-units:  "BLOCKS"  ! optional, default is BLOCKS
   to-units:   "BYTES"  ! optional, default resultant BYTES are autoscaled
```

V8.4:

```
    from-units:  unit      ! optional, default is BLOCKS
   to-units:  unit      ! optional, default resultant BYTES are autoscaled
```

where unit is one of:

```
  "BYTES","BLOCKS","KB","MB","GB","TB"
```

For large disks it may be more convenient to work or display values in units other than blocks.

The next example demonstrates usage of the F$CUNITS function to convert blocks to bytes. It calculates the percent of free disk space while exercising the F$CUNITS lexical. The example also contains an easy method to approximate the percentage.

```
$!
$ if p1 .nes. "" Then GOTO show_device_info
$ thisproc = f$environment("procedure")
$Loop:
$ disk = F$DEVICE("*","DISK")
$ if disk .eqs. "" then EXIT
$ @'thisproc' 'disk'
$ goto Loop
$show_device_info:
$! Example:
$!
$!   Demonstrate the use of the F$CUNITS() lexical and
$!   calculate the percent free of a disk
$!
$ disk = P1
$ if P1 .eqs. "" then disk = "sys$sysdevice"
$ if .NOT.  ( f$getdvi(disk,"EXISTS") .AND.  -
              f$getdvi(disk,"AVL")    .AND.  -
              f$getdvi(disk,"MNT")   ) then  EXIT
$ write sys$output "Results for disk : ''disk'"
$ line = f$fao("!4(16AS)","Units","Total", "FREE", "Percent")
$ write sys$output line
$!
$! do direct calculation or approximation first  (BLOCKS)
$!
$ TotalSize = f$fao("!UL",f$getdvi(disk,"MAXBLOCK"))
$ FreeSize = f$fao("!UL",f$getdvi(disk,"FREEBLOCKS"))
$ TotalDisplay = TotalSize
$ FreeDisplay = FreeSize
$ if f$length(TotalSize) .lt. 7 then GOTO No_approximation
$ TotalSize =  f$extract(0,7,f$fao("!10ZL",f$getdvi(disk,"MAXBLOCK")))
$ FreeSize  =  f$extract(0,7,f$fao("!10ZL",f$getdvi(disk,"FREEBLOCKS")))
$!
$No_approximation:
$ uindex = -1
$ units = "BLOCKS,KB,MB,GB,TB"
$!
$! show approximation
$!
$ unit = "Approx"
$!
$ GOTO Compute_Percent
$!
$Fcunits_display:
$!
$Get_Unit:
$!--------
$ unit = f$element(uindex,",",units)
$ if unit .eqs. "," then EXIT
$!
```

```
$ TotalSize  =  f$cunits(f$string(f$fao("!UL",f
$getdvi(disk,"MAXBLOCK"))),"BLOCKS",unit)  - "''unit'"
$ FreeSize   =  f$cunits(f$string(f$fao("!UL",f
$getdvi(disk,"FREEBLOCKS"))),"BLOCKS",unit) -  "''unit'"
$!
$ TotalDisplay = TotalSize
$ FreeDisplay = FreeSize
$!
$! for unit >= gigabytes retain the fractional value
$!
$ if uindex .lt. 4 then goto Edit_Units_002
$!
$ if f$locate(".",TotalSize) .eq. f$length(TotalSize)
$ then
$       TotalSize = TotalSize + "00"
$ else
$       TotalSize = TotalSize - "."
$ endif
$ if f$locate(".",FreeSize) .eq. f$length(FreeSize)
$ then
$       FreeSize = FreeSize + "00"
$ else
$       FreeSize = FreeSize - "."
$ endif
$!
$ GOTO Compute_Percent
$!
$Edit_Units_002:
$!
$! For units < Gigabytes truncate any fraction
$!
$ if f$locate(".",TotalSize) .lt. f$length(TotalSize)
$ then
$       TotalSize = f$extract(0,f$length(TotalSize)-3,TotalSize)
$ endif
$!
$ if f$locate(".",FreeSize) .lt. f$length(FreeSize)
$ then
$       FreeSize = f$extract(0,f$length(FreeSize)-3,FreeSize)
$ endif
$!
$Compute_Percent:
$!
$! check for overflow on percentage calculation
$!
$ PCTfree = -1
$ if (((f$integer(FreeSize)*100)/100) .ne. f$integer(FreeSize)) .OR. -
      f$length(FreeSize) .gt. 8 .OR f$length(TotalSize) .gt. 8
$ then
$       PCTfree = "Overflow"
$ else
$       if TotalSize .eq. 0 .OR. FreeSize .eq. 0
$       then
$               PCTfree = 0
$       else
$               PCTfree = f$integer( (FreeSize*100) / TotalSize)
$       endif
$ endif
```

```
$!
$ line =   f$fao("!4(16AS)",Unit, TotalDisplay, FreeDisplay, f
$string(PCTfree))
$ write sys$output line
$!
$ uindex = uindex + 1
$ GOTO Get_Unit
$!
$ EXIT
$!
B.4 Identifying Suspect DCL Code
This section describes some ways to identify DCL code and source code
 modules that may not function correctly when accessing devices over 1TB
 size.

B.4.1 DCL Code Search Example
This example demonstrates a simple search to identify DCL code lines that
 definitely need to be addressed. DCL code may supply lexical items as
 symbols or split lexical calls into multiple lines and this procedure will
 not identify such cases.

$ search := search/numbers/wind=1/MATCH=AND/statistics
$ filespec = 'your_filespec'
$!
$! The results of the search command on the next 3 lines
$! show DCL command lines that MUST be addressed on systems
$! accessing disks over 1tb.
$ search  'filespec'  "F$GETDVI","MAXBLOCK"
$ search  'filespec'  "F$GETDVI","VOLSIZE"
$ search  'filespec'  "F$GETDVI","EXPSIZE"
$!
$! The results of the search command on the next line
$! show DCL command lines that should be changed on
$! systems accessing disks over 1tb.
$! The DCL command is likely to return a value that will
$! not fit in a signed longword.
$ search  'filespec'  "F$GETDVI","FREEBLOCKS"
```

B.4. Identifying Suspect DCL Code

This section describes some ways to identify DCL code and source code modules that may not function correctly when accessing over 1 TB devices.

B.4.1. DCL Code Search Example

This example demonstrates a simple search to identify DCL code lines that definitely need to be addressed. DCL code may supply lexical items as symbols or split lexical calls into multiple lines and this procedure will not identify such cases.

```
$ search := search/numbers/wind=1/MATCH=AND/statistics
$ filespec = 'your_filespec'
$!
$! The results of the search command on the next 3 lines
$! show DCL command lines that MUST be addressed on systems
$! accessing disks over 1tb.
$ search  'filespec'  "F$GETDVI","MAXBLOCK"
```

```
$ search  'filespec'  "F$GETDVI","VOLSIZE"
$ search  'filespec'  "F$GETDVI","EXPSIZE"
$!
$! The results of the search command on the next line
$! show DCL command lines that should be changed on
$! systems accessing disks over 1tb.
$! The DCL command is likely to return a value that will
$! not fit in a signed longword.
$ search  'filespec'  "F$GETDVI","FREEBLOCKS"
```