VAX/VMS DCL Dictionary

Order Number: AA–Z200C–TE

April 1986

This manual provides detailed reference information and examples on all VAX/VMS DCL commands and lexical functions.

Revision/Update Information:	This is a revised manual.
Software Version:	VAX/VMS Version 4.4

digital equipment corporation
maynard, massachusetts

April 1986

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	UNIBUS
DEC/CMS	EduSystem	VAX
DEC/MMS	IAS	VAXcluster
DECnet	MASSBUS	VMS
DECsystem–10	PDP	VT
DECSYSTEM–20	PDT	
DECUS	RSTS	
DECwriter	RSX	digital

ZK–3032

HOW TO ORDER ADDITIONAL DOCUMENTATION
DIRECT MAIL ORDERS

USA & PUERTO RICO*

Digital Equipment Corporation
P.O. Box CS2008
Nashua, New Hampshire
03061

CANADA

Digital Equipment
of Canada Ltd.
100 Herzberg Road
Kanata, Ontario K2K 2A6
Attn: Direct Order Desk

INTERNATIONAL

Digital Equipment Corporation
PSG Business Manager
c/o Digital's local subsidiary
or approved distributor

In Continental USA and Puerto Rico call 800-258-1710.

In New Hampshire, Alaska, and Hawaii call 603-884-6660.

In Canada call 800-267-6215.

*Any prepaid order from Puerto Rico must be placed with the local Digital subsidiary (809-754-7575).

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Westminster, Massachusetts 01473.

This document was prepared using an in-house documentation production system. All page composition and make-up was performed by TEX, the typesetting system developed by Donald E. Knuth at Stanford University. TEX is a registered trademark of the American Mathematical Society.

Contents

COMMAND DESCRIPTIONS

INDEX

TABLES

Preface

Intended Audience

This manual is intended for all users of the VAX/VMS operating system, as it describes all DCL commands and lexical functions. If a command has any restrictions or requires special privileges, they are noted in reference information for that command.

If you are not familiar with an interactive computer system, you should read the *Introduction to VAX/VMS* before using this manual. For general information regarding the VAX/VMS DCL command language, see the *VAX/VMS DCL Concepts Manual.*

Structure of This Document

This manual contains detailed descriptions of each command and lexical function. The commands are listed is alphabetical order, with the command name appearing at the top of every page. The lexical functions are grouped under "Lexical Functions" (after the JOB command description), and are listed alphabetically within that grouping; the lexical function name appears at the top of each page.

Readers of this manual should be familiar with the material covered in *VAX/VMS DCL Concepts Manual.* Furthermore, while the *Guide to Using DCL and Command Procedures on VAX/VMS* is not a requirement for using this manual, it does help clarify some of the examples involving command procedures.

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

Those familiar with previous releases of the *VAX/VMS DCL Dictionary* should note that the general overview of DCL command language concepts (formerly, Part 1) has been removed. For a general discussion of the DCL command language, see the *VAX/VMS DCL Concepts Manual.*

Associated Documents

For an introduction to the VAX/VMS operating system and the use of the Digital Command Language, see the *Introduction to VAX/VMS.* This manual is especially recommended for novice users or users lacking experience with interactive computer systems.

The *VAX/VMS DCL Concepts Manual* provides a general overview of DCL command language concepts.

The *Guide to Using DCL and Command Procedures on VAX/VMS* defines and illustrates good practices in constructing command procedures with DCL commands and lexical functions.

The various VAX/VMS Utilities reference manuals document major VAX/VMS Utilities. These manuals describe the DCL commands that invoke the various utilities, any commands that you can enter while running a utility, and other information. For all utilities documented in these volumes, the *VAX/VMS DCL Dictionary* provides only a brief description and format information.

The *VAX/VMS System Messages and Recovery Procedures Reference Manual* explains any error and warning messages you may receive. In most cases, however, you will not need to refer to this manual, because it will be obvious from the message text and other information on your screen what action (if any) to take.

The *Introduction to the VAX/VMS Document Set* describes the new organization of the VAX/VMS document set. This manual shows how the individual manuals fit together and relate to each other.

Conventions Used in This Document

Convention	Meaning
RET	A symbol with a one- to three-character abbreviation indicates that you press a key on the terminal, for example, RET .
CTRL/x	The phrase CTRL/x indicates that you must press the key labeled CTRL while you simultaneously press another key, for example, CTRL/C, CTRL/Y, CTRL/O.
$ SHOW TIME 04-FEB-1986 11:55:22	Command examples show all output lines or prompting characters that the system prints or displays in black letters. All user-entered commands are shown in red letters.
$ TYPE MYFILE.DAT . . .	Vertical series of periods, or ellipsis, mean either that not all the data that the system would display in response to the particular command is shown or that not all the data a user would enter is shown.
file-spec,...	Horizontal ellipsis indicates that additional parameters, values, or information can be entered.
[logical-name]	Square brackets indicate that the enclosed item is optional. (Square brackets are not, however, optional in the syntax of a directory name in a file specification or in the syntax of a substring specification in an assignment statement.)
quotation marks apostrophes	The term quotation marks is used to refer to double quotation marks (″). The term apostrophe (′) is used to refer to a single quotation mark.

New and Changed Features

The DIGITAL Command Language has been enhanced for VAX/VMS Version 4.4. The "General Notes" section below lists general changes, as well as changes in the behavior of existing commands. Subsequent sections list new commands and new subroutine contructs.

General Notes

Two new features have been implemented for the SHOW SYMBOL command. When executing a SHOW SYMBOL/ALL command, the list of symbols will now be listed in alphabetic order. SHOW SYMBOL will also support the standard wildcard characters (* and %).

SHOW DEFAULT now expands a search list in the order that VAX RMS would process the list. It also displays the default directory specification separately if SYS$DISK contains a full directory specification. In addition, it will generate an informational message when the current default does not exist.

New Commands

The following new commands have been added since the Version 4.2 edition of the *VAX/VMS DCL Dictionary*:

CALL
GOSUB
SET RIGHTS_LIST
SET SYMBOL/SCOPE=([NO]LOCAL,[NO]GLOBAL)
RETURN

New Qualifiers and Keywords for Existing Commands

The following new command qualifiers and keywords have been added.

INITIALIZE/QUEUE/DEFAULT= FORM=type
SET PROCESS /CPU
SET QUEUE/DEFAULT= FORM=type
SHOW QUEUE /FILES
START/QUEUE/DEFAULT= FORM=type
START/QUEUE/MANAGER /RESTART

Command Descriptions

= (Assignment Statement)

Defines a symbolic name for a character string or integer value.

FORMAT

symbol-name =[=] expression

symbol-name[bit-position,size] =[=] replacement-expression

restrictions

None.

PARAMETERS

symbol-name

Defines a 1- through 255-character alphanumeric string name for the symbol. The symbol name must begin with an alphabetic character (uppercase and lowercase characters are equivalent), an underscore, or a dollar sign. After the first character, the name can contain any alphanumeric characters from the DEC Multinational Character Set, underscores, or dollar signs.

If you specify a single equal sign (=) in the assignment statement, the symbol name is placed in the local symbol table for the current command level.

If you specify double equal signs (==) in the assignment statement, the symbol name is placed in the global symbol table.

expression

Specifies the value on the right-hand side of an assignment statement. An expression can consist of a character string value, an integer value, a symbol name, a lexical function, or a combination of these entities. The components of an expression are evaluated, and the result is assigned to a symbol.

All literal character strings must be enclosed in quotation marks. If you specify an expression containing a symbol, the command interpreter uses the symbol's value in evaluating the expression.

The result of expression evaluation is either a character string or an integer value. The data type of a symbol is determined by the data type of the expression. If the expression evaluates to a string, the symbol is assigned a string value. If the expression evaluates to an integer, the symbol is assigned an integer value.

For a summary of operators used in expressions, details on how to specify expressions, and details on how expressions are evaluated, see Chapter 5 of the *VAX/VMS DCL Concepts Manual*.

DCL uses a buffer that is 1024 bytes long to hold an assignment statement, and to evaluate the expression. The length of the symbol name, the expression, and the expression's calculations cannot exceed 1024 bytes.

[bit-position,size]

Specifies that a binary overlay is to be inserted in the current 32-bit value of a symbol-name. This type of assignment statement evaluates the current value of the symbol-name and replaces the specified number of bits with the result of the replacement-expression.

The bit-position is the location relative to bit 0 at which the overlay is to occur. If the symbol that you are overlaying is an integer, then the bit position must be less than 32. Also, the sum of the bit position and the size must be less than or equal to 32.

If the symbol you are overlaying is a string, then the bit position must be less than 6152. Because each character is represented using 8 bits, you can begin an overlay at any character through the 768th. (The 768th character starts in bit position 6144.) The sum of the bit position and the size must be less than or equal to 6152.

The size is the number of bits to be overlaid. If you specify a size that is greater than 32, then DCL reduces the size to 32.

The square brackets are required notation; no spaces are allowed between the symbol name and the left bracket. Specify values for bit-position and size as integer expressions.

replacement-expression

Specifies the value that is used to overlay the symbol you are modifying. Specify the replacement-expression as an integer expression.

If the symbol you are modifying is an integer, the replacement-expression defines a bit pattern which is overlaid on the value assigned to the symbol. If the symbol you are modifying is a character string, the result of the replacement-expression defines a bit pattern which is overlaid on the specified bits of the character string. If the symbol you are modifying is undefined, the result of the replacement-expression is overlaid on a null string.

DESCRIPTION

Symbols defined using assignment statements allow you to extend the command language. At the interactive command level, you can use symbols to define synonyms for commands or command lines. In command procedure files, you can use symbols to provide for conditional execution and substitution of variables.

The maximum number of symbols that can be defined at any time depends on:

- The amount of space available to the command interpreter to contain symbol tables and labels for the current process. The amount of space is determined for each process by the SYSGEN parameter CLISYMTBL.
- The size of the symbol names and their values. The command interpreter allocates space for a symbol name and its value. In addition, a few bytes of overhead are allocated for each symbol.

EXAMPLES

1

```
$ LIST == "DIRECTORY"
$ TIME == "SHOW TIME"
$ QP == "SHOW QUEUE/DEVICE"
$ SS == "SHOW SYMBOL"
```

The file SYNONYM.COM contains the assignment statements shown; these are user-defined synonyms for commands. Execute this command procedure as follows:

```
$ @SYNONYM
```

After the command procedure creates the global symbol definitions, you can use these synonyms (LIST, TIME, QP, and SS) at the interactive command level. Note that the assignments are global; otherwise, the symbol names would have been deleted after the file SYNONYM.COM completed execution.

2

```
$ COUNT = 0
$ LOOP:
$       COUNT = COUNT + 1
          .
          .
          .
$       IF COUNT .LT.5 THEN GOTO LOOP
```

The symbol COUNT is initially assigned a numeric value of 0; a loop is established to increment the value of COUNT by 1 each time the loop is entered. Note that when the symbol name COUNT appears on the right-hand side of an arithmetic assignment statement, the command interpreter automatically substitutes its current value.

The IF command tests the value of COUNT; if it is less than 5, the procedure branches to the label LOOP and the statements between the label LOOP through the IF command are executed again. When the value of the symbol count reaches 5, the loop is not executed again and the command following the IF command is executed.

3

```
$ COUNT = 0
$ LOOP:
$       COUNT = COUNT + 1
$       IF P'COUNT' .EQS. "" THEN EXIT
$       APPEND/NEW &P'COUNT' SAVE.ALL
$       DELETE &P'COUNT';*
$       IF COUNT .LT. 8 THEN GOTO LOOP
$ EXIT
```

This command procedure, COPYDEL.COM, appends files (specified as parameters) to a file called SAVE.ALL. After a file has been appended, the command procedure deletes the file. Up to eight file names can be passed to the command procedure. The file names are assigned to the symbols P1, P2, and so on.

The command procedure uses a counter to refer to parameters that are passed to it. Each time through the loop, the procedure uses an IF command to check whether the value of the current parameter is a null string. When the IF command is scanned, the current value of the symbol COUNT is concatenated with the letter P. The first time through the loop, the IF command tests P1; the second time through the loop it tests P2, and so on. After the expression P'COUNT' is evaluated, the substitution of the file names that correspond to P1, P2, and so on, is automatic within the context of the IF command.

The APPEND and DELETE commands, however, do not automatically perform any substitution, because they expect and require file specifications as input parameters. The ampersand (&) precedes the P'COUNT' expression for these commands to force the appropriate symbol substitution. When these commands are initially scanned each time through the loop, COUNT is substituted with its current value. Then, when the commands execute, the ampersand causes another substitution: the first file specification is substituted for P1, the second file specification is substituted for P2, and so on.

You can invoke this procedure with the command:

```
$  @COPYDEL ALPHA.TXT BETA.DOC
```

The files ALPHA.TXT and BETA.DOC are each appended to the file SAVE.ALL and then deleted.

4
```
$ A = 25
$ CODE = 4 + F$INTEGER("6") - A
$ SHOW SYMBOL CODE
  CODE = -15   HEX = FFFFFFF1   Octal = 1777761
```

This example contains two assignment statements. The first statement assigns the value 25 to the symbol A. The second assignment statement evaluates an expression containing an integer (4), a lexical function (F$INTEGER("6")), and the symbol A. The result of the expression, –15, is assigned to the symbol CODE.

5
```
$ FILENAME = "JOBSEARCH" - "JOB"
$ FILETYPE = ".OBJ"
$ FILESPEC = FILENAME + FILETYPE
$ TYPE 'FILESPEC'
```

The first command in this example assigns the symbol FILENAME the value "SEARCH". Notice that the string "SEARCH" is the result of the string reduction operation performed by the expression. The second command assigns the symbol FILETYPE the character string ".OBJ". The symbols FILENAME and FILETYPE are then added together in an expression assigned to the symbol FILESPEC. Since the values of the symbols FILENAME and FILETYPE are concatenated, the resultant value assigned to FILESPEC is the character string "SEARCH.OBJ". The symbol FILESPEC is then used as a parameter for the TYPE command. The apostrophes request the command interpreter to replace the symbol FILESPEC with its value SEARCH.OBJ. Thus, the TYPE command types the file named SEARCH.OBJ.

6
```
$ BELL[0,32] = %X07
$ SHOW SYMBOL BELL
  BELL = ""
```

In this example, the symbol BELL is created with an arithmetic overlay assignment statement. Because the symbol BELL is previously undefined, the hexadecimal value 7 is inserted over a null character string and is interpreted as the ASCII code for the bell character on a terminal. When you issue the command SHOW SYMBOL BELL, the terminal beeps.

If the symbol BELL had been previously defined with an integer value, the result of displaying BELL would have been simply to show its new integer value.

:= (String Assignment)

Defines a symbolic name for a character string value.

FORMAT

symbol-name :=[=] string

symbol-name[offset,size] :=[=] replacement-string

restrictions

None.

PARAMETERS

symbol-name

Defines a 1- through 255-character string name for the symbol. The symbol name must begin with an alphabetic character (uppercase and lowercase characters are equivalent), an underscore, or a dollar sign. After the first character, the name can contain any alphanumeric characters from the DEC Multinational Character Set, underscores, or dollar signs.

If you specify a single equal sign (:=) in the assignment statement, the symbol name is placed in the local symbol table for the current command level.

If you specify double equal signs (:==) in the assignment statement, the symbol name is placed in the global symbol table.

string

Specifies a character string value to be equated to the symbol. The string can contain any alphanumeric or special characters. DCL uses a buffer that is 1024 bytes long to hold a string assignment statement. Therefore, the length of the symbol name, the string, and any symbol substitution within the string cannot exceed 1024 characters.

With the := string assignment statement, you do not need to enclose a string literal in quotation marks. String values are automatically converted to uppercase. Also, any leading and trailing spaces and tabs are removed, and multiple spaces and tabs between characters are compressed to a single space.

Note that, in general, it is easier to use the assignment statement (=) to create symbols with string values. The assignment statement does not automatically convert letters to uppercase and remove extra spaces. Also, the assignment statement allows you to perform string operations in expressions.

If you want to prohibit uppercase conversion and retain required space and tab characters in a string, you must place quotation marks around the string. To use quotation marks in a string, enclose the entire string in quotation marks and use a double set of quotation marks within the string. For example:

```
$ TEST := "this     is a ""test"" string"
$ SHOW SYMBOL TEST
  TEST = "this     is a "test" string"
```

In this example, the spaces, lowercase letters, and quotation marks are preserved in the symbol definition.

You can continue a symbol assignment on more than one line by using the hyphen as a continuation character. For example:

```
$ LONG_STRING := THIS_IS_A_VERY_LONG-
_$ _SYMBOL_STRING
```

To assign a null string to a symbol using the string assignment statement, do not specify a string. For example:

```
$ NULL :=
```

Specify the string as a string literal, or as a symbol or lexical function which evaluates to a string literal. If you use symbols or lexical functions, place apostrophes around them to request symbol substitution. See Chapter 6 of the *VAX/VMS DCL Concepts Manual* for more information on symbol substitution.

You can also use the string assignment statement to define a foreign command. See Section 5.11 of the *VAX/VMS DCL Concepts Manual* for more information about foreign commands.

[offset,size]

Specifies that a portion of a symbol value is to be overlaid with a replacement string. This form of the string assignment statement evaluates the value assigned to a symbol, and then replaces the portion of the value (defined by the offset and size) with the replacement string. The square brackets are required notation, and no spaces are allowed between the symbol name and the left bracket.

The offset specifies the character position relative to the beginning of the symbol-name's string value at which replacement is to begin. Offset values start at 0.

If the offset is greater than the offset of the last character in the string you are modifying, spaces are inserted between the end of the string and the offset where the replacement string is added. The maximum offset value you can specify is 768.

The size specifies the number of characters to replace. Size values start at 1.

Specify the offset and size as integer expressions. See Section 5.7.2 of the *VAX/VMS DCL Concepts Manual* for more information on integer expressions. The value of the size plus the offset must not exceed 769.

replacement-string

Specifies the string that is used to overlay the string you are modifying. If the replacement-string is shorter than the size argument, the replacement string is blank-filled on the right until it equals the specified size. Then the replacement string is used to overlay the string assigned to the symbol-name. If the replacement string is longer than the size argument, then the replacement string is truncated on the right to the specified size.

You can specify the replacement-string as a string literal, or as a symbol or lexical function which evaluates to a string literal. If you use symbols or lexical functions, place apostrophes around them to request symbol substitution. See Chapter 6 of the *VAX/VMS DCL Concepts Manual* for more information on symbol substitution.

EXAMPLES

1

```
$ TIME := SHOW TIME
$ TIME
  15-APR-1984 11:55:44
```

The symbol TIME is equated to the command string SHOW TIME. Because the symbol name appears as the first word in a command string, the command interpreter automatically substitutes it with its string value and executes the command SHOW TIME.

2

```
$ STAT := $DBA1:[CRAMER]STAT
$ STAT
```

This example shows how to define STAT as a foreign command. The symbol STAT is equated to a string that begins with a dollar sign followed by a file specification. The command interpreter assumes that the file specification is that of an executable image, that is, a file with a file type of EXE. Thus, the symbol STAT in this example becomes a synonym for the command:

```
$ RUN DBA1:[CRAMER]STAT.EXE
```

When you subsequently type STAT, the command interpreter executes the image.

3

```
$ A = "this is a big     space."
$ SHOW SYMBOL A
  A = "this is a big     space."
$ B := 'A'
$ SHOW SYMBOL B
  B = "THIS IS A BIG SPACE."
```

This example compares the assignment and the string assignment statements. The symbol A is defined using the assignment statement, so lowercase letters and multiple spaces are retained. The symbol B is defined using the string assignment statement. Note that the apostrophes are required; otherwise, the symbol name B would have been equated to the literal string A. However, when symbol A's value is assigned to symbol B, the letters are converted to uppercase and multiple spaces are compressed.

4

```
$ FILE_NAME := MYFILE
$ FILE_NAME[0,2]:= OL
$ SHOW SYMBOL FILE_NAME
  FILE_NAME = "OLFILE"
```

The substring expression in the assignment statement overlays the first two characters of the string assigned to the symbol FILE_NAME with the letters OL. The offset of 0 requests that the overlay begin with the first character in the string, and the size specification of 2 indicates the number of characters to overlay.

:= (String Assignment)

5
```
$ FILE_NAME := MYFILE
$ FILE_TYPE := .TST
$ FILE_NAME[F$LENGTH(FILE_NAME),4] := 'FILE_TYPE'
$ SHOW SYMBOL FILE_NAME
  FILE_NAME = "MYFILE.TST"
```

In this example, the symbol name FILE_NAME is equated to the string MYFILE and the symbol name FILE_TYPE is equated to the string .TST. The third assignment statement uses the lexical function F$LENGTH to define the offset value where the overlay is to begin. The symbol name FILE_TYPE is used to refer to the replacement string (.TST). Note that you must use apostrophes to request symbol substitution.

The F$LENGTH lexical function returns the length of the string equated to the symbol FILE_NAME; this length is used as the offset. The expression requests that four characters of the string currently equated to the symbol FILE_TYPE be placed at the end of the string currently equated to FILE_NAME. The resultant value of the symbol FILE_NAME is MYFILE.TST.

@ (Execute Procedure)

Executes a command procedure or requests the command interpreter to read subsequent command input from a specific file or device.

FORMAT

@ *file-spec [p1 [p2 [... p8]]]*

restrictions

None.

PARAMETERS *file-spec*

Specifies either the command procedure to be executed or the device or file from which input for the preceding command is to be read.

If you do not specify a file type, the system uses the default file type of COM.

No wildcard characters are allowed in the file specification.

p1 [p2 [... p8]]

Specifies from one to eight optional parameters to pass to the command procedure. The parameters assign character string values to the symbols named P1, P2, and so on in the order of entry, to a maximum of eight. The symbols are local to the specified command procedure.

Separate each parameter with one or more blanks. Use quotation marks (" ") to specify a null parameter. You can specify a parameter with a character string value containing alphanumeric or special characters, with the following restrictions:

- The command interpreter converts alphabetic characters to uppercase and uses blanks to delimit each parameter. To pass a parameter that contains embedded blanks or literal lowercase letters, place the parameter in quotation marks.
- If the first parameter begins with a slash character (/), you must enclose the parameter in quotation marks.
- To pass a parameter that contains literal quotation marks and spaces, enclose the entire string in quotation marks and use a double set of quotation marks within the string. For example:

```
$ @TEST "Never say ""quit"""
```

When the procedure TEST.COM executes, the parameter P1 is equated to the string:

```
Never say "quit"
```

If a string contains quotation marks and does not contain spaces, the quotation marks are preserved in the string and the letters within the quotation marks remain in lowercase. For example:

```
$ @ANOTHER_TEST abc"def"ghi
```

When the procedure ANOTHER_TEST.COM executes, the parameter P1 is equated to the string:

```
ABC"def"GHI
```

To use a symbol as a parameter, enclose the symbol in apostrophes to force symbol substitution. For example:

```
$ NAME = "JOHNSON"
$ @INFO 'NAME'
```

The apostrophes cause the value "JOHNSON" to be substituted for the symbol NAME. Therefore, the parameter "JOHNSON" is passed as P1 to INFO.COM.

DESCRIPTION

You can use the @ command to execute a command procedure containing:

- DCL command lines and/or data
- Qualifiers and/or parameters for a specific command line

To execute a command procedure containing commands and/or data, place the @ command at the beginning of a command line and then specify the name of the command procedure file. The command procedure can contain DCL commands and/or input data for a command or program that is currently executing. All DCL commands in a command procedure must begin with a dollar sign ($) character. If a command is continued with the continuation character (-), the subsequent lines must not begin with a dollar sign.

Any line in a command procedure that does not contain a dollar sign in the first character position (and is not a continuation line) is treated as input data for the command or program that is currently executing. The DECK command allows you to specify that data contains dollar signs in record position one.

A command procedure can also contain the @ command to execute another command procedure. The maximum command level you can achieve by nesting command procedures is sixteen, including the top-level command procedure. Command procedures can also be queued for processing as batch jobs, either by using the SUBMIT command or by placing a deck of cards containing the command procedure in the system card reader.

To execute a command procedure that contains qualifiers and/or parameters for a specific command line, place the @ command where the qualifiers and/or parameters would normally be in the command line. Then, specify the name of the command procedure file containing the qualifiers and/or parameters.

If the command procedure file begins with parameters for the command, the @ command must be preceded by a space. If the file begins with qualifiers for the command, the @ command must not be preceded with a space. In a file containing only parameters and/or qualifiers, the lines must not begin with dollar signs ($). Any additional data on the command line following @file-spec is treated as parameters for the procedure.

QUALIFIER

/OUTPUT=file-spec

Requests that all output directed to the logical device SYS$OUTPUT be written to the file or device specified. System responses and error messages are written to SYS$COMMAND as well as to the specified file.

If you specify /OUTPUT, the qualifier must immediately follow the file specification of the command procedure; otherwise, the qualifier is interpreted as a parameter to pass to the command procedure.

The default output file type is LIS.

No wildcard characters are allowed in the output file specification.

You can also redefine SYS$OUTPUT to redirect the output from a command procedure. If you place the command:

```
$ DEFINE SYS$OUTPUT file-spec
```

as the first line in a command procedure, output will be directed to the file you specify. When the procedure exits, SYS$OUTPUT will be restored to its original equivalence string. This produces the same result as using the /OUTPUT qualifier when you execute the command procedure.

EXAMPLES

1

```
$ CREATE DOFOR.COM
$ ON WARNING THEN EXIT
$ IF P1.EQS."" THEN INQUIRE P1 FILE
$ FORTRAN/LIST 'P1'
$ LINK 'P1'
$ RUN 'P1'
$ PRINT 'P1'
CTRL/Z
$ @DOFOR AVERAGE
```

This example shows a command procedure, named DOFOR.COM, that executes the FORTRAN, LINK, and RUN commands to compile, link, and execute a program. The ON command requests that the procedure not continue if any of the commands result in warnings or errors.

When you execute DOFOR.COM, you can pass the file specification of the FORTRAN program as the parameter P1. If you do not specify a value for P1 when you execute the procedure, the INQUIRE command issues a prompting message to the terminal and equates what you enter with the symbol P1. In this example, the file name AVERAGE is assigned to P1. The file type is not included because the commands FORTRAN, LINK, RUN, and PRINT provide default file types.

2

```
$ @MASTER/OUTPUT=MASTER.LOG
```

This command executes a procedure named MASTER.COM; all output is written to the file MASTER.LOG.

3

```
$ CREATE FILES.COM
*.FOR, *.OBJ
CTRL/Z
$ DIRECTORY @FILES
```

This example shows a command procedure, FILES.COM, that contains parameters for a DCL command line. You can execute this procedure after the DIRECTORY command to get a listing of all FORTRAN source and object files in your current default directory.

@ (Execute Procedure)

4
```
$ CREATE QUALIFIERS.COM
/DEBUG/SYMBOL_TABLE/MAP/FULL/CROSS_REFERENCE
CTRL/Z
$ LINK SYNAPSE@QUALIFIERS
```

This example shows a command procedure, QUALIFIERS.COM, that contains qualifiers for the LINK command. When you issue the LINK command, specify the command procedure immediately after the file specification of the file you are linking. Do not type a space between the file specification and the @ command.

5
```
$ CREATE SUBPROCES.COM
$ RUN 'P1' -
  /BUFFER_LIMIT=1024 -
  /FILE_LIMIT=4 -
  /PAGE_FILES=256 -
  /QUEUE_LIMIT=2 -
  /SUBPROCESS_LIMIT=2 -
  'P2'  'P3'  'P4'  'P5'  'P6'  'P7'  'P8'
^Z
$ @SUBPROCES  LIBRA  /PROCESS_NAME=LIBRA
```

This example shows a command procedure named SUBPROCES.COM. This procedure issues the RUN command to create a subprocess to execute an image, and also contains qualifiers defining quotas for subprocess creation. The name of the image to be run is passed as the parameter P1. P2 through P8 can be used to specify additional qualifiers.

In this example, the file name LIBRA is equated to P1; it is the name of an image to execute in the subprocess. The qualifier /PROCESS_NAME=LIBRA is equated to P2; it is an additional qualifier for the RUN command.

6
```
$ CREATE EDOC.COM
$ ASSIGN SYS$COMMAND:  SYS$INPUT
$ NEXT:
$       INQUIRE NAME "File name"
$       IF NAME.EQS."" THEN EXIT
$       EDIT/EDT 'NAME'.DOC
$       GOTO NEXT
CTRL/Z
$ @EDOC
```

This procedure, named EDOC.COM, invokes the EDT editor. When an edit session is terminated, the procedure loops to the label NEXT. Each time through the loop, the procedure requests another file name for the editor and supplies the default file type of DOC. When a null line is entered in response to the INQUIRE command, the procedure terminates with the EXIT command.

The ASSIGN command changes the equivalence name of SYS$INPUT for the duration of the procedure. This change allows the EDT editor to read input data from the terminal, rather than from the command procedure file (the default input data stream if SYS$INPUT had not been changed.) When the command procedure exits, SYS$INPUT is reassigned to its original value.

ACCOUNTING

Invokes the VAX/VMS Accounting Utility to collect, record, and report accounting data. For a complete description of the Accounting Utility, including information about the ACCOUNTING command, refer to the *VAX/VMS Accounting Utility Reference Manual*.

FORMAT **ACCOUNTING** *file-spec[,...]*

ALLOCATE

Provides exclusive access to a device and optionally establishes a logical name for the device. Once a device has been allocated, other users cannot access the device until you specifically deallocate it or log out.

FORMAT

ALLOCATE *device-name[:][,...] [logical-name[:]]*

restrictions

None.

PARAMETERS

device-name[:][,...]

Specifies the names of one or more devices to be searched for the allocation request. Device names can be generic device names, so that if no controller or unit number is specified, the system allocates the first available device that satisfies those components of the device names that are specified. If you specify "/GENERIC device-type," the system allocates the first available device of the specified name and type. If you specify more than one device name, the first nonallocated device in the list is allocated.

logical-name

Specifies a 1- through 255-character logical name to be associated with the allocated device. The logical name is placed in the process logical name table, with the name of the physical device allocated as its equivalence name. Subsequent references to the logical name result in automatic translation to the specified device name.

If you include a trailing colon (:) on the logical name, the colon is removed from the name before the name is placed in the logical name table.

QUALIFIERS

/GENERIC
/NOGENERIC (default)

Allows you to perform a generic allocation operation and specify a device type. The first free, nonallocated device of the specified name and type is allocated.

The /[NO]GENERIC qualifier is placed before the device-name parameter in the ALLOCATE command line. For example, you can allocate an RK07 device by entering the following command at the DCL prompt:

```
$ ALLOCATE/GENERIC RK07
```

The following table shows the device types that you can specify with the /GENERIC qualifier.

RA60/80/81	TA78/81
RC25/RCF25	TS11
RK06/7	TU16
RL01/2	TU77/78/80/81
RM03/05/80	
RP04/5/6/7	
RX01/2/4	
TU58	

/LOG (default)
/NOLOG

Displays a message indicating the name of the device allocated. If the operation specifies a logical name that is currently assigned to another device, displays the superseded value.

EXAMPLES

1
```
$ ALLOCATE  DMB2:
%DCL-I-ALLOC, DMB2: allocated
```

The ALLOCATE command requests the allocation of a specific RK06/RK07 disk drive, that is, unit 2 on controller B. The system response indicates that the device was successfully allocated.

2
```
$ ALLOCATE  MT,MF:   TAPE:
%DCL-I-ALLOC, MTB2: allocated
.
.
.
$ SHOW LOGICAL TAPE:
TAPE: = _MTB2:     (process)
$ DEALLOCATE TAPE:
$ DEASSIGN TAPE:
```

The ALLOCATE command requests the allocation of any tape device whose name begins with MT or MF, to be assigned the logical name TAPE. The ALLOCATE command locates an available tape device whose name begins with MT, and responds with the name of the device allocated. (If no tape device beginning with MT had been found, the ALLOCATE command would have searched for a device beginning with MF.) Subsequent references to the device TAPE in user programs or command strings are translated to the device name MTB2.

When the tape device is no longer needed, the DEALLOCATE command deallocates it and the DEASSIGN command deletes the logical name. Note that the logical name TAPE was specified with a colon on the ALLOCATE command, but that the logical name table entry does not have a colon.

3
```
$ ALLOCATE/GENERIC RL02 WORK
%DCL-I-ALLOC, _DLA1: allocated
%DCL-I-SUPERSEDE, previous value of WORK has been superseded
```

The ALLOCATE command requests the allocation of any RL02 disk device and assigns the logical name WORK to the device. The completion message identifies the allocated device and indicates that the assignment of the logical name WORK supersedes a previous assignment of that name.

ANALYZE/CRASH_DUMP

Invokes the System Dump Analyzer Utility (SDA) for analysis of a system dump file. The /CRASH_DUMP qualifier is required. For a complete description of the System Dump Analyzer Utility, including more information about the ANALYZE/CRASH_DUMP command and its qualifier, see the *VAX/VMS System Dump Analyzer Reference Manual*.

FORMAT **ANALYZE/CRASH_DUMP** *file-spec*

ANALYZE/DISK_STRUCTURE

Invokes the Verify Utility (VERIFY) to do the following:

- Check the readability and validity of Files–11 Structure Level 1 and Files–11 Structure Level 2 disk volumes
- Report errors and inconsistencies

The /DISK_STRUCTURE qualifier is required. For a complete description of the Verify Utility, including information about the DCL command ANALYZE/DISK_STRUCTURE and its qualifiers, see the *VAX/VMS Verify Utility Reference Manual*.

FORMAT

ANALYZE/DISK_STRUCTURE *device-name:*

ANALYZE/ERROR_LOG

Invokes the Errorlog Report Formatter (ERF) to selectively report the contents of an error log file. The /ERROR_LOG qualifier is required. For a complete description of the VAX/VMS Analyze Error Log Utility, including more information about the ANALYZE/ERROR_LOG command and its qualifiers, see the *VAX/VMS Error Log Utility Reference Manual*.

FORMAT

ANALYZE/ERROR_LOG *[file-spec[,...]]*

ANALYZE/IMAGE

Analyzes the contents of an executable image file or a shareable image file, and checks for obvious errors in the image file. The /IMAGE qualifier is required. (Use the ANALYZE/OBJECT to analyze the contents of an object file.)

FORMAT

ANALYZE/IMAGE *file-spec [,...]*

restrictions

None.

PARAMETER

file-spec[,...]

Specifies the name of one or more image files you want analyzed. You must specify at least one file name. If you specify more than one file, separate the file specifications with either commas or plus signs. The default file type is EXE.

Wildcard characters are allowed in the file specification.

DESCRIPTION

The ANALYZE/IMAGE command provides a description of the components of an executable image file or shareable image file. It also verifies that the structure of the major parts of the image file is correct. It cannot, however, ensure that program execution will be error-free.

If an error is found, the first error of the worst severity that is discovered is returned. In other words, if a warning (A) and two errors (B and C) are signaled, then the first error (B) is returned as the image exit status, which is placed in the DCL symbol $STATUS at image exit.

For additional information about image files in general, see the description of the Linker in the *VAX/VMS Linker Reference Manual*.

The ANALYZE/IMAGE command provides the following information:

- Image type—Identifies whether the image is executable or shareable.
- Image transfer addresses—Identify the addresses to which control is passed at image execution time.
- Image version—Identifies the revision level of the image.
- Patch information—Identifies whether the image has been patched (changed without having been recompiled or reassembled and relinked). If a patch is present, the actual patch code can be displayed.
- Location of the Debug Symbol Table (DST)—Identifies the location of the DST in the image file. DST information is present only in executable images that have been linked with the /DEBUG or /TRACEBACK command qualifiers.
- Location of the global symbol table (GST)—Identifies the location of the GST in the image file. GST information is present only in shareable image files.

- Image section descriptors—Identify portions of the image binary contents; these portions are grouped in clusters according to their attributes.

  An image section descriptor (ISD) contains information that the image activator needs when it initializes the address space for an image. For example, it tells whether or not the ISD is shareable, if it is readable or writable, if it is based or position-independent, and how much memory should be allocated.

- Fixup vectors—Contain information that the image activator needs in order to ensure the position-independence of shareable image references.

The ANALYZE/IMAGE command has both command qualifiers and positional qualifiers. By default, if you do not specify any positional qualifiers (for example, GST and HEADER), the entire image is analyzed. However, as soon as you specify any positional qualifier, you disable the others and then must explicitly request those you want in effect. The /HEADER qualifier is always in effect.

QUALIFIERS

/FIXUP_SECTION

Positional qualifier.

Specifies that the analysis should include all information in the fixup section of the image.

If you specify /FIXUP_SECTION after the ANALYZE/IMAGE command, the fixup section of each image file in the parameter list is analyzed.

If you specify /FIXUP_SECTION after a file specification, only the information in the fixup section of that image file is analyzed.

/GST

Positional qualifier.

Specifies that the analysis should include all global symbol table records. This qualifier is valid only for shareable images.

If you specify /GST after the ANALYZE/IMAGE command, the global symbol table records of each image file in the parameter list are analyzed.

If you specify /GST after a file specification, only the global symbol table records of that file are analyzed.

/HEADER

Positional qualifier.

Specifies that the analysis should include all header items and image section descriptions. The image header items are always analyzed.

/INTERACTIVE
/NOINTERACTIVE (default)

Controls whether the analysis is interactive. In interactive mode, as each item is analyzed, the results are displayed on the screen and you are asked whether you want to continue.

/OUTPUT=file-spec

Identifies the output file for storing the results of the image analysis. If you specify a file type and omit the file name, the default file name ANALYZE is used. The default file type is ANL. If you omit the qualifier, the results are output to the current SYS$OUTPUT device.

No wildcard characters are allowed in the file specification.

/PATCH_TEXT

Positional qualifier.

Specifies that the analysis should include all patch text records.

If you specify /PATCH_TEXT after the ANALYZE/IMAGE command, the patch text records of each image file in the parameter list are analyzed.

If you specify /PATCH_TEXT after a file specification, only the patch text records of that file are analyzed.

EXAMPLES

```
1   $ ANALYZE/IMAGE  LINEDT
```

The ANALYZE/IMAGE command produces a description and an error analysis of the image LINEDT.EXE. Output is to the current SYS$OUTPUT device. By default, the entire image is analyzed.

```
2   $ ANALYZE/IMAGE/OUTPUT=LIALPHEX -
    $_/FIXUP_SECTION/PATCH_TEXT  LINEDT, ALPHA
```

This command produces a description and an error analysis of the fixup sections and patch text records of LINEDT.EXE and ALPHA.EXE. Output is sent to the file LIALPHEX.ANL.

ANALYZE/MEDIA

Invokes the Bad Block Locator Utility (BAD), which analyzes block-addressable devices and records the location of blocks that cannot reliably store data. The /MEDIA qualifier is required. For a complete description of BAD, including information about the ANALYZE /MEDIA command and its qualifiers, see the *VAX/VMS Bad Block Locator Utility Reference Manual*.

FORMAT

ANALYZE/MEDIA *device*

ANALYZE/OBJECT

Analyzes the contents of an object file and checks for any obvious errors in the object file. The /OBJECT qualifier is required. (The ANALYZE/IMAGE command analyzes the contents of an image file.)

FORMAT

ANALYZE/OBJECT *file-spec[,...]*

restrictions

None.

PARAMETER

file-spec[,...]

Specifies the name of one or more object files or object module libraries you want analyzed. You must specify at least one file name. If you specify more than one file, separate the file specifications with either commas or plus signs. The default file type is OBJ.

Wildcard characters are allowed in the file specification.

DESCRIPTION

The ANALYZE/OBJECT command describes the contents of one or more object modules contained in one or more files. It also performs a partial error analysis. This analysis determines whether the records in an object module conform in content, format, and sequence to the specifications of the VAX Object Language.

ANALYZE/OBJECT is intended primarily for programmers of compilers, debuggers, or other software involving VAX/VMS object modules. It checks that the object language records generated by the object modules are acceptable to the VAX/VMS Linker, and it identifies certain errors that the file may have. It also provides a description of the records in the object file or object module library. For more information on the linker and on the VAX Object Language, refer to the *VAX/VMS Linker Reference Manual.*

The ANALYZE/OBJECT command analyzes the object modules in order, record by record, from the first to the last record in the object module. Fields in each record are analyzed in order from the first to the last field in the record. After the object module is analyzed, you should compare the content and format of each type of record to the required content and format of that record as described by the VAX Object Language. This comparison is particularly important if the analysis output contains a diagnostic message.

The linking of an object module differs from the analysis of an object module. Object language commands are not executed in an analysis, but they are executed in a linking operation. As a result, even if the analysis is error-free, the linking operation may not be. In particular, the analysis does not detect the following:

- That data arguments in TIR commands are in the correct format
- That "Store Data" TIR commands are storing within legal address limits

Therefore, you should still link an object module whose analysis is error-free as a final check before you assume it is correct.

If an error is found, however, the first error of the worst severity that is discovered is returned. In other words, if a warning (A) and two errors (B and C) are signaled, then the first error (B) is returned as the image exit status, which is placed in the DCL symbol $STATUS at image exit.

ANALYZE/OBJECT has both file and positional qualifiers. These positional qualifiers may be used as command qualifiers (in which case they affect every input file), or they may be used as file qualifiers (in which case they affect only the file directly preceding them). For example, specifying /TIR as a command qualifier causes TIR, HDR, EOM, and EOMW records (but not DBG, GSD, LNK, and TBT records) to be analyzed for every input file. On the other hand, specifying /TIR as a file qualifier limits this analysis to the file to which the qualifier was appended.

Typically, all records in an object module are analyzed. However, when any of the qualifiers /DBG, /EOM, /GSD, /LNK, /MHD, /TBT, or /TIR are specified, only the record types indicated by the qualifiers are analyzed; all other record types are ignored.

In other words, by default, if you do not specify any of the file qualifiers (such as /DBG, /EOM, and /GSD), all types of records are analyzed. However, when you specify one file qualifier, you disable the rest; you must explicitly request all the other file qualifiers you want.

Note: End-of-module (EOM) records and module header (MHD) records are always analyzed, no matter which qualifiers you specify.

QUALIFIERS

/DBG

Positional qualifier.

Specifies that the analysis should include all debugger information records.

If you specify /DBG after the ANALYZE/OBJECT command name, the debugger information records for each file in the parameter list are analyzed.

If you specify /DBG following a file specification, only the debugger information records for that file are analyzed.

/EOM

Positional qualifier.

Specifies that the analysis should include all EOM records. These records are always analyzed.

Note: End-of-module records may be EOM or EOMW records. See the *VAX/VMS Linker Reference Manual* for more information.

/GSD

Positional qualifier.

Specifies that the analysis should include all global symbol directory records.

If you specify /GSD after the ANALYZE/OBJECT command name, the global symbol directory records for each file in the parameter list are analyzed.

If you specify /GSD following a file specification, only global symbol directory records for that file are analyzed.

/INCLUDE[=(module[,...])]

Positional qualifier.

Indicates that the associated file is an object module library. If you specify only one module, you can omit the parentheses. If you omit the list or specify an asterisk, all modules are analyzed.

/INTERACTIVE
/NOINTERACTIVE (default)

Controls whether the analysis occurs interactively. In interactive mode, as each record is analyzed, the results are displayed on the screen, and you are asked whether you want to continue.

/LNK

Positional qualifier.

Specifies that the analysis should include all link option specification records.

If you specify /LNK after the ANALYZE/OBJECT command name, the link option specification records for each file in the parameter list are analyzed.

If you specify /LNK following a file specification, only the link option specification records for that file are analyzed.

/MHD

Positional qualifier.

Specifies that the analysis should include all module header records. Module header records are always analyzed.

If you specify /MHD following a file specification, only the module header records for that file are analyzed.

/OUTPUT[=file-spec]

Identifies the output file for storing the results of the analysis. If you specify a file type and omit the file name, the default file name ANALYZE is used. The default file type is ANL. If you omit the file-spec or omit the qualifier entirely, the results are output to the current SYS$OUTPUT device.

No wildcard characters are allowed in the file specification.

/TBT

Positional qualifier.

Specifies that the analysis should include all module traceback records.

If you specify /TBT after the ANALYZE/OBJECT command name, the traceback records for each file in the parameter list are analyzed.

If you specify /TBT following a file specification, only the traceback records for that file are analyzed.

/TIR

Positional qualifier.

Specifies that the analysis should include all text information and relocation records.

If you specify /TIR after the ANALYZE/OBJECT command name, the text information and relocation records for each file in the parameter list are analyzed.

If you specify /TIR following a file specification, only text information and relocation records for that file are analyzed.

EXAMPLES

1
```
$ ANALYZE/OBJECT/INTERACTIVE  LINEDT
```

This command produces a description and a partial error analysis of the object file LINEDT.OBJ. By default, all types of records are analyzed. Output is to the terminal, because the /INTERACTIVE qualifier has been used; as each item is analyzed, the results are displayed on the screen and you are asked whether or not you want to continue.

2
```
$ ANALYZE/OBJECT/OUTPUT=LIOBJ/DBG  LINEDT
```

This command analyzes only the debugger information records of the file LINEDT.OBJ. Output is to the file LIOBJ.ANL.

ANALYZE/PROCESS_DUMP

Invokes the VAX/VMS Debugger for analysis of a process dump file. The /PROCESS_DUMP qualifier is required. For a complete description of the debugger, including information about the DEBUG command and its qualifiers, refer to the *VAX/VMS Debugger Reference Manual*.

FORMAT

ANALYZE/PROCESS_DUMP *dump-file*

restrictions

Requires read (R) access to the dump file.

PARAMETERS

dump-file

Specifies the dump file to be analyzed by DEBUG.

DESCRIPTION

You can use ANALYZE/PROCESS_DUMP to examine the dump file of an image that failed during execution. To cause a dump to be taken for that process, you must use the /DUMP qualifier with the RUN command when invoking the image, or use the SET PROCESS/DUMP command before invoking the image.

QUALIFIER

/FULL

Causes all known information about the failing process to be displayed.

/IMAGE=image-name
/NOIMAGE

Designates the name of the image from which to take symbols to be used in analyzing this dump. If you use the /NOIMAGE qualifier, no symbols are taken from any image. The default is to take symbols from the image with the same name as the image that was running at the time the dump was taken.

/INTERACTIVE
/NOINTERACTIVE (default)

Causes the display of information to pause when your terminal screen is filled. Additional information is displayed when you press RETURN.

The default is /NOINTERACTIVE; that is, the display is continuous.

/MISCELLANEOUS

Causes the display of all the miscellaneous information in the dump.

/OUTPUT=file-spec

Causes the information output by the ANALYZE/PROCESS_DUMP command to be written to the specified file. No wildcard characters are permitted in the file specification.

/RELOCATION

Displays the addresses to which data structures saved in the dump are mapped in P0. (Examples of such data structures are the stacks.) The data structures in the dump must be mapped into P0 so that DEBUG can use those data structures in P1.

EXAMPLE

```
$ ANALYZE/PROCESS/FULL ZIPLIST
 R0 = 00018292  R1 = 8013DE20  R2 = 7FFE6A40   R3 = 7FFE6A98
 R4 = 8013DE20  R5 = 00000000  R6 = 7FFE7B9A   R7 = 0000F000
 R8 = 00000000  R9 = 00000000  R10 = 00000000  R11 = 00000000
 SP = 7FFAEF44  AP = 7FFAEF48  FP  = 7FFAEF84

 FREE_P0_VA  00001600    FREE_P1_VA  7FFAC600
 Active ASTs  00         Enabled ASTs 0F
 Current Privileges  FFFFFF80  1010C100
 Event Flags  00000000  E0000000
 Buffered I/O count/limit 6/6
 Direct I/O count/limit   6/6
 File count/limit         27/30
 Process count/limit      0/0
 Timer queue count/limit  10/10
 AST count/limit          6/6
 Enqueue count/limit      30/30
 Buffered I/O total 7      Direct I/O total 18

 Link Date  27-DEC-1983 15:02:00.48   Patch Date  17-NOV-1858 00:01:53.71
 ECO Level  0030008C  00540040  00000000  34303230

 Kernel stack 00000000 pages at 00000000 moved to 00000000
 Exec stack 00000000 pages at 00000000 moved to 00000000
 Vector page 00000001 page at 7FFEFE00 moved to 00001600
 PIO (RMS) area 00000005 pages at 7FFE1200 moved to 00001800
 Image activator context 00000001 page at 7FFE3400 moved to 00002200
 User writeable context 0000000A pages at 7FFE1C00 moved to 00002400
Creating a subprocess
        VAX DEBUG Version X4.0-2
DBG>
```

This example shows the output produced by the ANALYZE command when used with the /PROCESS and /FULL qualifiers, and the name of a file that contains the dump of a process that encountered a fatal error.

ANALYZE/RMS_FILE

Invokes the VAX/VMS Analyze/RMS_File Utility (ANALYZE/RMS_FILE) to inspect and analyze the internal structure of a VAX RMS file. The /RMS_FILE qualifier is required. For a complete description of the Analyze/RMS_File Utility, including more information about the VAX/VMS ANALYZE/RMS_FILE command and its qualifiers, see the *VAX/VMS Analyze/RMS_File Utility Reference Manual*.

FORMAT **ANALYZE/RMS_FILE** *file-spec[,...]*

ANALYZE/SYSTEM

Invokes the System Dump Analyzer (SDA) for analysis of the running system. The /SYSTEM qualifier is required. For a complete description of the System Dump Analyzer, including more information about the ANALYZE/SYSTEM command and its qualifiers, see the *VAX/VMS System Dump Analyzer Reference Manual*.

FORMAT ANALYZE/SYSTEM

APPEND

Adds the contents of one or more specified input files to the end of the specified output file.

FORMAT

APPEND *input-file-spec[,...] output-file-spec*

restrictions

None.

PARAMETERS

input-file-spec[,...]

Specifies the names of one or more input files to be appended.

If you specify more than one input file, separate the specifications with either commas or plus signs. (Commas and plus signs are equivalent.) All input files are appended, in the order specified, to the end of the output file.

You can use wildcard characters in the input file specifications.

output-file-spec

Specifies the name of the file to which the input files will be appended.

You must include at least one field in the output file specification. If you do not specify a device and/or directory, the APPEND command uses the current default device and directory. For other fields that you do not specify, the APPEND command uses the corresponding field of the input file specification.

If you use the asterisk wildcard character in any fields of the output file specification, the APPEND command uses the corresponding field of the input file specification. If you are appending more than one input file, APPEND uses the corresponding fields from the first input file.

DESCRIPTION

The APPEND command is similar in syntax and function to the COPY command. Normally, the APPEND command adds the contents of one or more files to the end of an existing file without incrementing the version number. The /NEW_VERSION qualifier causes the APPEND command to create a new output file if no file with that name exists.

QUALIFIERS

/ALLOCATION=n

Output-file-spec qualifier.

Forces the initial allocation of the output file to the number of 512-byte blocks specified by n.

This qualifier is valid in conjunction with the /NEW_VERSION qualifier. The allocation size is applied only if a new file is actually created. If a new file is created and you do not specify /ALLOCATION, the initial allocation of the output file is determined by the size of the input files.

/BACKUP

Selects files according to the dates of their most recent backup. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /BACKUP qualifier is incompatible with /CREATED, /EXPIRED, and /MODIFIED. The default is /CREATED.

/BEFORE[=time]

Selects only those files that are dated before the specified time. You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/BY_OWNER[=uic]

Selects one or more files only if their owner user identification code (UIC) matches the specified owner UIC.

Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual.*

If the /BY_OWNER qualifier is specified without a UIC, the UIC of the current process is assumed.

/CONFIRM
/NOCONFIRM (default)

Controls whether a request is issued before each individual APPEND operation to confirm that the operation should be performed on that file.

When the system issues the prompt, you can issue any of the following responses:

YES	NO	QUIT
TRUE	FALSE	CTRL/Z
1	0	ALL
	<RET>	

You can use any combination of upper- and lowercase letters for word responses. Word responses can be abbreviated to one or more letters (for example, T, TR, or TRU for TRUE). Affirmative answers are YES, TRUE, and 1. Negative answers are NO, FALSE, 0, and <RET> . QUIT or CTRL/Z indicates that you want to stop processing the command at that point. When you respond with 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, the prompt will be reissued.

/CONTIGUOUS
/NOCONTIGUOUS

Output-file-spec qualifier.

Indicates whether the output file is contiguous, that is, whether the file must occupy consecutive physical disk blocks.

By default, the APPEND command creates an output file in the same format as the corresponding input file. If an input file is contiguous, the APPEND command attempts to create a contiguous output file, but does not report an error if there is not enough space. If you append multiple input files of different formats, the output file may or may not be contiguous. You can use the /CONTIGUOUS qualifier to ensure that the output file is contiguous.

/CREATED (default)

Selects files based on their dates of creation. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /CREATED qualifier is incompatible with /BACKUP, /EXPIRED, and /MODIFIED.

/EXCLUDE=(file-spec[,...])

Any files that match the listed file specifications are excluded from the APPEND operation. If you specify only one file, you can omit the parentheses.

The file specification can contain a directory specification, but you cannot include the device in the file specifications you supply with the /EXCLUDE qualifier.

Wildcard characters are supported for file specifications. However, you cannot use relative version numbers to exclude a specific version.

/EXPIRED

Selects files according to the dates on which they will expire. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /EXPIRED qualifier is incompatible with /BACKUP, /CREATED, and /MODIFIED. The default is /CREATED.

/EXTENSION=n

Output-file-spec qualifier.

Specifies the number of blocks to be added to the output file each time the file is extended. When you specify /EXTENSION, the /NEW_VERSION qualifier is assumed and need not be typed on the command line.

The extension value is applied only if a new file is actually created.

/LOG
/NOLOG (default)

Controls whether the APPEND command displays the file specifications of each file appended.

When you use the /LOG qualifier, the APPEND command displays the file specifications of the input and output files as well as the number of blocks or the number of records appended after each append operation. At the end of command processing, the APPEND command displays the total number of new files created.

/MODIFIED

Selects files according to the dates on which they were last modified. This qualifier is only relevant when used with the /BEFORE or /SINCE qualifier. Use of the /MODIFIED qualifier is incompatible with /BACKUP, /CREATED, and /EXPIRED. The default is /CREATED.

/NEW_VERSION
/NONEW_VERSION (default)

Output-file-spec qualifier.

Controls whether the APPEND command creates a new file. By default, the specified output file must already exist. Use the /NEW_VERSION qualifier to have the APPEND command create a new output file if the specified output file does not already exist.

/PROTECTION=(code)

Output-file-spec qualifier.

Defines the protection to be applied to the output file.

Specify the protection code using the standard rules given in Section 7.1.2 of the *VAX/VMS DCL Concepts Manual*. Any protection attributes not specified are taken from the current protection of the output file if that file exists. If a new output file is created, the current default protection is used.

/READ_CHECK
/NOREAD_CHECK (default)

Input-file-spec qualifier.

Requests the APPEND command to read each record in the input files twice to verify that all records have been read correctly.

/SINCE[=time]

Selects only those files that are dated after the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/WRITE_CHECK
/NOWRITE_CHECK (default)

Output-file-spec qualifier.

Requests the APPEND command to read each record in the output file after the record is written to verify that it was appended successfully and that the output file can subsequently be read without error.

EXAMPLES

```
1    $ APPEND  TEST3.DAT TESTALL.DAT
```

The APPEND command appends the contents of the file TEST3.DAT from the default disk and directory to the file TESTALL.DAT, also located on the default disk and directory.

2
```
$ APPEND/NEW_VERSION/LOG *.TXT   MEM.SUM
%APPEND-I-CREATED, USE$:[MAL]MEM.SUM;1 created
%APPEND-S-COPIED, USE$:[MAL]A.TXT;2 copied to USE$:[MAL]MEM.SUM;1 (1 block)
%APPEND-S-APPENDED, USE$:[MAL]B.TXT;3 appended to USE$:[MAL]MEM.SUM;1 (3 records)
%APPEND-S-APPENDED, USE$:[MAL]G.TXT;7 appended to USE$:[MAL]MEM.SUM;1 (51 records)
%APPEND-S-NEWFILES, 1 file created
```

The APPEND command appends all files with file types of TXT to a file named MEM.SUM. The /LOG qualifier requests a display of the specifications of each input file appended. If the file MEM.SUM does not exist, the APPEND command creates it, as the output shows. The number of blocks or records shown in the output refers to the source file and not to the target file total.

3
```
$ APPEND/LOG A.DAT, B.MEM   C.*
%APPEND-S-APPENDED, USE$:[MAL]A.DAT;4 appended to USE$:[MAL]C.DAT;4 (2 records)
%APPEND-S-APPENDED, USE$:[MAL]B.MEM;5 appended to USE$:[MAL]C.DAT;4 (29 records)
```

The APPEND command appends the files A.DAT and B.MEM to the file C.DAT, which must already exist.

4
```
$ APPEND/LOG A.*   B.*
%APPEND-S-APPENDED, USE$:[MAL]A.DAT;5 appended to USE$:[MAL]B.DAT;1 (5 records)
%APPEND-S-APPENDED, USE$:[MAL]A.DOC;2 appended to USE$:[MAL]B.DAT;1 (1 record)
```

Both the input and output file specifications contain wildcard characters in the file type field. The APPEND command appends each file with a file name of A to an existing file with B as its file name. The file type of the first input file located determines the output file type.

5
```
$ APPEND BOSTON"JOHN_SMITH JKS"::DEMO01.DAT, DEMO2.DAT
$ _To:   DALLAS::DISK1:[MODEL.TEST]TEST.DAT
```

This APPEND command adds the contents of the files DEMO1.DAT and DEMO2.DAT at remote node BOSTON to the end of the file TEST.DAT at remote node DALLAS.

ASSIGN

Creates a logical name and assigns an equivalence string, or a list of strings, to the specified logical name.

FORMAT

ASSIGN *equivalence-name[,...] logical-name[:]*

restrictions

See qualifier descriptions.

PARAMETERS

equivalence-name[,...]

Defines the equivalence names to be associated with the logical name in the specified logical name table. The equivalence name string can contain from 1 to 255 characters. If the string contains any characters other than alphanumerics, the dollar sign, or the underscore character, enclose the string in quotation marks. If the equivalence name contains quotation marks, enclose the string in quotation marks and use two sets of quotation marks ("") in the places where you want a quotation mark to occur.

When you specify an equivalence name that will be used as a file specification, you must include the punctuation marks (colons, brackets, periods) that would be required if the equivalence name were used directly as a file specification. Therefore, if you specify a device name as an equivalence name, terminate the device name with a colon.

The ASSIGN command allows you to assign the same logical name to more than one equivalence name. When you specify more than one equivalence name for a logical name, you create a search list. See Section 4.8 of the *VAX/VMS DCL Concepts Manual* for more information on search lists.

logical-name

Specifies the logical name string. The logical name can have from 1 to 255 characters. (However, logical names entered in one of the logical name directory tables, LNM$PROCESS_DIRECTORY or LNM$SYSTEM_DIRECTORY, must be no more than 31 characters long, and must not contain characters other than alphanumerics, the dollar sign, and the underscore.)

If you terminate the logical name with a colon, the system removes the colon before placing the name in a logical name table. (This differs from the DEFINE command, which saves the colon.) By default, the logical name is placed in the process logical name table.

If the logical name contains any characters other than alphanumeric characters, the dollar sign, or the underscore, enclose the name in quotation marks. If the logical name contains quotation marks, enclose the name in quotation marks and use two sets of quotation marks in the places where you want one set of quotation marks to occur. Note that if you enclose a name in quotation marks, the case of alphabetic characters is preserved.

DESCRIPTION

The ASSIGN command creates an entry in a logical name table by defining a logical name to stand for one or more equivalence names. An equivalence name can be a device name, another logical name, a file specification, or any other string.

To specify the logical name table where you want to enter a logical name, use the /PROCESS, /JOB, /GROUP, /SYSTEM, or /TABLE qualifier. If you enter more than one of these qualifiers, only the last one entered is accepted. If you do not specify a table, the default is /TABLE=LNM$PROCESS (or /PROCESS).

To specify the access mode of the logical name you are creating, use the /USER_MODE, /SUPERVISOR_MODE, or /EXECUTIVE_MODE qualifiers. If you enter more than one of these qualifiers, only the last one entered is accepted. If you do not specify an access mode, then a supervisor mode name is created. You can create a logical name in the same mode as the table in which you are placing the name, or in an outer mode. (User mode is the outermost mode; executive mode is the innermost mode.)

You can enter more than one logical name with the same name in the same logical name table, as long as each name has a different access mode. (However, if an existing logical name within a table has the NO_ALIAS attribute, you cannot use the same name to create a logical name in an outer mode in this table.)

If you create a logical name with the same name, in the same table, and in the same mode as an existing name, the new logical name assignment replaces the existing assignment.

You can also use the DEFINE command to create logical names. To delete a logical name from a table, use the DEASSIGN command.

Note: Avoid assigning a logical name that matches the file name of an executable image in SYS$SYSTEM:. Such an assignment will prohibit you from invoking that image.

For additional information on how to create and use logical names, see Chapter 4 of the *VAX/VMS DCL Concepts Manual*.

QUALIFIERS

/EXECUTIVE_MODE

Requires the user privilege SYSNAM to create an executive mode logical name.

Creates an executive mode logical name in the specified table.

If you specify the /EXECUTIVE_MODE qualifier and you do not have SYSNAM, then the ASSIGN command ignores the qualifier and creates a supervisor mode logical name.

/GROUP

Requires the user privilege GRPNAM or SYSPRV to place a name in the group logical name table.

Places the logical name in the group logical name table. Other users who have the same group number in their user identification codes (UICs) can access the logical name. The /GROUP qualifier is synonymous with /TABLE=LNM$GROUP.

/JOB

Places the logical name in the job-wide logical name table. All processes within the same job tree as the process creating the logical name can access the logical name. The /JOB qualifier is synonymous with /TABLE=LNM$JOB.

/LOG (default)
/NOLOG

Controls whether a message is displayed when you define a logical name that supersedes an existing name.

/NAME_ATTRIBUTES[=(keyword[,...])]

Specifies attributes for a logical name. By default, no attributes are set. You can specify the following keywords for attributes:

CONFINE	The name is not to be copied into a subprocess by the SPAWN command. This keyword is meaningful only when you create logical names in a private table. If you specify this keyword for a logical name in a shareable table, it is ignored. The CONFINE attribute is also inherited from the logical name table where the name is entered; if the logical name table is "confined", then all names in the table are "confined".
NO_ALIAS	The logical name cannot be duplicated in this table with a less privileged (outer) access mode. If another logical name with the same name and an outer access mode already exists in this table, the name is deleted.

If you specify only one keyword, you can omit the parentheses. Only the attributes you specify are set.

/PROCESS (default)

Places the logical name in the process logical name table. The /PROCESS qualifier is synonymous with /TABLE=LNM$PROCESS.

/SUPERVISOR_MODE (default)

Creates a supervisor mode logical name in the specified table.

/SYSTEM

Requires the user privilege SYSNAM or SYSPRV to place a name in the group logical name table.

Places the logical name in the system logical name table. All system users can access the logical name. The /SYSTEM qualifier is synonymous with /TABLE=LNM$SYSTEM.

/TABLE=name

Requires WRITE access to the table if the table is shareable.

Specifies the name of the logical name table in which the logical name is to be entered. You can use the /TABLE qualifier to specify a user-defined logical name table (created with the CREATE/NAME_TABLE command), to specify the process, job, group, or system logical name tables, or to specify the process or system logical name directory tables.

If you specify the table name using a logical name that has more than one translation, the logical name is placed in the first table found. For example, if you specify ASSIGN/TABLE=LNM$FILE_DEV and LNM$FILE_DEV is equated to LNM$PROCESS, LNM$JOB, LNM$GROUP, and LNM$SYSTEM, then the logical name is placed in LNM$PROCESS.

If you do not explicitly specify the /TABLE qualifier, the default is /TABLE=LNM$PROCESS (or /PROCESS).

/TRANSLATION_ATTRIBUTES[=(keyword[,...])]

Equivalence-name qualifier.

Specifies one or more attributes that modify an equivalence string of the logical name. You can specify the following keywords for translation attributes:

CONCEALED	Indicates that the equivalence string is a concealed device name. When a concealed device name is defined, the system displays the logical name, rather than the equivalence string, in messages that refer to the device. If you specified the CONCEALED attribute, then the equivalence string must be a physical device name.
TERMINAL	Indicates that the equivalence string should not be translated iteratively; logical name translation should terminate with the current equivalence string.

If you specify only one keyword, you can omit the parentheses. Only the attributes you specify are set.

Note that different equivalence strings of the same logical name can have different translation attributes specified.

/USER_MODE

Creates a user mode logical name in the specified table.

User mode logical names created within the process logical name table are used for the execution of a single image; for example, you can create a user mode logical name to allow an image executing in a command procedure to redefine SYS$INPUT. User mode entries are deleted from the process logical name table when any image executing in the process exits (that is, after any DCL command or user program that executes an image completes execution).

EXAMPLES

1

```
$ ASSIGN XXX1:[CHARLES]  CHARLIE
$ PRINT  CHARLIE:TEST.DAT
Job 274 entered on queue SYS$PRINT
```

The ASSIGN command associates the logical name CHARLIE with the directory name [CHARLES] on the disk XXX1. Subsequent references to the logical name CHARLIE result in the correspondence between the logical name CHARLIE and the disk and directory specified. The PRINT command queues a copy of the file XXX1:[CHARLES]TEST.DAT to the system printer.

2
```
$ ASSIGN YYY2:  TEMP:
$ SHOW LOGICAL TEMP
  "TEMP" = "YYY2:" (LNM$PROCESS_TABLE)
$ DEASSIGN TEMP
```

The ASSIGN command equates the logical name TEMP to the device YYY2. TEMP is created in supervisor mode and placed in the process logical name table. The SHOW LOGICAL command verifies that the logical name assignment was made. Note that the logical name TEMP was terminated with a colon in the ASSIGN command, but that the command interpreter deleted the colon before placing the name in the logical name table. Thus, you can specify TEMP without a colon in the subsequent DEASSIGN command. You should omit the colon in the SHOW LOGICAL command (for example, SHOW LOGICAL TEMP).

3
```
$ MOUNT TTT1: MASTER TAPE
$ ASSIGN TAPE:NAMES.DAT PAYROLL
$ RUN PAYROLL
    .
    .
    .
```

The MOUNT command establishes the logical name TAPE for the device TTT1, which has the volume labelled MASTER mounted on it. The ASSIGN command equates the logical name PAYROLL with the file named NAMES.DAT on the logical device TAPE. Thus, an OPEN request in a program referring to the logical name PAYROLL results in the correspondence between the logical name PAYROLL and the file NAMES.DAT on the tape whose volume label is MASTER.

4
```
$ CREATE/NAME_TABLE TABLE1
$ ASSIGN/TABLE=LNM$PROCESS_DIRECTORY TABLE1,-
_$ LNM$PROCESS,LNM$JOB,LNM$GROUP,LNM$SYSTEM LNM$FILE_DEV
$ ASSIGN/TABLE=TABLE1 -
_$ /TRANSLATION_ATTRIBUTES=CONCEALED  DBA1:  WORK_DISK
```

The CREATE/NAME_TABLE command creates the process private logical name table TABLE1.

The first ASSIGN command ensures that TABLE1 is searched first in any logical name translation of a file specification or device name (because TABLE1 is the first item in the equivalence string for the logical name LNM$FILE_DEV, which determines the default search sequence of logical name tables whenever a device or file specification is translated).

The second ASSIGN command assigns the logical name WORK_DISK to the physical device DBA1, and places the name in TABLE1. The logical name has the concealed attribute. Therefore, the logical name WORK_DISK will be displayed in system messages.

5
```
$ ASSIGN/TABLE=LNM$PROCESS/TABLE=LNM$GROUP  DBA0:  SYSFILES
$ SHOW LOGICAL  SYSFILES
  "SYSFILES" = "DBA0:" (LNM$GROUP_000240)
```

This ASSIGN command contains conflicting qualifiers. When you specify conflicting qualifiers, the ASSIGN command uses the last qualifier specified. The response from the SHOW LOGICAL command indicates that the name was placed in the group logical name table.

```
6   $ ASSIGN/TABLE=LNM$GROUP 'F$TRNLNM("SYS$COMMAND")' TERMINAL
    %DCL-I-SUPERSEDE, previous value of TERMINAL has been superseded
```

The ASSIGN command uses the lexical function F$TRNLNM to translate the logical name SYS$COMMAND and use the result as the equivalence name for the logical name TERMINAL. The message from the ASSIGN command indicates that an entry for the logical name TERMINAL already existed in the group logical name table, and that the new entry has replaced the previous one.

If this command is used in a LOGIN.COM file, the entry for TERMINAL will be redefined at the beginning of each terminal session. The current process and any subprocesses it creates can execute images that use the logical name TERMINAL to write messages to the current terminal device.

```
7   $ ASSIGN DALLAS::DMA1:  DATA
```

The ASSIGN command associates the logical name DATA with the device specification DMA1 on remote node DALLAS. Subsequent references to the logical name DATA result in references to the disk on the remote node.

```
8   $ CREATE AVERAGE.COM
    $ ASSIGN/USER_MODE SYS$COMMAND:  SYS$INPUT
    $ EDIT/EDT AVERAGE.FOR
    $ FORTRAN AVERAGE
    $ LINK AVERAGE
    $ RUN AVERAGE
    87
    80
    90
    9999
    $ EXIT
    CTRL/Z
    $ @AVERAGE.COM
```

This example starts with the CREATE command to create the command procedure AVERAGE.COM. Then the command procedure is executed.

First the command procedure uses the ASSIGN command with the /USER_MODE qualifier to temporarily change the value of SYS$INPUT. When the EDT editor is invoked, it will accept input from the terminal. This allows you to create or modify AVERAGE.FOR interactively.

When you exit from EDT, SYS$INPUT is reassigned to its original value (the input stream provided by the command procedure). Thus, when the program AVERAGE.FOR is ready to accept input, it will look for that input in the command procedure.

```
9   $ ASSIGN [MARY.CH1], [MARY.CH2] BOOK
    $ TYPE BOOK:EXAMPLES.RNO
```

In this example, the ASSIGN command is used to create a logical name with two equivalence strings. When the TYPE command is issued, the system searches for the file EXAMPLES.RNO in the directories [MARY.CH1] and [MARY.CH2]. If files names EXAMPLES.RNO exist in both directories, then both files are typed.

ASSIGN/MERGE

Removes all jobs from one queue and places them in another existing queue. The /MERGE qualifier is required.

FORMAT

ASSIGN/MERGE *target-queue[:] source-queue[:]*

restrictions

Requires operator (OPER) privilege or execute (E) access to both queues.

PARAMETERS

target-queue[:]

Specifies the name of the queue to which the jobs will be directed.

source-queue[:]

Specifies the name of the queue whose jobs are to be reassigned to a new queue.

DESCRIPTION

The ASSIGN/MERGE command removes jobs that are pending in one queue and places them in another queue. This command does not affect any jobs that are executing in either the target queue or the source queue. Jobs currently running in the source queue complete in that queue. This command is generally used with printer queues, although it can be used with batch queues.

The ASSIGN/MERGE command is particularly useful when a line printer malfunctions. By issuing an ASSIGN/MERGE command, you can reroute existing jobs to a different print device. To perform the merge operation without losing or disrupting any jobs, you should stop the source queue with the STOP/QUEUE/NEXT command. You can then issue the STOP /QUEUE/REQUEUE command to ensure that the current job on the source queue is requeued for processing on the target queue. (If the STOP/QUEUE /REQUEUE command fails to requeue the job, use the STOP/QUEUE /RESET command to regain control of the queue.) Once the STOP commands have been issued, enter the ASSIGN/MERGE command.

EXAMPLE

```
$ STOP/QUEUE/NEXT LPB0
$ STOP/QUEUE/REQUEUE=LPA0 LPB0
$ ASSIGN/MERGE LPA0 LPB0
```

The STOP/QUEUE/NEXT command prevents another job from executing on queue LPB0. The STOP/QUEUE/REQUEUE command requeues the current job that was running on LPB0 to the target queue LPA0. The ASSIGN /MERGE command removes the remaining jobs from the LPB0 printer queue and places them in the LPA0 printer queue.

ASSIGN/QUEUE

Assigns, or redirects, a logical queue to a single execution queue. ASSIGN/QUEUE can be used only with printer or terminal queues. The /QUEUE qualifier is required.

FORMAT

ASSIGN/QUEUE *queue-name[:]*
logical-queue-name[:]

restrictions

- Requires operator (OPER) privilege or execute (E) access to both queues.
- Cannot be used with batch queues.

PARAMETERS

queue-name[:]

Specifies the name of an execution queue to be associated with the specified logical queue name. The queue cannot be a logical queue, a generic queue, or a batch queue.

logical-queue-name[:]

Specifies the name of the logical queue that is to be associated with the specified execution queue.

DESCRIPTION

The ASSIGN/QUEUE command sets up a one-to-one correspondence between a logical queue and an execution queue. Jobs submitted to the logical queue are always queued to the specified execution queue for eventual printing.

When you issue the ASSIGN/QUEUE command, the logical queue cannot be running.

Once you have initialized a logical queue, you use the ASSIGN/QUEUE command to associate the logical queue with an existing execution queue. You must follow these steps to set up a logical queue and enable jobs in that queue to be executed:

1. Initialize the logical queue with the INITIALIZE/QUEUE command. (Do not use the /START qualifier.)
2. Assign the logical queue name to an existing execution queue.
3. Start the logical queue with the START/QUEUE command.

After you have issued the START/QUEUE command for the logical queue, jobs can be sent to the logical queue for processing.

EXAMPLES

1
```
$ INITIALIZE/QUEUE/DEFAULT=FLAG=ONE/START LPA0
$ INITIALIZE/QUEUE TEST_QUEUE
$ ASSIGN/QUEUE LPA0 TEST_QUEUE
$ START/QUEUE TEST_QUEUE
```

This example first initializes and starts the printer queue LPA0. The LPA0 queue is set to have a flag page precede each job. The second INITIALIZE /QUEUE command creates the logical queue TEST_QUEUE. The ASSIGN /QUEUE command assigns the logical queue TEST_QUEUE to the printer queue LPA0. The START/QUEUE command starts the logical queue.

2
```
$ INITIALIZE/QUEUE/START LPB0
```

The ASSIGN/QUEUE command is not needed here because a logical queue is not being initialized. A printer queue is being initialized; LPB0 is the name of a line printer. After you issue the INITIALIZE/QUEUE/START command, jobs can be queued to LPB0 for printing.

ATTACH

Enables you to switch control from your current process to another process in your job.

FORMAT

ATTACH *[process-name]*

restrictions

The SPAWN and ATTACH commands cannot be used if your terminal has an associated mailbox.

PARAMETER

process-name

Specifies the name of the process to which the connection is to be made. Process names can contain from 1 to 15 alphanumeric characters. If a connection to the specified process cannot be made, an error message is displayed. You cannot connect to the process if:

- The process is your current process
- The process is not part of your current job
- The process does not exist

If you specify the /IDENTIFICATION qualifier, you cannot use the process name parameter. If you omit the /IDENTIFICATION qualifier, you must specify a process name.

DESCRIPTION

The ATTACH command allows you to connect your input stream to another process. You can use ATTACH to change control from one subprocess to another subprocess or to the parent process.

When you issue the ATTACH command, the parent or "source" process is put into a hibernation state, and your input stream is connected to the specified destination process. You can use ATTACH to connect to a subprocess that is part of a current job left hibernating as a result of the SPAWN/WAIT command or another ATTACH command as long as the connection is valid. (No connection can be made to the current process, to a process that is not part of the current job, or to a process that does not exist.)

You can also use ATTACH in conjunction with the SPAWN/WAIT command to return to a parent process without terminating the created subprocess. See the description of the SPAWN command for more details.

QUALIFIER

/IDENTIFICATION=pid

Specifies the process identification (PID) of the process to which terminal control will be switched. When you specify a PID, you can omit the leading zeros.

If the process is not part of the current job or does not exist, an error message is displayed.

When you use the /IDENTIFICATION qualifier, do not specify the process name parameter. If you omit the /IDENTIFICATION qualifier, you must specify a process name.

EXAMPLES

1
```
$ ATTACH JONES_2
```

The ATTACH command switches control to the process JONES_2.

2
```
$ ATTACH/IDENTIFICATION=30019
```

The ATTACH command switches control from the current process to a process having the PID 30019. Notice that since the /IDENTIFICATION qualifier is specified, the process-name parameter is omitted.

BACKUP

Invokes the Backup Utility (BACKUP) to perform one of the following BACKUP operations:

- Make copies of disk files.
- Save disk files as data in a file created by BACKUP on disk or magnetic tape. (Files created by BACKUP are called save sets.)
- Restore disk files from a BACKUP save set.
- Compare disk files or files in a BACKUP save set with other disk files.
- List information about files in a BACKUP save set to an output device or file.

Note that stand-alone BACKUP cannot be invoked this way, but must be bootstrapped in order to run.

For a complete description of the Backup Utility, including information about the BACKUP command and its qualifiers, as well as using stand-alone BACKUP, see the *VAX/VMS Backup Utility Reference Manual*.

FORMAT

BACKUP *input-specifier output-specifier*

CALL

Transfers control to a labeled subroutine in a command procedure and creates a new procedure level.

FORMAT

CALL *label [p1[p2[... p8]]]*

restrictions

None.

PARAMETERS

label

Specifies a 1- through 255-alphanumeric character label appearing as the first item on a command line. A label may not contain embedded blanks. When the CALL command is executed, control passes to the command following the specified label.

The label can precede or follow the CALL statement in the current command procedure. When you use a label in a command procedure, it must be terminated with a colon.

All labels are procedure level dependent except for those labels that define subroutine entry points. These labels are local to the current command procedure file level. Subroutine entry points labels must be unique.

p1 [p2 [... p8]]

Specifies from one to eight optional parameters to pass to the command procedure. The parameters assign character string values to the symbols named P1, P2, and so on in the order of entry, to a maximum of eight.

Separate each parameter with one or more blanks. Use quotation marks (" ") to specify a null parameter. You can specify a parameter with a character string value containing alphanumeric or special characters, with the following restrictions:

- The command interpreter converts alphabetic characters to uppercase and uses blanks to delimit each parameter. To pass a parameter that contains embedded blanks or literal lowercase letters, place the parameter in quotation marks.
- If the first parameter begins with a slash character (/), you must enclose the parameter in quotation marks.
- To pass a parameter that contains literal quotation marks and spaces, enclose the entire string in quotation marks and use a double set of quotation marks within the string. For example:

```
$ CALL SUB1 "Never say ""quit"""
```

When control transfers to SUB1, the parameter P1 is equated to the string:

```
Never say "quit"
```

If a string contains quotation marks and does not contain spaces, the quotation marks are preserved in the string and the letters within the quotation marks remain in lowercase. For example:

```
$ CALL SUB2 abc"def"ghi
```

When control transfers to SUB2, the parameter P1 is equated to the string:

```
ABC"def"GHI
```

To use a symbol as a parameter, enclose the symbol in apostrophes to force symbol substitution. For example:

```
$ NAME = "JOHNSON"
$ CALL INFO 'NAME'
```

The apostrophes cause the value "JOHNSON" to be substituted for the symbol NAME. Therefore, the parameter "JOHNSON" is passed as P1 to the subroutine INFO.

DESCRIPTION

The CALL command is similar to the @ (Execute Procedure) command in that it creates a new procedure level. The advantage of the CALL command is that it does not require files to be opened and closed to process the procedure. Using CALL also makes managing a set of procedures easier since they can all exist in one file rather than in several files.

When you use the CALL command to transfer control to a subroutine, a new procedure level is created and the symbols P1 through P8 are assigned the values of the supplied arguments. Execution then proceeds until an EXIT command is encountered. At this point, control is transferred to the command line following the CALL command.

Procedures can be nested to a maximum of 32 levels. This includes any combination of command procedure and subroutine calls. Local symbols and labels defined within a nested subroutine structure are treated the same way as if the routines had been invoked with the @ command. Labels are only valid for the subroutine level in which they are defined. Local symbols defined in an outer subroutine level are available to any subroutine levels at an inner nesting level.

The SUBROUTINE and ENDSUBROUTINE commands define the beginning and end of a subroutine. The label defining the entry point to the subroutine must appear either before the SUBROUTINE command or on the same command line.

A subroutine can have only one entry point. The subroutine must begin with the SUBROUTINE command as the first executable statement. The ENDSUBROUTINE command functions as an EXIT command if an EXIT command is not specified in the procedure.

The SUBROUTINE verb performs two distinctly different functions depending on the context in which it is executed. If executed as the result of a CALL command, it initiates a new procedure level, defines the P1-P8 parameters as specified in the CALL statement, and begins execution of the subroutine. If the SUBROUTINE verb is encountered in the execution flow of the procedure without having been invoked by a CALL command, all the commands following the SUBROUTINE command are skipped until the corresponding ENDSUBROUTINE command is encountered. Although commands are skipped, all subroutine entry point labels will be defined in the symbol table.

Note: The SUBROUTINE and ENDSUBROUTINE commands cannot be abbreviated to less than four characters.

QUALIFIER

/OUTPUT=file-spec

Requests that all output directed to the logical device SYS$OUTPUT be written to the file or device specified. System responses and error messages are written to SYS$COMMAND as well as to the specified file. If you specify /OUTPUT, the qualifier must immediately follow the CALL command.

The default output file type is LIS.

No wildcard characters are allowed in the output file specification.

You can also redefine SYS$OUTPUT to redirect the output from a command procedure. If you place the command:

```
$ DEFINE SYS$OUTPUT file-spec
```

as the first line in a command procedure, output will be directed to the file you specify. When the procedure exits, SYS$OUTPUT will be restored to its original equivalence string. This produces the same result as using the /OUTPUT qualifier when you execute the command procedure.

EXAMPLE

```
$
$! CALL.COM
$
$! Define subroutine SUB1
$!
$ SUB1: SUBROUTINE
    .
    .
    .
$ CALL SUB2 !Invoke SUB2 from within SUB1
    .
    .
    .
$ @FILE !Invoke another procedure command file
    .
    .
    .
$ EXIT
$ ENDSUBROUTINE !End of SUB1 definition
$!
$! Define subroutine SUB2
$!
$ SUB2: SUBROUTINE
    .
    .
    .
$ EXIT
$ ENDSUBROUTINE !End of SUB2 definition
$!
$! Start of main routine. At this point, both SUB1 and SUB2
$! have been defined but none of the previous commands have
$! been executed.
$!
$ START:
$ CALL/OUTPUT=NAMES.LOG SUB1 "THIS IS P1"
    .
    .
    .
$ CALL SUB2 "THIS IS P1" "THIS IS P2"
    .
    .
    .
```

```
$ EXIT !Exit this command procedure file
```

This command procedure shows how to use CALL to transfer control to labeled subroutines. The example also shows that you can call a subroutine or another command file from within a subroutine. The CALL command invokes the subroutine SUB1, directing output to the file NAMES.LOG and allowing other users write access to the file. The subroutine SUB2 is called from within SUB1. The procedure executes SUB2 and then uses the @ (Execute Procedure) command to invoke the command procedure FILE.COM. When all the commands in SUB1 have executed, the CALL command in the main procedure calls SUB2 a second time. The procedure continues until SUB2 has executed.

CANCEL

Cancels scheduled wake-up requests for a specified process, including both wakeups scheduled with the RUN command and with the Schedule Wakeup ($SCHDWK) system service.

FORMAT

CANCEL *[process-name]*

restrictions

The user privilege GROUP is required to cancel scheduled wakeups for any processes that you do not own in the same group; the user privilege WORLD is required to cancel scheduled wakeups for any process in the system.

PARAMETER

process-name

Specifies the name of the process for which wake-up requests are to be canceled. Process names are strings consisting of 1–15 alphanumeric characters. The specified process must have the same group number in its user identification code (UIC) as the current process.

If you specify both the /IDENTIFICATION qualifier and the process name, the process name is ignored. If you specify neither the process-name parameter nor the /IDENTIFICATION qualifier, the CANCEL command cancels scheduled wake-up requests for the current (that is, the issuing) process.

DESCRIPTION

The CANCEL command is used to cancel scheduled wake-up requests for the specified process.

The CANCEL command does not delete the specified process. If the process is executing an image when the CANCEL command is issued for it, the process hibernates instead of exiting after the image completes execution.

To delete a process which is hibernating and for which wake-up requests have been canceled, use the STOP command. You can determine whether a subprocess has been deleted by issuing the SHOW PROCESS command with the /SUBPROCESSES qualifier.

QUALIFIER

/IDENTIFICATION=pid

Specifies the process identification code (PID) that the system assigned to the process when the process was created. When you specify the process identification, you can omit leading zeros.

EXAMPLES

1

```
$ RUN/SCHEDULE=14:00 STATUS
%RUN-S-PROC_ID, identification of created process is 0013012A
    .
    .
    .
$ CANCEL/IDENTIFICATION=13012A
```

The RUN command creates a process to execute the image STATUS. The process hibernates and is scheduled to be awakened at 14:00. Before the process is awakened, the CANCEL command cancels the wake-up request.

2

```
$ RUN/PROCESS_NAME=LIBRA/INTERVAL=1:00    LIBRA
%RUN-S-PROC_ID, identification of created process is 00130027
    .
    .
    .
$ CANCEL LIBRA
$ STOP LIBRA
```

The RUN command creates a subprocess named LIBRA to execute the image LIBRA.EXE at hourly intervals.

Subsequently, the CANCEL command cancels the wake-up request. The process continues to exist, but in a state of hibernation, until the STOP command deletes it.

CLOSE

Closes a file that was opened for input or output with the OPEN command, and deassigns the logical name specified when the file was opened.

FORMAT

CLOSE *logical-name[:]*

restrictions

None.

PARAMETER

logical-name[:]

Specifies the logical name assigned to the file when it was opened with the OPEN command.

DESCRIPTION

Files that are opened for reading or writing at the command level remain open until explicitly closed with the CLOSE command, or until the process is deleted at logout. 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.

QUALIFIER

/ERROR=label

Specifies a label in the command procedure to receive control if an error occurs when you are closing a file that was opened with the OPEN command. The error routine specified for this qualifier takes precedence over any action statement indicated in an ON command. If /ERROR is not specified and an error occurs during the closing of the file, 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.

/LOG (default)
/NOLOG

Controls whether a warning message is generated when you attempt to close a file that was not opened by DCL. If you specify the /ERROR qualifier, the /LOG qualifier has no effect. If the file has not been opened by DCL, the error branch is taken and no message is displayed.

EXAMPLES

1

```
$ OPEN/READ INPUT_FILE   TEST.DAT
$ READ_LOOP:
$ READ/END_OF_FILE=NO_MORE  INPUT_FILE  DATA_LINE
    .
    .
    .
$ GOTO READ_LOOP
$ NO_MORE:
$ CLOSE INPUT_FILE
```

The OPEN command opens the file TEST.DAT and assigns it the logical name of INPUT_FILE. The /END_OF_FILE qualifier on the READ command requests that, when the end-of-file is reached, the command interpreter should transfer control to the line at the label NO_MORE. The CLOSE command closes the input file.

2

```
$ @READFILE
^Y
$ STOP
$ SHOW LOGICAL/PROCESS
    .
    .
    .
  "INFILE" = "_DB1"
  "OUTFILE" = "_DB1"
$ CLOSE INFILE
$ CLOSE OUTFILE
```

CTRL/Y interrupts the execution of the command procedure READFILE.COM. Then, the STOP command stops the procedure. The SHOW LOGICAL/PROCESS command displays the names that currently exist in the process logical name table. Among the names listed are the logical names INFILE and OUTFILE, assigned by OPEN commands in the procedure READFILE.COM.

The CLOSE commands close these files and deassign the logical names.

CONNECT

Connects your physical terminal to a virtual terminal that is connected to another process.

FORMAT

CONNECT *virtual-terminal-name*

restrictions

- You can connect only to a virtual terminal that is connected to a process with your user identification code (UIC).
- No other physical terminal may be connected to the virtual terminal.

PARAMETER

virtual-terminal-name

The name of the virtual terminal to which you are connecting. A virtual terminal name always begins with VTA. To determine the name of the virtual terminal that is connected to a process, enter the SHOW USERS command.

DESCRIPTION

The CONNECT command connects you to a separate process, as opposed to the SPAWN and ATTACH commands, which create and attach subprocesses.

The CONNECT command is useful when you are logged into the system using telecommunications lines. If there is noise over the line and you lose the carrier signal, your process does not terminate. After you log in again, you can reconnect to the original process and log your second process out.

In order to use the CONNECT command, the virtual terminal feature must be enabled for your system with the System Generation Utility (SYSGEN). If virtual terminals are allowed on your system, then the SET TERMINAL /PERMANENT command is used to enable the virtual terminal characteristic for a particular physical terminal. When this characteristic is enabled, a virtual terminal will be created when a user logs in on the physical terminal. The physical terminal is connected to the virtual terminal which is, in turn, connected to the process.

When the connection between the physical terminal and the virtual terminal is broken, the process remains connected to the virtual terminal. If the process is executing an image, it continues until it needs terminal input or attempts to write to the terminal. At that point, it waits.

You can connect to a virtual terminal even if you are not currently using a virtual terminal. However, you must use the CONNECT command with the /LOGOUT qualifier to log your current process out. If you connect to a virtual terminal from another virtual terminal, you can save your current process by using the /NOLOGOUT qualifier.

QUALIFIERS

/CONTINUE
/NOCONTINUE (default)

Controls whether the CONTINUE command is executed in the current process just before connecting to another process. This allows an interrupted image to continue processing after you connect to another process.

The /CONTINUE qualifier is incompatible with the /LOGOUT qualifier.

/LOGOUT (default)
/NOLOGOUT

Controls whether your current process is logged out when you connect to another process using a virtual terminal.

When you issue the CONNECT command from a process that is not connected to a virtual terminal, you must explicitly specify the /LOGOUT qualifier. Otherwise, DCL will issue an error message.

The /LOGOUT qualifier is incompatible with the /CONTINUE qualifier.

EXAMPLES

1

```
$ RUN AVERAGE
^Y
$ CONNECT/CONTINUE VTA72
```

In this example, the RUN command is used to execute the image AVERAGE.EXE. This command is issued from a terminal that is connected to a virtual terminal. Next, CTRL/Y is used to interrupt the image. After you interrupt the image, you issue the CONNECT command with the /CONTINUE qualifier. This issues the CONTINUE command so the image continues to run, and then connects you to another virtual terminal. You can connect back to the process later.

2

```
$ SHOW USERS
          VAX/VMS Interactive Users
          23-DEC-1985 15:25:30.75
    Total number of interactive users = 5

 Username     Process Name      PID     Terminal
 REICH        REICH           2040055C  VTA267:        TXC13:
 GLASS        Phil            20400560  VTA270:        LTA102:
 ADAMS        ADAMS           20400551  VTA261:        TTC7:
 JANZEN       JANZEN          2040056D  VTA272:        Disconnected
 JANZEN       _VTA273:        2040056E  VTA273:        TTB5:
$ CONNECT VTA273
  JANZEN        logged out at 23-DEC-1985 15:26:56.53
$
```

This example shows how to reconnect to your original process after you have lost the carrier signal. First, you must log in again and create a new process. After you log in, issue the SHOW USERS command to determine the virtual terminal name for your initial process. Then issue the CONNECT command to connect to the virtual terminal associated with your original process. The process from which you issue the CONNECT command is logged out because no qualifiers are specified.

When you connect back to your original process, you continue running the image you were running when you lost the carrier signal. In this example, the user JANZEN was at DCL level when the connection was broken.

CONTINUE

Resumes execution of a DCL command, a program, or a command procedure that was interrupted by pressing CTRL/Y or CTRL/C. The CONTINUE command serves as the target command of an IF or ON command in a command procedure. CONTINUE is also a target command when it follows a label that is the target of a GOTO command. In addition, you can use the CONTINUE command to resume processing of a program that has executed either a VAX FORTRAN PAUSE statement or a VAX COBOL-74 STOP literal statement.

You can abbreviate the CONTINUE command to a single letter, C.

FORMAT

CONTINUE

restrictions *None.*

PARAMETERS

None.

DESCRIPTION

The CONTINUE command enables you to resume processing an image after certain kinds of interruptions. If the interruption involved execution of another image, you cannot use the CONTINUE command on the original image. Similarly, if you have invoked a command procedure after interrupting the original image, that image cannot resume processing. However, you can use CONTINUE after commands that do not execute separate images; see Table DCL–2-1 in the *VAX/VMS DCL Concepts Manual* for a list of these commands.

QUALIFIERS

None.

EXAMPLES

```
1   $ RUN  MYPROGRAM_A
    ^Y
    $ SHOW TIME
      15-APR-1984 13:40:12
    $ CONTINUE
```

The RUN command executes the program MYPROGRAM_A. While the program is running, pressing CTRL/Y interrupts the image. The SHOW TIME command requests a display of the current date and time. The CONTINUE command resumes the image.

```
2    $ ON SEVERE_ERROR THEN CONTINUE
```

This statement in a command procedure requests the command interpreter to continue executing the procedure if any warning, error, or severe error status value is returned from the execution of a command or program. This ON statement overrides the default action, which is to exit from a procedure following errors or severe errors.

CONVERT

Invokes the VAX/VMS Convert Utility (CONVERT) to copy records from one file to another, changing the organization and format of the input file to those of the output file. For a complete description of the VAX/VMS Convert Utility, including more information about the CONVERT command and its qualifiers, see the *VAX/VMS Convert and Convert/Reclaim Utility Reference Manual*.

FORMAT

CONVERT *input-file-spec[,...] output-file-spec*

CONVERT/RECLAIM

Invokes the VAX/VMS Convert/Reclaim Utility (CONVERT /RECLAIM) to make empty buckets in Prolog 3 indexed files available so that new records can be written in them. If all the records in a bucket have been deleted, that bucket is locked until CONVERT/RECLAIM makes it available. Unlike CONVERT, CONVERT/RECLAIM maintains record file addresses (RFAs). The /RECLAIM qualifier is required. For a complete description of the Convert/Reclaim Utility, including more information about the CONVERT/RECLAIM command and its qualifier, see the VAX/VMS Convert Utility in the *VAX/VMS Convert and Convert/Reclaim Utility Reference Manual*.

FORMAT **CONVERT/RECLAIM** *file-spec*

COPY

Creates a new file from one or more existing files. The COPY command can:

- Copy an input file to an output file
- Concatenate two or more input files into a single output file
- Copy a group of input files to a group of output files

FORMAT

COPY *input-file-spec[,...] output-file-spec*

restrictions

None.

PARAMETERS

input-file-spec[,...]

Specifies the names of one or more input files to be copied. If you specify more than one input file, you can separate the names with either commas or plus signs.

You can use wildcard characters in the file specifications.

output-file-spec

Specifies the name of the output file into which the input files will be copied.

You must specify at least one field in the output file specification. If the device or directory is not specified, your current default device and directory are used. The COPY command replaces any other missing fields (file name, file type, version number) with the corresponding field of the input file specification. If you specify more than one input file, COPY generally uses the fields from the first input file to determine any missing fields in the output file.

The asterisk wildcard character can be used in place of the file name, file type, and/or version number. The COPY command uses the corresponding field in the related input file to name the output file. The wildcard character can also be used in the output file specification to have COPY create more than one output file. For example:

```
$ COPY A.A;1, B.B;1 *.C
```

This COPY command creates the files A.C;1 and B.C;1 in the current default directory.

Full use of wildcard characters is allowed for directories in the output file specification.

DESCRIPTION

The COPY command, by default, creates a single output file. When more than one input file is specified, the first input file is copied to the output file, and each subsequent input file is appended to the end of the output file. If a field of the output file specification is missing or contains an asterisk wildcard character, the COPY command uses the corresponding field from the first, or only, input file to name the output file.

If multiple input files with maximum record lengths are specified, the output file is given the maximum record length of the first input file. Should the COPY command encounter a record in a subsequent input file that is longer than the maximum record length of the output file, it will issue a message noting the incompatible file attributes, and begin copying the next file.

To create multiple output files, specify multiple input files and use at least one of the following:

- An asterisk wildcard character in the output directory specification, file name, file type, and/or version number field
- Only a node name, a device name, or a directory specification as the output file specification
- The /NOCONCATENATE qualifier

When multiple output files are created, the corresponding field from each input file is used in the output file name.

You can use the /LOG qualifier when you specify multiple input and output files to verify that the files were copied as you intended.

Version Numbers

If no version numbers are specified for input and output files, the COPY command (by default) assigns a version number to the output files that is either of the following:

- The version number of the input file
- A version number one greater than the highest version number of an existing file with the same file name and file type

When the output file version number is specified by an asterisk wildcard character, the COPY command uses the version numbers of the associated input files as the version numbers of the output files.

If the output file specification has an explicit version number, the COPY command uses that number for the output file specification. If a higher version of the output file already exists, a warning message is issued, but the file is still copied. If an equal version of the output file already exists, a message is issued and the input file is *not* copied.

File Protection and Creation/Revision Dates

The COPY command considers an output file to be new when any portion of the output file name is explicitly specified. The creation date for a new file is set to the current time and date.

If the output file specification has one or more wildcard characters, the creation date of the input file is used.

The revision date of the output file is always set to the current time and date; the backup date is set to zero. The output file is assigned a new expiration date. (Expiration dates are set by the file system if retention is enabled; otherwise they are set to zero.)

The protection and access control list (ACL) of the output file is determined by the following parameters, in the following order: protection of previously existing versions of the output file, the default protection and ACL of the output directory, and the process default file protection. (Note that the

BACKUP command takes the creation and revision dates as well as the file protection from the input file.)

Use the /PROTECTION qualifier to change the output file protection.

Normally, the owner of the output file will be the same as the creator of the output file. However, if a user with extended privileges creates the output file, the owner will be the owner of the parent directory or a previous version of the output file if it exists.

"Extended privileges" include any of the following:

- SYSPRV or BYPASS
- System UIC
- GRPPRV if the owner of the parent directory (or previous version of the output file) is in the same group as the creator of the new output file
- An identifier (with the resource attribute) representing the owner of the parent directory (or previous version of the output file)

Copying Directory Files

If you copy a file that is a directory, a new empty directory is created as a subdirectory of the named directory. Note that even if the input directory contained files, none of those files are copied to the new subdirectory. For example:

```
$ COPY  [SMITH]CATS.DIR  [JONES]
```

This COPY command creates the new subdirectory [JONES]CATS.DIR, which is empty.

QUALIFIERS

/ALLOCATION=n

Output-file-spec qualifier.

Forces the initial allocation of the output file to the number of 512-byte blocks specified by n. If not specified, the initial allocation of the output file is determined by the size of the input file being copied.

/BACKUP

Selects files according to the dates of their most recent backup. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /BACKUP qualifier is incompatible with /CREATED, /EXPIRED, and /MODIFIED. /CREATED is the default.

/BEFORE[=time]

Selects only those files that are dated before the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/BY_OWNER[=uic]

Selects one or more files only if their owner user identification code (UIC) matches the specified owner UIC.

Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

If the /BY_OWNER qualifier is specified without a UIC, the UIC of the current process is assumed.

/CONCATENATE (default)
/NOCONCATENATE

Controls whether a single output file is to be created from all the input files when a wildcard character is used in any field of the output file specification.

By default, a wildcard character in an input file specification results in a single output file consisting of the concatenation of all input files matching the file specification.

When you concatenate files from Files–11 Structure Level 2 disks, the COPY command concatenates the files in alphanumeric order. If you specify a wildcard character in the file version field, files are copied in descending order by version number. When you concatenate files from Files–11 Structure Level 1 disks, the COPY command concatenates the files in random order.

/CONFIRM
/NOCONFIRM (default)

Controls whether a request is issued before each individual COPY operation to confirm that the operation should be performed on that file.

When the system issues the prompt, you can issue any of the following responses:

YES	NO	QUIT
TRUE	FALSE	CTRL/Z
1	0	ALL
	<RET>	

You can use any combination of upper- and lowercase letters for word responses. Word responses can be abbreviated to one or more letters (for example, T, TR, or TRU for TRUE). Affirmative answers are YES, TRUE, and 1. Negative answers are NO, FALSE, 0, and <RET> . QUIT or CTRL/Z indicates that you want to stop processing the command at that point. When you respond with 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, the prompt will be reissued.

/CONTIGUOUS
/NOCONTIGUOUS

Output-file-spec qualifier.

Indicates whether the output file is to be contiguous, that is, whether the file must occupy consecutive physical disk blocks. This qualifier can be applied only to an output file.

By default, the COPY command creates an output file in the same format as the corresponding input file. If an input file is contiguous, the COPY command attempts to create a contiguous output file, but it does not report an error if there is not enough space. If you copy multiple input files of different

formats, the output file may or may not be contiguous. You can use the /CONTIGUOUS qualifier to ensure that files are copied contiguously.

The /CONTIGUOUS qualifier has no effect when you copy files to or from tapes, because the size of the file on tape cannot be determined until after it is copied to the disk. If you copy a file from a tape and want the file to be contiguous, use the COPY command twice: once to copy the file from the tape, and a second time to create a contiguous file.

/CREATED (default)

Selects files based on their dates of creation. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /CREATED qualifier is incompatible with /BACKUP, /EXPIRED, and /MODIFIED.

/EXCLUDE=(file-spec[,...])

Any files that match the listed file specifications are excluded from the COPY operation. If you specify only one file, you can omit the parentheses.

Wildcard characters are supported for file specifications. However, you cannot use relative version numbers to exclude a specific version. The file specification can contain a directory specification, but not a device specification.

/EXPIRED

Selects files according to the dates on which they will expire. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /EXPIRED qualifier is incompatible with /BACKUP, /CREATED, and /MODIFIED. /CREATED is the default.

/EXTENSION=n

Output-file-spec qualifier.

Specifies the number of blocks to be added to the output file each time the file is extended. If you do not specify /EXTENSION, the default extension attribute of the output file is determined by the extension attribute of the corresponding input file.

/LOG
/NOLOG (default)

Controls whether the COPY command displays the file specifications of each file copied.

When you use the /LOG qualifier, the COPY command displays the following for each copy operation: (1) the file specifications of the input and output files, (2) the number of blocks or the number of records copied (depending on whether the file is copied on a block-by-block or record-by-record basis), and (3) the total number of new files created.

/MODIFIED

Selects files according to the dates on which they were last modified. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /MODIFIED qualifier is incompatible with /BACKUP, /CREATED, and /EXPIRED. /CREATED is the default.

/OVERLAY
/NOOVERLAY (default)

Output-file-spec qualifier.

Requests that data in the input file be copied into an existing output file, overlaying the existing data. The physical location of the file on disk does not change.

The /OVERLAY qualifier is ignored if the output file is written to a non-file-structured device.

/PROTECTION=(code)

Output-file-spec qualifier.

Defines the protection to be applied to the output file.

Specify the protection code using the rules given in Section 7.1.2 of the *VAX/VMS DCL Concepts Manual*. Any protection field you specify overrides any currently existing protection field.

/READ_CHECK
/NOREAD_CHECK (default)

Input-file-spec qualifier.

Requests the COPY command to read each record in the specified input files twice to verify that all records were correctly read.

/REPLACE
/NOREPLACE (default)

Output-file-spec qualifier.

Requests that, if a file already exists with the same file specification as that entered for the output file, the existing file is to be deleted. The COPY command allocates new space for the output file. In general, when you use the /REPLACE qualifier, you will want to include version numbers with the file specifications.

By default, the COPY command creates a new version of a file if a file with that specification already exists, incrementing the version number. Under /NOREPLACE, when a conflict in version numbers occurs, an error is signaled.

/SINCE[=time]

Selects only those files that are dated after the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/TRUNCATE
/NOTRUNCATE (default)

Output-file-spec qualifier.

Controls whether the COPY command truncates an output file at the end-of-file when copying it. By default, the COPY command uses the allocation of the input file to determine the size of the output file.

/VOLUME=n

Output-file-spec qualifier.

Requests that the COPY command place the entire output file on the specified relative volume number of a multivolume set.

If the /VOLUME qualifier is not specified, the file is placed in an arbitrary position within the multivolume set.

/WRITE_CHECK
/NOWRITE_CHECK (default)

Output-file-spec qualifier.

Requests the COPY command to read each record in the output file after it was written to verify that the record was successfully copied and that the file can subsequently be read without error.

EXAMPLES

1
```
$ COPY TEST.DAT NEWTEST.DAT
```

The COPY command copies the contents of the file TEST.DAT from the default disk and directory to a file named NEWTEST.DAT on the same disk and directory. If a file named NEWTEST.DAT already exists, the COPY command creates a new version of it.

2
```
$ COPY ALPHA.TXT TMP
$ COPY ALPHA.TXT .TMP
```

The first COPY command copies the file ALPHA.TXT into a file named TMP.TXT. The COPY command uses the file type of the input file to complete the file specification for the output file. The second COPY command creates a file named ALPHA.TMP. The COPY command uses the file name of the input file to name the output file.

3
```
$ COPY/LOG TEST.DAT  NEW.DAT;1/REPLACE
%COPY-I-REPLACED, DBA0:[MAL]NEW.DAT;1 being replaced
%COPY-S-COPIED, DBA0:[MAL]TEST.DAT;1 copied to DBA0:[MAL]NEW.DAT;1 (1 block)
```

The /REPLACE qualifier requests that the COPY command replace an existing version of the output file with the new file. The first message from the COPY command indicates that it is replacing an existing file. The version number in the output file must be explicit; otherwise, the COPY command creates a new version of the file NEW.DAT.

4
```
$ COPY *.COM [MALCOLM.TESTFILES]
```

The COPY command copies the highest versions of files in the current default directory with the file type COM to the subdirectory MALCOLM.TESTFILES.

5
```
$ COPY/LOG *.TXT *.OLD
%COPY-S-COPIED, DBA0:[MAL]A.TXT;2 copied to DBA0:[MAL]A.OLD;2 (1 block)
%COPY-S-COPIED, DBA0:[MAL]B.TXT;2 copied to DBA0:[MAL]B.OLD;2 (1 block)
%COPY-S-COPIED, DBA0:[MAL]G.TXT;2 copied to DBA0:[MAL]G.OLD;2 (4 blocks)
%COPY-S-NEWFILES, 3 files created
```

The COPY command copies the highest versions of files with file types of TXT into new files. Each new file has the same file name as an existing file, but a file type of OLD. The last message from the COPY command indicates the number of new files that have been created.

6
```
$ COPY/LOG  A.DAT,B.MEM C.*
%COPY-S-COPIED, DBA0:[MAL]A.DAT;5 copied to DBA0:[MAL]C.DAT;11 (1 block)
%COPY-S-COPIED, DBA0:[MAL]B.MEM;2 copied to DBA0:[MAL]C.MEM;24 (58 records)
%COPY-S-NEWFILES, 2 files created
```

The two input file specifications are separated with a comma. The asterisk wildcard character in the output file specification indicates that two output files are to be created. For each copy operation, the COPY command uses the file type of the input file to name the output file.

7
```
$ COPY/LOG *.TXT TXT.SAV
%COPY-S-COPIED, DBA0:[MAL]A.TXT;2 copied to DBA0:[MAL]TXT.SAV;1 (1 block)
%COPY-S-APPENDED, DBA0:[MAL]B.TXT;2 appended to DBA0:[MAL]TXT.SAV;1 (3 records)
%COPY-S-APPENDED, DBA0:[MAL]G.TXT;2 appended to DBA0:[MAL]TXT.SAV;1 (51 records)
%COPY-S-NEWFILES, 1 file created
```

The COPY command copies the highest versions of all files with the file type TXT to a single output file named TXT.SAV. After the first input file is copied, the messages from the COPY command indicate that subsequent files are being appended to the output file.

Note that, if you use the /NOCONCATENATE qualifier in this example, the COPY command creates as many TXT.SAV files as there are input files. Each TXT.SAV file has a different version number.

8
```
$ COPY MASTER.DOC DBA1:[BACKUP]
```

The COPY command copies the highest version of the file MASTER.DOC to the device DBA1. If no file named MASTER.DOC already exists in the directory [BACKUP], the COPY command assigns the version number of the input file to the output file. You must have W (write) access to the directory [BACKUP] on device DBA1 for the command to work.

9
```
$ COPY SAMPLE.EXE DALLAS::DISK2:[000,000]SAMPLE.EXE/CONTIGUOUS
```

The COPY command copies the file SAMPLE.EXE on the local node to a file with the same name at remote node DALLAS. The /CONTIGUOUS qualifier indicates that the output file is to occupy consecutive physical disk blocks. You must have W (write) access to the device DISK2 on remote node DALLAS for the command to work.

10
```
$ COPY *.* PRTLND::*.*
```

The COPY command copies all files within the user directory at the local node to the remote node PRTLND. The new files will have the same names as the input file. You must have W (write) access to the default directory on remote node PRTLND for the command to work.

11
```
$ COPY BOSTON::DISK2:TEST.DAT;5
_To: DALLAS"SAM SECRET"::DISKO:[MODEL.TEST]TEST.DAT/ALLOCATION=50
```

The COPY command copies the file TEST.DAT;5 on the device DISK2 at node BOSTON to a new file named TEST.DAT at remote node DALLAS. The /ALLOCATE qualifier initially allocates 50 blocks for the new file TEST.DAT at node DALLAS. The access control string SAM SECRET is used to access the remote directory.

12
```
$ MOUNT   TAPED1:   VOL025   TAPE:
$ COPY  TAPE:*.*   *
```

The MOUNT command requests that the volume labeled VOL025 be mounted on the magnetic tape device TAPED1, and assigns the logical name TAPE to the device.

The COPY command uses the logical name TAPE as the input file specification, requesting that all files on the magnetic tape be copied to the current default disk and directory. All the files copied retain their file names and file types.

13
```
$ ALLOCATE CR:
  _CR1: ALLOCATED
$ COPY CR1: CARDS.DAT
$ DEALLOCATE CR1:
```

The ALLOCATE command allocates a card reader for exclusive use by the process. The response from the ALLOCATE command indicates the device name of the card reader, CR1.

After the card reader is allocated, you can place a deck of cards in the reader and issue the COPY command specifying the card reader as the input file. The COPY command reads the cards into the file CARDS.DAT. The end-of-file in the card deck must be indicated with an EOF card (12-11-0-1-6-7-8-9 overpunch).

The DEALLOCATE command relinquishes use of the card reader.

CREATE

Creates one or more sequential disk files from the records that follow the command in the input stream.

FORMAT

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

restrictions

See qualifier descriptions.

PARAMETER

file-spec[,...]

Specifies the name of one or more input files to be created.

If you omit either the file name or the file type, the CREATE command does not supply any defaults; the file name or file type is null. If you do not specify a file version number, and if a file already exists with the same file name and file type as the file specification, the CREATE command creates a new version of the file.

No wildcard characters are allowed in the file specifications.

DESCRIPTION

The CREATE command creates a new sequential disk file. The contents of the file are determined by what you type after the command line. Each separate line that you type becomes a record in the newly created file. When you have finished entering the records, use CTRL/Z to signal the end of the input.

When you issue the CREATE command from a command procedure file, the system reads all subsequent records in the command procedure file into the new file until it encounters a dollar sign in the first position in a record.

If you use an existing file specification with the CREATE command, the newly created file has a higher version number than any existing files with the same specification.

Normally, the owner of the output file will be the same as the creator of the output file. However, if a user with extended privileges creates the output file, the owner will be the owner of the parent directory or a previous version of the output file if it exists.

"Extended privileges" include any of the following:

- SYSPRV or BYPASS
- System UIC
- GRPPRV if the owner of the parent directory (or previous version of the output file) is in the same group as the creator of the new output file
- An identifier (with the resource attribute) representing the owner of the parent directory (or previous version of the output file)

QUALIFIERS

/LOG
/NOLOG (default)

Controls whether the CREATE command displays the file specification of each file that it has created.

/OWNER_UIC=uic

Requires SYSPRV privilege to specify a UIC other than your own.

Specifies the user identification code (UIC) to be associated with the file being created. Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

/PROTECTION=(code)

Defines the protection to be applied to the file. Specify the protection code according to the rules given in Section 7.1.2 of the *VAX/VMS DCL Concepts Manual*.

If you do not specify a value for each access category, or if you omit the /PROTECTION qualifier, the command applies the current default protection for each unspecified category.

If you specify an existing file specification, and you do not specify a value for each access category, or if you omit the /PROTECTION qualifier, the command applies the protection of the existing file to the new file.

/VOLUME=n

Requests that each file be placed on the specified relative volume number of a multivolume set.

If you omit the /VOLUME qualifier, files are placed arbitrarily within the multivolume set.

EXAMPLES

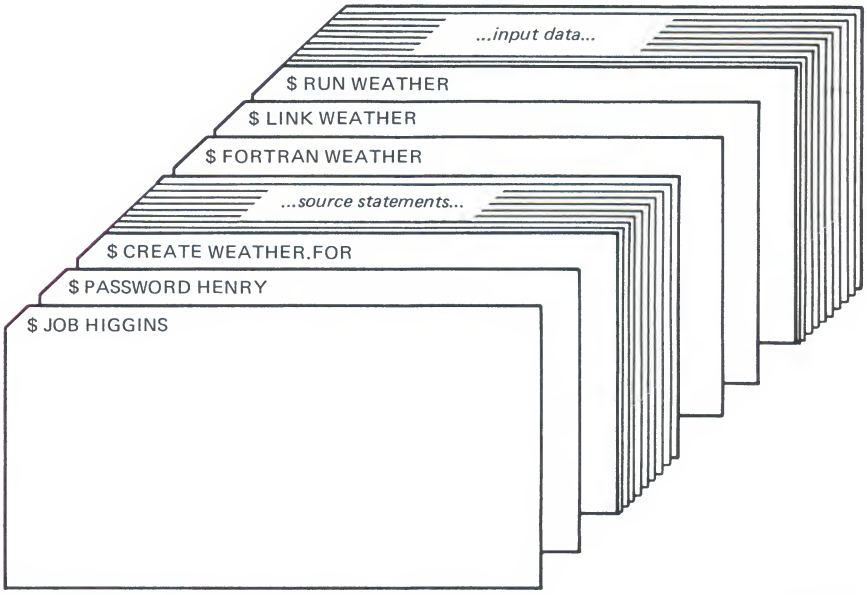
1
```
$ CREATE   A.DAT, B.DAT
Input line one for A.DAT...
Input line two for A.DAT...
   .
   .
   .
^Z
Input line one for B.DAT...
Input line two for B.DAT...
   .
   .
   .
^Z
$
```

After you issue the CREATE command from the terminal, the system reads input lines into the sequential file A.DAT until CTRL/Z terminates the first input. The next set of input data is placed in the second file, B.DAT. Again, CTRL/Z terminates the input.

2
```
$ CREATE  TOLEDO::XXX1:[MODEL.TEST]TEST.DAT
A.1, A.2, A.3, A.4, A.5
52974,53785,78935,57832,78549
B.1, B.2, B.3, B.4, B.5
76984,67486,39076,38676,57681
^Z
$
```

The CREATE command creates a sequential file named TEST.DAT on the remote node TOLEDO. The contents of the files are the four records that follow the CREATE command line. The CTRL/Z entry indicates the end of the file.

3

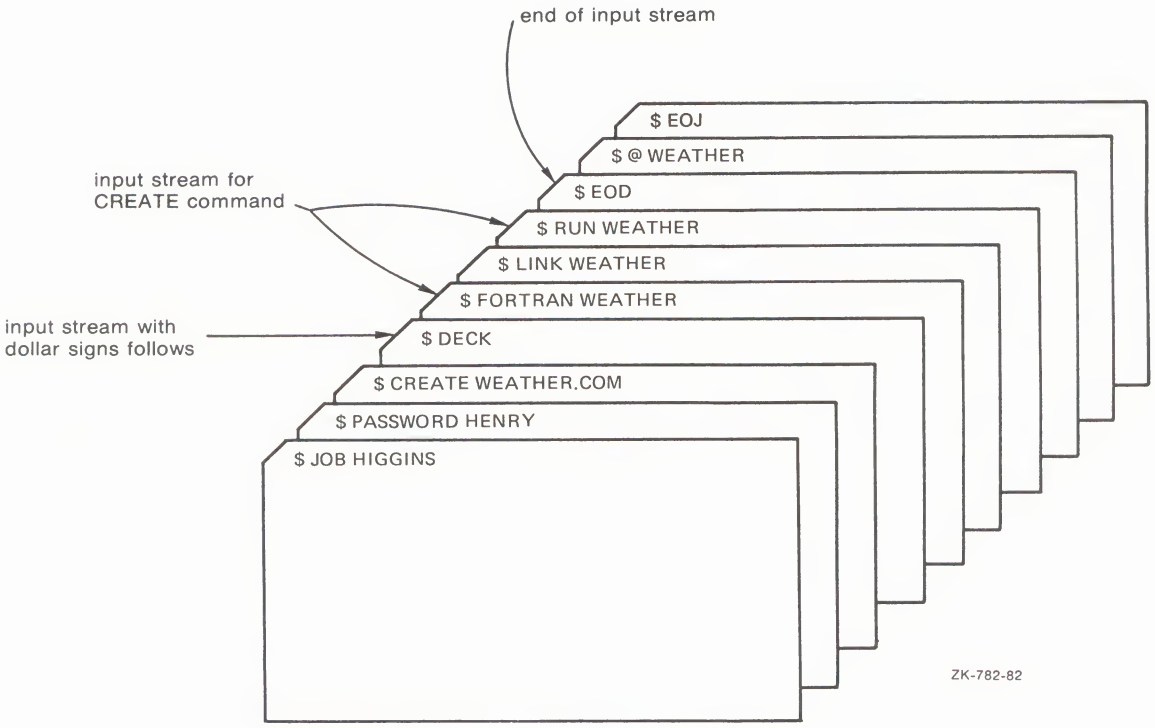


ZK-781-82

In this batch job example, the CREATE command creates a FORTRAN source file WEATHER.FOR. Records are read into that file until the system encounters a dollar sign in the first position of the record $ FORTRAN WEATHER. The next commands compile, link, and run the file just created. Input data follows the RUN command.

CREATE

4



This batch job example uses the CREATE command to create a command procedure from data in the input stream. The DECK command is required so that subsequent lines that begin with a dollar sign are not executed as commands, but are accepted as input records. The EOD command signals the end-of-file for the data records. Then the WEATHER procedure is executed with the @ (Execute Procedure) command.

CREATE/DIRECTORY

Creates a new directory or subdirectory for cataloging files. The /DIRECTORY qualifier is required.

FORMAT

CREATE/DIRECTORY *directory-spec[,...]*

restrictions

- To create a first-level directory, you must have write access to the master file directory (MFD) of the volume on which you are creating the directory. On a system volume, generally only six users with a system UIC or the SYSPRV or BYPASS user privileges have write access to the MFD to create a first-level directory.
- To create a subdirectory, you must have write access to the lowest-level directory that currently exists.

PARAMETER

directory-spec[,...]

Specifies the name of one or more directories or subdirectories to be created.

The directory specifications must contain a directory name. A device name is optional. When you create a subdirectory, separate the names of the directory levels with periods.

Note that it is possible to create a series of nested subdirectories with a single CREATE/DIRECTORY command. For example, [a.b.c] can be created, even though neither [a.b] nor [a] exists at the time the command is issued. Each subdirectory will be created, starting with the highest level and proceeding downwards.

No wildcard characters are allowed in the directory specification.

DESCRIPTION

The CREATE/DIRECTORY command can be used to create new directories as well as subdirectories. Special privileges are needed to create new first-level directories (see restrictions). Generally, users have sufficient privileges to create subdirectories in their own directories. Use the SET DEFAULT command to change from one directory to another.

QUALIFIERS

/LOG
/NOLOG (default)

Controls whether the CREATE/DIRECTORY command displays the directory specification of each directory after creating it.

/OWNER_UIC[=option]

Specifies the user identification code (UIC) to be associated with the directory being created. Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

If you do not specify the /OWNER_UIC qualifier when you create a directory, the command assigns ownership as follows:

- If you specify the directory name in either alphanumeric or subdirectory format, ownership defaults to your UIC (unless you have privileges, in which case the UIC defaults to the parent directory).
- If you specify the directory name in UIC format, ownership defaults to the UIC in the directory name.

/PROTECTION=(code)

Defines the protection to be applied to the directory. Specify the protection code according to the rules given in Section 7.1.2 of the *VAX/VMS DCL Concepts Manual*.

If you do not specify a value for each access category or if you omit the /PROTECTION qualifier when you create a directory, the command uses the protection in effect for the next-higher-level directory, less any delete access for each unspecified category. If you are creating a first-level directory, then the next-higher-level directory is the MFD. (The protection of the MFD is established by the INITIALIZE command.)

/VERSION_LIMIT=n

Specifies that, by default, no more than n versions of each file created in this directory are to be kept. Whenever n versions exist and a new version is created, the lowest version is automatically deleted. If you omit the /VERSION_LIMIT qualifier, the default is the number of versions permitted for the directory at the next-higher-level.

When you specify /VERSION_LIMIT=0, the system creates a directory with no default version limit.

When you change the version limit setting, the new limit applies only to files created after the setting was changed. New versions of files created before the change are subject to the previous version limit.

/VOLUME=n

Requests that the directory file be placed on the specified relative volume number of a multivolume set.

If you omit the /VOLUME qualifier, the file is placed arbitrarily within the multivolume set.

EXAMPLES

1
```
$ CREATE/DIRECTORY DISK2:[MALCOLM]
```

The CREATE/DIRECTORY command creates a directory named [MALCOLM] on the device DISK2. Special privileges are required to create a first-level directory. (See the Restrictions section.)

2
```
$ CREATE/DIRECTORY [MALCOLM.SUB]
$ SET DEFAULT [MALCOLM.SUB]
```

The CREATE/DIRECTORY command creates a subdirectory named [MALCOLM.SUB]. This directory file is placed in the directory named [MALCOLM]. The command SET DEFAULT [MALCOLM.SUB] changes the current default directory to this subdirectory. All files subsequently created are cataloged in [MALCOLM.SUB].

3
```
$ CREATE/DIRECTORY/PROTECTION=(SYSTEM:RWED,OWNER:RWED,GROUP,WORLD) -
_$[MALCOLM.SUB.HLP]
```

The CREATE/DIRECTORY command creates a subdirectory named [MALCOLM.SUB.HLP]. The protection on the subdirectory allows read, write, execute and delete access for the system and owner categories, but prohibits all access for the group or world categories.

4
```
$ CREATE/DIRECTORY [FRED.SUB1.SUB2.SUB3]
```

The CREATE/DIRECTORY command will create a top-level directory ([FRED]) and three subdirectories ([FRED.SUB1], [FRED.SUB1.SUB2], and [FRED.SUB1.SUB2.SUB3]).

CREATE/FDL

Invokes the Create/FDL Utility (CREATE/FDL) to use the specifications in an FDL file to create a new, empty data file. You use this utility when you want to create a data file from a particular FDL specification. The /FDL qualifier is required. For a complete description of the Create/FDL Utility, including more information about the CREATE/FDL command and its qualifier, see the FDL Utility document in the *VAX/VMS File Definition Language Facility Reference Manual*.

FORMAT

CREATE/FDL *=fdl-file-spec [file-spec]*

CREATE/NAME_TABLE

Creates a new logical name table. The /NAME_TABLE qualifier is required.

FORMAT

CREATE/NAME_TABLE *table-name*

restrictions

See qualifier descriptions.

PARAMETER

table-name

The name of the logical name table you are creating. This name is entered as a logical name in either the process directory logical name table (LNM$PROCESS_DIRECTORY) or the system directory logical name table (LNM$SYSTEM_DIRECTORY).

The table name can have from 1 through 31 characters, and must not contain any characters other than alphanumerics, dollar signs, or underscores.

DESCRIPTION

You can use the CREATE/NAME_TABLE command to create a new logical name table. The name of the table is contained within the LNM$PROCESS_DIRECTORY directory table if the table is process-private, and within the LNM$SYSTEM_DIRECTORY directory table if the table is shareable.

Every new table has a parent table, which determines whether the new table is process-private or shareable. To create a process-private table, use the /PARENT_TABLE qualifier to specify LNM$PROCESS_DIRECTORY (the process directory table). To create a shareable table, specify the parent as LNM$SYSTEM_DIRECTORY.

If you do not explicitly provide a parent table, the CREATE/NAME_TABLE command creates a process-private table whose parent is LNM$PROCESS_DIRECTORY; that is, the name of the table is entered in the process directory.

Every table has a size quota. The quota may either constrain the potential growth of the table or indicate that the table's size can be virtually unlimited. The description of the /QUOTA qualifier explains how to specify a quota.

To specify an access mode for the table you are creating use the /USER_MODE, /SUPERVISOR_MODE, or /EXECUTIVE_MODE qualifiers. If you specify more than one of these qualifiers, only the last one entered is accepted. If you do not specify an access mode, then a supervisor mode table is created.

To delete a logical name table, use the DEASSIGN command, specify the name of the table you want to delete, and use the /TABLE qualifier to specify the directory table where the name of the table was entered.

QUALIFIERS

/ATTRIBUTES[=(keyword[,...])]

Specifies attributes for the logical name table. If you do not specify the /ATTRIBUTES qualifier, no attributes are set.

You can specify the following keywords for attributes:

CONFINE	The table is not to be copied into a subprocess by the SPAWN command. You can use this keyword only when you are creating a private logical name table. If a table is created with the CONFINE attribute, all names entered into the table are also confined.
NO_ALIAS	No identical names (either logical names or names of logical name tables) may be entered in an outer access mode in the current directory. If you do not specify NO_ALIAS, then the table may be "aliased" by an identical name created in an outer access mode. If you create a table with the NO_ALIAS attribute, and an identical name in the same or in an outer mode already exists within that logical name directory table, then this name is deleted.
SUPERSEDE	Creates a new table that supersedes any previous (existing) table with the same name, access mode, and directory table that you specify. The new table is created regardless of whether the previous table exists. (If you do not specify the SUPERSEDE attribute, the new table is not created if the previous table exists.) If you specify or accept the default for the qualifier /LOG, you receive a message indicating the result.

If you specify only one keyword, you can omit the parentheses. Only the attributes you specify are set.

/EXECUTIVE_MODE

Requires SYSNAM privilege to create a logical name table in executive mode.

Creates an executive mode logical name table.

/LOG
/NOLOG (default)

Controls whether an informational message indicating the result of the command is generated when the SUPERSEDE attribute was specified, or when the table already exists but the SUPERSEDE attribute was not specified. The default is /LOG; that is, the informational message is displayed.

/PARENT_TABLE=table

Requires enable (E) access to the parent table and SYSPRIV privilege to create a shareable logical name table.

Specifies the name of the parent table. If you do not specify a parent table, the default table is LNM$PROCESS_DIRECTORY. A shareable table has LNM$SYSTEM_DIRECTORY as its parent table.

/PROTECTION

Defines the protection to be applied to the table. By default, the system and the owner have read, write, enable and delete access; the group and the world have no access.

The format for specifying a protection code is described in Section 7.1.2 of the *VAX/VMS DCL Concepts Manual*.

Protection applies only to shareable logical name tables; it does not apply to process-private logical name tables.

/QUOTA=bytes

Specifies the size limit of the table, in bytes. If you do not specify the /QUOTA qualifier, or if you specify /QUOTA=0, the table has unlimited quota.

/SUPERVISOR_MODE (default)

Creates a supervisor mode logical name table. If you do not explicitly specify a mode, a supervisor mode logical name table is created.

/USER_MODE

Creates a user mode logical name table.

EXAMPLES

1

```
$ CREATE/NAME_TABLE TEST_TAB
$ SHOW LOGICAL TEST_TAB
%SHOW-S-NOTRAN, no translation for logical name TEST_TAB
$ SHOW LOGICAL/TABLE=LNM$PROCESS_DIRECTORY TEST_TAB
```

The CREATE/NAME_TABLE command creates a new table called TEST_TAB. By default, the name of the table is entered in the process directory. The first SHOW LOGICAL command does not find the name TEST_TAB because it does not, by default, search the process directory table. You must use the /TABLE qualifier to request that the process directory be searched.

2

```
$ CREATE/NAME_TABLE/ATTRIBUTES=CONFINE EXTRA
$ DEFINE/TABLE=EXTRA MYDISK DISK4:
$ DEFINE/TABLE=LNM$PROCESS_DIRECTORY LNM$FILE_DEV -
_$ EXTRA, LNM$PROCESS, LNM$JOB, LNM$GROUP, LNM$SYSTEM
$ TYPE MYDISK:[COHEN]EXAMPLE1.LIS
```

This example creates a new logical name table called EXTRA. EXTRA is created with the CONFINE attribute. Therefore, the EXTRA table and the names it contains will not be copied to subprocesses.

Next, the logical name MYDISK is placed into the table EXTRA. In order to use the name MYDISK in file specifications, you need to make sure that the table EXTRA is searched when RMS parses file specifications. To do this, you can define a process-private version of the logical name LNM$FILE_DEV to include the name EXTRA as one of its equivalence strings. (The system uses LNM$FILE_DEV to determine the tables to search during logical name translation for device or file specifications, and will use the process-private version of the logical name before using the default system version.) After you define LNM$FILE_DEV, the system searches the following tables during logical name translation: EXTRA, your process table, your job table, your group table, and the system table. Now, you can use the name MYDISK in a file specification and the equivalence string DISK4 will be substituted.

DEALLOCATE

Returns a previously allocated device to the pool of available devices in the system.

FORMAT

DEALLOCATE *device-name[:]*

restrictions

None.

PARAMETER

device-name[:]

Specifies the name of the device to be deallocated. The device name can be a physical device name or a logical name.

If you omit the controller designator and/or unit number, the defaults are controller A and unit 0, respectively.

QUALIFIER

/ALL

Requests that all devices you have currently allocated be deallocated. If you specify /ALL, you cannot specify a device name.

EXAMPLES

1
```
$ DEALLOCATE  DMB1:
```

The DEALLOCATE command deallocates unit 1 of the RK06/RK07 devices on controller B.

2
```
$ ALLOCATE   MT:   TAPE
%DCL-I-ALLOC, _MTB1:  allocated
    .
    .
    .
$ DEALLOCATE  TAPE:
```

The ALLOCATE command requests that any magnetic tape drive be allocated and assigns the logical name TAPE to the device. The response to the ALLOCATE command indicates the successful allocation of the device MTB1. The DEALLOCATE command specifies the logical name TAPE to release the tape drive.

3
```
$ DEALLOCATE/ALL
```

The DEALLOCATE command deallocates all devices that are currently allocated.

DEASSIGN

Cancels logical name assignments made with the ALLOCATE, ASSIGN, DEFINE, or MOUNT command. The DEASSIGN command also deletes logical name tables created with the CREATE/NAME_TABLE command.

FORMAT

DEASSIGN *[logical-name[:]]*

restrictions

See qualifier descriptions.

PARAMETER

logical-name[:]

Specifies the logical name to be deassigned. Logical names can have from 1 to 255 characters. If the logical name contains any characters other than alphanumerics, dollar signs, or underscores, enclose it in quotation marks.

If you terminate the logical-name parameter with a colon, the command interpreter ignores the colon. (Note that the ASSIGN and ALLOCATE commands remove a trailing colon, if present, from a logical name before placing the name in a logical name table.) If a colon is present in the actual logical name, you must type two colons on the logical-name parameter for the DEASSIGN command (for example, DEASSIGN FILE::).

The logical-name parameter is required unless you use the /ALL qualifier.

To delete a logical name table, specify the table name as the logical name parameter. You must also use the /TABLE qualifier to indicate the logical name directory table where the table name is entered.

DESCRIPTION

The DEASSIGN command cancels a logical name assignment that was made with one of the following commands: ALLOCATE, ASSIGN, CREATE /NAME_TABLE, DEFINE, or MOUNT. You can use the /ALL qualifier with DEASSIGN to cancel all logical names in a specified table. If you use the /ALL qualifier and do not specify a table, then all names in the process table (except names created by the command interpreter) are deassigned; that is, all names entered at the indicated access mode or an outer access mode are deassigned.

To specify the logical name table from which you want to deassign a logical name, use the /PROCESS, /JOB, /GROUP, /SYSTEM, or /TABLE qualifier. If you enter more than one of these qualifiers, only the last one entered is accepted. If entries exist for the specified logical name in more than one logical name table, the name is deleted from only the last logical name table specified on the command line. If you do not specify a logical name table, the default is /TABLE=LNM$PROCESS (or /PROCESS).

To specify the access mode of the logical name you want to deassign, use the /USER_MODE, /SUPERVISOR_MODE, or /EXECUTIVE_MODE qualifiers. If you enter more than one of these qualifiers, only the last one is accepted. If you do not specify a mode, then the DEASSIGN command deletes a supervisor mode name. When you deassign a logical name, any identical

names created with outer access modes in the same logical name table are also deleted.

You must have SYSNAM privilege to deassign an executive mode logical name. If you specify /EXECUTIVE_MODE and you do not have SYSNAM privilege, then the DEASSIGN command ignores the qualifier and attempts to deassign a supervisor mode logical name.

All process-private logical names and logical name tables are deleted when you log off the system. User mode entries within the process logical name table are deassigned when any image exits. The logical names in the job table, and the job table itself, are deleted when you log off the system.

Names in all other shareable logical name tables remain there until they are explicitly deassigned, regardless of whether they are user, supervisor, or executive mode names. You must have write (W) access to a shareable logical name table to delete any name in that table.

If you delete a logical name table, all the logical names in the table are also deleted. Also, any descendant tables are deleted. To delete a shareable logical name table, you must have the user privilege SYSPRV or you must have delete (D) access to the table.

QUALIFIERS

/ALL

Specifies that all logical names with the same or an outer access mode in the specified logical name table are to be deleted. If no logical name table is specified, the default is the process table, LNM$PROCESS.

If you specify /ALL, you cannot enter a logical-name parameter.

/EXECUTIVE_MODE

Requires SYSNAM privilege to deassign executive mode logical names.

Deletes entries in the specified logical name table that were created in executive mode. If you specify the /EXECUTIVE_MODE qualifier, the DEASSIGN command also deletes supervisor and user mode entries with the same name.

If you specify the /EXECUTIVE_MODE qualifier and you do not have SYSNAM, then the DEASSIGN command ignores the qualifier and attempts to deassign a supervisor mode logical name.

/GROUP

Requires the user privilege GRPNAM or SYSPRV to delete entries from the group logical name table.

Indicates that the specified logical name is in the group logical name table. The /GROUP qualifier is synonymous with /TABLE=LNM$GROUP.

/JOB

Indicates that the specified logical name is in the job-wide logical name table. The /JOB qualifier is synonymous with /TABLE=LNM$JOB. If you do not explicitly specify a logical name table, the default is /PROCESS.

You should not deassign job-wide logical name entries that were made by the system at login time, for example, SYS$LOGIN, SYS$LOGIN_DEVICE, and SYS$SCRATCH. However, if you assign new equivalence names for these logical names (that is, create new logical names in outer access modes), you can deassign the names you explicitly created.

/PROCESS (default)

Indicates that the specified logical name is in the process logical name table. The /PROCESS qualifier is synonymous with /TABLE=LNM$PROCESS.

You cannot deassign logical name table entries that were made by the command interpreter, for example, SYS$INPUT, SYS$OUTPUT, and SYS$ERROR. However, if you assign new equivalence names for these logical names (that is, you create new logical names in outer access modes), you can deassign the names you explicitly created.

/SUPERVISOR_MODE (default)

Deletes entries in the specified logical name table that were created in supervisor mode. If you specify the /SUPERVISOR_MODE qualifier, the DEASSIGN command also deassigns user mode entries with the same name.

/SYSTEM

Requires the user privilege SYSNAM or SYSPRV to delete entries from the system logical name table.

Indicates that the specified logical name is in the system logical name table. The /SYSTEM qualifier is synonymous with /TABLE=LNM$SYSTEM.

/TABLE=name

Requires write (W) access to the table to delete a shareable logical name. Requires SYSPRV or delete (D) access to delete a shareable logical name table.

Specifies the name of a logical name table from which the logical name is to be deleted. You can specify a user-defined table, or one of the process, job, group, or system tables. (The process, job, group, and system logical name tables should be referred to by the logical names LNM$PROCESS, LNM$JOB, LNM$GROUP, and LNM$SYSTEM, respectively.) You can also specify one of the logical name directory tables.

The /TABLE qualifier also can be used to delete a logical name table. To delete a process-private table, specify:

```
$ DEASSIGN/TABLE=LNM$PROCESS_DIRECTORY table-name
```

To delete a shareable table, specify:

```
$ DEASSIGN/TABLE=LNM$SYSTEM_DIRECTORY table-name
```

To delete a shareable logical name table, you must have delete (D) access to the table or write (W) access to the directory table in which the name of the shareable table is cataloged.

If you do not explicitly specify the /TABLE qualifier, the default is /TABLE=LNM$PROCESS (or /PROCESS).

/USER_MODE

Deletes entries in the process logical name table that were created in user mode. If you specify the /USER_MODE qualifier, the DEASSIGN command can deassign only user mode entries.

EXAMPLES

```
1   $ SHOW LOGICAL TEST_CASES
      "TEST_CASES" = "USER_DISK:[HARVEY]FILES.DAT" (LNM$PROCESS_TABLE)
    $ DEASSIGN   TEST_CASES
    $ SHOW LOGICAL TEST_CASES
    %SHOW-S-NOTRAN, no translation for logical name TEST_CASES
```

The SHOW LOGICAL command displays the current equivalence name for the logical name TEST_CASES. The DEASSIGN command deassigns the equivalence name; the next SHOW LOGICAL command indicates that the name has been deassigned.

```
2   $ ASSIGN USER_DISK:   COPY
    $ DEASSIGN COPY
```

The ASSIGN command equates the logical name COPY with the device USER_DISK and places the names in the process logical name table. The DEASSIGN command deletes the logical name.

```
3   $ DEFINE SWITCH:  TEMP
    $ DEASSIGN SWITCH::
```

The DEFINE command places the logical name SWITCH: in the process logical name table. The trailing colon is retained as part of the logical name. Two colons are required on the DEASSIGN command to delete this logical name because the DEASSIGN command removes one trailing colon, and the other colon is needed to match the characters in the logical name.

```
4   $ ASSIGN/TABLE=LNM$GROUP DBA1: GROUP_DISK
    $ DEASSIGN/PROCESS/GROUP GROUP_DISK
```

The ASSIGN command places the logical name GROUP_DISK in the group logical name table. A subsequent DEASSIGN command specifies conflicting qualifiers; because the /GROUP qualifier is last, the name is successfully deassigned.

```
5   $  DEASSIGN/ALL
```

The DEASSIGN command deletes all names from the process logical name table that were created in supervisor or user mode. This command does not, however, delete the names that were placed in the process logical name table in executive mode by the command interpreter (for example, SYS$INPUT, SYS$OUTPUT, SYS$ERROR, SYS$DISK, and SYS$COMMAND).

```
6   $ ASSIGN DALLAS::USER_DISK:   DATA
      .
      .
      .
    $ DEASSIGN DATA
```

The ASSIGN command associates the logical name DATA with the device specification USER_DISK on remote node DALLAS. Subsequent references to the logical name DATA result in references to the disk on the remote node. The DEASSIGN command cancels the logical name assignment.

7 `$ DEASSIGN/TABLE=LNM$PROCESS_DIRECTORY TABLE1`

This command deletes the logical name table TABLE1 from the process directory table. When you delete a logical name table, you must specify either /TABLE=LNM$PROCESS_DIRECTORY or /TABLE=LNM$SYSTEM_DIRECTORY, because the names of all tables are contained in these directories.

DEASSIGN/QUEUE

Deassigns a logical queue from a printer or terminal queue and stops the logical queue. The /QUEUE qualifier is required. The DEASSIGN/QUEUE command is the complement of the ASSIGN /QUEUE command described in this manual.

FORMAT

DEASSIGN/QUEUE *logical-queue-name[:]*

restrictions

- Requires operator (OPER) privilege or execute (E) access to the queue.
- Cannot be used with batch queues.

PARAMETER

logical-queue-name[:]

Specifies the name of the logical queue that is to be deassigned from a specific printer or terminal queue.

DESCRIPTION

Once you issue the DEASSIGN/QUEUE command, the jobs in the logical queue remain in a pending state until the queue is reassigned to another printer queue or device with the ASSIGN/QUEUE command.

EXAMPLE

```
$ ASSIGN/QUEUE LPAO  ASTER
    .
    .
    .
$ DEASSIGN/QUEUE  ASTER
$ ASSIGN/MERGE LPBO  ASTER
```

The ASSIGN/QUEUE command associates the logical queue ASTER with the print queue LPA0. Later, you deassign the logical queue with the DEASSIGN /QUEUE command. The ASSIGN/MERGE command reassigns the jobs from ASTER to the print queue LPB0.

DEBUG

Invokes the VAX/VMS Debugger after program execution is interrupted by CTRL/C or CTRL/Y. For a complete description of the VAX/VMS Debugger, including more information about the DEBUG command, see the *VAX/VMS Debugger Reference Manual*.

FORMAT DEBUG

DECK

Marks the beginning of an input stream for a command or program.

FORMAT

$ DECK

restrictions

Can be used only after a request to execute a command or program that requires input data.

PARAMETERS

None.

DESCRIPTION

The DECK command marks the following data as input for a command or program. DECK is required in command procedures when the first nonblank character in any data record in the input stream is a dollar sign.

The DECK command must be preceded by a dollar sign; the dollar sign must be in the first character position (column 1) of the input record.

The DECK command defines an end-of-file indicator only for a single data stream. Use of the DECK command enables you to place data records beginning with dollar signs in the input stream. You can place one or more sets of data in the input stream following a DECK command, if each is terminated by an end-of-file indicator.

After an end-of-file indicator specified with the /DOLLARS qualifier is encountered, the end-of-file indicator is reset to the default, that is, to any record beginning with a dollar sign. The default is also reset if an actual end-of-file occurs for the current command level.

QUALIFIER

/DOLLARS[=string]

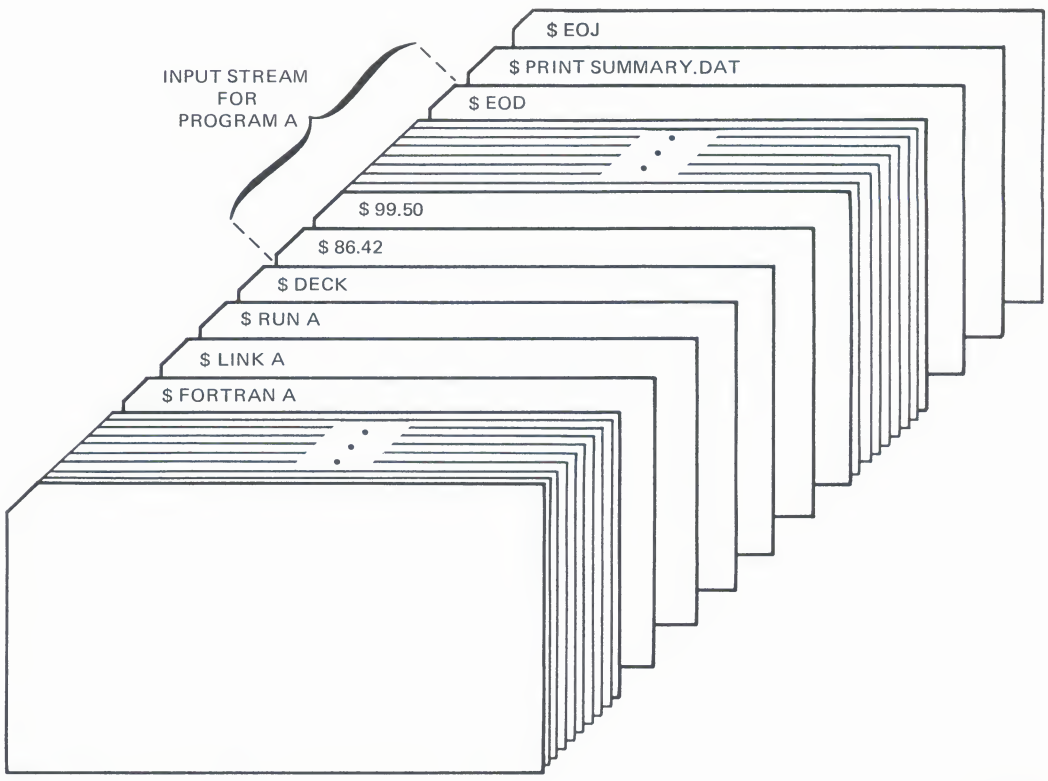
Sets the end-of-file indicator to the specified string.

If you do not specify /DOLLARS, or if you specify /DOLLARS without specifying a string, you must use the EOD command to signal the end-of-file.

Specify a string if the input data contains one or more records beginning with the string $EOD. The string can have from 1 through 15 characters. Enclose it in quotation marks if you want to specify an end-of-file indicator that contains literal lowercase letters, multiple spaces, or tabs.

EXAMPLES

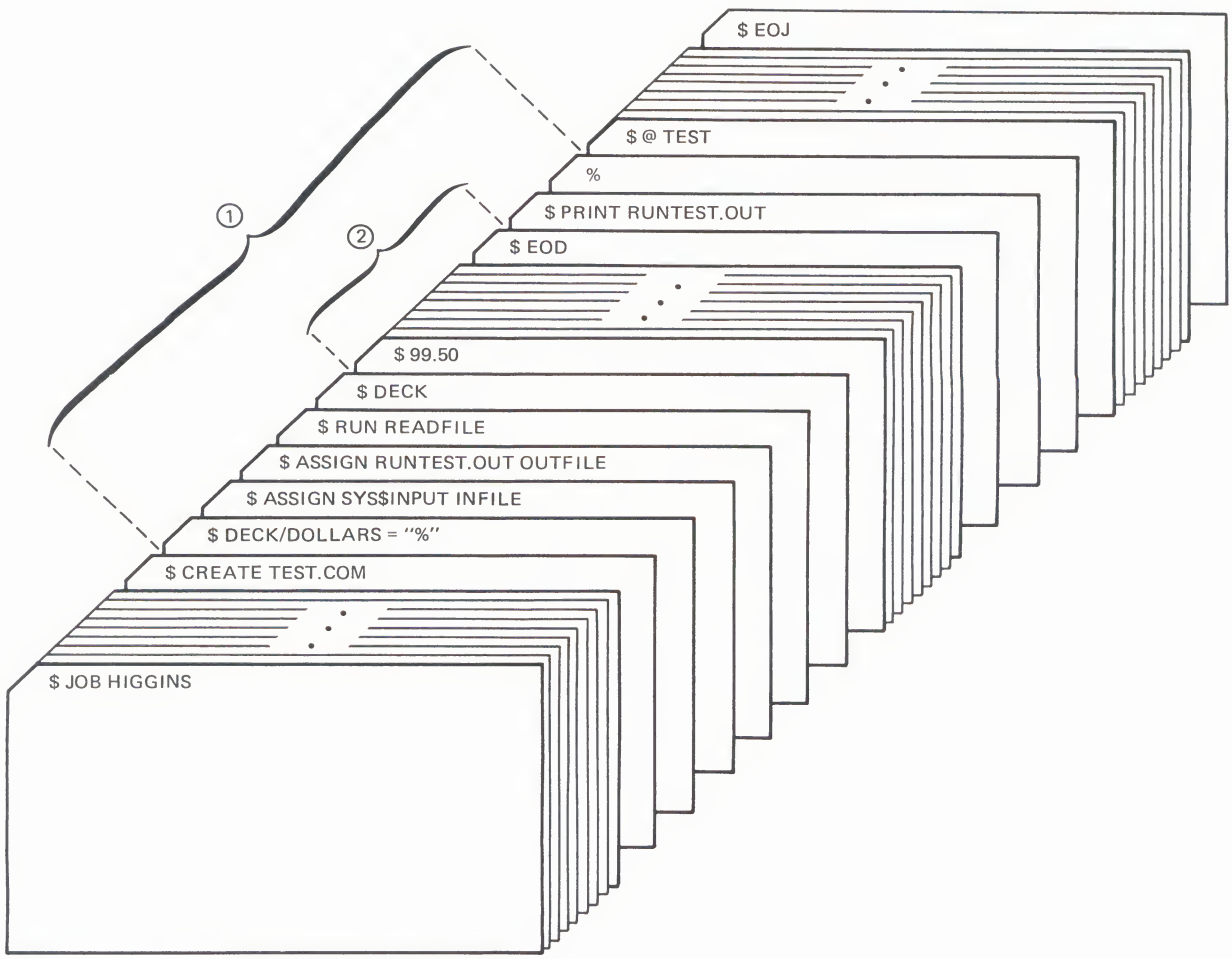
1



The FORTRAN and LINK commands compile and link program A. When the program is run, any data the program reads from the logical device SYS$INPUT is read from the command stream. The DECK command indicates that the input stream can contain dollar signs in column 1 of the record. The EOD command signals end-of-file for the data.

DECK

2



(1) INPUT STREAM FOR CREATE COMMAND

(2) INPUT STREAM FOR PROGRAM READFILE

ZK-784-82

The CREATE command creates the command procedure file TEST.COM from lines entered into the input stream. The DECK/DOLLARS command indicates that the percent sign character is the end-of-file indicator for the CREATE command. This allows the string $EOD to be read as an input record, signaling the end of the input for the RUN command.

DEFINE

Creates a logical name entry and assigns an equivalence name string, or a list of strings, to the specified logical name.

FORMAT

DEFINE *logical-name equivalence-name[,...]*

restrictions

See qualifier descriptions.

PARAMETERS

logical-name

Specifies the logical name string. The logical name string can contain from 1 to 255 characters. (However, logical names entered in one of the logical name directory tables, LNM$SYSTEM_DIRECTORY or LNM$PROCESS_DIRECTORY, must be no more than 31 characters long, and must not contain characters other than alphanumerics, the dollar sign, or the underscore.) If you specify a colon at the end of a logical name, the DEFINE command saves the colon as part of the logical name. (This is in contrast to the ASSIGN command, which removes the colon before placing the name in a logical name table.) By default, the logical name is placed in the process logical name table.

If the string contains any characters other than alphanumerics, the dollar sign, or the underscore character, enclose the string in quotation marks. If the logical name contains quotation marks, enclose the name in quotation marks and use two sets of quotation marks ("") in the places where you want a quotation mark (") to occur. Note that if you enclose a name in quotation marks, the case of alphabetic characters is preserved.

equivalence-name[,...]

Defines the equivalence names to be associated with the logical name in the specified logical name table. An equivalence name string can contain from 1 to 255 characters. If the string contains any characters other than alphanumerics, the dollar sign, or the underscore character, enclose the string in quotation marks. If the equivalence name contains quotation marks, enclose the string in quotation marks and use two sets of quotation marks ("") in the places where you want a quotation mark (") to occur.

When you specify an equivalence name that will be used as a file specification, you must include the punctuation marks (colons, brackets, periods) that would be required if the equivalence name were used directly as a file specification. Therefore, if you specify a device name as an equivalence name, you must terminate the equivalence name with a colon.

The DEFINE command allows you to assign the same logical name to more than one equivalence name. For example, you can use the same logical name to access different directories on different disks, or to access different files in different directories. When you specify more than one equivalence name for a logical name, you create a search list. See Section 4.8 of the *VAX/VMS DCL Concepts Manual* for more information on search lists.

DESCRIPTION

The DEFINE command creates an entry in a logical name table by defining a logical name to stand for one or more equivalence names. An equivalence name can be a device name, another logical name, a file specification, or any other string.

To specify the logical name table where you want to enter a logical name, use the /PROCESS, /GROUP, /SYSTEM, /JOB or /TABLE qualifier. If you enter more than one of these qualifiers, only the last one entered is accepted. If you do not specify a table, the default is /TABLE=LNM$PROCESS (or /PROCESS).

To specify the access mode of the logical name you are creating, use the /USER_MODE, /SUPERVISOR_MODE, or /EXECUTIVE_MODE qualifiers. If you enter more than one of these qualifiers, only the last one entered is accepted. If you do not specify an access mode, then a supervisor mode name is created. You can create a logical name in the same mode as the table in which you are placing the name, or in an outer mode. (User mode is the outermost mode; executive mode is the innermost mode.)

You can enter more than one logical name with the same name in the same table, as long as each name has a different access mode. (However, if an existing logical name within a table has the NO_ALIAS attribute, you cannot use the same name to create a logical name in an outer mode in this table.)

If you create a logical name with the same name, in the same table, and in the same mode as an existing name, the new logical name assignment replaces the existing assignment.

You can also use the ASSIGN command to create logical names. To delete a logical name from a table, use the DEASSIGN command.

Note: Avoid assigning a logical name that matches the file name of an executable image in SYS$SYSTEM:. Such an assignment will prohibit you from invoking that image.

For additional information on how to create and use logical names, see Chapter 4 of the *VAX/VMS DCL Concepts Manual.*

QUALIFIERS

/EXECUTIVE_MODE

Requires the user privilege SYSNAM to create an executive mode logical name.

Creates an executive mode logical name in the specified table.

If you specify the /EXECUTIVE_MODE qualifier and you do not have SYSNAM, then the DEFINE command ignores the qualifier and creates a supervisor mode logical name.

/GROUP

Requires the user privilege GRPNAM or SYSPRV to place a name in the group logical name table.

Places the logical name in the group logical name table. Other users who have the same group number in their UICs (user identification codes) can access the logical name. The /GROUP qualifier is synonymous with /TABLE=LNM$GROUP.

/JOB

Places the logical name in the job-wide logical name table. All processes in the same job tree as the process that created the logical name can access the logical name. The /JOB qualifier is synonymous with /TABLE=LNM$JOB.

/LOG (default)
/NOLOG

Controls whether a message is displayed when you define a logical name that supersedes an existing name.

/NAME_ATTRIBUTES[=(keyword[,...])]

Specifies attributes for a logical name. By default, no attributes are set. You can specify the following keywords for attributes:

CONFINE	The name is not to be copied into a subprocess by the SPAWN command. This keyword is meaningful only when you create logical names in a private table. If you specify this keyword for a logical name in a shareable table, it is ignored. The CONFINE attribute is also inherited from the logical name table where the name is entered; if the logical name table is "confined", then all names in the table are "confined".
NO_ALIAS	The logical name cannot be duplicated in this table with a less privileged (outer) access mode. If another logical name with the same name and an outer access mode already exists in this table, the name is deleted.

If you specify only one keyword, you can omit the parentheses. Only the attributes you specify are set.

/PROCESS (default)

Places the logical name in the process logical name table. The /PROCESS qualifier is synonymous with /TABLE=LNM$PROCESS.

/SUPERVISOR_MODE (default)

Creates a supervisor mode logical name in the specified table.

/SYSTEM

Requires the user privilege SYSNAM or SYSPRV to place a name in the system logical name table.

Places the logical name in the system logical name table. All system users can access the logical name. The /SYSTEM qualifier is synonymous with /TABLE=LNM$SYSTEM.

/TABLE=name

Requires write (W) access to the table to specify the name of a shareable logical name table.

Specifies the name of the logical name table in which the logical name is to be entered. You can use the /TABLE qualifier to specify a user-defined logical name table (created with the CREATE/NAME_TABLE command); to specify the process, job, group, or system logical name tables; or to specify the process or system logical name directory tables.

If you specify the table name using a logical name that has more than one translation, the logical name is placed in the first table found. For example, if you specify DEFINE/TABLE=LNM$FILE_DEV and LNM$FILE_DEV is equated to LNM$PROCESS, LNM$JOB, LNM$GROUP, and LNM$SYSTEM, then the logical name is placed in LNM$PROCESS.

If you do not explicitly specify the /TABLE qualifier, the default is /TABLE=LNM$PROCESS (or /PROCESS).

/TRANSLATION_ATTRIBUTES[=(keyword[,...])]

Equivalence-name qualifier.

Specifies one or more attributes that modify an equivalence string of the logical name. You can specify the following keywords for translation attributes:

CONCEALED	Indicates that the equivalence string is a concealed device name. When a concealed device name is defined, the system displays the logical name, rather than the equivalence string, in messages that refer to the device.
TERMINAL	Indicates that the equivalence string should not be translated iteratively; logical name translation should terminate with the current equivalence string.

If you specify only one keyword, you can omit the parentheses. Only the attributes you specify are set.

Note that different equivalence strings of a logical name can have different translation attributes.

/USER_MODE

Creates a user mode logical name in the specified table.

User mode logical names created within the process logical name tables are used for the execution of a single image; for example, you can create a user mode logical name to allow an image executing in a command procedure to redefine SYS$INPUT. User mode entries are deleted from the process logical name table when any image executing in the process exits (that is, after any DCL command or user program that executes an image completes execution).

EXAMPLES

1

```
$ DEFINE CHARLIE XXX1:[CHARLES]
$ PRINT  CHARLIE:TEST.DAT
Job 274 entered on queue SYS$PRINT
```

The DEFINE command associates the logical name CHARLIE with the directory name [CHARLES] on the disk XXX1. Subsequent references to the logical name CHARLIE result in the correspondence between the logical name CHARLIE and the disk and directory specified. The PRINT command queues a copy of the file XXX1:[CHARLES]TEST.DAT to the system printer.

2

```
$ DEFINE PROCESS_NAME  LIBRA
$ RUN WAKE
```

The DEFINE command places the logical name PROCESS_NAME in the process logical name table with an equivalence name of LIBRA. The logical name is created in supervisor mode. The program WAKE translates the logical name PROCESS_NAME to perform some special action on the process named LIBRA.

3

```
$ DEFINE TEMP:  XXX1:
   .
   .
   .
$ DEASSIGN TEMP::
```

The DEFINE command creates an equivalence name for the logical name TEMP: and places the name in the process logical name table. The colon is retained as part of the logical name. The DEASSIGN command deletes the logical name. Note that two colons are required on the logical name in the DEASSIGN command. One colon is stripped by the DEASSIGN command. The other colon is kept as part of the logical name.

4

```
$ DEFINE PORTLAND PRTLND::YYY0:[DECNET.DEMO.COM]
```

The DEFINE command places the logical name PORTLAND in the process logical name table with an equivalence name of PRTLND::YYY0:[DECNET.DEMO.COM].

5

```
$ DEFINE LOCAL "BOSTON""JOHN_SMITH JKS""::"
```

The DEFINE command places the logical name LOCAL in the process logical name table with a remote node equivalence name of BOSTON"JOHN_SMITH JKS"::. To satisfy conventions for local DCL command string processing, you must use three sets of quotation marks, so that access control information will be enclosed in one set of quotation marks in the equivalence name.

6

```
$ DEFINE MYDISK XXX0:[MYDIR], YYY0:[TESTDIR]
```

This DEFINE command places the logical name MYDISK in the process logical name table with two equivalence names: XXX0:[MYDIR] and YYY0:[TESTDIR].

7

```
$ CREATE/NAME_TABLE TABLE1
$ DEFINE/TABLE=LNM$PROCESS_DIRECTORY LNM$FILE_DEV -
_$ TABLE1,LNM$PROCESS,LNM$JOB,LNM$GROUP,LNM$SYSTEM
$ DEFINE/TABLE=TABLE1 -
_$ /TRANSLATION_ATTRIBUTES=CONCEALED  WORK_DISK DBA1:
```

The CREATE/NAME_TABLE command creates the process private logical name table TABLE1.

The first DEFINE command ensures that TABLE1 is searched first in any logical name translation of a device or file specification (because TABLE1 is the first item in the equivalence string for the logical name LNM$FILE_DEV, which determines the default search sequence of logical name tables whenever a device or file specification is translated).

The second DEFINE command assigns the logical name WORK_DISK to the physical device DBA1, and places the name in TABLE1. The logical name has the concealed attribute. Therefore, the logical name WORK_DISK will be displayed in system messages.

8
```
$ CREATE/NAME_TABLE SPECIAL
$ DEFINE/TABLE=LNM$PROCESS_DIRECTORY LNM$FILE_DEV -
_$ SPECIAL,LNM$PROCESS,LNM$JOB,LNM$GROUP,LNM$SYSTEM
$ DEFINE/TABLE=LNM$PROCESS_DIRECTORY TAB SPECIAL
$ DEFINE/TABLE=TAB REPORT [CHELSEA]STORES
$ SHOW LOGICAL/TABLE=SPECIAL REPORT
  "REPORT" = "[CHELSEA]STORES" (SPECIAL)
```

In this example, the CREATE/NAME_TABLE command is used to create a new logical name table called SPECIAL. This table is defined in the process directory, LNM$PROCESS_DIRECTORY. The first DEFINE command ensures that the logical name table SPECIAL is searched first during iterative logical name translation of a device or file specification, because the logical name LNM$FILE_DEV is used by the system is determining the sequence for searching logical name tables in this case, and because LNM$FILE_DEV was placed in the process logical name directory table (thus taking precedence over any definition of LNM$FILE_DEV in the system directory table).

With the next DEFINE command, a new logical name, TAB, is defined. TAB translates to the string SPECIAL, which identifies a logical name table. You must define TAB in the process directory because it translates iteratively to a logical name table.

Next, the logical name REPORT is placed into the logical name table TAB. Because TAB translates to the table SPECIAL, the name REPORT is entered into SPECIAL table. The SHOW LOGICAL command verifies that the name REPORT has been entered into the table SPECIAL.

Note that you can redefine TAB so it translates to a different table. Therefore, if you run different programs that use the name TAB as a table name, you can change the actual tables where the names are entered or referenced.

DEFINE/CHARACTERISTIC

Defines a characteristic name and associated characteristic number in the system characteristics table. If the characteristic name is already defined, the request alters the definition of the characteristic. The /CHARACTERISTIC qualifier is required.

FORMAT

DEFINE/CHARACTERISTIC *characteristic-name characteristic-number*

restrictions

Requires operator (OPER) privilege.

PARAMETERS

characteristic-name

Assigns a name to the characteristic being defined. Characteristic names can have from 1 to 31 characters, including all upper and lower case letters, digits, the dollar sign, and the underscore. The name, however, must contain at least one non-numeric character.

characteristic-number

Assigns a number to the characteristic being defined. Numbers can range from 0 to 127.

DESCRIPTION

The system manager or operator uses the DEFINE/CHARACTERISTIC command to assign a name and number to a particular characteristic for queues in the system. Characteristics can refer to any attribute of a print or batch job that is meaningful for your environment. The name and number of a characteristic are arbitrary, but must be unique for that characteristic. You can use the DEFINE/CHARACTERISTIC command to add a new characteristic or change the number of a previously defined characteristic. Use the SHOW QUEUE/CHARACTERISTICS command to find out what characteristics are currently defined for the system. The DELETE /CHARACTERISTIC command deletes a previously defined characteristic.

When queues are initialized or started, you can use either characteristic names or numbers with the /CHARACTERISTICS qualifier to specify characteristics that are to be associated with the queue. Similarly, when users issue the PRINT or SUBMIT command with the /CHARACTERISTICS qualifier, they can use either the characteristic name or number to specify which queue characteristics must match before the job is executed.

You can use the SET QUEUE command to change the characteristics of a queue. If you need to change the physical setup of the queue, use the STOP /QUEUE/NEXT command to stop the queue. Then change the setup and issue the START/QUEUE command with the new characteristics specified.

When users include the /CHARACTERISTICS qualifier with a PRINT or SUBMIT command, all the characteristics they specify must also be specified for the queue that will be executing the job. If not, the job will remain pending in the queue until the queue characteristics are changed or they delete the entry with the DELETE/ENTRY command. Users need not specify

every characteristic of a queue with a PRINT or SUBMIT command as long as the ones they specify are a subset of the characteristics set for that queue. The job will also run if no characteristics are specified.

You can use the SHOW QUEUE/CHARACTERISTICS command to find out which characteristics are available on the system. Use the SHOW QUEUE /FULL command to find out which characteristics have been specified for a particular queue.

EXAMPLE

```
$ DEFINE/CHARACTERISTIC REDINK 3
```

The example defines the characteristic REDINK with the number 3. When a user issues the command PRINT/CHARACTERISTICS=REDINK (or PRINT /CHARACTERISTICS=3), the job will be printed only if the printer queue has been established with the REDINK or 3 characteristic.

DEFINE/FORM

Defines a form name and number as well as the type of the physical paper stock. If the form name is already defined, DEFINE/FORM alters the definition of the form. Each form must have a unique number. The /FORM qualifier is required.

FORMAT

DEFINE/FORM *form-name form-number*

restrictions

Requires operator (OPER) privilege.

PARAMETERS

form-name

Assigns a name to the form being defined. Form names can have from 1 to 31 characters, including all upper and lowercase letters, digits, the dollar sign, and the underscore. The name, however, must contain at least one non-numeric character.

form-number

Assigns a number to the form being defined. Numbers can range from 0 to 999. The DEFAULT form, which is automatically defined when the system is bootstrapped is assigned number 0.

DESCRIPTION

The system manager or operator uses the DEFINE/FORM command to assign a name and number to a type of paper stock or printing area for use with printer or terminal queues. When a new queue file is created, the system defines a form named DEFAULT with a form number of zero and all the default attributes.

The DEFINE/FORM qualifiers specify the area for printing. The /LEFT, /RIGHT, and /WIDTH qualifiers determine the number of characters per line. Using the /RIGHT and /WIDTH qualifiers, you can affect the point at which lines of text in the file will wrap. (These qualifiers cannot be used for filling or formatting the text, however.)

You can also use the DEFINE/FORM command to specify different types of paper stock. The /DESCRIPTION qualifier enables you to describe more fully the form name.

When a printer or terminal queue is initialized, you can use either the form name or number to specify the form for the queue. Similarly, when users issue a PRINT command with the /FORM qualifier, they can use either the form name or number to specify which form they want. The default form is number 0.

To change the form type of a queue, you should stop the queue using the STOP/QUEUE/NEXT command, change the physical form, and then restart the queue specifying the appropriate form with the START/QUEUE command.

When you include the /FORM qualifier with a PRINT command, the form you specify must match the one specified for the queue that will be executing the job. If not, the job will remain pending in the queue until the queue characteristics are changed or you delete the entry with the DELETE/ENTRY command. If you omit the /FORM qualifier from your PRINT command, your job will be printed using the default form definition.

You can use the SHOW QUEUE/FORM command to find out what forms are available on the system. Use the SHOW QUEUE/FULL command to find out what form has been specified for a particular queue.

QUALIFIERS

/DESCRIPTION=string

Specifies a string of up to 255 characters. The default string is the specified form name.

The string can be used to define the form type more specifically. For example, if you have form names such as LETTER1, LETTER2, and LETTER3, the /DESCRIPTION qualifier could be used to let the users and operators know that LETTER1 refers to the standard corporate letterhead paper (8.5 x 11), LETTER2 refers to the smaller corporate letterhead paper (6 x 9), and LETTER3 refers to the president's personalized letterhead paper.

/LENGTH=n

Specifies the physical length of a form page in lines. The default page length is 66 lines, which assumes a standard page length of 11 inches with 6 lines of print per inch. The n parameter must be a positive integer greater than 0 and not more than 255.

The print symbiont sets the page length of the device equal to the form length. This enables the driver to compute the number of line feeds for devices lacking mechanical form feed.

/MARGIN=(option[,...])

Specifies one or more of the four margin options: BOTTOM, LEFT, RIGHT, and TOP.

BOTTOM=n — Specifies the number of blank lines to leave between the end of the print image on a page and the end of the physical page of paper. The n parameter must be a positive integer that is less than the /LENGTH parameter. The default value is 6, which generally means a one-inch bottom margin.

LEFT=n — Specifies the number of columns to be left blank between the leftmost printing position and the actual print image area. The n parameter must be a positive integer that is between 0 and the value of the /WIDTH parameter. The default value is 0, which means that the print image area starts as far to the left of the paper as the printer can go.

RIGHT=n — Specifies the number of columns to be left blank between the /WIDTH setting and the actual print image area. The n parameter must be a positive integer that is between 0 and the value of the /WIDTH parameter. When determining the /RIGHT parameter, start at the /WIDTH value and count to the left. The default value is 0, which means that the print image extends as far to the right as the /WIDTH value.

TOP=n — Specifies the number of blank lines to leave between the top of the physical page of paper and the start of the print image. The n parameter must be a positive integer between 0 and the /LENGTH parameter. The default value is 0, which generally means that there is no top margin.

/PAGE_SETUP=(module[,...])
/NOPAGE_SETUP (default)

Specifies one or more modules that set up the device before every page. The modules are located in the device control library. When a new page is detected, the system extracts the appropriate modules from the device control library and copies them to the printer before the page is printed.

/SETUP=(module[,...])

Specifies one or more modules that set up the device appropriately for the specified form. The modules are located in the device control library. When the form is mounted, the system extracts the appropriate modules from the device control library and copies them to the printer before the file is printed.

/SHEET_FEED
/NOSHEET_FEED (default)

Specifies that the output print symbiont pause at the end of every physical page so that a new piece of paper can be inserted.

/STOCK=string

Enables you to specify a type of paper stock to be associated with the form you are defining. The default for string is the form name.

This qualifier is useful when you have several forms that use the same paper stock, but differ in other ways, such as margin specifications, wrapping, or page dimension. The system will change from one form to another automatically if those forms have an identical /STOCK qualifier. If the /STOCK qualifiers are different, you need to stop the queue, change the form, and restart the queue to print on another stock.

The string parameter with /STOCK can be from 1 to 31 characters, including all letters, all digits, the dollar sign, and the underscore. You can create any string that you want. However, when you are creating forms with the same stock, be sure that the /STOCK string is identical in all the DEFINE/FORM commands that refer to the same type of paper.

/TRUNCATE (default)
/NOTRUNCATE

Determines how the printer will accommodate lines that exceed either the /WIDTH value or the /MARGIN=RIGHT value. If you specify /TRUNCATE, the default, the printer will discard any characters that exceed the line length that is currently in effect.

If you specify /TRUNCATE, you cannot specify /WRAP. The /TRUNCATE qualifier forces /NOWRAP.

If you specify both /NOTRUNCATE and /NOWRAP, the printer prints all characters on a line insofar as it is able. This combination of qualifiers is useful for some types of graphics output.

/WRAP
/NOWRAP (default)

Determines how the printer will accommodate lines that exceed either the /WIDTH value or the /MARGIN=RIGHT value. The /WRAP qualifier causes print lines that exceed the current line length to wrap onto the next line.

If you specify /WRAP, you cannot specify /TRUNCATE. The /WRAP qualifier forces /NOTRUNCATE.

If you specify both /NOWRAP and /NOTRUNCATE, the printer prints all characters on a line insofar as it is able. This combination of qualifiers is useful for some types of graphics output.

/WIDTH=n

Specifies the physical width of the paper in terms of columns or character positions. Any lines exceeding this value will be wrapped if /WRAP is in effect or truncated if /TRUNCATE is in effect. (If both /NOTRUNCATE and /NOWRAP are in effect, lines will be printed as far as is possible.) If the /MARGIN=RIGHT qualifier has been specified, its value overrides the /WIDTH value in determining when lines of text will be wrapped. The n parameter must be a positive integer greater than 0 and not more than 65535. The default value for n is 132.

EXAMPLE

```
$ DEFINE/FORM /MARGIN=(TOP=6,LEFT=10) CENTER 3
```

The example defines the form CENTER to have a top margin of 6 and a left margin of 10. The defaults remain in effect for both bottom margin (6) and right margin (0). The form is assigned the number 3.

DEFINE/KEY

Associates an equivalence string and a set of attributes with a key on the terminal keyboard. The /KEY qualifier is required.

FORMAT

DEFINE/KEY *key-name equivalence-string*

restrictions

None.

PARAMETERS

key-name

Specifies the name of the key that you are defining. The following table lists the key names in column one. The remaining three columns indicate the key designations on the keyboards for the three different types of terminals that allow key definitions. All definable keys on VT52 terminals are located on the numeric keypad. On VT100-type terminals, you can define the LEFT and RIGHT arrow keys as well as all the keys on the numeric keypad. There are three types of keys that can be defined on terminals with LK201 keyboards: (1) keys on the numeric keypad, (2) keys on the editing keypad (except the UP and DOWN arrow keys), and (3) keys on the function key row across the top of the terminal. (Note that you cannot define function keys F1 through F5.)

Key-name	LK201	VT100-type	VT52
PF1	PF1	PF1	[blue]
PF2	PF2	PF2	[red]
PF3	PF3	PF3	[gray]
PF4	PF4	PF4	- -
KP0, KP1, ..., KP9	0, 1, ..., 9	0, 1, ..., 9	0, 1, ..., 9
PERIOD	.	.	.
COMMA	,	,	n/a
MINUS	-	-	n/a
ENTER	Enter	ENTER	ENTER
LEFT	←	←	←
RIGHT	→	→	→
Find (E1)	Find	- -	- -
Insert Here (E2)	Insert Here	- -	- -
Remove (E3)	Remove	- -	- -
Select (E4)	Select	- -	- -
Prev Screen (E5)	Prev Screen	- -	- -
Next Screen (E6)	Next Screen	- -	- -

Key-name	LK201	VT100-type	VT52
HELP	Help	- -	- -
DO	Do	- -	- -
F6, F7, ..., F20	F6, F7, ..., F20	- -	- -

Some defineable keys are enabled for definition all the time. Others, including KP0 through KP9, PERIOD, COMMA, and MINUS, need to be enabled for definition purposes. You must issue either the SET TERMINAL /APPLICATION or SET TERMINAL/NONUMERIC command before using these keys.

The LEFT and RIGHT arrow keys and the F6 through F14 VT200 keys are reserved for command line editing. You must issue the SET TERMINAL /NOLINE_EDITING command before defining these keys. You can also press CTRL/V to enable keys F7 through F14. Note that CTRL/V will not enable the F6 key.

equivalence-string

Specifies the string which is to be processed when you press the key. If the string contains any spaces, enclose the equivalence string in quotation marks.

DESCRIPTION

The DEFINE/KEY command enables you to assign definitions to the peripheral keys on certain terminals. The terminals include VT52s, the VT100 series, and terminals with LK201 keyboards.

To define keys on the numeric keypads of these terminals, you must first issue the SET TERMINAL/APPLICATION or SET TERMINAL/NONUMERIC command. When your terminal has this setting, the system interprets the keystrokes from keypad keys differently. For example, with SET TERMINAL /NONUMERIC in effect, pressing the 1 key on the keypad does not send the character "1" to the system.

The equivalence string definition can contain different types of information. Definitions often consist of DCL commands. For example, you can assign SHOW TIME to the zero key. When you press 0, the system displays the current date and time. Other definitions can consist of text strings that are to be appended to command lines. When you define a key to insert a text string, use the /NOTERMINATE qualifier so that you can continue typing more data after the string has been inserted.

In most instances you will want to take advantage of the echo feature. The default setting is /ECHO. With /ECHO set, the key definition is displayed on the screen each time you press the key.

You can use the /STATE qualifier to increase the number of key definitions available on your terminal. The same key can be assigned any number of definitions, as long as each definition is associated with a different state. State names can contain any alphanumeric characters, dollar signs, and underscores. Generally, you will want to create a state name that is easy to remember and type and, if possible, one that might remind you of the types of definitions you created for that state. For example, you can create a state called SETSHOW. The key definitions for this state might all refer to various DCL SET and SHOW commands. If you are used to the EDT Editor, you might define a state as GOLD. Then, using the /IF_STATE qualifier, you can assign different definitions to keys used in combination with a key defined as GOLD.

You can use the SET KEY command to change the keypad state. Use the SHOW KEY command to display key definitions and states.

QUALIFIERS

/ECHO (default)
/NOECHO

Determines whether the equivalence string is displayed on your screen after the key has been pressed. You cannot use /NOECHO with the /NOTERMINATE qualifier.

/ERASE
/NOERASE (default)

Determines whether the current line is erased before the key translation is inserted.

/IF_STATE=(state-name,...)
/NOIF_STATE

Specifies a list of one or more states, one of which must be in effect for the key definition to be in effect. If you omit the /IF_STATE qualifier or use /NOIF_STATE, the current state is used. The state name is an alphanumeric string. States are established with the /SET_STATE qualifier or the SET KEY command. If you specify only one state name, you can omit the parentheses. By including several state names, you can define a key to have the same function in all the specified states.

/LOCK_STATE
/NOLOCK_STATE (default)

Specifies that the state set by the /SET_STATE qualifier remain in effect until explicitly changed. If you use the /NOLOCK_STATE qualifier, the state set by /SET_STATE is in effect only for the next defineable key that you press or for the next read terminating character that you type.

The /LOCK_STATE qualifier can only be specified with the /SET_STATE qualifier.

/LOG (default)
/NOLOG

Controls whether the system displays a message indicating that the key definition has been successfully created.

/SET_STATE=state-name
/NOSET_STATE (default)

Causes the specified state-name to be set when the key is pressed. The state name can be any alphanumeric string.

If you omit the SET_STATE qualifier or use /NOSET_STATE, the current state that was locked remains in effect. If you have not included this qualifier with a key definition, you can use the SET KEY command to change the current state.

/TERMINATE
/NOTERMINATE (default)

Specifies whether the current equivalence string is to be terminated (that is, processed) when the key is pressed. Pressing RETURN has the same effect as using /TERMINATE.

The /NOTERMINATE qualifier allows you to create key definitions that insert text into command lines, after prompts, or into other text that you are typing.

EXAMPLES

1
```
$ DEFINE/KEY PF3 "SHOW TIME" /TERMINATE
%DCL-I-DEFKEY, DEFAULT key PF3 has been defined
$ SHOW TIME
  15-APR-1984 14:43:59
```

The DEFINE/KEY command defines the PF3 key on the keypad to perform the SHOW TIME command. DEFAULT refers to the default state.

2
```
$ DEFINE/KEY PF1 "SHOW " /SET_STATE=GOLD/NOTERMINATE/ECHO
%DCL-I-DEFKEY, DEFAULT key PF1 has been defined
$ DEFINE/KEY PF1 " DEFAULT" /TERMINATE/IF_STATE=GOLD/ECHO
%DCL-I-DEFKEY, GOLD key PF1 has been defined
$ SHOW DEFAULT
DISK1:[JOHN.TEST]
```

The first DEFINE/KEY command defines the PF1 key to be the string SHOW. The state is set to GOLD for the subsequent key. The /NOTERMINATE qualifier instructs the system not to process the string when the key is pressed. The second DEFINE/KEY command defines the use of the PF1 key when the keypad is in the GOLD state. When the keypad is in the GOLD state, pressing PF1 will cause the current read to be terminated.

If you press the PF1 key twice, the system displays and processes the SHOW DEFAULT command.

The word DEFAULT in the second line of the example refers to the fact that the key PF1 has been defined in the default state. Note the space before the word DEFAULT in the second DEFINE/KEY command. If the space is omitted, the system fails to recognize DEFAULT as the keyword for the SHOW command.

3
```
$ SET KEY/STATE=ONE
%DCL-I-SETKEY, keypad state has been set to ONE
$DEFINE/KEY PF1 "ONE"
%DCL-I-DEFKEY, ONE key PF1 has been defined
$DEFINE/KEY/IF_STATE=ONE PF1 "ONE"
%DCL-I-DEFKEY, ONE key PF1 has been defined
```

The above two examples both define the PF1 key to be "ONE" for state ONE. However, the first method is somewhat error-prone, in that it is possible to forget which state you are in and to define a key for a state other than you intended. The second example eliminates this possibility of error by specifying the state in the same command as the key definition. This is the preferred method for defining keys.

DELETE

Deletes one or more files from a mass storage disk volume.

FORMAT

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

restrictions

None.

PARAMETER

file-spec[,...]

Specifies the names of one or more files to be deleted from a mass storage disk volume. The first file specification must contain an explicit or default directory specification plus an explicit file name, file type, and version number; subsequent file specifications must contain at least a version number. You can use wildcard characters in any of the file specification fields.

If you omit the directory specification or device name, the current default device and directory are assumed.

A semicolon followed by no file version number, by a version number of 0, or by one or more spaces in the version number of a file specification results in the deletion of the latest version of the file.

To delete more than one file, separate the file specifications with commas or plus signs.

QUALIFIERS

/BACKUP

Selects files according to the dates of their most recent backup. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /BACKUP qualifier is incompatible with /CREATED, /EXPIRED, and /MODIFIED. /CREATED is the default.

/BEFORE[=time]

Selects only those files that are dated before the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/BY_OWNER[=uic]

Selects one or more files only if their owner user identification code (UIC) matches the specified owner UIC.

Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

If the /BY_OWNER qualifier is specified without a UIC, the UIC of the current process is assumed.

/CONFIRM
/NOCONFIRM (default)

Controls whether a request is issued before each individual DELETE operation to confirm that the operation should be performed on that file.

When the system issues the prompt, you can issue any of the following responses:

YES	NO	QUIT
TRUE	FALSE	CTRL/Z
1	0	ALL
	<RET>	

You can use any combination of upper- and lowercase letters for word responses. Word responses can be abbreviated to one or more letters (for example, T, TR, or TRU for TRUE). Affirmative answers are YES, TRUE, and 1. Negative answers are NO, FALSE, 0, and <RET> . QUIT or CTRL/Z indicates that you want to stop processing the command at that point. When you respond with 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, the prompt will be reissued.

/CREATED (default)

Selects files based on their dates of creation. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /CREATED qualifier is incompatible with /BACKUP, /EXPIRED, and /MODIFIED.

/ERASE
/NOERASE (default)

When you simply 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. With the /ERASE qualifier, the storage location is overwritten with a system specified pattern so that the data no longer exists.

/EXCLUDE=(file-spec[,...])

Any files that match the listed file specifications are excluded from the DELETE operation. If you specify only one file, you can omit the parentheses.

The file specification can contain a directory specification, but you cannot include the device in the file specifications you supply with the /EXCLUDE qualifier.

Wildcard characters are supported for file specifications. However, you cannot use relative version numbers to exclude a specific version.

/EXPIRED

Selects files according to the dates on which they will expire. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /EXPIRED qualifier is incompatible with /BACKUP, /CREATED, and /MODIFIED. /CREATED is the default.

/LOG
/NOLOG (default)

Controls whether the DELETE command displays the file specification of each file after its deletion.

/MODIFIED

Selects files according to the dates on which they were last modified. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /MODIFIED qualifier is incompatible with /BACKUP, /CREATED, and /EXPIRED. /CREATED is the default.

/SINCE[=time]

Selects only those files that are dated after the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

EXAMPLES

1

```
$ DELETE   COMMON.SUM;2
```

The DELETE command deletes the file COMMON.SUM;2 from the current default disk and directory.

2

```
$ DELETE *.OLD;*
```

The DELETE command deletes all versions of files with file type OLD from the default disk directory.

3

```
$ DELETE  ALPHA.TXT;*, BETA;*, GAMMA;*
```

The DELETE command deletes all versions of the files ALPHA.TXT, BETA.TXT, and GAMMA.TXT. The command uses the file type of the first input file as a temporary default. Note, however, that some form of version number (here specified as wildcards) must be included in each file specification.

4

```
$ DELETE /BEFORE=15-APR/LOG *.DAT;*
%DELETE-I-FILDEL, DISK2:[MALCOLM]ASSIGN.DAT;1 deleted (5 block)
%DELETE-I-FILDEL, DISK2:[MALCOLM]BATCHAVE.DAT;3 deleted (4 blocks)
%DELETE-I-FILDEL, DISK2:[MALCOLM]BATCHAVE.DAT;2 deleted (4 blocks)
%DELETE-I-FILDEL, DISK2:[MALCOLM]BATCHAVE.DAT;1 deleted (4 blocks)
%DELETE-I-FILDEL, DISK2:[MALCOLM]CANCEL.DAT;1 deleted (2 blocks)
%DELETE-I-FILDEL, DISK2:[MALCOLM]DEFINE.DAT;1 deleted (3 blocks)
%DELETE-I-FILDEL, DISK2:[MALCOLM]EXIT.DAT;1 deleted (1 block)
%DELETE-I-TOTAL, 7 files deleted (23 blocks)
```

The DELETE command deletes all versions of all files with file type DAT that were either created or updated before April 15 of this year. The /LOG qualifier not only displays the name of each file deleted, but also the total number of files deleted.

5

```
$ DELETE A.B;
```

The DELETE command deletes the file A.B with the highest version number.

6
```
$ DELETE/CONFIRM/SINCE=TODAY [MALCOLM.TESTFILES]*.OBJ;*
DISKO:[MALCOLM.TESTFILES]AVERAG.OBJ;1, delete? [N]:Y
DISKO:[MALCOLM.TESTFILES]SCANLINE.OBJ;4, delete? [N]:N
DISKO:[MALCOLM.TESTFILES]SCANLINE.OBJ;3, delete? [N]:N
DISKO:[MALCOLM.TESTFILES]SCANLINE.OBJ;2, delete? [N]:N
DISKO:[MALCOLM.TESTFILES]WEATHER.OBJ;3, delete? [N]:Y
```

The DELETE command examines all versions of files with file type OBJ in the subdirectory [MALCOLM.TESTFILES], and locates those that were created or modified today. Before deleting each file, it requests confirmation that the file should be deleted. The default response—N—is given in square brackets.

7
```
$ DIRECTORY [.SUBTEST]
%DIRECT-W-NOFILES, no files found
$ SET PROTECTION SUBTEST.DIR/PROTECTION=OWNER:D
$ DELETE SUBTEST.DIR;1
```

Before the directory file SUBTEST.DIR is deleted, the DIRECTORY command is used to verify that there are no files cataloged in the directory. The SET PROTECTION command redefines the protection for the directory file so that it can be deleted; then the DELETE command deletes it.

8
```
$ DELETE DALLAS"THOMAS SECRET"::DISKO:[000,000]DECODE.LIS;1
```

This DELETE command deletes the file DECODE.LIS;1 from the directory [000,000] on device DISK0 at remote node DALLAS. The user name and password follow the remote node name.

9
```
$ DELETE QUEBEC::"DISK1:DEAL.BIG"
$ DELETE QUEBEC::DISK1:DEAL.BIG;
```

Either of these DELETE commands can be used to delete the file DEAL.BIG on device ZZZ1 at remote node QUEBEC. Note that the DELETE command requires an explicit version number in a file specification, but the file to be deleted is on a remote node whose file syntax does not recognize version numbers. (QUEBEC is an RT–11 node.) Therefore, the file specification must either be enclosed in quotation marks or entered with a null version number (that is, a trailing semicolon).

DELETE/CHARACTERISTIC

Deletes the definition of a queue characteristic that was previously established with the DEFINE/CHARACTERISTIC command. The /CHARACTERISTIC qualifier is required.

FORMAT

DELETE/CHARACTERISTIC *characteristic-name*

restrictions

Requires operator (OPER) privilege.

PARAMETER

characteristic-name

Specifies the name that was assigned to the characteristic by a DEFINE /CHARACTERISTIC command.

DESCRIPTION

You can use this command to delete a characteristic from the system characteristic table.

Note that if you want to change the number of an existing characteristic, you can use the DEFINE/CHARACTERISTIC command. It is not necessary to delete the characteristic before changing it.

EXAMPLE

```
$ DEFINE/CHARACTERISTIC BLUE 7
   .
   .
   .
$ DELETE/CHARACTERISTIC BLUE
$ DEFINE/CHARACTERISTIC BLUE_INK 7
```

The DEFINE/CHARACTERISTIC command establishes the characteristic BLUE, with number 7, to mean blue ink ribbons for printers. Later you decide to change the name of the characteristic. First, issue the DELETE/CHARACTERISTIC command. Then, issue another DEFINE /CHARACTERISTIC command to rename the characteristic to BLUE_INK, using the characteristic number 7.

DELETE/ENTRY

Deletes one or more job entries from a queue. The /ENTRY qualifier is required.

FORMAT

DELETE/ENTRY=*(entry-number[,...]) queue-name[:]*

restrictions

To delete a job from a queue, you must have operator (OPER) privilege, execute (E) access to the specified queue, or delete (D) access to the specified job.

PARAMETERS

entry-number[,...]

Specifies the entry number of a job to be deleted from the queue. The /ENTRY qualifier requires at least one entry-number parameter, specifying one or more jobs to be deleted from a single printer or batch queue. If you specify only one entry number, you can omit the parentheses.

queue-name[:]

Specifies the name of the queue where the jobs are located. The queue name can refer either to the queue to which the job was submitted or to the queue where the job is executing.

DESCRIPTION

The DELETE/ENTRY command deletes one or more jobs from a queue. If you specify more than one entry number with a DELETE/ENTRY command, all the jobs must be located in the same queue.

You can delete jobs that are currently executing, as well as jobs that are in other states. For example, DELETE/ENTRY can stop a job that is currently printing.

EXAMPLES

1

```
$ PRINT/HOLD   ALPHA.TXT
Job ALPHA (queue SYS$PRINT, entry 110) holding
   .
   .
   .
$ DELETE/ENTRY=110  SYS$PRINT
```

The PRINT command queues a copy of the file ALPHA.TXT in a HOLD status, to defer its printing until later. The system displays the job name, entry number, name of the queue in which the job was entered, and the status. Later, the DELETE/ENTRY command requests that the entry be deleted from the queue SYS$PRINT.

2
```
$ SUBMIT/HOLD/PARAMETERS=SCANLINE  DOFOR
  Job DOFOR (queue SYS$BATCH, entry 203) holding
$ SUBMIT/AFTER=18:00  WEATHER
  Job WEATHER (queue SYS$BATCH, entry 210) holding until 15_APR-1984 18:00
    .
    .
    .
$ DELETE/ENTRY=(203,210)   SYS$BATCH
```

The SUBMIT commands queue the command procedures DOFOR.COM and WEATHER.COM for processing as batch jobs. DOFOR.COM is queued in a HOLD status and cannot execute until you issue a SET QUEUE/ENTRY /RELEASE command. WEATHER.COM is queued for execution after 6:00 P.M. Later, the DELETE/ENTRY command requests that both these entries be deleted from the queue SYS$BATCH.

3
```
$ PRINT CHAPTER8.MEM
  Job CHAPTER8 (queue SYS$PRINT, entry 25) pending on queue LPA0
    .
    .
    .
$ SHOW QUEUE SYS$PRINT
Printer queue SYS$PRINT, on LPA0:
Jobname         Username        Entry   Blocks  Status
-------         --------        -----   ------  ------
CHAPTER7        SMITH              24      274  Pending
CHAPTER8        SMITH              25      976  Pending
$ DELETE/ENTRY=25 SYS$PRINT
```

The PRINT command submits the file CHAPTER8.MEM to the generic printer queue SYS$PRINT. Later, user Smith needs to edit the file again before printing it. Using the SHOW QUEUE command, Smith verifies that the job is still pending and that the entry number for the job is 25, and then issues the DELETE/ENTRY command to delete the job from the queue.

DELETE/FORM

Deletes the definition of a form type for printer or terminal queues that was previously established with the DEFINE/FORM command. The /FORM qualifier is required.

FORMAT

DELETE/FORM *form-name*

restrictions

Requires operator (OPER) privilege.

PARAMETER

form-name

Specifies the name that was assigned to the form by a DEFINE/FORM command.

DESCRIPTION

You can use this command to delete a form definition from the system forms table. When you delete a form, there can be no outstanding references to the form either in queues that have been mounted with the form or by jobs requesting that form. Use the SHOW QUEUE/FULL qualifier to locate all references to the form.

If you want to change the number or attributes of an existing form, you can use the DEFINE/FORM command. It is not necessary to delete the form before changing it.

EXAMPLE

```
$ DEFINE/FORM /DESCRIPTION="letter size continuous form paper" CFLET 7
    .
    .
    .
$ DELETE/FORM CFLET
$ DEFINE/FORM /DESCRIPTION="letter size continuous form paper" LETTER_CONT 7
```

The DEFINE/FORM command establishes the form CFLET with number 7, to mean 8.5 x 11 inch continuous form feed paper. Later, you decide to change the name of the form. To do this, delete the form named CFLET and define a new one named LETTER_CONT.

DELETE/INTRUSION_RECORD

Removes an entry from the breakin database.

FORMAT

DELETE/INTRUSION_RECORD *source*

restrictions

Requires CMKRNL and SECURITY privileges.

PARAMETERS

source

Source field of the entry to be removed from the breakin database.

DESCRIPTION

If you determine that an entry in the breakin database resulted from a user error and not a breakin attempt, you can remove an entry from the breakin database with the DELETE/INTRUSION command. For example, if the user Hammer repeatedly attempted to log in on terminal TTA24 with an expired password, the SHOW INTRUSION command would display the following entry:

```
Intrusion   Type       Count    Expiration     Source
TERM_USER   INTRUDER     9      10:29:39.16    TTA24:HAMMER
```

Since the login failure limit has been reached, the terminal is locked out of the system. When Hammer approaches you and you identify the problem as an expired password, you can then use the DELETE/INTRUSION command to remove the record from the breakin database.

EXAMPLES

1

```
$ DELETE/INTRUSION_RECORD TTC2:
```

This command removes all intrusion records generated by breakin attempts on TTC2. No username is specified because none of the login failures occurred for valid users.

2

```
$ DELETE/INTRUSION_RECORD GALAXY::HAMMER
```

This command removes all intrusion entries generated from node GALAXY for user HAMMER.

DELETE/KEY

Deletes key definitions that have been established by the DEFINE /KEY command. The /KEY qualifier is required.

FORMAT

DELETE/KEY *[key-name]*

restrictions

None.

PARAMETER

key-name

Specifies the name of the key whose definition you want to delete.

QUALIFIERS

/ALL

Specifies that all key definitions in the specified state be deleted. If you use the /ALL qualifier, do not specify a key name. If no state is specified, all key definitions in the current state are deleted. Use the /STATE qualifier to specify one or more states.

/LOG (default)
/NOLOG

Controls whether messages are displayed indicating that the specified key definitions have been deleted.

/STATE=(state-name[,...])
/NOSTATE (default)

Specifies the names of states for which the specified key definitions are to be deleted. If you specify only one state name, you can omit the parentheses. State names can be any appropriate alphanumeric string.

If you omit the /STATE qualifier or use /NOSTATE, key definitions in the current state are deleted.

EXAMPLES

1

```
$ DEFINE/KEY PF3 "SHOW TIME" /TERMINATE
%DCL-I-DEFKEY, DEFAULT key PF3 has been defined
(PF3)
$ SHOW TIME
 15-APR-1985 14:43:59
   .
   .
   .
$ DELETE/KEY PF3
%/DCL-I-DELKEY, DEFAULT key PF3 has been deleted
(PF3)
$
```

The DEFINE/KEY command defines the PF3 key on the keypad to perform the SHOW TIME command. Later, you use the DELETE/KEY command so

that the PF3 key no longer has a definition. Now when you press that key, only the system prompt appears.

```
2   $ DELETE/KEY/ALL
    %DCL-I-DELKEY, DEFAULT key PF1 has been deleted
    %DCL-I-DELKEY, DEFAULT key PF2 has been deleted
    %DCL-I-DELKEY, DEFAULT key PF3 has been deleted
    %DCL-I-DELKEY, DEFAULT key PF4 has been deleted
    $
```

This example assumes that you had defined keys PF1 through PF4 in the default state. The DELETE/KEY command deletes all key definitions in the current state, which is the default state.

DELETE/QUEUE

Deletes the specified queue. The /QUEUE qualifier is required.

FORMAT

DELETE/QUEUE *queue-name[:]*

restrictions

Requires operator (OPER) privilege.

PARAMETER

queue-name[:]

Specifies the name of the queue to be deleted.

DESCRIPTION

The DELETE/QUEUE command takes effect only if the specified queue has been stopped. In order to delete a queue, you must do the following:

1. Be sure that there are no outstanding references to the queue.
2. Stop the queue with the STOP/QUEUE/NEXT command.

Queue references are made with the /PRINTER=queue-name and /GENERIC=queue-name qualifiers of the INITIALIZE/QUEUE, SET QUEUE, and START/QUEUE commands. The ASSIGN/QUEUE command also makes queue references.

After you have determined that there are no outstanding references to the queue, stop the queue with the STOP/QUEUE/NEXT command. Wait for any current jobs to complete. You are now ready to issue the DELETE /QUEUE command. Note that any pending jobs in the queue will be deleted when the queue is deleted.

EXAMPLE

```
$ INITIALIZE/QUEUE/DEFAULT=FLAG/START LPA0
    .
    .
    .
$ STOP/QUEUE/NEXT LPA0
$ DELETE/QUEUE LPA0
```

The first command initializes and starts the printer queue LPA0. Later, when you decide to delete that queue, first stop the queue with the STOP/QUEUE /NEXT command. Then delete the queue by issuing the DELETE/QUEUE command.

DELETE/SYMBOL

Deletes a symbol definition from a local or global symbol table or deletes all symbol definitions in a symbol table. The /SYMBOL qualifier is required.

FORMAT

DELETE/SYMBOL *[symbol-name]*

restrictions

None.

PARAMETER

symbol-name

Specifies the name of the symbol to be deleted. Symbol names can have from 1 to 255 characters. By default, the DELETE/SYMBOL command assumes that the symbol is in the local symbol table for the current command procedure.

The symbol-name parameter is required unless /ALL is specified.

DESCRIPTION

The DELETE/SYMBOL command deletes a symbol definition from a symbol table. If you do not specify either the global or local symbol table, the symbol is deleted from the local table. If you specify both /GLOBAL and /LOCAL, only the last specified qualifier is accepted. The /SYMBOL qualifier must always follow immediately after the DELETE command name.

QUALIFIERS

/ALL

Specifies that all symbol names in the specified symbol table be deleted. If you do not specify either /LOCAL or /GLOBAL, all symbols defined at the current command level are deleted.

/GLOBAL

Indicates that the symbol name is to be deleted from the global symbol table of the current process.

/LOCAL (default)

Indicates that the symbol name is to be deleted from the local symbol table of the current command level.

/LOG
/NOLOG (default)

Controls whether an informational message listing each symbol being deleted is displayed.

EXAMPLES

1
```
$ DELETE/SYMBOL/ALL
```

The DELETE/SYMBOL command deletes all symbol definitions at the current command level.

2
```
$ DELETE/SYMBOL/LOG FOO
%DCL-I-DELSYM, LOCAL symbol FOO has been deleted
```

The DELETE/SYMBOL command deletes the symbol FOO from the local symbol table for the current process. In addition, the /LOG qualifier causes an informational message, listing the symbol being deleted, to be displayed.

3
```
$ DELETE/SYMBOL/GLOBAL PDEL
```

The DELETE/SYMBOL command deletes the symbol named PDEL from the global symbol table for the current process.

DEPOSIT

Replaces the contents of a specified location or series of locations in virtual memory.

The DEPOSIT command, together with the EXAMINE command, aids in debugging programs interactively. The DCL DEPOSIT command is similar to the DEPOSIT command of the VAX/VMS Debugger.

FORMAT

DEPOSIT *location=data[,...]*

restrictions

Requires user mode R (read) and W (write) access to the location in virtual memory whose contents you want to change.

PARAMETERS

location

Specifies the starting virtual address of a location or series of locations whose contents are to be changed.

The specified location must be within the virtual address space of the image currently running in the process. In addition, you must have user mode R (read) and W (write) access to the location.

You can specify the location using any valid integer expression. The expression can consist of an integer value, a symbol name, a lexical function, or a combination of these entities.

The DEPOSIT and EXAMINE commands maintain a pointer to a current memory location. The DEPOSIT command sets this pointer to the byte following the last byte modified; you can refer to this pointer by using a period (.) in subsequent EXAMINE and DEPOSIT commands. If the DEPOSIT command cannot deposit the specified data, the pointer does not change. The EXAMINE command does not change the value of the pointer.

data[,...]

Defines the data to be deposited into the specified locations. If you specify a list, separate the items with commas; the DEPOSIT command writes the data in consecutive locations, beginning with the address specified.

By default, the data is assumed to be in hexadecimal format; the DEPOSIT command converts the data to binary format before writing it into the specified location.

When non-ASCII data is deposited, you can specify each item of data using any valid integer expression.

When ASCII data is deposited, only one item of data is allowed. All characters to the right of the equal sign are considered to be part of a single string. The characters are converted to uppercase, and all spaces are compressed.

DEPOSIT

DESCRIPTION

When the DEPOSIT command completes, it displays both the virtual memory address into which data is deposited and the new contents of the location, as follows:

```
address:  contents
```

If the specified address can be read but not written by the current access mode, the DEPOSIT command displays the original contents of the location. If the specified address can be neither read nor written, the DEPOSIT command displays asterisks in the data field.

If you specify a list of numeric values, some but not all of the values may be successfully deposited before an access violation occurs. If an access violation occurs while ASCII data is being deposited, nothing is deposited.

Radix Qualifiers: The radix default for a DEPOSIT or EXAMINE command determines how the command interpreter interprets numeric literals. The initial default radix is hexadecimal; all numeric literals in the command line are assumed to be hexadecimal values. If a radix qualifier modifies the command, that radix becomes the default for subsequent EXAMINE and DEPOSIT commands, until another qualifier overrides it. For example:

```
$ DEPOSIT/DECIMAL 900=256
00000384:  256
```

The DEPOSIT command interprets both the location 900 and the value 256 as decimal. All subsequent DEPOSIT and EXAMINE commands assume that numbers you enter for addresses and data are decimal. Note that the DEPOSIT command always displays the address location in hexadecimal.

Symbol values defined by = (Assignment Statement) commands are always interpreted in the radix in which they were defined.

Note that hexadecimal values entered as deposit locations or as data to be deposited must begin with a numeric character (0 through 9). Otherwise, the command interpreter assumes that you have entered a symbol name and attempts symbol substitution.

You can use the radix operators %X, %D, or %O to override the current default when you enter the DEPOSIT command. For example:

```
$ DEPOSIT/DECIMAL %X900=10
```

This command deposits the decimal value 10 in the location specified as hexadecimal 900.

Length Qualifiers: The initial default length unit for the DEPOSIT command is a longword. If a list of data values is specified, the data is deposited into consecutive longwords beginning at the specified location. If a length qualifier modifies the command, that length becomes the default for subsequent EXAMINE and DEPOSIT commands, until another qualifier overrides it. If you specify data values that are longer than the specified length, an error occurs.

Length qualifiers are ignored when ASCII values are deposited.

Restriction on Placement of Qualifiers: The DEPOSIT command analyzes expressions arithmetically. Therefore, qualifiers, which must be preceded by a slash (/), must appear immediately after the command name to be interpreted correctly.

QUALIFIERS

/ASCII

Indicates that the specified data is ASCII. The DEPOSIT command converts the data to its binary equivalent before placing it in virtual memory.

When you specify ASCII data, the command interpreter compresses multiple spaces to a single space and changes all letters to uppercase before writing the data in memory. To deposit an ASCII string containing consecutive multiple spaces or lowercase characters, enclose the string in quotation marks.

When you specify /ASCII, or when ASCII mode is the default, the location you specify is assumed to be hexadecimal.

/BYTE

Requests that data be deposited one byte at a time.

/DECIMAL

Indicates that the specified data is decimal; the DEPOSIT command converts the data to its binary equivalent before placing it in virtual memory.

/HEXADECIMAL

Indicates that the specified data is hexadecimal. The DEPOSIT command converts the data to its binary equivalent before placing it in virtual memory.

/LONGWORD

Requests that data be deposited a longword at a time.

/OCTAL

Indicates that the specified data is octal; the DEPOSIT command converts the data to its binary equivalent before placing it in virtual memory.

/WORD

Requests that the data be deposited one word at a time.

EXAMPLES

```
1   $ RUN MYPROG
      .
      .
      .
    ^Y
    $ EXAMINE 2780
    00002780:  1C50B344
    $ DEPOSIT .=0
    00002780:  00000000
    $ CONTINUE
```

The RUN command executes the image MYPROG.EXE; subsequently, CTRL/Y interrupts the program. Assuming that the initial defaults of /HEXADECIMAL and /LONGWORD are in effect, the DEPOSIT command places a longword of zeros in virtual memory location 2780.

Since the EXAMINE command sets up a pointer to the current memory location which, in this case, is virtual address 2780, you are able to refer to this location with "." in the DEPOSIT command.

The CONTINUE command resumes execution of the image.

2
```
$ DEPOSIT/ASCII   2C00=FILE: NAME: TYPE:
00002C00:  FILE: NAME: TYPE:...
```

The DEPOSIT command deposits character data at hexadecimal location 2C00 and displays the contents of the location after modifying it. Since the current default length is a longword, the response from the DEPOSIT command displays full longwords. Trailing dots (ellipses) indicate that the remainder of the last longword of data contains information that was not modified by the DEPOSIT command.

3
```
$ EXAMINE 9C0          !  Look at Hex location 9C0
000009C0:  8C037DB3
$ DEPOSIT .=0          !  Deposit longword of 0
000009C0:  00000000
$ DEPOSIT/BYTE .=1     !  Put 1 byte at next location
000009C4:  01
$ DEPOSIT .+2=55       !  Deposit 55 next
000009C7:  55
$ DEPOSIT/LONG .=0C,0D,0E !  Deposit longwords
000009C8:  0000000C 0000000D 0000000E
```

The sequence of DEPOSIT commands in the above example illustrates how the DEPOSIT command changes the current position pointer. Note that after you specify /BYTE, all data is deposited and displayed in bytes, until the /LONGWORD qualifier restores the system default.

4
```
$ BASE=%X200           !  Define a base address
$ LIST=BASE+%X40       !  Define offset from base
$ DEPOSIT/DECIMAL LIST=1,22,333,4444
00000240:  00000001 00000022 00000333 00004444
$ EXAMINE/HEX LIST:LIST+0C !  Display results in hex
00000240:  00000001 00000016 0000014D 0000115C
```

The assignment statements define a base address in hexadecimal and a label at a hexadecimal offset from the base address. The DEPOSIT command reads the list of values and deposits each value into a longword, beginning at the specified location. The EXAMINE command requests a hexadecimal display of these values.

DIFFERENCES

Compares the contents of two disk files and creates a listing of those records that do not match.

FORMAT

DIFFERENCES *master-file-spec [revision-file-spec]*

restrictions

None.

PARAMETERS

master-file-spec

Specifies the name of the primary input file to be compared.

The file specification must include a file name and a file type. No wildcard characters are allowed in the file specification.

revision-file-spec

Specifies the name of the secondary input file to be compared. Any nonspecified fields default to the corresponding fields of the primary input file specification.

If you do not specify a secondary input file, the DIFFERENCES command uses the next lower version of the primary input file.

No wildcard characters are allowed in the file specification.

DESCRIPTION

Use the DIFFERENCES command to find out whether two files are identical and, if not, how they differ. DIFFERENCES compares the two specified files on a record-by-record basis and produces an output file that lists the differences, if any.

The qualifiers for the DIFFERENCES command can be categorized according to their functions, as follows:

- Qualifiers that request DIFFERENCES to ignore data in each record:

  /COMMENT_DELIMITERS
  /IGNORE

  These qualifiers allow you to define characters that denote comments or to designate characters or classes of characters to ignore when comparing files. For example, you can have DIFFERENCES ignore extra blank lines or extra spaces within lines.

  By default, DIFFERENCES compares every character in each record.

- Qualifiers that control the format of the information contained in the list of differences:

  /CHANGE_BAR
  /IGNORE
  /MERGED
  /MODE
  /PARALLEL
  /SEPARATED

/SLP
/WIDTH

By default, DIFFERENCES merges the differences it finds in the files being compared. It lists each record in the file that has no match in the other input file and then lists the next record that it finds that does have a match.

By default, DIFFERENCES also supplies a line number with each listed record, and it lists the records with all designated ignore characters deleted.

You can specify combinations of qualifiers to request an output listing that includes the comparison in more than one format. Note that SLP output is incompatible with all other types of output; PARALLEL output can be generated only in ASCII mode.

- Qualifiers that control the extent of the comparison:

  /MATCH
  /MAXIMUM_DIFFERENCES
  /WINDOW

  By default, DIFFERENCES reads every record in the master input file, and looks for a matching record in the revision input file. A search for a match between the two input files continues until either a match is found or the ends of the two files are reached. Sections of the two files are considered a match only if three sequential records are found to be identical in each file.

By default, DIFFERENCES output is written to the current SYS$OUTPUT device. Use the /OUTPUT qualifier to request DIFFERENCES to write the output to an alternate file or device.

DIFFERENCES terminates with an exit status. The following severity levels indicate the result of the comparison:

SUCCESS	Files are identical.
INFORMATIONAL	Files are different.
WARNING	User-specified maximum number of differences has been exceeded.
ERROR	Insufficient virtual memory to complete comparison.

All statuses other than SUCCESS indicate that the two input files are different.

QUALIFIERS

/CHANGE_BAR[=(format[,...])]

Requests that the output contain a listing of the associated files with a change bar character next to the lines that do not match. You can specify any of the following format keywords:

c	A change bar character.
NUMBER	Include line numbers.
NONUMBER	Do not include line numbers.

In ASCII output mode, the change bar character parameter specifies a one-character code that will appear in the left margin next to records that do not have a match. By default, an exclamation point is used as the change bar character.

In HEXADECIMAL and OCTAL output mode, the change bar character is ignored. Instead, the string "***CHANGE***" appears in the record header.

You can control whether the change bar listing includes line numbers by specifying either NUMBER or NONUMBER. If neither is specified, the default is controlled by the /[NO]NUMBER command qualifier.

To specify both a change bar character and either NUMBER or NONUMBER, separate the options with a comma and enclose the list in parentheses, for example, /CHANGE_BAR=($,NUMBER).

/COMMENT_DELIMITER[=(delimiter[,...])]

Specifies one or more comment delimiters. This qualifier is used with or without the /IGNORE=COMMENTS qualifier to indicate which comments are to be ignored.

You can specify a comment delimiter either by listing the character itself or by using one of the following keywords:

Keyword	Character
COLON	Colon (:)
COMMA	Comma (,)
EXCLAMATION	Exclamation point (!)
FORM_FEED	Form feed
LEFT	Left bracket ([)
RIGHT	Right bracket (])
SEMI_COLON	Semicolon (;)
SLASH	Slash (/)
SPACE	Space
TAB	Horizontal tab

You can specify up to 32 comment characters. If you specify only one delimiter, you can omit the parentheses.

When you specify a keyword, you must type at least the first two letters of the keyword, since single letters are treated as delimiters. Lowercase letters are automatically changed to uppercase unless they are enclosed in quotation marks. If both the uppercase and lowercase forms of a letter are to be used as delimiters, the letter must be specified twice, once in uppercase and once in lowercase. Non-alphanumeric characters (for example, "!" and ",") must be enclosed in quotation marks.

If you do not include either a comment character or a keyword with the /COMMAND_DELIMITER qualifier, DIFFERENCES assumes a default comment character based on the file type. For some file types (COB and FOR), the default comment characters are considered valid delimiters only if they appear in the first column of a line. Multicharacter comment delimiters are not allowed.

File type	Default Comment Character
B2S, B32, BAS, BLI	!
CBL, CMD	! and ;
COB	* or / in first column
COM, COR	!
FOR	! anywhere; C,D,c,d in first column
HLP	!
MAC, MAR	;
R32, REQ	!

/IGNORE=(option[,...])

Specifies that one or more special characters, strings, or records be ignored during the comparison. The /IGNORE qualifier also controls whether the comparison records are output to the listing file exactly as they appeared in the input file or as edited records. The option parameter refers either to a character or a keyword. The following keywords are valid options for the /IGNORE qualifier:

BLANK_LINES	Blank lines between data lines.
COMMENTS	Data that follow a comment delimiter. (Use the /COMMENT_DELIMITER qualifier to designate one or more non-default comment delimiters.)
FORM_FEEDS	Form feed characters.
HEADER[=n]	First n records beginning with a record whose first character is a form feed. (The first record is not ignored if the only character it contains is a form feed.) The default value of n is 2.
TRAILING_SPACES	Space and tab characters at the end of a line of data.
SPACING	Multiple spaces or tabs within lines of data. (Each set of contiguous spaces or tabs is replaced by a single space.)

Each data line is checked for COMMENTS, FORM_FEEDS, HEADER, and SPACING before it is tested for TRAILING_SPACES and then BLANK_LINES. Therefore, if you direct DIFFERENCES to ignore COMMENTS, TRAILING_SPACES, and BLANK_LINES, it will totally ignore a record that contains several spaces or blank lines followed by a comment.

By default, the DIFFERENCES command compares every character in each file and reports all differences. Also, by default, DIFFERENCES lists records in the output file with all ignored characters deleted.

Although output records can contain any characters that appear in the input files, you can choose to have output records formatted as shown in the following table.

Character	Formatted Output
Tab (CTRL/I)	1–8 spaces
RET (CTRL/M)	<CR>
Line feed (CTRL/J)	<LF>
Vertical tab (CTRL/K)	<VT>
Form feed (CTRL/L)	<FF>
Other nonprinting characters	. (period)

If you specify /PARALLEL, output formatting is always performed.

You can choose how you want the records to appear by specifying one of the following keywords with the /IGNORE qualifier:

EDITED	Output records are listed with ignored strings deleted.
EXACT	Output records are listed exactly as they appear in the input file.
PRETTY	Output records are formatted.

/MATCH=size

Specifies the number of records that constitutes a match.

By default, after DIFFERENCES finds unmatched records, it assumes that the files once again match after it finds three sequential records that match. Use the /MATCH qualifier to override the default match size of 3.

You can increase the /MATCH value if you feel that DIFFERENCES is incorrectly matching sections of the master and revision input files after it has detected a difference.

/MAXIMUM_DIFFERENCES=n

Specifies that DIFFERENCES is to terminate after a specified number of unmatched records has been found.

The number of unmatched records is determined by finding the maximum number of difference records for each difference section and adding them together.

If DIFFERENCES reaches the maximum number of differences that you specify, it will output only those records that were detected before the maximum was reached. Also, it will output, at most, one listing format and return a warning message.

By default, there is no maximum number of differences. All records in the specified input files are compared.

/MERGED[=n]

Requests that the output file contain a merged list of differences. The value n indicates the number of matched records to list after each list of unmatched records; it should be a decimal number less than or equal to the size of a match (see the /MATCH=size qualifier).

By default, DIFFERENCES produces a merged listing with one matched record listed after each set of unmatched records (that is, /MERGED=1).

Use the /MERGED qualifier to override the default value of n, or to include a merged listing with other types of output.

/MODE=(radix[,...])

Specifies the format of the output listing. You can request that the output be formatted in one or more radix modes by specifying the following keywords:

ASCII
HEXADECIMAL
OCTAL

Enclose the keywords in parentheses. You can truncate any of these keywords to one or more characters.

By default, DIFFERENCES writes the output file in ASCII. If you specify more than one radix, the output listing contains the file comparison in each specified radix. When you specify two or more radix modes, separate them with commas.

If you specify the /PARALLEL or /SLP qualifier, the /MODE qualifier is ignored for that listing form.

/NUMBER (default)
/NONUMBER

Controls whether line numbers are to be generated for records in the listing.

/OUTPUT[=file-spec]

Defines an output file to receive the output difference list. If you omit the /OUTPUT qualifier, the output is written to the current SYS$OUTPUT device. If you use the /OUTPUT qualifier without a file specification, the output is directed to a file with the same name as the master file with the file type DIF.

When you specify /OUTPUT, you can control the defaults applied to the output file specification as described in Section 2.4 of the *VAX/VMS DCL Concepts Manual*. The default output file type is DIF.

No wildcard characters are allowed in the file specification.

/PARALLEL[=n]

Requests that the output file contain a parallel list of differences. The value n indicates the number of matched records to output after each list of unmatched records; it should be a decimal number less than or equal to the size of the match (see the /MATCH=size qualifier).

By default, DIFFERENCES does not list records after each list of unmatched records. Also by default, DIFFERENCES creates only a list of merged differences.

/SEPARATED[=(input-file[,...])]

Requests that the output file contain sequential lists of unmatched records from the specified input files. You can designate the desired input files by specifying the following keywords:

MASTER
REVISION

If you specify only one input file, you can omit the parentheses.

If you specify the /SEPARATED qualifier without designating any input files, DIFFERENCES will generate a separate listing for each input file.

By default, DIFFERENCES creates only a merged list of differences.

/SLP

Requests that DIFFERENCES produce an output file suitable for input to the SLP editor. If you use the /SLP qualifier, you cannot specify any of the following output file qualifiers: /MERGED, /PARALLEL, /SEPARATED, or /CHANGE_BAR.

Use the output file produced by the SLP qualifier as input to SLP to update the master input file, that is, to make the master input file match the revision input file.

When you specify /SLP and you do not specify /OUTPUT, DIFFERENCES writes the output file to a file with the same file name as the master input file with the file type DIF.

/WIDTH=n

Specifies the width of lines in the output listing.

By default, output is 132 characters wide, unless output is directed to the terminal. In that case, the output line width is controlled by the terminal line width. Use the SET TERMINAL command to change the terminal line width.

/WINDOW=size

Controls the number of records to search before listing a record as unmatched and continuing with the next record in the master input file.

The window size is the minimum size of a differences section that will cause DIFFERENCES to lose synchronization between the two input files.

By default, DIFFERENCES searches to the ends of both input files before listing a record as unmatched.

EXAMPLES

1

```
$ DIFFERENCES EXAMPLE.TXT
************
File DISK1:[GEORGE.TEXT]EXAMPLE.TXT;2
    1   DEMONSTRATION
    2   OF V3.0 DIFFERENCES
    3   UTILITY
******
File DISK1:[GEORGE.TEXT]EXAMPLE.TXT;1
    1   DEMONSTRATION
    2   OF VMS DIFFERENCES
    3   UTILITY
************
Number of difference sections found: 1
Number of difference records found: 2
DIFFERENCES/MERGED=1-
       DISK1:[GEORGE.TEXT]EXAMPLE.TXT;2
       DISK1:[GEORGE.TEXT]EXAMPLE.TXT;1
```

The DIFFERENCES command compares the contents of the two most recent versions of the file EXAMPLE.TXT in the current default directory. DIFFERENCES compares every character in every record and displays the results at the terminal.

```
2   $ DIFFERENCES/PARALLEL/WIDTH=80/COMMENT_DELIMITER="V" EXAMPLE.TXT
    ------------------------------------------------------------------------------
    File DISK1:[GEORGE.TEXT]EXAMPLE.TXT;2  |  File DISK1:[GEORGE.TEXT]EXAMPLE.TXT;1
    ------------------- 1 ------------------------------------ 1 -----------------
    DEMONSTRATION                          |  <LF>DEMONSTRATION
    ------------------------------------------------------------------------------

    Number of difference sections found: 1
    Number of difference records found: 1
    DIFFERENCES/IGNORE=(COMMENTS)/COMMENT_DELIMITER=("V")/WIDTH=80/PARALLEL-
           DISK1:[GEORGE.TEXT]EXAMPLE.TXT;2-
           DISK1:[GEORGE.TEXT]EXAMPLE.TXT;1
```

The DIFFERENCES command compares the same files as in Example 1, but ignores all comments following the first "V" encountered by DIFFERENCES. The command also specifies that an 80-column parallel list of differences be displayed.

```
3   $ DIFFERENCES/WIDTH=80/MODE=(HEX,ASCII) EXAMPLE.TXT/CHANGE_BAR
    ************
    File DISK1:[GEORGE.TEXT]EXAMPLE.TXT;2
        1 ! DEMONSTRATION
        2 ! OF V3.0 DIFFERENCES
        3   UTILITY
    ************
    ************
    File DISK1:[GEORGE.TEXT]EXAMPLE.TXT;2
    RECORD NUMBER 1 (00000001) LENGTH 14 (0000000E)  ***CHANGE***
         204E 4F495441 5254534E 4F4D4544 DEMONSTRATION .. 000000
    RECORD NUMBER 2 (00000002) LENGTH 19 (00000013)  ***CHANGE***
     4E455245 46464944 20302E33 5620464F OF V3.0 DIFFEREN 000000
                               534543 CES............. 000010
    RECORD NUMBER 3 (00000003) LENGTH 7 (00000007)
                          595449 4C495455 UTILITY..... 000000
    ************
    Number of difference sections found: 1
    Number of difference records found: 2
    DIFFERENCES /WIDTH=80/MODE=(HEX,ASCII)
           DISK1:[GEORGE.TEXT]EXAMPLE.TXT;2/CHANGE_BAR-
           DISK1:[GEORGE.TEXT]EXAMPLE.TXT;1
```

The DIFFERENCES command compares the same files as in Example 1, but lists the differences in both hexadecimal and ASCII formats. The command also specifies that default change bars be used in the output. The default change bar notation for the hexadecimal output is ***CHANGE***. For the ASCII output, the default change bar character is the exclamation point.

```
4   $ DIFFERENCES/OUTPUT BOSTON::DISK2:TEST.DAT OMAHA::DISK1:[PGM]TEST.DAT
```

The DIFFERENCES command compares two remote files and displays any differences found. The first file is TEST.DAT on remote node BOSTON. The second file is also named TEST.DAT on remote node OMAHA. The DIFFERENCES output is located in the file DISK1:[PGM]TEST.DIF.

DIRECTORY

Provides a list of files or information about a file or group of files.

FORMAT

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

restrictions

Requires read (R) access to the directories or sufficient privilege to override the protection to obtain information other than the file name.

PARAMETER

file-spec[,...]

Specifies one or more files to be listed. The syntax of a file specification determines which files will be listed, as follows:

- If you do not enter a file specification, the DIRECTORY command lists all versions of the files in the current default directory.
- If you specify only a device name, the DIRECTORY command uses your default directory specification.
- Whenever the file specification does not include a file name and file type, all versions of all files in the specified directory are listed.
- If a file specification contains a file name and/or file type and no version number, the DIRECTORY command lists all versions.
- If a file specification contains only a file name, the DIRECTORY command lists all files in the current default directory with that name, regardless of file type and version number.
- If a file specification contains only a file type, the DIRECTORY command lists all files in the current default directory with that file type, regardless of file name and version number.

If you specify more than one file, separate the file specifications with either commas or plus signs. You can use wildcard characters in the directory specification, file name, file type, or version number fields of a file specification to list all files that satisfy the components you specify.

DESCRIPTION

The DIRECTORY command lists the files contained in a directory. When you use certain qualifiers with the command, additional information is displayed, along with the names of the files.

The output of the DIRECTORY command depends on certain formatting qualifiers and their defaults. These qualifiers are: /COLUMNS, /DATE, /FULL, /OWNER, /PROTECTION, and /SIZE. However, the files that are listed always appear in alphabetical order, with the highest-numbered versions first. The page width is adjusted automatically to the number of columns requested.

In studying the qualifiers and the capabilities they offer, watch for qualifiers that work together and for qualifiers that override other qualifiers. For example, if you specify the /FULL format, the system cannot display all the information in more than one column. Thus, if you specify both /COLUMNS and /FULL, the number of columns you request is ignored.

QUALIFIERS

/ACL

Controls whether the access control list (ACL) for each file is displayed. By default, DIRECTORY does not display the ACL for each file. If you specify both the /ACL qualifier and the /COLUMNS qualifier, the /COLUMNS qualifier is ignored.

/BACKUP

Selects files according to the dates of their most recent backup. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /BACKUP qualifier is incompatible with /CREATED, /EXPIRED, and /MODIFIED. /CREATED is the default.

/BEFORE[=time]

Selects only those files that are dated before the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/BRIEF (default)

Includes only the file name, type, and version number of each file to be listed. The brief format lists the files in alphabetical order from left to right on each line, in descending version number order. However, the /BRIEF qualifier, whether specified explicitly or in effect by default, is overridden whenever one of the following formatting qualifiers is specified in the command: /ACL, /DATE, /FILE-ID, /FULL, /NOHEADING, /OWNER, /PROTECTION, /SECURITY, or /SIZE.

/BY_OWNER[=uic]

Selects one or more files only if their owner user identification code (UIC) matches the specified owner UIC.

Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

If the /BY_OWNER qualifier is specified without a UIC, the UIC of the current process is assumed.

/COLUMNS=n

Lists the files using the specified number of columns on each line of the display. By default, the number of columns in the brief format is four. However, you can request as many columns as you like.

The number of columns actually displayed depends on the amount of information requested for each column and the DISPLAY value of the /WIDTH qualifier. The system will display only as many columns as can fit within the default or specified display width, regardless of how many columns you specify with /COLUMNS.

The DIRECTORY command truncates long file names only when you have asked for additional information to be included in each column. The default file name size is 19; use the /WIDTH qualifier to change the default. When a file name is truncated, the system will display one less character than the file name field size and will insert a vertical bar in the last position. For

example, if the file name is SHOW_QUEUE_CHARACTERISTICS, and if you requested DIRECTORY to display both file name and size in each column, the display for that file would be: SHOW_QUEUE_CHARACT| 120.

The /COLUMNS qualifier is not available with /ACL or /FULL.

/CREATED (default)

Selects files based on their date of creation. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /CREATED qualifier is incompatible with /BACKUP, /EXPIRED, and /MODIFIED.

/DATE[=option]
/NODATE

Includes the backup, creation, expiration, or modification date for each specified file. If you omit this qualifier, the default is /NODATE. If you use the /DATE qualifier without an option, the creation date is provided.

You can specify one of the following options with the /DATE qualifier:

ALL	Lists all four file dates, in this order: CREATED, MODIFIED, EXPIRED, BACKUP.
BACKUP	Lists the date of the last backup with each file.
CREATED	Lists the creation date with each file.
EXPIRED	Lists the expiration date with each file.
MODIFIED	Lists the last date the file was written.

/EXCLUDE=(file-spec[,...])

Any files that match the listed file specifications are excluded from the DIRECTORY operation. If you specify only one file, you can omit the parentheses.

Wildcard characters are supported for file specifications. However, you cannot use relative version numbers to exclude a specific version. The file specification can contain a directory specification, but not a device specification.

/EXPIRED

Selects files according to the dates on which they will expire. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /EXPIRED qualifier is incompatible with /BACKUP, /CREATED, and /MODIFIED. /CREATED is the default.

/FILE_ID

Controls whether the file identification of the file is displayed. By default, DIRECTORY does not display the file identification, except in the /FULL display.

/FULL

Lists the following items for each file:

File name
File type
Version number
Number of blocks used
Number of blocks allocated

Date of creation
Date of last backup
Date last modified
Date of expiration
File owner's UIC
File protection
File identification number (FID)
File organization
Other file attributes
Record attributes
Record format
Access control list (ACL)

The /FULL qualifier overrides the default brief listing format.

/GRAND_TOTAL

Suppresses both the per-directory total and individual file information. /GRAND_TOTAL displays only the total line for all files and directories that have been selected. (See the /TRAILING qualifier for information on displaying directory totals.)

/HEADING
/NOHEADING

Controls whether heading lines consisting of a device description and directory specification are printed. The default output format provides this heading.

When you specify /NOHEADING, the output appears in single-column format. In addition, the output contains the full file specification on every file. If you specify the /NOHEADING qualifier and also specify a value other than 1 with /COLUMNS, the number of columns you specify is disregarded.

You may find the combination of the /NOHEADING and /NOTRAILING qualifiers useful in command procedures where you want to create a list of complete file specifications for later operations.

/MODIFIED

Selects files according to the dates on which they were last modified. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /MODIFIED qualifier is incompatible with /BACKUP, /CREATED, and /EXPIRED. /CREATED is the default.

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT with a partial file specification (for example, /OUTPUT=[JONES]), DIRECTORY is the default file name and LIS the default file type. If you enter a file specification, it may not include any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

/OWNER
/NOOWNER (default)

Controls whether the file owner's UIC is listed.

The default size of the owner field is 20 characters. If the file owner's UIC exceeds the length of the owner field, the information will be truncated. The size of this field can be altered by specifying /WIDTH=OWNER, along with a value for the OWNER field. For more information, see the description of the /WIDTH qualifier.

/PRINTER

Queues the DIRECTORY command output for printing under the name given by the /OUTPUT qualifier. If you specify /PRINTER without the /OUTPUT qualifier, the output is directed to a file named DIRECTORY.LIS, which is automatically queued for printing and then deleted.

/PROTECTION
/NOPROTECTION (default)

Controls whether the file protection for each file is listed.

/SECURITY

Controls whether information about file security is displayed. Using the /SECURITY qualifier provides the same information as the three qualifiers /ACL, /OWNER, and /PROTECTION together.

/SELECT=(keyword[,...])

Allows you to select files based on the /SELECT keywords.

/SELECT=SIZE=MAXIMUM=n selects files that have fewer blocks than the specified parameter.

/SELECT=SIZE=MINIMUM=n selects files that have more blocks than the specified parameter.

/SELECT=SIZE=(MAXIMUM=n,MINIMUM=n) selects files whose blocksize falls within the range specified by the two parameters.

By default, file selection is based on other criteria.

/SINCE[=time]

Selects only those files that are dated after the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/SIZE[=option]
/NOSIZE (default)

Provides the file size in blocks used and/or allocated for each file listed, according to the option you specify. If you specify /SIZE without an option,

the listing provides the file size in blocks used. The options you can specify are:

ALL	Lists the file size both in blocks used and blocks allocated
ALLOCATION	Lists the file size in blocks allocated
USED	Lists the file size in blocks used

The size of this field can be altered by supplying the SIZE value of the /WIDTH qualifier.

/TOTAL

Suppresses the listing of all individual file information and displays only the trailing lines, as described under the /TRAILING qualifier.

By default, the output format is /BRIEF, which gives this total, but also lists all the file names, file types, and their version numbers.

/TRAILING
/NOTRAILING

Controls whether trailing lines that summarize the following information are output:

- Number of files listed
- Total number of blocks used per directory
- Total number of blocks allocated
- Total number of directories and total blocks used and/or allocated in all directories (only if more than one directory is listed)

By default, the output format includes most of this summary information. The /SIZE and /FULL qualifiers determine more precisely what summary information is included. If you omit /SIZE or /FULL, only the number of files is displayed and possibly the total number of directories, if applicable. If you specify the /SIZE qualifier, the number of blocks is also displayed, according to the size option selected (USED and/or ALLOCATION). If you specify the /FULL qualifier, the number of files as well as the number of blocks used and allocated are displayed.

/VERSIONS=n

Causes the latest n versions of each of the files selected to be listed. If you omit the /VERSIONS qualifier, by default the listing includes all versions of each file. A value less than 1 is not allowed.

/WIDTH=(keyword[,...])

Allows you to change the format of the directory display for four elements: file name field width, entire display width, owner field width, file size field width. If you specify only one keyword, you can omit the parentheses.

DISPLAY=n	Determines the total width for the DIRECTORY display. The value of n can range from 1 through 255. The default value for n is 0, which means that DIRECTORY sets its display width to the terminal width. If the total width of the display exceeds the terminal width, the information will be truncated.

FILENAME=n — Determines the width of the file name field. The default value for n is 19. If you have requested another piece of information to be displayed along with the file name in each column, file names that exceed the n parameter will cause the line to wrap, after the file name field. (See the /COLUMNS qualifier.)

OWNER=n — Determines the width of the owner field. The default value for n is 20. If the owner's UIC exceeds the length of the owner field, the information will be truncated.

SIZE=n — Determines the width of the size field. The default value for n is 6. If the file size exceeds the length of the size field, the information will be truncated.

EXAMPLES

1

```
$ DIRECTORY AVERAGE.*
Directory DISK$DOCUMENT:[MALCOLM]
AVERAGE.EXE;6       AVERAGE.FOR;6       AVERAGE.LIS;4       AVERAGE.OBJ;12
Total of 4 files.
```

2

```
$ DIRECTORY/SIZE=USED/DATE=CREATED/VERSIONS=1/PROTECTION  AVERAGE
Directory DISK$DOCUMENT:[MALCOLM]
AVERAGE.EXE;6        6        10-APR-1985 15:43 (RWED,RWED,RWED,RE)
AVERAGE.FOR;6        2        2-APR-1985 10:29 (RWED,RWED,RWED,RE)
AVERAGE.LIS;4        5         9-APR-1985 16:27 (RWED,RWED,RWED,RE)
AVERAGE.OBJ;6        2         9-APR-1985 16:27 (RWED,RWED,RWED,RE)
Total of 4 files, 15 blocks.
```

3

```
$ DIRECTORY/FULL [MALIK.9188]9188DIRE.GNC
Directory DISK$STARWORK05:[MALIK.9188]
9188DIRE.GNC;5                File ID:  (14086,53689,0)
Size:           47/48         Owner:    [DOC1,MALIK]
Created:  20-MAR-1985 13:39   Revised:  20-MAR-1985 13:39 (2)
Expires:   <None specified>   Backup:   21-MAR-1985 06:25
File organization:  Sequential
File attributes:    Allocation: 48, Extend: 0, Global buffer count: 0, No version limit
Record format:      Variable length, maximum 88 bytes
Record attributes:  Carriage return carriage control
File protection:    System:RWED, Owner:RWED, Group:RE, World:
Access Cntrl List:  None
Total of 1 file, 47 blocks.
```

4

```
$ DIRECTORY
```

The DIRECTORY command lists all versions of all files in the current default disk and directory in the brief format. The heading identifies the disk and directory, and the trailing line gives the total number of files.

5

```
$ DIRECTORY/VERSIONS=1/COLUMNS=1 AVERAGE.*
```

The DIRECTORY command lists only the highest version of each file named AVERAGE in the current default directory. The format is brief and restricted to one column. Heading and trailing lines are provided.

```
6  $ DIRECTORY BLOCK%%%
```

The DIRECTORY command locates all versions and types of files in the default device and directory whose names begin with the letters BLOCK and end with any three additional characters. The default output format is brief, four columns, with heading and trailing lines.

```
7  $ DIRECTORY/TOTAL/SIZE=ALL
```

The DIRECTORY command outputs only a header and a trailing line that identifies the total number of files and the blocks used and allocated for all versions of all files in the default disk and directory.

```
8  $ DIRECTORY/EXCLUDE=(AVER.DAT;*,AVER.EXE;*) [*...]AVER
```

The DIRECTORY command lists and totals all versions and types of files named AVER in all directories and subdirectories on the default disk, except any files named AVER.DAT and AVER.EXE.

```
9  $ DIRECTORY/SIZE=ALL FRESNO::DISK1:[TAYLOR]*.COM
```

This DIRECTORY command lists all versions of all files with the file type COM in the directory TAYLOR on node FRESNO and device DISK1. The listing includes the file size both in blocks used and in blocks allocated for each file.

```
10 $ DIRECTORY PEORIA::
```

This DIRECTORY command lists all the files cataloged in the directory associated with the default account being accessed at remote node PEORIA.

```
11 $ DIRECTORY-
$_/MODIFIED/SINCE=15-APR-1985:01:30/SIZE=ALL/OWNER-
$_/PROTECTION/OUTPUT=UPDATE/PRINTER [A*]
```

The DIRECTORY command locates all files that have been modified since 1:30 A.M. on April 15, 1985 and that reside on the default disk in all directories whose names begin with the letter A. It formats the output to include all versions, the size used and size allocated, the date last modified, the owner, and the protection codes. The output is directed to a file named UPDATE.LIS that is queued automatically to the default printer queue and then deleted when done.

DISCONNECT

Disconnects your physical terminal from a virtual terminal that is connected to a process. After the physical terminal is disconnected, the virtual terminal and the process remain on the system.

FORMAT

DISCONNECT

restrictions

Requires that your physical terminal is connected to a virtual terminal.

PARAMETERS

None.

DESCRIPTION

You can use the DISCONNECT command to disconnect a physical terminal from a virtual terminal and its associated process. The virtual terminal and the process remain on the system, so you can use the CONNECT command to reconnect to the process later. (See the description of the CONNECT command for more information about virtual terminals and how to connect to them.) To terminate a process connected to a virtual terminal, use the LOGOUT command.

After you are disconnected from a virtual terminal, you can use the physical terminal to log in again.

You can use the DISCONNECT command only if your physical terminal is connected to a virtual terminal.

QUALIFIER

/CONTINUE
/NOCONTINUE (default)

Controls whether the CONTINUE command is executed in the current process just before connecting to another process. This allows an interrupted image to continue processing after you connect to another process.

EXAMPLES

1

```
$ DISCONNECT
```

This command disconnects a physical terminal from a virtual terminal, but does not log the process out. The physical terminal can now be used to log in again.

DISCONNECT

```
2   $ RUN PAYROLL
    ^Y
    $ DISCONNECT/CONTINUE
```

In this example, the RUN command is issued from a physical terminal that is connected to a virtual terminal. After the image PAYROLL.EXE is interrupted, the DISCONNECT command disconnects the physical and the virtual terminals without logging the process out. The /CONTINUE qualifier allows the image PAYROLL.EXE to continue to execute. However, the terminal can be used to log in again and perform other work.

DISMOUNT

Releases a disk or magnetic tape volume that was previously mounted with the VAX/VMS Mount Utility.

FORMAT

DISMOUNT *device-name[:]*

restrictions

Requires the group name table (GRPNAM) and system name table (SYSNAM) user privileges to dismount group and system volumes, respectively.

PARAMETERS

device-name[:]

Specifies the name of the device to be dismounted. You can specify a physical device name or a logical name assigned to a physical device name. If you omit a controller designation and/or a unit number, the defaults are controller A and unit 0, respectively.

If the volume that is currently mounted on the device is a member of a disk or tape volume set, all volumes in the set are dismounted, unless the /UNIT qualifier is specified.

DESCRIPTION

The DISMOUNT command (which invokes the $DISMOU system service) performs the following operations:

- Removes the volume from the user's list of mounted volumes, deletes the logical name (if any) associated with the volume, and decrements the mount count.
- If the mount count equals 0 after being decremented, DISMOUNT marks the volume for dismounting, and then waits until the volume is idle before dismounting it. (A native volume is idle when no user has an open file to the volume; a foreign volume is idle when no channels are assigned to the volume.)
- If the mount count does not equal 0 after being decremented, DISMOUNT does not mark the volume for dismount (since the volume must have been mounted shared). In this case, the total effect for the issuing process is that the process is denied access to the volume and a logical name entry is deleted.
- Once a volume is actually dismounted, nonpaged pool is returned to the system. Paged pool is also returned if the volume had been mounted using the /GROUP or /SYSTEM qualifiers.
- If a volume is part of a Files–11 volume set and the flag bit DMT$V_UNIT is not set, the entire volume set will be dismounted.

If the volume was mounted with the /SHARE qualifier, it is not actually dismounted until all users who mounted it dismount it or log off. However, the DISMOUNT command deassigns the logical name associated with the device.

If the device was allocated with an ALLOCATE command, it remains allocated after the volume is dismounted with the DISMOUNT command. If the device was implicitly allocated by the MOUNT command, the DISMOUNT command deallocates it.

If the volume was mounted with the /GROUP or the /SYSTEM qualifier, it is dismounted even if other users are currently accessing it. The GRPNAM and SYSNAM user privileges are required to dismount group and system volumes, respectively.

Note that the file system performs volume dismounting and all open files on the volume must be closed for the dismount to finish. Note also, that the file system cannot dismount a volume while any known file lists associated with it contain entries. Thus, a substantial amount of time can pass between the time you issue the DISMOUNT command and the completion of the dismount. Always wait for the drive to unload before you remove the volume. (You can verify that the dismount has completed by issuing the SHOW DEVICES command.)

QUALIFIERS

/ABORT

Requires volume ownership or the user privilege VOLPRO to use this qualifier with a volume that is mounted neither group nor system.

Specifies that the volume is to be dismounted, regardless of who actually mounted it. The primary purpose of the /ABORT qualifier is to terminate mount verification. DISMOUNT/ABORT also cancels any outstanding I/O requests. If the volume was mounted with the /SHARE qualifier, the /ABORT qualifier causes the volume to be dismounted for all of the users who mounted it.

You must have the user privileges GRPNAM and SYSNAM in order to dismount group and system volumes, respectively. To use the /ABORT qualifier with a volume that is mounted neither group nor system, you must be the owner of the volume or have the user privilege VOLPRO.

/CLUSTER

Specifies that after the dismount command successfully dismounts the volume on the local node, the volume is to be dismounted on every other node in the existing VAXcluster (that is, the volume is dismounted cluster-wide). If the system is not a member of a VAXcluster, the /CLUSTER qualifier has no effect.

/UNIT

Specifies, for disk volume sets, that only the volume on the specified device is to be dismounted. By default, the DISMOUNT command dismounts all volumes in a volume set.

Note that there are few instances in which it is advisable to dismount part of a volume set, since individual files in a volume set may be distributed over more than one volume. However, if you are working with large volume sets and find that you must make a drive available, you may want to use the /UNIT qualifier.

Note also that it is not recommended to dismount the root volume of a volume set, since the master file directory (MFD) for the volume set is on the root volume. It may be impossible to access files on a volume set if the MFD is not accessible.

/UNLOAD (default)
/NOUNLOAD

Controls whether the DISMOUNT command unloads the physical device on which the volume is mounted and makes the device not ready. Use the /NOUNLOAD qualifier to keep the device and volume in a ready state.

EXAMPLES

```
1   $ MOUNT MT: PAYVOL TAPE
      .
      .
      .
    $ DISMOUNT TAPE:
```

The MOUNT command mounts the tape whose volume identification is PAYVOL on the device MTA0: and assigns the logical name TAPE to the device. By default, the volume is not shareable. The DISMOUNT command releases access to the volume, deallocates the device, and deletes the logical name TAPE.

```
2   $ MOUNT/SHARE DBA3:  DOC_FILES
      .
      .
      .
    $ DISMOUNT DBA3:
```

The MOUNT command mounts the volume labeled DOC_FILES on the device DBA3. Other users can issue MOUNT commands to access the device. The DISMOUNT command shown in this example deaccesses the device for the process issuing the command. If other users still have access to the volume, the volume remains mounted for their process(es).

```
3   $ DISMOUNT/NOUNLOAD  DMA2:
```

The DISMOUNT command dismounts the volume; the /NOUNLOAD qualifier requests that the volume remain in a ready state.

```
4   $ MOUNT/BIND=PAYROLL  DMA1:,DMA2:  PAYROLL01,PAYROLL02
      .
      .
      .
    $ DISMOUNT/UNIT  DMA2:
```

The MOUNT command mounts PAYROLL, a two-volume set. The DISMOUNT command will dismount only PAYROLL02, leaving PAYROLL01 accessible. Note that it is not recommended to dismount the root volume (in this case, PAYROLL01) of the volume set, since the master file directory (MFD) for the volume set is on the root volume.

DUMP

Displays or prints the contents of files or volumes in ASCII, decimal, hexadecimal, or octal representation.

FORMAT

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

restrictions

None.

PARAMETER

file-spec

Specifies the files or the volume to be displayed.

If the specified device is not a disk, tape, or network device, or if the device is mounted with the /FOREIGN qualifier, the file specification must contain only the device name.

If the specified device is a network device, a disk device, or tape device that is mounted without the /FOREIGN qualifier, the file specification can contain wildcards.

DESCRIPTION

By default, the DUMP command formats the output both in ASCII characters and in hexadecimal longwords. You can specify another format for the dump by using a radix qualifier (/OCTAL, /DECIMAL, or /HEXADECIMAL) and /or a length qualifier (/BYTE, /WORD, or /LONGWORD).

Dumping Files

If the input medium is a network device, a disk device, or tape device that is mounted without the /FOREIGN qualifier, the DUMP command operates on files. You can dump files by either records or blocks. Wildcard specifications can be used to select a group of files for processing.

Dumping Volumes

If the input medium is not a disk or tape device, or if it is mounted with the /FOREIGN qualifier, the DUMP command operates on the input device as a non-file-structured medium. Disk devices are dumped by 512-byte logical blocks. Other devices are dumped by physical blocks. No repositioning of the input medium occurs; thus consecutive blocks on a tape can be dumped by a single DUMP command.

If you have LOG_IO (logical I/O) privilege, you can dump random blocks on a Files–11 volume. For example, using the /BLOCKS qualifier you could dump block 100 on the system disk.

Reading Dumps

The ASCII representation is read left to right. The hexadecimal, decimal, and octal representations are read right to left.

Specifying Numeric Qualifier Values

The numeric values for the /BLOCKS, /RECORDS, and /NUMBER qualifiers can be specified either as decimal numbers or with a leading %X, %O, or %D to signify hexadecimal, octal, or decimal numbers respectively. For example, the following are all valid ways to specify decimal value 24:

```
24
%X18
%O30
%D24
```

QUALIFIERS

/ALLOCATED

Specifies that the dump include all blocks allocated to the file. By default, the dump does not include blocks following the end-of-file block.

You can specify /ALLOCATED if the input is a disk that is mounted without the /FOREIGN qualifier. If you specify /ALLOCATED, you cannot specify /RECORDS.

/BLOCKS[=(option[,...])]

Specifies that the input medium be dumped one block at a time. This is the default for all devices except network devices. You cannot specify /BLOCKS for network devices.

You can use one or more of the following options to select a range of blocks to be dumped:

START:n	Specifies the number of the first block to be dumped. By default, the dump begins with the first block of the file or device.
END:n	Specifies the number of the last block to be dumped. By default, the dump ends with the last block of the file or device. If the input is a disk file, the /ALLOCATED qualifier determines whether the last block is the end of file block or the last allocated block.
COUNT:n	Specifies the number of blocks to be dumped. This option provides an alternate way to specify the last block to be dumped.

If you specify only one option, you can omit the parentheses. You cannot specify both END and COUNT.

Blocks are usually numbered beginning with 1. However, for a disk device that is mounted with the /FOREIGN qualifier, blocks are numbered beginning with 0.

If you specify /BLOCKS, you cannot specify /RECORDS.

Use the /BLOCKS qualifier to dump random blocks from Files–11 volumes. This requires LOG-IO (logical I/O) privilege.

/BYTE

Specifies that the dump is to be formatted in bytes. The default format is composed of longwords.

If you specify /BYTE, you cannot specify /WORD or /LONGWORD.

/DECIMAL

Specifies that the dump is to be formatted in decimal. The default format is in hexadecimal representation.

If you specify /DECIMAL, you cannot specify /HEXADECIMAL or /OCTAL.

/FILE_HEADER

Specifies that data blocks which are valid Files–11 file headers are to be printed in an interpreted representation. All other data blocks are printed in the selected radix and length.

/FORMATTED (default)
/NOFORMATTED

Specifies whether the file header is displayed in a formatted or unformatted representation. This qualifier is meaningful only when the /HEADER qualifier is specified. If you specify /FORMATTED, the file header is printed in an interpreted format. If you specify /NOFORMATTED, the file header is printed in the selected radix and length.

/HEADER

Specifies that the output include the file header as well as the access control list. You can use the /FORMATTED qualifier to control the format of the display. You can display the file header without displaying file contents by also specifying /BLOCKS=COUNT=0.

You can use the /HEADER qualifier if the input is a disk or tape mounted without the /FOREIGN qualifier.

You can use the /FILE_HEADER qualifier with /HEADER to have Files–11 file headers printed in an interpreted representation.

By default, the file header is not displayed.

/HEXADECIMAL (default)

Specifies that the dump is to be formatted in hexadecimal.

If you specify /HEXADECIMAL, you cannot specify /DECIMAL or /OCTAL.

/LONGWORD (default)

Specifies that the dump is to be formatted in longwords.

If you specify /LONGWORD, you cannot specify /BYTE or /WORD.

/NUMBER[=n]

Specifies how byte offsets are assigned to the lines of output. If you specify /NUMBER, the byte offsets increase continuously through the dump, beginning with the specified value. If you omit the value, the initial byte offset is zero.

By default, the byte offset is reset to zero at the beginning of each block or record.

/OCTAL

Specifies that the dump is to be formatted in octal. The default format is in hexadecimal representation.

If you specify /OCTAL, you cannot specify /DECIMAL or /HEXADECIMAL.

/OUTPUT[=file-spec]

Specifies that the DUMP output be written to the specified file. By default, the DUMP command writes output to SYS$OUTPUT. If you specify /OUTPUT without a file specification, the DUMP command writes output to a file with the same file name as the input file and the file type DMP.

No wildcard characters are allowed in the file specification.

If you specify /OUTPUT, you cannot specify /PRINTER.

/PRINTER

Specifies that output be queued to the system printer. By default, the DUMP command writes output to SYS$OUTPUT. If you specify /PRINTER, the DUMP command writes output to a file with the same file name as the input file and a file type DMP, and queues the file to SYS$PRINT.

If you specify /PRINTER, you cannot specify /OUTPUT.

/RECORDS[=(option[,...])]

Specifies that a file be dumped one record at a time. By default, input is dumped one block at a time for all devices except network devices. The default for network devices is /RECORDS.

You can specify one or more of the following options to select a range of records to be dumped:

START:n	Specifies the number of the first record to be dumped. By default, the dump begins with the first record of the file.
END:n	Specifies the number of the last record to be dumped. By default, the dump ends with the last record of the file.
COUNT:n	Specifies the number of records to be dumped. This option provides an alternate way to specify the last record to be dumped.

Records are numbered beginning with 1.

If you specify only one option, you can omit the parentheses. You cannot specify both END and COUNT.

You can use the /RECORDS qualifier if the input is a network device or a disk or tape mounted without the /FOREIGN qualifier. If you specify /RECORDS, you cannot specify /ALLOCATED or /BLOCKS.

/WORD

Specifies that the dump is to be formatted in words. The default format is composed of longwords.

If you specify /WORD, you cannot specify /BYTE or /LONGWORD.

EXAMPLES

1

```
$ DUMP TEST.DAT
Dump of file DISKO:[NORMAN]TEST.DAT;1 on 15-APR-1985 15:43:26.08
File ID (3134,818,2)    End of file block 1 / Allocated 3

Virtual block number 1 (00000001), 512 (0200) bytes

 706D6173 20612073 69207369 68540033 3.This is a samp 000000
 73752065 62206F74 20656C69 6620656C le file to be us 000010
 61786520 504D5544 2061206E 69206465 ed in a DUMP exa 000020
 00000000 00000000 0000002E 656C706D mple..........  000030
 00000000 00000000 00000000 00000000 ................ 000040
 00000000 00000000 00000000 00000000 ................ 000050
 00000000 00000000 00000000 00000000 ................ 000060
          .
          .
          .
 00000000 00000000 00000000 00000000 ................ 0001E0
 00000000 00000000 00000000 00000000 ................ 0001F0
```

The DUMP command displays the contents of TEST.DAT both in hexadecimal longword format and in ASCII beginning with the first block in the file.

2

```
$ DUMP TEST.DAT/OCTAL/BYTE
Dump of file DISKO:[NORMAN]TEST.DAT;1 on 15-APR-1985 15:45:33.58
File ID (74931,2,1)    End of file block 1 / Allocated 3

Virtual block number 1 (00000001), 512 (0200) bytes

 151 040 163 151 150 124 000 063 3.This i 000000
 160 155 141 163 040 141 040 163 s a samp 000010
 040 145 154 151 146 040 145 154 le file  000020
 163 165 040 145 142 040 157 164 to be us 000030
 040 141 040 156 151 040 144 145 ed in a  000040
 141 170 145 040 120 115 125 104 DUMP exa 000050
 377 377 000 056 145 154 160 155 mple.... 000060
 000 000 000 000 000 000 000 000 ........ 000070
 000 000 000 000 000 000 000 000 ........ 000100
 000 000 000 000 000 000 000 000 ........ 000110
          .
          .
          .
 000 000 000 000 000 000 000 000 ........ 000760
 000 000 000 000 000 000 000 000 ........ 000770
```

The DUMP command displays the image of the file TEST.DAT, formatted both in octal bytes and in ASCII characters beginning with the first block.

3

```
$ DUMP NODE3::DISK2:[STATISTICS]RUN1.DAT
```

This command line dumps the file RUN1.DAT that is located at remote node NODE3. The default DUMP format will be used.

EDIT/ACL

Invokes the Access Control List (ACL) Editor to create or modify an access control list for a specified object. For a complete description of the VAX/VMS Access Control List Editor, including information about the EDIT/ACL command and its qualifiers, see the *VAX/VMS Access Control List Editor Reference Manual*.

The /ACL qualifier is required.

FORMAT EDIT/ACL *file-spec*

EDIT/EDT

Invokes the VAX EDT interactive text editor. The /EDT qualifier is not required, because EDT is the VAX/VMS default editor.

FORMAT

EDIT *file-spec*

restrictions

None.

PARAMETER

file-spec

Specifies the file to be created or edited using the EDT editor. If the file does not exist, it is created by EDT.

The EDT editor does not provide a default file type when creating files; if you do not include a file type, it is null. The file must be a disk file on a Files–11 formatted volume.

No wildcard characters are allowed in the file specification.

DESCRIPTION

The EDT editor creates or edits text files. You can use EDT to enter or edit text in three modes: keypad, line, or nokeypad. Keypad editing, which is screen-oriented, is available on VT200 Series, VT100, and VT52 terminals. A screen-oriented editor allows you to see several lines of text at once and move the cursor throughout the text in any direction. Line editing operates on all terminals. In fact, if you have a terminal other than a VT200 Series, VT100, or VT52, line editing is the only way you can use EDT. You might prefer line editing if you are accustomed to editing by numbered lines. Nokeypad mode is a command oriented screen editor available on VT200 Series, VT100, and VT52 terminals. You can use line mode and nokeypad mode to redefine keys for use in keypad mode.

When you invoke EDT, you are in line mode, by default. If you are editing an existing file, EDT prints the line number and text for the first line of the file. If you are creating a new file, EDT prints the following message:

```
Input file does not exist
[EOB]
```

In either case, EDT then displays the line mode prompt, which is the asterisk (*).

For complete details on the EDT editor, see the *VAX EDT Reference Manual*.

QUALIFIERS

/COMMAND[=file-spec]
/NOCOMMAND

Determines whether or not EDT uses a startup command file. The /COMMAND file qualifier should be followed by an equal sign and the specification of the command file. The default file type for command files is EDT.

The following command line invokes EDT to edit a file named MEMO.DAT and specifies that EDT use a startup command file named XEDTINI.EDT:

```
$ EDIT/COMMAND=XEDTINI.EDT MEMO.DAT
```

If you do not include the /COMMAND=command file qualifier, EDT looks for the EDTSYS logical name assignment. If EDTSYS is not defined, EDT processes the systemwide startup command file SYS$LIBRARY:EDTSYS.EDT. If this file does not exist, EDT looks for the EDTINI logical name assignment. If EDTINI is not defined, EDT looks for the file named EDTINI.EDT in your default directory. If none of these files exists, EDT begins your editing session in the default state.

To prevent EDT from processing either the systemwide startup command file or the EDTINI.EDT file in your default directory, use the /NOCOMMAND qualifier as follows:

```
$ EDIT/NOCOMMAND MEMO.DAT
```

No wildcard characters are allowed in the file specification.

/CREATE (default)
/NOCREATE

Controls whether EDT creates a new file when the specified input file is not found. Normally, EDT creates a new file to match the input file specification if it cannot find the requested file name in the specified directory. When you use /NOCREATE in the EDT command line and type a specification for a file that does not exist, EDT prints an error message and returns you to the DCL command level as follows:

```
$ EDIT/NOCREATE NEWFILE.DAT
Input file does not exist
$
```

/JOURNAL[=journal-file]
/NOJOURNAL

Determines whether EDT keeps a journal file during your editing session. The default file name for the journal file is the same as the input file name. The default file type is JOU. The /JOURNAL qualifier enables you to use a different file specification for the journal file.

The following command line invokes EDT to edit a file named MEMO.DAT and specifies the name SAVE.JOU for the journal file:

```
$ EDIT/JOURNAL=SAVE MEMO.DAT
```

If you are editing a file from another directory and want the journal file to be located in that directory, you must use the /JOURNAL qualifier with a file specification that includes the directory name. Otherwise, EDT creates the journal file in the default directory.

The directory that is to contain the journal file should not be write protected.

To prevent EDT from keeping a record of your editing session, use the /NOJOURNAL qualifier in the EDT command line as follows:

```
$ EDIT/NOJOURNAL MEMO.DAT
```

Once you have created a journal file, use the /RECOVER qualifier to have EDT process the commands contained in the journal file.

No wildcard characters are allowed in the file specification.

/OUTPUT=output-file
/NOOUTPUT

Determines whether EDT creates an output file at the end of your editing session. The default file specification for both the input file and the output file is the same. Use the /OUTPUT qualifier to give the output file a different file specification from the input file.

The following command line invokes EDT to edit a file named MEMO.DAT and gives the resulting output file the name OUTMEM.DAT:

```
$ EDIT/OUTPUT=OUTMEM.DAT MEMO.DAT
```

You can include directory information as part of your output file specification to send output to another directory as follows:

```
$ EDIT/OUTPUT=[BARRETT.MAIL]MEMO.DAT MEMO.DAT
```

The /NOOUTPUT qualifier suppresses the creation of an output file, but not the creation of a journal file. If you are testing some edits and are not sure you want an output file, you can use /NOOUTPUT as follows:

```
$ EDIT/NOOUTPUT MEMO.DAT
```

A system interruption will not prevent you from recreating your editing session because a journal file is still being maintained. If you decide you want to save your editing session, you can do so, even though you specified /NOOUTPUT, by using the line mode command WRITE to put the text in an external file before you end the session.

No wildcard characters are allowed in the file specification.

/READ_ONLY
/NOREAD_ONLY (default)

Determines whether EDT keeps a journal file and creates an output file. With the default /NOREAD_ONLY, EDT maintains the journal file and creates an output file when it processes the line mode command EXIT. Using the /READ_ONLY qualifier is like using both the /NOJOURNAL and /NOOUTPUT qualifiers.

The following command line invokes EDT to edit a file named CALENDAR.DAT, but does not create a journal file or an output file:

```
$ EDIT/READ_ONLY CALENDAR.DAT
```

Use /READ_ONLY when you are searching a file. If you then want to modify the file, use the line mode command WRITE to save your changes. Remember, however, that you have no journal file.

/RECOVER
/NORECOVER (default)

Determines whether or not EDT reads a journal file at the start of the editing session.

When you use the /RECOVER qualifier, EDT reads the appropriate journal file and processes whatever commands it contains. The appropriate syntax follows:

```
$ EDIT/RECOVER MEMO.DAT
```

If the journal file type is not JOU or the file name is not the same as the input file name, you must include both the /JOURNAL qualifier and the /RECOVER qualifier as follows:

```
$ EDIT/RECOVER/JOURNAL=SAVE.XXX MEMO.DAT
```

Because /NORECOVER is the default for EDT, you do not need to specify it in a command line.

EXAMPLES

1

```
$ EDIT/OUTPUT=NEWFILE.TXT OLDFILE.TXT
    1       This is the first line of the file OLDFILE.TXT.
*
```

This EDIT command invokes the EDT editor to edit the file OLDFILE.TXT. EDT looks for the EDTSYS logical name assignment. If EDTSYS is not defined, EDT processes the systemwide startup command file SYS$LIBRARY:EDTSYS.EDT. If this file does not exist, EDT looks for the EDTINI logical name assignment. If EDTINI is not defined, EDT looks for the file named EDTINI.EDT in your default directory. If none of these files exists, EDT begins your editing session in the default state. When the session ends, the edited file has the name NEWFILE.TXT.

2

```
$ EDIT/RECOVER OLDFILE.TXT
```

This EDIT command invokes the EDT editor to recover from an abnormal exit during a previous editing session. EDT opens the file OLDFILE.TXT, and then processes the journal file OLDFILE.JOU. Once the journal file has been processed, you can resume interactive editing.

EDIT/FDL

Invokes the VAX/VMS FDL Editor (EDIT/FDL) to create and modify File Definition Language (FDL) files. The /FDL qualifier is required. For a complete description of the FDL Utility, including more information about the EDIT/FDL command and its qualifiers, see the FDL Facility in the *VAX/VMS File Definition Language Facility Reference Manual*.

FORMAT **EDIT/FDL** *file-spec*

EDIT/SUM

Invokes the SUMSLP batch-oriented editor, to update a single input file with multiple files of edit commands. The /SUM qualifier is required.

For a complete description of the SUMSLP editor, including information about the EDIT/SUM command and its qualifiers, see the *VAX/VMS SUMSLP Utility Reference Manual*.

FORMAT **EDIT/SUM** *input-file*

EDIT/TECO

Invokes the TECO interactive text editor. The /TECO qualifier is required.

FORMAT

EDIT/TECO *[file-spec]*
EDIT/TECO/EXECUTE=command-file *[argument]*

restrictions

None.

PARAMETER

file-spec

Specifies the file to be created or edited using the TECO editor. If the file does not exist, it is created by TECO, unless you specify /NOCREATE. No wildcard characters are allowed in the file specification.

If you do not enter a file specification and you have specified /MEMORY (the default), TECO edits the file identified by the logical name TEC$MEMORY. If there is no equivalence string for TEC$MEMORY, or if you have specified /NOMEMORY, TECO starts in command mode and does not edit an existing file.

If you enter a file specification and have specified /MEMORY, the file specification is equated to the logical name TEC$MEMORY.

DESCRIPTION

The TECO editor creates or edits text files. For detailed information on the use of TECO, see the *PDP–11 TECO Editor Reference Manual*.

QUALIFIERS

/COMMAND[=file-spec]
/NOCOMMAND

Determines whether or not TECO uses a startup command file. The /COMMAND file qualifier may be followed by an equal sign and the specification of the command file. The default file type for command files is TEC.

The following command line invokes TECO to edit a file named MEMO.DAT and specifies that TECO use a startup command file named XTECOINI.TEC:

```
$ EDIT/TECO/COMMAND=XTECOINI.TEC MEMO.DAT
```

If you do not include the /COMMAND qualifier, or if you enter /COMMAND without specifying a command file, TECO looks for the TEC$INIT logical name assignment. If TEC$INIT is not defined, no startup commands are executed.

The logical name TEC$INIT can equate either to a string of TECO commands or to a dollar sign followed by a file specification. If TEC$INIT translates to a string of TECO commands, the string is executed; if it translates to a dollar sign followed by a file specification, the contents of the file are executed as a TECO command string. For further information, see the *PDP–11 TECO Editor Reference Manual*.

To prevent TECO from using any startup command file, use the /NOCOMMAND qualifier as follows:

```
$ EDIT/TECO/NOCOMMAND MEMO.DAT
```

If you enter a file specification, no wildcard characters are allowed.

/CREATE (default)
/NOCREATE

Controls whether TECO creates a new file when it cannot find the specified input file or, if no input file is specified, the file specified by the logical name TEC$MEMORY, if any and if /MEMORY is specified. Normally, TECO creates a new file to match the input file specification if it cannot find the requested file name in the specified directory. When you use /NOCREATE in the TECO command line and type a specification for a file that does not exist, TECO prints an error message and returns you to the DCL command level.

If you specify the /EXECUTE qualifier, the /CREATE and /NOCREATE qualifiers are ignored.

/EXECUTE=command-file [argument]

Causes TECO to be invoked to execute a TECO macro. The command file contains the TECO macro to be executed. The argument, if specified, appears in the text buffer when macro execution starts. If the argument contains blanks or special characters, it must be enclosed in quotes.

For detailed information on the use of TECO macros, see the *PDP–11 TECO Editor Reference Manual*.

If you specify the /EXECUTE qualifier, the /CREATE and /MEMORY qualifiers are ignored.

/MEMORY (default)
/NOMEMORY

Specifies whether the last file that you edited with TECO, identified by the logical name TEC$MEMORY, will be edited if you omit the file specification on the EDIT/TECO command.

/OUTPUT=output-file
/NOOUTPUT (default)

Determines the specification of the output file at the end of your editing session. The default is /NOOUTPUT; that is, the output file specification is the same as for the input file, but the version number is increased by 1. Use the /OUTPUT qualifier to give the output file a different file specification from the input file.

The following command line invokes TECO to edit a file named MEMO.DAT and gives the resulting output file the name OUTMEM.DAT:

```
$ EDIT/TECO/OUTPUT=OUTMEM.DAT MEMO.DAT
```

You can include directory information as part of your output file specification to send output to another directory as follows:

```
$ EDIT/TECO/OUTPUT=[BARRETT.MAIL]MEMO.DAT MEMO.DAT
```

No wildcard characters are allowed in the file specification.

/READ_ONLY
/NOREAD_ONLY (default)

Controls whether an output file is written. If you specify /READ_ONLY, no output file is written.

EXAMPLES

1
```
$ EDIT/TECO/OUTPUT=NEWFILE.TXT OLDFILE.TXT
```

This EDIT command invokes the TECO editor to edit the file OLDFILE.TXT. TECO looks for the TEC$INIT logical name assignment. If TEC$INIT is not defined, TECO begins the editing session without using a command file. When the session ends, the edited file has the name NEWFILE.TXT.

2
```
$ EDIT/TECO/EXECUTE=FIND_DUPS "TEMP, ARGS, BLANK"
```

The /EXECUTE qualifier causes the TECO macro contained in the file FIND_DUPS.TEC to be executed, with the argument string "TEMP, ARGS, BLANK" located in the text buffer.

EDIT/TPU

Invokes the VAX Text Processing Utility (VAXTPU). By default, the Extensible VAX Editor (EVE) is used as the interface for VAXTPU. To invoke VAXTPU with the EDT Keypad Emulator interface, define the logical TPUSECINI to point to the section file for that interface as follows:

```
$ DEFINE TPUSECINI EDTSECINI
```

FORMAT

EDIT/TPU *[file-spec]*

restrictions

None.

PARAMETER

file-spec

Specifies the file to be created or edited using one of the VAXTPU editing interfaces. If the file you specify does not exist, the VAXTPU interfaces provide a buffer in which to create the file. If you write out the contents of the buffer (either with the built-in procedure WRITE_FILE or by exiting from the editor), VAXTPU creates the file in the appropriate directory.

If you do not provide a file specification as a parameter when you invoke VAXTPU, when you try to leave the editor, you are prompted for a file name for any buffer that you modified.

VAXTPU does not provide a default file type when creating files. If you do not specify a file type, the file type is null. The file must be a disk file on a Files–11 formatted volume.

Whether or not VAXTPU recognizes wildcard characters as a part of the input file specification depends on the interface you are using. EVE handles wildcard characters if there is a single match for the wildcard character. Otherwise, EVE prompts within the editor for possible file names. EVE treats multiple files in a search list in the same way as multiple files in a wildcard specification. The EDT Keypad Emulator does not recognize wildcard characters.

DESCRIPTION

The VAX Text Processing Utility (VAXTPU) consists of a procedural language, a compiler, an interpreter, and two editing interfaces—the Extensible VAX Editor (EVE) and the VAXTPU EDT Keypad Emulator. Both of the interfaces are written in VAXTPU and are layered on VAXTPU. You can use the editing interfaces to create or edit text files, as well as to access VAXTPU functions to customize or extend the interfaces.

For information on VAXTPU, see the *VAX Text Processing Utility Reference Manual*. To learn how to use one of the editing interfaces provided with VAXTPU, read the appropriate manual. The *VAX Text Processing Utility Reference Manual* contains an appendix of editing commands for each of the two interfaces. The *Guide to Text Processing on VAX/VMS* contains chapters describing both the Extensible VAX Editor (EVE) interface and the EDT Keypad Emulator.

EDIT/TPU

VAXTPU provides qualifiers for the command EDIT/TPU so that interfaces layered on VAXTPU can control command line processing. The following descriptions explain what the basic qualifier does and indicate any differences between the basic behavior and the way EVE and the EDT Keypad Emulator implement the qualifiers.

QUALIFIERS

/COMMAND[=command-file] (default)
/NOCOMMAND

Determines whether VAXTPU reads a user-written command file for initialization purposes. The default file type for command files is TPU. By default, VAXTPU tries to read a file called TPUINI.TPU from your default directory. You can use a full file specification after the qualifier /COMMAND or define the logical name TPUINI to point to a command file other than the default one.

To prevent VAXTPU from processing a command file, use the qualifier /NOCOMMAND. Using /NOCOMMAND when you do not want to use a command file decreases startup time by eliminating the search for a command file.

/CREATE (default)
/NOCREATE

Controls whether VAXTPU creates a new file when the specified input file is not found. The interface that is layered on VAXTPU is responsible for processing this qualifier. The interface can get information on whether the qualifier was specified when VAXTPU was invoked by using the built-in procedure GET_INFO, for example, x := GET_INFO (COMMAND_LINE, "create"). For information on GET_INFO, see Section 4 of the *VAX Text Processing Utility Reference Manual*.

By default, the VAXTPU interfaces provide a buffer in which to create a file. If you write out the contents of the buffer (either with the built-in procedure WRITE_FILE or by exiting from the editor), VAXTPU creates a new file for the input file specification. When you use the qualifier /NOCREATE and enter a file specification that does not exist, EVE and the EDT Keypad Emulator clear the screen, display an error message, and return you to the DCL command level as follows:

```
$ EDIT/TPU/NOCREATE NEWFILE.DAT

Input file does not exist: DISK$:[USER]NEWFILE.DAT;
$
```

/DISPLAY[=file-spec] (default)
/NODISPLAY

Determines whether a VAXTPU session is run from a supported terminal and uses terminal functions such as the screen display and the keyboard. By default your VAXTPU session is run with the screen management file, TPU$CCTSHR.EXE, for terminals that respond to ANSI control functions and that operate in ANSI mode.

VAXTPU expects sessions to be run from a supported terminal unless you specify /NODISPLAY. If an unsupported input device is used when /DISPLAY is active, VAXTPU returns an error message and the session is terminated. (See Appendix B of the *VAX Text Processing Utility Reference Manual* for information on terminals supported by VAXTPU.)

The qualifier /NODISPLAY causes VAXTPU to run without using the screen display and the keyboard functions of a terminal. Use the qualifier /NODISPLAY under the following conditions:

- When running VAXTPU procedures in a batch job
- When using VAXTPU on an unsupported terminal

If you use /NODISPLAY, window and screen manipulation commands and key definitions cause errors. Initialization files that contain screen manipulation commands (ADJUST_WINDOW, CREATE_WINDOW, MAP) and key definitions can run when the /NODISPLAY feature is active. However, the commands are meaningless and may even return error messages in the batch log file or on your screen. Use a special initialization file (either a section file or a command file) for sessions in which you use the qualifier /NODISPLAY. This file should not include key definitions or screen manipulation commands (except for READ_LINE, MESSAGE, and LAST_KEY, which work with some restrictions). See the descriptions of READ_LINE and LAST_KEY in Section 4 of the *VAX Text Processing Utility Reference Manual* for a list of the restrictions. The file should be a complete VAXTPU session; it should end with the command EXIT or the command QUIT.

/JOURNAL[=journal-file] (default) /NOJOURNAL

Determines whether VAXTPU keeps a journal file of your editing session so that you can recover from an interrupted session. The interface that is layered on VAXTPU is responsible for processing this qualifier. The interface can get information on whether the qualifier was specified when VAXTPU was invoked by using the built-in procedure GET_INFO, for example, x := GET_INFO (COMMAND_LINE, "journal"). For information on GET_INFO, see Section 4 of the *VAX Text Processing Utility Reference Manual.*

By default, the VAXTPU interfaces maintain a journal file that has the same name as the input file and the file type TJL. If you invoke VAXTPU without a file specification, the default name for the journal file is TPU.TJL. Use a full file specification with the qualifier /JOURNAL to specify a different name for the journal file.

If you edit a file from another directory and you want the journal file to be located in that directory, you must use the qualifier /JOURNAL with a file specification that includes the directory name. Otherwise, the VAXTPU interfaces create the journal file in the default directory.

Once you create a journal file, invoke VAXTPU with the qualifier /RECOVER to cause VAXTPU to process the journal file.

To prevent VAXTPU from keeping a journal file for your editing session, use the qualifier /NOJOURNAL.

/OUTPUT=output-file (default) /NOOUTPUT

Determines whether VAXTPU creates an output file at the end of your editing session. The interface that is layered on VAXTPU is responsible for processing this qualifier. The interface can get information on whether the qualifier was specified when VAXTPU was invoked with the built-in procedure GET_INFO, for example, x := GET_INFO (COMMAND_LINE, "output"). For information on GET_INFO, see Section 4 of the *VAX Text Processing Utility Reference Manual.*

By default, the VAXTPU interfaces use the same file specification for both the input file and the output file. The output file has a version number one higher than the highest existing version of the input file. Use a file specification with the qualifier /OUTPUT to specify a file name that is different from the input file. You can include directory information as part of your output file specification to send output to another directory.

The qualifier /NOOUTPUT suppresses the creation of an output file, but not the creation of a journal file. If you invoke VAXTPU with /NOOUTPUT and then decide you want an output file, use the command WRITE FILE in EVE, or the built-in procedure WRITE_FILE in the EDT Keypad Emulator interface before you end the editing session to write out the contents of a buffer.

/READ_ONLY
/NOREAD_ONLY (default)

Determines whether VAXTPU keeps a journal file and creates an output file from the contents of the main buffer. With /NOREAD_ONLY, VAXTPU maintains a journal file and creates an output file from the contents of the main buffer if you modified it.

Using the qualifier /READ_ONLY is like using the qualifier /NOJOURNAL for the editing session and the qualifier /NOOUTPUT for the main buffer. When you specify /READ_ONLY, VAXTPU does not maintain a journal file for your editing session, and the NO_WRITE attribute is set for the main buffer. When a buffer is set to NO_WRITE, the contents of the buffer are not written out when you leave VAXTPU. Both the built-in procedures EXIT and QUIT end the editing session without creating a new file from the contents of the main buffer (even if you modified it).

Use the qualifier /READ_ONLY when you are merely reading or searching a file. If you change your mind and want to save any edits you make to the file, use the command WRITE FILE in EVE, or the built-in procedure WRITE_FILE in the EDT Keypad Emulator to write the modified buffer to an external file. Remember, however, that you do not have a journal file.

/RECOVER
/NORECOVER (default)

Determines whether VAXTPU reads a journal file at the start of an editing session.

When you use the qualifier /RECOVER, VAXTPU reads the appropriate journal file and processes whatever commands it contains. If the journal file type is not TJL, or if the file name is not the same as the input file name, you must include both the qualifier /JOURNAL=journal-file and the qualifier /RECOVER.

When you recover a session, all files must be in the same state as they were at the start of the editing session being recovered. All terminal characteristics must also be in the same state as they were at the start of the editing session being recovered. Check especially the following terminal characteristics:

Device_Type
Edit_mode
Eightbit
Page
Width

/SECTION[=file-spec] (default)
/NOSECTION

Determines whether VAXTPU reads a special initialization file that is stored in binary form. The default file type for section files is TPU$SECTION. By default, VAXTPU tries to read the file SYS$LIBRARY:TPUSECINI.TPU$SECTION, the section file that creates the EVE editing interface. You can specify a different file for initialization purposes by defining the logical name TPUSECINI to point to a section file other than the default one. This is the preferred method. However, you can also supply a full file specification for the qualifier /SECTION.

The file used as the value for /SECTION=section-file must be compiled by running the source code version of your initialization file through VAXTPU and then using the built-in procedure SAVE. This process converts the file to the proper binary form.

If you specify /NOSECTION, VAXTPU does not read a binary initialization file. If no section file is read, and no command file is read, VAXTPU will not have a user interface and no keys will be defined. In this state, the only way to exit from VAXTPU is by pressing CTRL/Y.

EXAMPLES

1
```
$ EDIT/TPU/OUTPUT=NEWFILE.TXT OLDFILE.TXT
```

Invokes VAXTPU to edit the file OLDFILE.TXT. By default, VAXTPU uses the section file SYS$LIBRARY:TPUSECINI.TPU$SECTION that creates the EVE editing interface. VAXTPU then tries to read the command file TPUINI.TPU in your default directory. If you do not define the logical name TPUINI to point to a user-written command file, and if you do not have a file named TPUINI.TPU in your default directory, VAXTPU does not read a command file. If you modify the main buffer and then use the command EXIT to end the session, the modified file is named NEWFILE.TXT.

2
```
$ EDIT/TPU/SECTION=EDTSECINI OLDFILE.TXT
```

Invokes VAXTPU with the EDT Keypad Emulator editing interface. VAXTPU makes a copy of the file OLDFILE.TXT available for editing. If you modify the main buffer, when you use the command EXIT to leave the editing session, VAXTPU creates a new version of the file with a version number one higher than the highest existing version number for that file.

3
```
$ EDIT/TPU/SECTION=DISK$USER:[SMITH]VT100INI
```

Causes VAXTPU to read the section file VT100INI.TPU$SECTION.

4
```
$ EDIT/TPU/RECOVER OLDFILE.TXT
```

Invokes VAXTPU with the EVE editing interface to recover from an abnormal exit during a previous editing session. VAXTPU opens the file OLDFILE.TXT and then processes the journal file OLDFILE.TJL. Once the journal file has been processed, you can resume interactive editing.

5
```
$ EDIT/TPU/RECOVER/JOURNAL=SAVE.XXX MEMO.DAT
```

Invokes VAXTPU with the EVE editing interface to recover from an abnormal exit during a previous session. VAXTPU opens the file MEMO.DAT and then processes the journal file SAVE.XXX.

EOD

Signals the end of a data stream when a command or program is reading data from an input device other than an interactive terminal.

FORMAT

$ EOD

restrictions *None.*

PARAMETERS

None.

DESCRIPTION

Use the EOD (End of Deck) command in a command procedure or in a batch job to:

- Terminate input data lines that begin with dollar signs. The DECK command indicates that the following lines begin with dollar signs and should be interpreted as data, not as commands; the EOD command indicates the end of the data lines.
- Terminate an input file if multiple input files are contained in the command stream without intervening commands. The program or command reading the data receives an end-of-file condition when the EOD command is read.

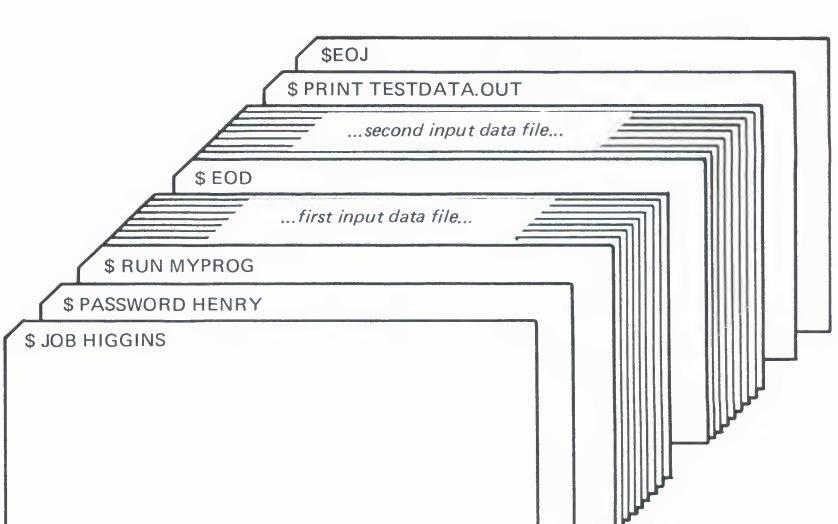
The EOD command must be preceded by a dollar sign; the dollar sign must be in the first character position (column 1) of the input record.

EXAMPLES

1
```
$ CREATE WEATHER.COM
$ DECK
$ FORTRAN WEATHER
$ LINK WEATHER
$ RUN WEATHER
$ EOD
$ @WEATHER
```

This command procedure creates a command procedure called WEATHER.COM. The lines delimited by the DECK and EOD commands are written to the file WEATHER.COM. Then the command procedure executes WEATHER.COM.

2



ZK-785-82

The program MYPROG requires two input files; these are read from the logical device SYS$INPUT. The EOD command signals the end of the first data file and the beginning of the second. The next line that begins with a dollar sign (a PRINT command in this example) signals the end of the second data file.

EOJ

Marks the end of a batch job submitted through a card reader. An EOJ card is not required; however, if present, the first nonblank character in the command line must be a dollar sign ($). If issued in any other context, the EOJ command logs the process out. The EOJ command cannot be abbreviated.

The EOF card is equivalent to the EOJ card.

FORMAT

$ EOJ

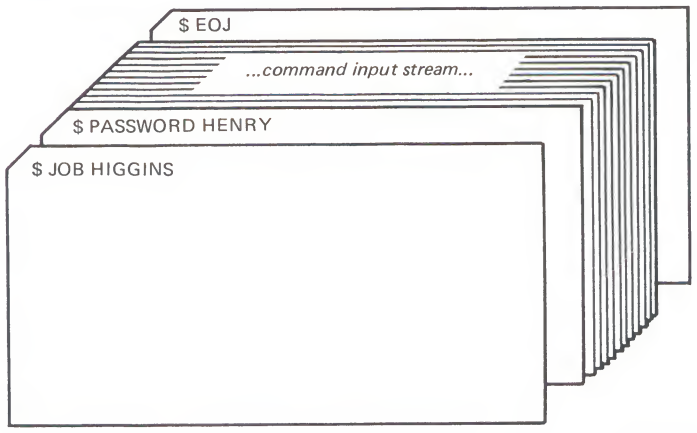
restrictions

None.

PARAMETERS

None.

EXAMPLE



ZK-786-82

The JOB and PASSWORD commands mark the beginning of a batch job submitted through the card reader; the EOJ command marks the end of the job.

EXAMINE

Displays the contents of virtual memory.

FORMAT

EXAMINE *location[:location]*

restrictions

Requires user mode read (R) and write (W) access to the location in virtual memory whose contents you want to examine.

PARAMETER

location[:location]

Specifies a virtual address or a range of virtual addresses whose contents you want to examine. If you specify a range of addresses, separate the first and last with a colon. The second address must be larger than the first.

You can specify locations using any valid arithmetic expression that contains arithmetic or logical operators or symbol names that have been previously given values with DCL assignment statements.

The DEPOSIT and EXAMINE commands maintain a pointer to the current memory location. The EXAMINE command sets this pointer to the last location examined when you specify an EXAMINE command. You can refer to this location using the period (.) in a subsequent EXAMINE or DEPOSIT command.

DESCRIPTION

When the EXAMINE command is executed, it displays the virtual memory address in hexadecimal format and the contents in the radix requested as follows:

```
address:  contents
```

If the address specified is not accessible to user mode, four asterisks are displayed in the contents field.

Radix Qualifiers: The radix default for a DEPOSIT or EXAMINE command determines how the command interprets numeric literals. The initial default radix is hexadecimal; all numeric literals in the command line are assumed to be hexadecimal values. If a radix qualifier modifies an EXAMINE command, that radix becomes the default for subsequent EXAMINE and DEPOSIT commands, until another qualifier overrides it. For example:

```
$ EXAMINE/DECIMAL 900
00000384:  0554389621
```

The EXAMINE command interprets the location 900 as a decimal number and displays the contents of that location in decimal. All subsequent DEPOSIT and EXAMINE commands assume that numbers you enter for addresses and data are decimal. Note that the EXAMINE command always displays the address location in hexadecimal format.

Symbol names defined by = (Assignment Statement) commands are always interpreted in the radix in which they were defined.

Note that hexadecimal values entered as examine locations or as data to be deposited must begin with a numeric character (0 through 9). Otherwise, the command interpreter assumes that you have entered a symbol name, and attempts symbol substitution.

You can use the radix operators %X, %D, or %O to override the current default when you enter the EXAMINE command. For example:

```
$ EXAMINE/DECIMAL %X900
00000900:  321446536
```

This command requests a decimal display of the data in the location specified as hexadecimal 900.

Length Qualifiers: The initial default length unit for the EXAMINE command is a longword. The EXAMINE command displays data, one longword at a time, with blanks between longwords. If a length qualifier modifies the command, that length becomes the default length of a memory location for subsequent EXAMINE and DEPOSIT commands, until another qualifier overrides it.

Restriction on Placement of Qualifiers: The EXAMINE command analyzes expressions arithmetically. Therefore, qualifiers are interpreted correctly only when they appear immediately after the command name.

QUALIFIERS

/ASCII

Requests that the data at the specified location be displayed in ASCII.

Binary values that do not have ASCII equivalents are displayed as periods (.).

When you specify /ASCII, or when ASCII mode is the default, hexadecimal is used as the default radix for numeric literals that are specified on the command line.

/BYTE

Requests that data at the specified location be displayed one byte at a time.

/DECIMAL

Requests that the contents of the specified location be displayed in decimal format.

/HEXADECIMAL

Requests that the contents of the specified location be displayed in hexadecimal format.

/LONGWORD

Requests that data at the specified location be displayed one longword at a time.

/OCTAL

Requests that the contents of the specified location be displayed in octal format.

/WORD

Requests that data at the specified location be displayed one word at a time.

EXAMPLES

1

```
$ RUN   MYPROG
^Y
$ EXAMINE   2678
0002678:  1F4C5026
$ CONTINUE
```

The RUN command begins execution of the image MYPROG.EXE. While MYPROG is running, CTRL/Y interrupts its execution, and the EXAMINE command requests a display of the contents of virtual memory location 2678 (hexadecimal).

2

```
$ BASE = %X1C00
$ READBUF = BASE + %X50
$ ENDBUF = BASE + %XA0
$ RUN   TEST
^Y
$ EXAMINE/ASCII READBUF:ENDBUF
00001C50:  BEGINNING OF FILE MAPPED TO GLOBAL SECTION
   .
   .
   .
```

Before executing the program TEST.EXE, symbolic names are defined for the program's base address, and for labels READBUF and ENDBUF; all are expressed in hexadecimal format using the radix operator %X. READBUF and ENDBUF define offsets from the program base.

While the program is executing, CTRL/Y interrupts it, and the EXAMINE command requests a display in ASCII of all data between the specified memory locations.

EXCHANGE

Invokes the VAX/VMS Exchange Utility (EXCHANGE) to manipulate mass storage volumes that are written in formats other than those normally recognized by VAX/VMS.

EXCHANGE allows you to perform any of the following tasks:

- Create foreign volumes
- Transfer files to and from the volume
- List directories of the volume

For block-addressable devices, such as RT–11 disks, EXCHANGE performs additional operations such as renaming and deletion of files. The Exchange Utility can also manipulate Files–11 files that are images of foreign volumes; these files are called *virtual devices.*

The /[NO]MESSAGE qualifier determines whether EXCHANGE displays information related to EXCHANGE INITIALIZE, MOUNT, and DISMOUNT subcommands. You can also use this qualifier with any of these three subcommands to reverse the default. Normally, EXCHANGE displays the information.

For a complete description of the VAX/VMS Exchange Utility, see the *VAX/VMS Exchange Utility Reference Manual.*

FORMAT

EXCHANGE *[subcommand] [file-spec] [file-spec]*

EXIT

Terminates processing of the current command procedure. If the command procedure was executed from within another command procedure, control returns to the calling procedure.

If a command procedure is not being executed, the EXIT command terminates the current image.

FORMAT

EXIT *[status-code]*

restrictions

None.

PARAMETER

status-code

Defines a numeric value for the reserved global symbol $STATUS. You can specify the status-code as an integer or an expression equivalent to an integer value. The value can be tested by the next outer command level. The low-order three bits of the longword integer value change the value of the reserved global symbol $SEVERITY.

If you specify a status-code, DCL will interpret 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 EXIT and STOP commands both provide a way to terminate the execution of a procedure. The EXIT command terminates execution of the current command procedure and returns control to the calling command level. If you issue the EXIT command from a noninteractive process (such as a batch job), at command level 0, then the process terminates.

The STOP command returns control to command level 0, regardless of the current command level. If you execute the STOP command from a command procedure or from a noninteractive process (such as a batch job), the process terminates.

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 an EXIT 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 three 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 zero through four.

Value	Severity
0	Warning
1	Success
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.

When any command procedure exits and returns control to another level, the command interpreter tests the current value of $STATUS. If $STATUS contains an even numeric value and if its high-order digit is 0, the command interpreter will display the system message associated with that status code, if one exists. (If no message exists, the message NOMSG will be displayed.) If the high order digit is 1, the message is not displayed.

When a command procedure exits following a warning or error condition that has already been displayed by a DCL command, the command interpreter sets the high-order digit of $STATUS to 1, leaving the remainder of the value intact. This ensures that error messages are not displayed by both the command that caused the error, and by the command procedure.

The EXIT command, when used after you interrupt an image with CTRL/Y, causes a normal termination of the image that is currently executing. If the image declared any exit-handling routines, they are given control. This is in contrast to the STOP command, which does not execute exit-handling routines. For this reason, the EXIT command is generally preferable to the STOP command.

EXAMPLES

1
```
$ ON WARNING THEN EXIT
$ FORTRAN 'P1'
$ LINK 'P1'
$ RUN 'P1'
```

The EXIT command is used as the target of an ON command; this statement ensures that the command procedure terminates whenever any warnings or errors are issued by any command in the procedure.

The procedure exits with the status value of the command or program that caused the termination.

2

```
$ START:
$       IF (P1 .EQS. "TAPE") .OR. (P1 .EQS. "DISK") THEN GOTO 'P1'
$       INQUIRE P1 "Enter device (TAPE or DISK)"
$       GOTO START
$ TAPE: !  Process tape files
   .
   .
   .
$       EXIT
$ DISK:  ! Process disk files
   .
   .
   .
$       EXIT
```

This command procedure shows how to use the EXIT command to terminate different command paths within the procedure. To execute the procedure, you must enter either TAPE or DISK as a parameter. The IF command uses a logical OR to test whether either of these strings was entered. If the result is true, the GOTO command branches to the corresponding label. If P1 was neither TAPE nor DISK, the INQUIRE command prompts for a correct parameter.

The commands following each of the labels TAPE and DISK provide different paths through the procedure. The EXIT command before the label DISK ensures that the commands after the label DISK are executed only if the procedure explicitly branches to DISK.

Note that the EXIT command at the end of the procedure is not required because the end of the procedure causes an implicit EXIT command. Use of the EXIT command, however, is recommended.

3

```
$ IF P1. EQS. "" THEN -
     INQUIRE P1 "Enter file-spec (null to exit)"
$ IF P1 .EQS. "" THEN EXIT
$ PRINT 'P1'/AFTER=20:00/COPIES=50/FORMS=6
```

This command procedure tests whether a parameter was passed to it; if the parameter was not passed, the procedure prompts for the required parameter. Then it retests the parameter P1. If a null string, indicated by a carriage return for a line with no data, is entered, the procedure exits. Otherwise, it executes the PRINT command with the current value of P1 as the input parameter.

4

```
$ IF P1 .EQS. "" THEN INQUIRE P1 "Code"
$ CODE = %X'P1'
$ EXIT CODE
```

This command procedure, E.COM, illustrates how to determine the system message, if any, associated with a hexadecimal system status code. The procedure requires a parameter and prompts if none is entered. Then it prefixes the value with the radix operator %X and assigns this string to the symbol CODE. Finally, it issues the EXIT command with the hexadecimal value. For example, if the procedure is in the file E.COM:

```
$ @E 1C
%SYSTEM-F-EXQUOTA, exceeded quota
```

When the procedure exits, the value of $STATUS is %X1C, which equates to the EXQUOTA message. Note that you can also use the F$MESSAGE lexical function to determine the message that corresponds to a status code.

EXIT

```
5    $ RUN MYPROG
     ^Y
     $ EXIT
```

The RUN command initiates execution of the image MYPROG.EXE. Then the CTRL/Y interrupts the execution. The EXIT command that follows calls any exit handlers declared by the image before terminating MYPROG.EXE.

GOSUB

Transfers control to a labeled subroutine in a command procedure.

FORMAT

GOSUB *label*

restrictions

None.

PARAMETER

label

Specifies a 1- through 255-alphanumeric character label appearing as the first item on a command line. A label may not contain embedded blanks. When the GOSUB command is executed, control passes to the command following the specified label.

The label can precede or follow the GOSUB statement in the current command procedure. When you use a label in a command procedure, it must be terminated with a colon.

DESCRIPTION

Use the GOSUB command in command procedures to transfer control to a subroutine specified by the label. If the command stream is not being read from a random access device (that is, a disk device), the GOSUB command performs no operation.

The RETURN command terminates the GOSUB subroutine procedure, returning control to the command following the calling GOSUB statement. The RETURN command accepts an optional status value.

The GOSUB command does not cause the creation of a new procedure level. Therefore, it is referred to as a "local" subroutine call. Any labels and local symbols defined in the current command procedure level are available to a subroutine invoked with a GOSUB command. The GOSUB command can be nested up to a maximum of 16 levels per procedure level.

As the command interpreter encounters labels, it enters them in a label table, which is allocated from space available in the local symbol table. If a label is encountered that already exists in the table, the new definition replaces the existing one. Therefore, if you use duplicate labels, control is always given to the label most recently read by DCL. In general:

- If duplicate labels precede and follow the GOSUB command, control is given to the label preceding the command.
- If duplicate labels all precede the GOSUB command, control is given to the most recent label, that is, the one nearest the GOSUB command.
- If duplicate labels all follow the GOSUB command, control is given to the one nearest the GOSUB command.

If a label does not exist in the current command procedure, the procedure cannot continue and is forced to exit.

Note that the amount of space available for labels is limited. If a command procedure uses many symbols and contains many labels, the command interpreter may run out of table space and issue an error message.

EXAMPLE

```
$!
$! GOSUB.COM
$!
$ SHOW TIME
$ GOSUB TEST1
$ WRITE SYS$OUTPUT "success completion"
$ EXIT
$!
$! TEST1 GOSUB definition
$!
$ TEST1:
$     WRITE SYS$OUTPUT "This is GOSUB level 1."
$     GOSUB TEST2
$     RETURN %X1
$!
$! TEST2 GOSUB definition
$!
$ TEST2:
$     WRITE SYS$OUTPUT "This is GOSUB level 2."
$     GOSUB TEST3
$     RETURN
$!
$! TEST3 GOSUB definition
$!
$ TEST3:
$     WRITE SYS$OUTPUT "This is GOSUB level 3."
$     RETURN
```

This command procedure shows how to use the GOSUB command to transfer control to labeled subroutines. The GOSUB command transfers control to the subroutine labeled TEST1. The procedure executes the commands in subroutine TEST1, branching to the subroutine labeled TEST2. The procedure then executes the commands in subroutine TEST2, branching to the subroutine labeled TEST3. Each subroutine is terminated by the RETURN command. After TEST3 is executed, the RETURN command returns control back to the command line following each calling GOSUB statement. At this point, the procedure has been successfully executed.

GOTO

Transfers control to a labeled statement in a command procedure.

FORMAT

GOTO *label*

restrictions

None.

PARAMETER

label

Specifies a 1- through 255-alphanumeric character label appearing as the first item on a command line. A label may not contain embedded blanks. When the GOTO command is executed, control passes to the command following the specified label.

The label can precede or follow the GOTO statement in the current command procedure. When you use a label in a command procedure, it must be terminated with a colon.

DESCRIPTION

Use the GOTO command in command procedures to transfer control to a line that is not the next line in the procedure. If the command stream is not being read from a random access device (that is, a disk device), the GOTO command performs no operation.

As the command interpreter encounters labels, it enters them in a label table, which is allocated from space available in the local symbol table. If a label is encountered that already exists in the table, the new definition replaces the existing one. Therefore, if you use duplicate labels, control is always given to the label most recently read by DCL. In general:

- If duplicate labels precede and follow the GOTO command, control is given to the label preceding the command.
- If duplicate labels all precede the GOTO command, control is given to the most recent label, that is, the one nearest the GOTO command.
- If duplicate labels all follow the GOTO command, control is given to the one nearest the GOTO command.

If a label does not exist in the current command procedure, the procedure cannot continue and is forced to exit.

Note that the amount of space available for labels is limited. If a command procedure uses many symbols and contains many labels, the command interpreter may run out of table space and issue an error message.

EXAMPLES

1

```
$ IF P1 .EQS. "HELP" THEN GOTO TELL
$ IF P1 .EQS. "" THEN GOTO TELL
    .
    .
    .
$ EXIT
$ TELL:
$ TYPE SYS$INPUT
To use this procedure, you must enter a value for P1.
    .
    .
    .
$ EXIT
```

The IF command checks the first parameter passed to the command procedure; if this parameter is the string HELP or if the parameter is not specified, the GOTO command is executed and control is passed to the line labeled TELL. Otherwise, the procedure continues executing until the EXIT command is encountered. At the label TELL, a TYPE command displays data in the input stream that documents how to use the procedure.

2

```
$ ON ERROR THEN GOTO CHECK
    .
    .
    .
$ EXIT
$ CHECK:  ! Error handling routine
    .
    .
    .
$ END:
$ EXIT
```

The ON command establishes an error-handling routine. If any command or procedure subsequently executed in the command procedure returns an error or severe error return, the GOTO command transfers control to the label CHECK.

HELP

Displays information from the system HELP files or from any HELP library you specify.

FORMAT

HELP *[keyword ...]*

restrictions

None.

PARAMETER

keyword ...

Specifies one or more keywords that refer to the topic or subtopic on which you want information from a HELP library. Information within HELP libraries is arranged in a hierarchical manner. The levels are:

1. None—If you do not specify a keyword, HELP describes the HELP command and lists the topics that are documented in the root library. Each item in the list is a keyword in the first level of the hierarchy.
2. Topic-name—If you specify a keyword by naming a topic, HELP describes the topic as it is documented in either the root library or one of the other enabled default libraries. Keywords for additional information available on this topic are listed.
3. Topic-name subtopic—If you specify a subtopic following a topic, HELP provides a description of the specified subtopic.
4. @file-spec followed by any of the above—If you specify a HELP library to replace the current root library, HELP searches that library for a description of the topic or subtopic specified. The file specification must take the same form as the file specification included with the /LIBRARY command qualifier. However, if the specified library is an enabled user-defined default library, the file specification can be abbreviated to any unique leading substring of that default library's logical name translation.

If you use an asterisk in place of any keyword, the HELP command displays all information available at the level that the asterisk replaces. For example, HELP COPY * displays all the subtopics under the topic COPY.

If you use an ellipsis immediately after any primary keyword, HELP displays all the information on the specified topic and all subtopics of that topic. For example, HELP COPY... displays information on the COPY topic as well as information on all the subtopics under COPY.

You can use percent signs and asterisks in the keyword as wildcard characters.

DESCRIPTION

To use the HELP facility in its simplest form, issue the HELP command from your terminal. HELP displays a list of topics at your terminal and the prompt Topic?. If you want to see information on one of the topics, type the topic name after the prompt. The system will display information on that topic.

If the topic has subtopics, HELP will list the subtopics and display the Subtopic? prompt. If you want information on one of the subtopics, type the name after the prompt. If you want information on another topic, press RETURN. You can now ask for information on another topic when HELP displays the Topic? prompt. Or if you are finished with the HELP facility and want to exit, you can press RETURN again to return to the DCL command level.

QUALIFIERS

/INSTRUCTIONS (default)
/NOINSTRUCTIONS

Controls whether the HELP command displays information on how to use the HELP facility. By default, the HELP command display includes a description of the facility and the format, along with the list of topics. If you specify /NOINSTRUCTIONS, only the list of topics is displayed.

/LIBLIST (default)
/NOLIBLIST

Controls whether a list of additional HELP libraries available default HELP libraries is output at the topic level.

/LIBRARY=file-spec
/NOLIBRARY

Controls whether an alternate HELP library is used in place of the default system library, SYS$HELP:HELPLIB.HLB. The specified library is used as the main HELP library, and is searched for HELP information before any user-defined default HELP libraries are checked.

If you omit the device and directory specification, the default is SYS$HELP, the logical name of the location of the system HELP libraries. The default file type is HLB.

The /NOLIBRARY qualifier is used to exclude the default HELP library from the library search order.

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT with a partial file specification (for example, /OUTPUT=[JONES]), HELP is the default file name and LIS the default file type. If you enter a file specification, it may not include any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

/PAGE (default)
/NOPAGE

Controls whether output to the screen stops after each screen full ("page") of information is displayed.

If you specify /NOPAGE, output continues until the information display ends or until you manually control the scrolling.

/PROMPT (default)
/NOPROMPT

Controls whether HELP initiates an interactive session once the initial HELP request has been processed. If you specify /NOPROMPT, HELP returns you to the DCL command level after displaying the requested information.

If /PROMPT is in effect, one of four different prompts is displayed, requesting you to specify a particular HELP topic or subtopic. Each prompt represents a different level in the hierarchy of HELP information. The four prompt levels are:

1. Topic?—The root library is the main library and you are not currently examining HELP for a particular topic.
2. [library-spec] Topic?—The root library is a library other than the main library and you are not currently examining HELP for a particular topic.
3. [keyword] subtopic?—The root library is the main library and you are currently examining HELP for a particular topic (and/or subtopic).
4. A combination of 2 and 3.

When you encounter one of these prompts, you can enter any one of the responses described below:

Response	Action in the Current Prompt Environment
keyword[...]	(1,2) Searches all enabled libraries for the keyword. (3,4) Searches additional HELP libraries for the current topic (and /or subtopic) for the keyword.
@file-spec keyword[...]	(1,2) Same as above, except that the library specified by @file-spec is now the root library. If the specified library does not exist, HELP treats @file-spec as a normal keyword. (3,4) Same as above, treats @file-spec as a normal keyword.
?	(1,2) Displays a list of topics available in the root library. (3,4) Displays the list of subtopics of the current topic (and /or subtopics) for which HELP exists.
<RET>	(1) Exits from HELP. (2) Changes root library to main library. (3,4) Prompts for a topic or subtopic at the next higher level.
CTRL/Z	(1,2,3,4) Exits from HELP.

/USERLIBRARY=(table[,...])
/NOUSERLIBRARY

Controls whether HELP searches one or more user-specified root libraries after it has searched the root HELP library.

When you specify the /USERLIBRARY qualifier, HELP searches the process, group, and system logical name tables for logical names of the form: HLP$LIBRARY, HLP$LIBRARY_1, HLP$LIBRARY_2, and so on, to find the file specifications of any user-defined libraries. When HELP searches for information on a given topic, it proceeds in the following search order: root library, main library (if not the same as the root library), process libraries,

group libraries, system libraries, and root library. If the search fails, the root library is searched a second time so that the context is returned to the root library from which the search was initiated.

If you specify only one table for HELP to search, you can omit the parentheses. You can specify any of the following tables or keywords:

ALL	HELP searches the process, group, and system logical name tables for user-defined library definitions.
NONE	HELP does not search any of the logical name tables. This specification is the same as /NOUSERLIBRARY.
PROCESS	HELP searches the process logical name table for user-defined library definitions.
GROUP	HELP searches the group logical name table for user-defined library definitions.
SYSTEM	HELP searches the system logical name table for user-defined library definitions.

By default, if you omit the qualifier, /USERLIBRARY=ALL is assumed.

EXAMPLES

1

```
$ HELP
HELP
  .
  .  (HELP message text and list of topics)
  .
Topic?
```

Issuing the HELP command without any qualifiers or parameters produces a display of the HELP topics available from the root HELP library, SYS$HELP:HELPLIB.HLB.

If you type one of the listed topics in response to the Topic? prompt, HELP displays information about that topic and a list of subtopics (if there are any). If one or more subtopics exist, HELP will prompt you for a subtopic.

```
Topic? ASSIGN
ASSIGN
  .
  .  (HELP message text and subtopics)
  .
ASSIGN Subtopic?
```

If you type a subtopic name, HELP displays information about that subtopic:

```
ASSIGN Subtopic? Name
ASSIGN
  Name
    .
    .  (HELP message text and subtopics, if any)
    .
ASSIGN Subtopic?
```

If one or more sub-subtopics exist, HELP will prompt you for a sub-subtopic; otherwise, as in the above example, the facility will prompt you for another subtopic of the topic you are currently inspecting.

Typing a question mark will redisplay the HELP message and options at your current level. Pressing RETURN will either: (1) move you back to the previous HELP level if you are in a subtopic level, or (2) terminate HELP if you are at the first level. Pressing CTRL/Z terminates HELP at any level.

```
2  $ HELP/NOPROMPT ASSIGN/GROUP
     .
     .  (ASSIGN/GROUP HELP message)
     .
   $
   $ HELP/NOPROMPT/PAGE EDIT *
     .
     .  (HELP messages on all first-level EDIT subtopics)
     .
   $
```

You can get HELP on a specific topic or subtopic, or on a range of topics or subtopics. You can also choose not to be prompted for additional topics or subtopics.

The two HELP commands request HELP on specific topics. In each case, HELP displays the HELP message you request and then returns you to DCL command level and the dollar sign prompt.

The first command requests HELP on the /GROUP qualifier of the ASSIGN command. The asterisk in the second example is a wildcard character. It signals HELP to display information about all EDIT subtopics, which HELP then displays in alphabetical order. The /NOPROMPT qualifier suppresses prompting in both sample commands. The /PAGE qualifier on the second HELP command causes output to the screen to stop after each screenful of information is displayed.

```
3  $ HELP FILL
   Sorry, no documentation on FILL
   Additional information available:
     .
     .  (list of first-level topics )
     .
   Topic? @EDTHELP FILL
   FILL
     .
     .  (FILL HELP message)
     .
   @EDTHELP Topic?
```

When you issue a request for HELP on a topic that is not in the default HELP library, you can instruct HELP to search another HELP library for the topic. In this example, entering the command @EDTHELP FILL instructs HELP to search the HELP library SYS$HELP:EDTHELP.HLB for information on FILL, an EDT editor command. HELP displays the message and prompts you for another EDT editor topic.

```
4   $ DEFINE HLP$LIBRARY EDTHELP
    $ DEFINE HLP$LIBRARY_1 MAILHELP
    $ DEFINE HLP$LIBRARY_2 BASIC
    $ DEFINE HLP$LIBRARY_3 DISK2:[MALCOLM]FLIP
    $ HELP REM
```

You can use logical names to define libraries for HELP to search automatically if it does not find the specified topic in the VAX/VMS root HELP library. This sequence of commands instructs HELP to search libraries besides the default root library, SYS$HELP:HELPLIB.HLB.

The four DEFINE statements create logical names for the four user-defined HELP libraries that HELP is to search after it has searched the root library. The first three entries are HELP libraries in the directory. HELP searches by default for user-defined HELP libraries, SYS$HELP. The fourth is the HELP library FLIP.HLB in the directory DISK2:[MALCOLM]. Note that the logical names that you use to define these HELP libraries must be numbered consecutively; that is, you cannot skip any numbers.

HELP first searches the root library for REM. It then searches the libraries HLP$LIBRARY, HLP$LIBRARY_1, HLP$LIBRARY_2, and so on, until it finds REM or exhausts the libraries it knows it can search. When it finds REM in the BASIC.HLB library, it displays the appropriate HELP information and prompts you for a subtopic in that library. If you request HELP on a topic not in the BASIC.HLB library, HELP once again searches the HELP libraries you have defined.

IF

Tests the value of an expression and executes the command following the THEN keyword if the test is true.

FORMAT

IF *expression THEN [$] command*

restrictions

None.

PARAMETERS

expression

Defines the test to be performed. The expression can consist of one or more numeric constants, string literals, symbolic names, or lexical functions separated by logical, arithmetic, or string operators.

Expressions in IF commands are automatically evaluated during the execution of the command. All character strings beginning with alphabetic characters that are not enclosed in quotation marks are assumed to be symbol names or lexical functions, and the IF command replaces them with their current values.

Symbol substitution in expressions in IF commands is not iterative; that is, each symbol is replaced only once. However, if you want iterative substitution, you can precede a symbol name with an apostrophe or ampersand.

The command interpreter does not execute an IF command when it contains an undefined symbol. Instead, the command interpreter issues a warning message and executes the next command in the procedure.

For a summary of operators and details on how to specify expressions, see Section 5.7 of the *VAX/VMS DCL Concepts Manual.*

command

Defines the action to take if the result of the expression is true.

You can specify any valid DCL command following the THEN keyword. Optionally, you can precede the command with a dollar sign.

DESCRIPTION

The IF command tests the value of an expression and executes a given command if the result of the expression is true. The expression is true if the result:

- Has an odd integer value
- Has a character string value that begins with any of the letters Y, y, T, or t
- Has an odd numeric string value

The expression is false if the result:

- Has an even integer value
- Has a character string value that begins with any letter except Y, y, T, or t
- Has an even numeric string value

If the expression is false, the THEN command string is not executed and the procedure continues with the next command line.

EXAMPLES

1
```
$ COUNT = 0
$ LOOP:
$ COUNT = COUNT + 1
    .
    .
    .
$ IF COUNT .LE. 10 THEN GOTO LOOP
$ EXIT
```

This example shows how to establish a loop in a command procedure, using a symbol named COUNT and an IF statement. The IF statement checks the value of COUNT and performs an EXIT command when the value of COUNT is greater than 10.

2
```
$ IF P1 .EQS. "" THEN GOTO DEFAULT
$ IF (P1 .EQS. "A") .OR. (P1 .EQS. "B") THEN GOTO 'P1'
$ WRITE SYS$OUTPUT "Unrecognized parameter option ''P1' "
$ EXIT
$ A:        !  Process option a
    .
    .
    .
$ EXIT
$ B:        !  Process option b
    .
    .
    .
$ EXIT
$ DEFAULT: !  Default processing
    .
    .
    .
$ EXIT
```

This example shows a command procedure that tests whether a parameter was passed. The GOTO command passes control to the label specified as the parameter.

If the procedure is executed with a parameter, the procedure uses that parameter to determine the label to branch to. For example:

```
$ @TESTCOM A
```

When the procedure executes, it determines that P1 is not null, and branches to the label A. Note that the EXIT command causes an exit from the procedure before the label B.

```
3    $ SET NOON
        .
        .
        .
     $ LINK CYGNUS,DRACO,SERVICE/LIBRARY
     $ IF .NOT. $STATUS THEN EXIT
     $ RUN CYGNUS
```

This command procedure uses the SET NOON command to disable error checking by the command procedure. After the LINK command, the IF command tests the value of the reserved global symbol $STATUS. If the LINK command returns an error status value, the command procedure exits.

INITIALIZE

Formats and writes a label on a mass storage volume.

FORMAT

INITIALIZE *device-name[:] volume-label*

restrictions

Requires volume protection (VOLPRO) privilege for most INITIALIZE operations.

PARAMETERS

device-name[:]

Specifies the name of the device on which the volume to be initialized is physically mounted.

The device does not have to be currently allocated; however, allocating the device before initializing it is the recommended practice.

volume-label

Specifies the identification to be encoded on the volume. For a disk volume, you can specify a maximum of 12 alphanumeric characters; for a magnetic tape volume, you can specify a maximum of 6 alphanumeric characters. Letters are automatically changed to uppercase. Nonalphanumeric characters are not allowed in the volume-label specification on disk.

In order to use ANSI "a" characters on the volume-label on magnetic tape, the volume name must be enclosed in quotation marks. For an explanation of ANSI "a" characters, see the description of the /LABEL qualifier.

DESCRIPTION

The default format for disk volumes in the VAX/VMS operating system is called the Files–11 Structure Level 2. The default for magnetic tape volumes is based on Level 3 of the ANSI standard for magnetic tape labels and file structure for informational interchange (ANSI X3.27-1978).

The INITIALIZE command can also initialize disk volumes in the Files–11 Structure Level 1 format.

You do not need special privileges to override logical protection on the following devices:

- A blank disk or magnetic tape volume; that is, a volume that has never been written
- A disk volume that is owned by your current UIC or by the UIC [0,0]
- A magnetic tape volume that allows write access to your current UIC that was not protected when it was initialized

In all other cases, you must have the volume protection (VOLPRO) privilege to initialize a volume.

When the INITIALIZE command initializes a magnetic tape volume, it always attempts to read the volume. A blank magnetic tape can sometimes cause unrecoverable errors, such as the following:

- The message:

  ```
  %INIT-F-VOLINV, volume is invalid
  ```

- A runaway magnetic tape (this frequently occurs with new magnetic tapes that have never been written or that have been run through verifying machines). You can stop a runaway magnetic tape only by setting the magnetic tape drive off line and then putting it back on line.

If this type of unrecoverable error occurs, you can successfully initialize a magnetic tape by repeating the INITIALIZE command from an account that has the volume protection (VOLPRO) privilege and specifying the following qualifier in the command:

```
/OVERRIDE=(ACCESSIBILITY,EXPIRATION)
```

This qualifier ensures that the INITIALIZE command will not attempt to verify any labels on the magnetic tape.

Many of the INITIALIZE command qualifiers allow you to specify parameters that can maximize input/output efficiency.

QUALIFIERS

/ACCESSED=n

Requires OPER (operator) privilege.

Specifies, for disk volumes, the number of directories to be maintained in system space for ready access.

Legal values for n are 0 through 255. If /ACCESSED is not specified, the INITIALIZE command uses the default value of 3. This value is ignored by VAX/VMS, but is present for use on RSX systems.

/BADBLOCKS=(area[,...])

Specifies, for disk volumes, specific areas on the volume that are faulty. The INITIALIZE command marks the areas as allocated so that no data will be written in them.

You can specify one or more areas, using one or both of the formats shown below. If you specify only one area, you can omit the parentheses.

lbn[:count]	Specifies a logical block number on the disk volume, and optionally a count of logical blocks beginning with the logical block specified, to be marked as allocated
sector.track.cyl[:count]	Specifies a specific sector, track, and cylinder on the disk volume, and optionally a count of blocks, beginning with the first block specified, to be marked as allocated

All media supplied by DIGITAL and supported on the VAX/VMS operating system, except floppy disks and TU58 cartridges, are factory formatted and contain bad block data. The Bad Block Locator Utility (BAD) or the diagnostic formatter EVRAC can be used to refresh the bad block data or to construct it for the media exceptions above. The /BADBLOCKS qualifier is necessary

only to enter bad blocks that are not identified in the volume's bad block data.

Digital Storage Architecture (DSA) disks (for example, disks attached to UDA-50 and HSC50 controllers) have bad blocks handled by the controller, and appear logically perfect to the file system.

For information on how to run BAD, see the *VAX/VMS Bad Block Locator Utility Reference Manual*.

/CLUSTER_SIZE=n

Defines, for disk volumes, the minimum allocation unit, in blocks. The maximum size you can specify for a volume is one-hundredth the size of the volume; the minimum size you can specify is calculated with the formula:

```
disk size  (number of blocks)
---------
255*4096
```

For Files–11 Structure Level 2 disks, the cluster size default depends on the disk capacity; disks that are 50,000 blocks or larger have a default cluster size of 3, while those smaller than 50,000 blocks have a default value of 1.

For Files–11 Structure Level 1 disks the cluster size must always be 1.

/DATA_CHECK[=(option[,...])]

Defines a default for data check operations following all reads and/or writes to the volume. You can specify one or both of the following options:

READ	Performs checks following all read operations
WRITE	Performs checks following all write operations

If you specify /DATA_CHECK without specifying an option, the system assumes the default of /DATA_CHECK=WRITE. If you do not specify /DATA_CHECK, the system performs no checking as the default. You can override the checking you specify at initialization for disks when you issue a MOUNT command to mount the volume.

If you specify only one option, you can omit the parentheses.

/DENSITY=density-value

For magnetic tape volumes, specifies the density in bytes per inch (bpi) at which the magnetic tape is to be written.

For floppy disk volumes that are to be initialized on RX02 dual-density disk drives, specifies the density at which the floppy disk is to be formatted.

For magnetic tape volumes, specifies the density in bytes per inch (bpi) at which the magnetic tape is to be written. For magnetic tape volumes, the density value specified can be 800 bpi, 1600 bpi, or 6250 bpi, as long as the density is supported by the magnetic tape drive. If you do not specify a density value for a blank magnetic tape, the system uses a default density of the highest value allowed by the tape drive. If the drive allows 6250, 1600, and 800 bpi operation, the default density is 6250. If the drive allows only 1600 and 800 bpi operation then the default density is 1600. If you do not specify a density value for a magnetic tape that has been previously written, the system uses the density of the first record on the volume. The magnetic tape density will not default on an unusually short record.

RX02 dual-density disk drives allow floppy disks to be initialized at single or double density. To specify single-density formatting of a floppy disk, specify the density value SINGLE. To specify double-density formatting of a floppy disk, specify the density value DOUBLE.

If you do not specify a density value for a floppy disk being initialized on an RX02 drive, the system leaves the volume at the density to which the volume was last formatted. Floppy disks purchased from DIGITAL are formatted in single density.

Note: Floppy disks formatted in double density cannot be read or written by the console block storage device (an RX01 drive) of a VAX/780 until they have been reformatted in single density.

/DIRECTORIES=n

Specifies, for disk volumes, the number of entries to preallocate for user directories.

The legal values are in the range 16 through 16000; if you do not specify a value, the INITIALIZE command uses the default value of 16.

/ERASE
/NOERASE (default)

Controls the Data Security Erase (DSE) operation on the volume before initializing it. The /ERASE qualifier applies to ODS-2 disk and ANSI magnetic tape volumes, and is valid for magnetic tape devices that support the hardware erase function, such as TU78 and MSCP magnetic tapes.

If you specify /ERASE, a DSE operation is performed on the volume. For disk devices, the ERASE volume attribute is set. In effect, each file on the volume is erased when it is deleted.

Note that the amount of time taken by the DSE operation depends on the volume size; INITIALIZE/ERASE is always slower than INITIALIZE /NOERASE.

/EXTENSION=n

Affects Files-11 Structure Level 1 disks ONLY

Specifies, for disk volumes, the number of blocks to use as a default extension size for all files on the volume. The extension default is used when a file increases to a size greater than its initial default allocation during an update.

You can specify a value in the range 0 through 65,535. If you do not specify a default extension size, the INITIALIZE command uses a value of 5. Note that this attribute is not used when the volume is being used on a VAX/VMS system, but is provided to control the process's use of the volume on RSX-11M systems. In VAX/VMS, the default file extension is specified using the SET RMS command.

/FILE_PROTECTION=code

Affects Files-11 Structure Level 1 disks ONLY

Defines, for disk volumes, the default protection to be applied to all files on the volume.

Specify the code according to the standard syntax rules for specifying protection given in Section 7.1.2 of the *VAX/VMS DCL Concepts Manual*. Any attributes not specified are taken from the current default protection.

Note that this attribute is not used when the volume is being used on a VAX/VMS system, but is provided to control the process's use of the volume on RSX–11M systems. VAX/VMS systems always use the default file protection. Use the SET PROTECTION/DEFAULT command to change the default file protection.

/GROUP

Defines a disk volume as a group volume. The owner UIC of the volume defaults to the group number of the user issuing the command and a member number of 0.

If this qualifier is specified in conjunction with the /NOSHARE qualifier, the volume protection is RWED for the system, owner, and group. However, the /GROUP qualifier specified alone defines the volume protection as RWED for all user categories.

/HEADERS=n

Specifies, for disk volumes, the number of file headers to be allocated initially for the index file. The minimum value you can specify is 16; the maximum value is the value set with the /MAXIMUM_FILES qualifier.

By default, the INITIALIZE command allocates 16 file headers.

/HIGHWATER (default)
/NOHIGHWATER

Controls highwater marking on the specified volume. This qualifier is applicable to ODS-2 disk volumes only.

If you specify /HIGHWATER, INITIALIZE sets the file highwater mark (FHM) volume attribute. FHM is a mechanism that guarantees that a user cannot read data that he has not written.

The /NOHIGHWATER qualifier disables FHM for the volume.

/INDEX=position

Requests, for disk volumes, that the index file for the volume's directory structure be placed in a specific location on the volume.

You can specify one of the following options:

BEGINNING	Places the index file at the beginning of the volume
END	Places the index file at the end of the volume
MIDDLE	Places the index file in the middle of the volume
BLOCK:n	Places the index file at the beginning of the logical block specified by the logical block number n

By default, the INITIALIZE command places the index file in the middle of the volume.

/LABEL=option

Defines characteristics for the magnetic tape volume label, as directed by the included option. The available options are as follows:

- OWNER_IDENTIFIER:"(14 ANSI characters)"

  Allows you to specify the Owner Identifier field in the volume label. The field specified can accept up to 14 ANSI characters.

- VOLUME_ACCESSIBILITY:"character"

  Specifies the character to be written in the volume accessibility field of the VAX/VMS ANSI volume label VOL1 on an ANSI magnetic tape. The character may be any valid ANSI "a" character. This set of characters includes numeric characters, uppercase letters, and any one of the following nonalphanumeric characters:

  ```
  ! " % ' ( ) * + , - . / : ; < = > ?
  ```

  By default, VAX/VMS provides a routine that checks this field in the following manner.

  - If the magnetic tape was created on a version of VAX/VMS that conforms to Version 3 of ANSI, then this option must be used to override any character other than an ASCII space.
  - If a VAX/VMS protection is specified and the magnetic tape conforms to an ANSI standard that is later than Version 3, then this option must be used to override any character other than an ASCII 1.

  If you specify any character other than the default, you must specify the /OVERRIDE=ACCESSIBILITY qualifier on the INITIALIZE and MOUNT commands in order to access the magnetic tape.

/MAXIMUM_FILES=n

Restricts, for disk volumes, the maximum number of files that the volume can contain (overriding the default value). The default is calculated from the volume size in blocks as follows:

```
     volume size
-----------------------
(cluster factor + 1) *2
```

The maximum size you can specify for any volume is:

```
    volume size
--------------------
(cluster factor + 1)
```

The minimum value is 0. Note, however, that you should specify a low file maximum only after careful consideration. Once set, the maximum can be increased only by reinitializing the volume.

/OVERRIDE=(option[,...])

Requests the INITIALIZE command to ignore data on a magnetic tape volume that protects it from being overwritten. You may specify one or more of the following options:

ACCESSIBILITY

For magnetic tapes only. If the installation allows, this option overrides any character in the Accessibility Field of the volume. The necessity of this option is defined by the installation. That is, each installation has the option of specifying a routine that the magnetic tape file system will use to process this field. By default, VAX/VMS provides a routine that checks this field in the following manner:

- If the magnetic tape was created on a version of VAX/VMS that conforms to Version 3 of ANSI, then this option must be used to override any character other than an ASCII space.
- If a VAX/VMS protection is specified and the magnetic tape conforms to an ANSI standard that is later than Version 3, then this option must be used to override any character other than an ASCII 1.

The ACCESSIBILITY option, requires the user privilege VOLPRO or volume ownership.

EXPIRATION

Overrides the expiration date on the volume (as indicated by the expiration date of the first file on the volume), and of any files on the volume.

OWNER_IDENTIFIER

Allows you to override processing of the Owner Identifier field in the volume label. You may need to do this for magnetic tapes that were created before VAX/VMS Version 4.0 on DIGITAL operating systems using the D% format in the volume Owner Identifier field.

If you specify only one option, you can omit the parentheses.

In order to initialize a volume that was initialized previously with the /PROTECTION qualifier, your UIC must match the UIC written on the volume, or you must have VOLPRO privilege.

/OWNER_UIC=uic

Specifies the user identification code to be assigned ownership of the volume and files. Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

If you do not specify /OWNER_UIC, your current UIC is assigned ownership of the disk volume.

For magnetic tapes, no UIC is written unless protection on the magnetic tape is specified. If protection is specified, but no owner UIC is specified, your current UIC is assigned ownership of the volume.

/PROTECTION=code

Specifies the protection to be applied to the volume. The protection controls who can read, write, create, and delete files on the volume. If you do not specify a protection code, protection defaults to all types of access to all user categories. Note that the /GROUP, /SHARE, and /SYSTEM qualifiers can also be used to define protection for disk volumes.

For magnetic tape, the protection code is written to a VAX/VMS-specific volume label. The system only applies read and write access restrictions; execute and delete access are meaningless. Moreover, the system and the owner are always given both read and write access to magnetic tapes, regardless of what you specify in a protection code.

Specify the code according to the standard syntax rules for specifying protection given in Section 7.1.2 of the *VAX/VMS DCL Concepts Manual*.

When you specify a protection code for an entire disk volume, access type E (execute) indicates create access.

/SHARE (default)
/NOSHARE

Controls whether a disk volume is shareable. The protection code for the volume defaults to all types of access for all user categories. If you specify /NOSHARE, the protection code defaults to no access for group and world, unless /GROUP has been specified.

/STRUCTURE=level

Specifies, for disk volumes, whether the volume should be formatted in Files–11 Structure Level 1 or Structure Level 2. By default, disk volumes are formatted in Files–11 Structure Level 2.

If you specify /STRUCTURE=1, the /CLUSTER_SIZE and /DATA_CHECK qualifiers are not allowed. The default protection for a Structure Level 1 disk is all types of access to system, owner, and group, and R (read) access to all other users.

/SYSTEM

Defines a disk volume as a system volume. The owner UIC of the volume defaults to [1,1] and default protection provides all types of access to the volume to all users.

No user privilege is required to use the /SYSTEM qualifier; however, only users with system UICs can create directories on system volumes.

/USER_NAME=string

Specifies, for disk volumes, a user name of up to 12 characters to be recorded on the volume. If /USER_NAME is not specified, the INITIALIZE command takes the user name under which you logged in.

/VERIFIED
/NOVERIFIED

Indicates, for disk volumes, whether the disk has bad block data on it. The default is /VERIFIED for disks with 4096 blocks or more; the INITIALIZE

command assumes that disks contain bad block data, and uses the data to mark the bad blocks as allocated. Use /NOVERIFIED to request INITIALIZE to ignore bad block data on the disk. (The default is /NOVERIFIED for disks with less than 4096 blocks.)

/WINDOWS=n

Specifies, for disk volumes, the number of mapping pointers to be allocated for file windows. When a file is opened, the file system uses the mapping pointers to access data in the file. You can specify a value in the range of 7 through 80. The default number of pointers is 7.

EXAMPLES

1
```
$ ALLOCATE DMA2:  TEMP
  _DMA2: ALLOCATED
$ INITIALIZE  TEMP:  BACK_UP_FILE
$ MOUNT  TEMP:  BACK_UP_FILE
%MOUNT-I-MOUNTED, BACK_UP_FILE mounted on _DMA2:
$ CREATE/DIRECTORY  TEMP:[ARCHIE]
```

The previous sequence of commands shows how to initialize an RK06 /RK07 volume. First, the device is allocated, to ensure that no one else can access it. Then, when the volume is physically mounted on the device, the INITIALIZE command initializes it. When the volume is initialized, the MOUNT command makes the file structure available. Before you can place any files on the volume, you must create a directory, as shown by the CREATE/DIRECTORY command.

2
```
$ ALLOCATE MT:
  _MTB1:  ALLOCATED
$ INITIALIZE MTB1:  SOURCE
$ MOUNT MTB1:  SOURCE
%MOUNT-I-MOUNTED, SOURCE mounted on _MTB1:
$ COPY *.FOR  MTB1:
$ DIRECTORY MTB1:
    .
    .
    .
$ DISMOUNT MTB1:
```

These commands show the procedure necessary to initialize a magnetic tape. After allocating a drive, the magnetic tape is loaded on the device, and the INITIALIZE command writes the label SOURCE on it. Then, the MOUNT command mounts the magnetic tape so that files can be written on it.

INITIALIZE/QUEUE

Creates or initializes queues. You use this command to create queues and to assign them names and attributes.

When you create a batch queue, the qualifier /BATCH is required.

FORMAT

INITIALIZE/QUEUE *queue-name[:]*

restrictions

This command requires operator (OPER) privilege.

PARAMETERS

queue-name[:]

Specifies the name of an execution queue or a generic queue.

DESCRIPTION

Initializing a Queue

Printer and batch queues are normally created by entering the necessary INITIALIZE command in a site-specific system start-up command procedure. However, once the system is running, you can use the INITIALIZE/QUEUE command to create additional queues as they are needed. The INITIALIZE /QUEUE command can also be used to update existing queue parameters of a stopped queue without affecting jobs in the queue.

When you want to change a queue parameter for an existing queue, it is normally easier to use the SET QUEUE command. Use the INITIALIZE /QUEUE command when you need to change queue parameters that are unavailable with SET QUEUE.

If you find that you need to initialize an existing queue, do the following: (1) stop the queue (use the STOP/QUEUE/NEXT command); (2) initialize the queue; and (3) start the queue. Once a queue has been stopped, you can specify new parameters to replace existing queue attributes. Unspecified parameters mean that these queue attributes remain as they were when the queue was previously initialized, started, or set.

If you want to start the queue at the same time you initialize it, you can use the /START qualifier. Alternately, you can issue only the INITIALIZE /QUEUE command to get the queue ready and then later issue the START /QUEUE command to begin queue operations.

If the specified queue is already running, the INITIALIZE/QUEUE command is ignored. Use SET QUEUE to change the attributes of a running queue. Note that initializing an existing queue does not delete any jobs currently in that queue. Any new queue settings established by the new INITIALIZE /QUEUE command apply to all jobs waiting in the queue or subsequently entering the queue. Any jobs that are executing in the queue when it was stopped complete their execution under the old settings.

The following qualifiers apply to generic and execution queues:

/OWNER_UIC
/PROTECTION

/[NO]RETAIN
/[NO]START

These qualifiers apply to all types of execution queues:

/BASE_PRIORITY
/[NO]CHARACTERISTICS
/[NO]ENABLE_GENERIC
/ON
/WSDEFAULT
/WSEXTENT
/WSQUOTA

Qualifiers that apply only to batch execution queues are:

/[NO]BATCH
/CPUDEFAULT
/CPUMAXIMUM
/[NO]DISABLE_SWAPPING
/JOB_LIMIT

Qualifiers that apply only to printer, terminal, or server queues are:

/[NO]BLOCK_LIMIT
/[NO]DEFAULT
/FORM_MOUNTED
/[NO]LIBRARY
/[NO]PROCESSOR
/[NO]RECORD_BLOCKING
/SCHEDULE
/[NO]SEPARATE

The /[NO]GENERIC qualifier is used to distinguish a generic queue from an execution queue.

The /TERMINAL qualifier can be used only with generic terminal queues.

Types of Queues

There are several different types of queues on the system. In general, queues can be divided into two major types: generic and execution. When a job is sent to an execution queue, it is executed in that queue; on the other hand, no processing takes place in generic queues. Generic queues hold jobs that will execute on an execution queue when one is available.

There are four types of generic queues:

- Generic batch queue
- Generic printer queue
- Generic terminal queue
- Generic server queue

Generic batch queues hold batch jobs for processing on batch execution queues. Generic printer queues hold print jobs for processing on printer execution queues. Generic terminal queues hold print jobs for processing on terminal execution queues. Generic server queues hold jobs for execution on a server queue that has been defined with the same processor.

The system manager or operator uses the /GENERIC qualifier to specify which execution queues can be accessed by a generic queue. The /ENABLE_GENERIC qualifier can be used when initializing an execution queue to enable jobs to be placed in that queue by a generic queue even if that exeucution queue name was not specified with the /GENERIC qualifier of the generic queue.

There are several types of execution queues:

- Execution batch queue
- Printer queue
- Terminal queue
- Execution server queue

Execution batch queues execute batch jobs. Batch jobs are those that request the execution of one or more command procedures in a batch process.

Output queues execute print jobs. A print job requests the processing of one or more files by a symbiont executing in a symbiont process. The default system symbiont is designed to print files on hardcopy devices (printers or terminals). Customer-written symbionts can be designed for this or any other file processing activity.

Output queues include both printer and terminal execution queues. These execution queues execute print jobs: printer queues execute the jobs on print devices; terminal queues execute the jobs at terminals that have been designed for receiving print jobs.

Execution server queues execute jobs using the server processor specified with the /PROCESSOR qualifier. Server queue processors are customer-written.

Another type of queue is the logical queue. A logical queue is a special type of generic queue in that it can place work only into the execution queue specified in the ASSIGN/QUEUE command. The logical queue's relation to an execution queue remains in effect until a DEASSIGN/QUEUE command is issued which negates or changes the assignment.

QUALIFIERS

/BASE_PRIORITY=n

Specifies the base process priority at which jobs are initiated from a batch queue or the base priority of the symbiont process for a printer, terminal, or server 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. The n specifier can be any decimal value from 0 through 15.

/BATCH
/NOBATCH (default)

Indicates that you are initializing a batch queue. For an existing queue, you can use the /BATCH qualifier only if the queue is already a batch queue. If you are initializing an existing batch queue, you can omit the /BATCH qualifier.

/BLOCK_LIMIT=([lowlim,]uplim)
/NOBLOCK_LIMIT (default)

Limits the size of print jobs that can be executed on a printer or terminal 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 will be 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, enabling it to execute.

The uplim parameter is a decimal number referring to the maximum number of blocks that will be 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, enabling it to execute.

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 will execute in the queue. In order to specify only a lower job limit, you must use two double quotation marks to indicate the upper specifier. For example, /BLOCK_LIMIT=(500,"") means any job with 500 or more blocks will execute 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 will not run in the queue.

The /NOBLOCK_LIMIT qualifier cancels the /BLOCK_LIMIT setting previously established for that queue.

/CHARACTERISTICS=(characteristic[,...])
/NOCHARACTERISTICS (default)

Specifies one or more characteristics for processing jobs on the queue. If only one characteristic is specified, you can omit the parentheses.

Each time you specify /CHARACTERISTICS, all previously set characteristics are erased. Only the ones specified with the qualifier are now established for the queue.

Queue characteristics are installation-specific. The characteristic parameter can be either a value from 0 through 127 or a characteristic name that has been defined by the DEFINE/CHARACTERISTIC command.

When users include the /CHARACTERISTICS qualifier with a PRINT or SUBMIT command, all the characteristics they specify must also be specified for the queue that will be executing the job. If not, the job will remain pending in the queue until the queue characteristics are changed, or until the entry is deleted with the DELETE/ENTRY command. Users need not specify every characteristic of a queue with a PRINT or SUBMIT command as long as the ones they specify are a subset of the characteristics set for that queue. The job will also run if no characteristics are specified.

The /NOCHARACTERISTICS qualifier cancels any /CHARACTERISTICS settings previously established for that queue.

/CPUDEFAULT=time

Defines the default CPU time limit for batch jobs.

Specify the CPU default time as a delta time value, the numeric value 0, or the keyword NONE or INFINITE. The value 0 and the keyword INFINITE allow a job unlimited CPU time, subject to restrictions imposed by the /CPUMAXIMUM qualifier or the user authorization file (UAF). Specify NONE when a default CPU time limit is not needed.

The time value cannot exceed the CPU time limit set by the /CPUMAXIMUM qualifier.

/CPUMAXIMUM=time

Defines the maximum CPU time limit for batch jobs. Use this qualifier to override the CPU time limit specified in the user authorization file (UAF).

Specify the CPU maximum time as a delta time value, the numeric value 0, or the word NONE or INFINITE. Specify NONE when a maximum CPU time limit is not desired. Refer to Table DCL–1 for more information on specifying CPU time limits.

A CPU time limit for processes is specified by each user record in the system UAF. You can also specify the following: a default CPU time limit for all jobs in a given queue and a maximum CPU time limit for all jobs in a given queue. Table DCL–1 shows the action taken for each value specified and possible combinations of specifications.

Table DCL–1 CPU Time Limit Specifications and Actions

CPU Time Limit Specified By The SUBMIT Command?	Default CPU Time Limit Specified For The Queue?	Maximum CPU Time Limit Specified For The Queue?	Action Taken
No	No	No	Use the UAF value.
Yes	No	No	Use the smaller of SUBMIT command and UAF values
Yes	Yes	No	Use the smaller of SUBMIT command and UAF values
Yes	No	Yes	Use the smaller of SUBMIT command and queue's maximum values
Yes	Yes	Yes	Use the smaller of SUBMIT command and queue's maximum values
No	Yes	Yes	Use the smaller of queue's default and maximum values

Table DCL–1 (Cont.) CPU Time Limit Specifications and Actions

CPU Time Limit Specified By The SUBMIT Command?	Default CPU Time Limit Specified For The Queue?	Maximum CPU Time Limit Specified For The Queue?	Action Taken
No	No	Yes	Use the maximum value
No	Yes	No	Use the smaller of UAF and queue's default values

QUALIFIERS

/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. Once an option is set for the queue by the /DEFAULT qualifier, users do not have to specify that option in their PRINT commands.

The options are:

[NO]BURST[=keyword]	Specifies whether file burst pages will be printed. If the keyword is ALL (the default), burst pages are placed before each file in the print job. If the keyword is ONE, a burst page is placed before the first copy of the first file in the job.
[NO]FEED	Specifies whether a form-feed is automatically inserted at the end of a page. (The default is FEED.)
[NO]FLAG[=keyword]	Specifies whether file flag pages will be printed. If the keyword is ALL (the default), flag pages are placed before each file in the print job. If the keyword is ONE, a flag page is placed before the first copy of the first file in the job.
FORM=type	Specifies the default form for a printer, terminal, or server queue. If a job is not submitted with an explicit form definition, then this form will be used to process the job. The systemwide default form, form=0, is the default value for this keyword. See also /FORM_MOUNTED.
[NO]TRAILER[=keyword]	Specifies whether file trailer pages will be printed. If the keyword is ALL (the default), trailer pages are placed at the end of each file in the print job. If the keyword is ONE, a trailer page is placed after the last copy of the last file in the job.

If you specify any of the keywords BURST, FLAG, or TRAILER without specifying a value, the value ALL is used by default.

The /DEFAULT qualifier is incompatible with the /GENERIC qualifier.

/DISABLE_SWAPPING
/NODISABLE_SWAPPING (default)

Controls whether batch jobs executed from a queue can be swapped in and out of memory.

/ENABLE_GENERIC (default)
/NOENABLE_GENERIC

This execution queue qualifier specifies whether files queued to a generic queue that does not have specific associated execution queues (named with the /GENERIC qualifier) can be placed in this execution queue for processing. (See the description of the /GENERIC qualifier for more information.)

/FORM_MOUNTED=type

Specifies the form type for a printer, terminal, or server queue. If the stock of the mounted form is not identical to the stock of the default form, as indicated by the DCL command qualifier /DEFAULT=FORM=type, then all jobs submitted to this queue without an explicit form definition will enter a pending state. 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, then the job will enter a pending state. In both cases, the pending state will be maintained until the stock of the mounted form of the queue is identical to the stock of the form associated with the job.

Specify the form type using either a numeric value or a form name that has been defined by the DEFINE/FORM command. Form types are installation-specific. The /FORM_MOUNTED qualifier is incompatible with the /GENERIC qualifier.

/GENERIC[=(queue-name[,...])]
/NOGENERIC (default)

Specifies that this is a generic queue and that jobs placed in it can be moved for processing to compatible execution queues.

The /GENERIC qualifier optionally accepts a list of existing execution queues into which the generic queue can place work. If you list one or more queues, jobs can only be moved to the specified queues. The /BATCH qualifier of the generic queue and all the specified execution queues must match.

If you do not specify any queue names with the /GENERIC qualifier, jobs can be moved to any execution queue that is initialized with /ENABLE_GENERIC in effect, and that is the same type (batch, printer, terminal, or server) as the generic queue. For a generic server queue, the /PROCESSOR qualifier of the generic queue and the execution queue must also match.

By default, a generic queue is a generic printer queue. You establish a generic batch queue by including the /BATCH qualifier, a generic terminal queue by including the /TERMINAL qualifier, and a generic server queue by including the /PROCESSOR qualifier. Use of the /GENERIC qualifier is incompatible with the /DEFAULT, /FORM_MOUNTED, and /SEPARATE qualifiers.

The /BATCH qualifier determines that an execution queue is a batch queue. The symbiont process determines whether queues are printer, terminal, or server queues; the standard symbiont sets this characteristic, depending upon whether the output device is a printer or a terminal.

/JOB_LIMIT=n

Indicates the number of batch jobs that can be executed concurrently from the queue. The job limit default value for n is 1.

/LIBRARY=file-name
/NOLIBRARY

Specifies the file name for the device control library. When you are initializing a output queue, you can use the /LIBRARY qualifier to specify an alternate device control library. The default library is SYS$LIBRARY:SYSDEVCTL.TLB. Only a file name can be used as the parameter of the /LIBRARY qualifier. The system always assumes that the location of the file is in SYS$LIBRARY and that the file type is TLB.

/ON=[node::]device[:] (printer, terminal, server queue)
/ON=node:: (batch queue)

Specifies the node and/or device on which this execution queue is located. For batch queues, only the node name can be specified. You can include both the node name and the device name for printer and terminal queues. By default, a queue executes on the same node from which you first start the queue. The default device parameter is the same as the queue name.

The node name is used only in VAXcluster systems; it must match the node name specified by the SYSGEN parameter SCSNODE for the processor on which the queue executes.

/OWNER_UIC=uic

Enables you to change the UIC of the queue. Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*. The default UIC is [1,4].

/PROCESSOR=file-name
/NOPROCESSOR

Allows users to specify their own print symbionts. The file name specifier can be any valid file name. Only a file name can be used as a parameter of the /PROCESSOR qualifier. The system supplies the device and directory name SYS$SYSTEM as well as the file type EXE.

If you use this qualifier for a output queue, it specifies that the symbiont image to be executed is SYS$SYSTEM:file-name.EXE. By default, SYS$SYSTEM:PRTSMB.EXE is executed.

If you use this qualifier for a generic queue, it specifies that the generic queue can place jobs only on queues that have been established as server queues and that are executing the specified symbiont image.

The /NOPROCESSOR qualifier cancels the effect of a previous /PROCESSOR setting.

/PROTECTION=(codes)

Specifies the protection of the queue. By default, the queue protection is (SYSTEM:E, OWNER:D, GROUP:R, WORLD:W). If you include only one protection code, you can omit the parentheses. Specify the protection using the rules given in Section 7.1.2 of the *VAX/VMS DCL Concepts Manual*.

/RECORD_BLOCKING (default)
/NORECORD_BLOCKING

Determines whether the symbiont can concatenate (or block together) output records for transmission to the output device. If you specify /NORECORD_BLOCKING, the symbiont is directed to send each formatted record in a separate I/O request to the output device. For the standard VMS print symbiont, record blocking can have a significant performance advantage over single-record mode.

/RETAIN[=option]
/NORETAIN (default)

Specifies that jobs are to be retained in the queue in a completed status after they have executed.

The option parameter can be the keyword ALL or ERROR. If ALL is specified, all jobs are retained in the queue after execution. If ERROR is specified, only jobs that have completed unsuccessfully are retained in the queue. If you omit the option parameter, ALL is assumed.

The /NORETAIN qualifier enables you to reset the queue to the default.

/SCHEDULE=[NO]SIZE

Specifies whether pending jobs in a printer, terminal, or server queue are scheduled for printing based on the size of the job. When the default, /SCHEDULE=SIZE, is in effect, shorter jobs will print before longer ones.

If you issue this command while there are pending jobs in any queue, its effect on future jobs is unpredictable.

/SEPARATE=(option[,...])
/NOSEPARATE (default)

Specifies the job separation defaults for a printer or terminal queue. The /SEPARATE qualifier is incompatible with the /GENERIC qualifier. The job separation options are:

[NO]BURST	Specifies whether a burst page will be printed at the beginning of every job. Specifying BURST also results in a flag page being printed.
[NO]FLAG	Specifies whether a flag page will be printed at the beginning of every job.
[NO]TRAILER	Specifies whether a trailer page will be printed at the end of every job.
[NO]RESET=(module[,...])	Specifies a job reset sequence for the queue. The specified modules from the device control library are used to reset the device each time a job reset occurs.

/START
/NOSTART (default)

Specifies that the queue being initialized will start at the conclusion of this INITIALIZE command.

/TERMINAL
/NOTERMINAL (default)

Indicates that a generic queue will be associated with terminal queues instead of with printer queues. The /TERMINAL qualifier allows all jobs entered in the generic queue to be moved to terminal queues with matching characteristics.

The /NOTERMINAL qualifier cancels the effect of a previous /TERMINAL setting.

/WSDEFAULT=n

Defines a working set default for a batch 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 a positive integer in the range 1 through 65,535, 0, or the word NONE as the value for n. If 0 or NONE is specified for n, the working set default value defaults to the value specified either in the UAF or by the SUBMIT command (if specified). For more information, refer to Table DCL–2.

You can also specify this qualifier for an output queue. Used in this context, it establishes the working set default of the symbiont process for a printer, terminal, or server queue when the symbiont process is created.

/WSEXTENT=n

Defines a working set extent for the batch 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 a positive integer in the range 1 through 65,535, 0, or the word NONE as the value for n. If 0 or NONE is specified for n, the working set value defaults to the value specified either in the UAF or by the SUBMIT command (if specified). For more information, refer to Table DCL–2.

You can also specify this qualifier for an output queue. Used in this context, it establishes the working set extent of the symbiont process for a printer, terminal, or server queue when the symbiont process is created.

/WSQUOTA=n

Defines the working set page size (working set quota) for a batch 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 a positive integer in the range 1 through 65,535, 0, or the word NONE as the value for n. If 0 or NONE is specified for n, the working set quota defaults to the value specified either in the UAF or by the SUBMIT command (if specified). For more information, refer to Table DCL–2.

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), and can be specified for individual jobs and/or for all jobs in a given queue. The decision table (Table DCL–2) shows the action taken for different combinations of specifications that involve working set size and working set quota values.

You can also specify this qualifier for an output queue. Used in this context, it establishes the working set quota of the symbiont process for a printer, terminal, or server queue when the symbiont process is created.

Table DCL–2 Working Set Default, Extent, and Quota Decision Table

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

```
$ INITIALIZE/QUEUE/START/DEFAULT=FLAG SYS$PRINT/ON=LPA0:
$ INITIALIZE/QUEUE/START/BATCH/JOB_LIMIT=4 SYS$BATCH
```

These two commands initialize and start the printer queue SYS$PRINT on device LPA0 and then the batch queue SYS$BATCH. The /DEFAULT=FLAG qualifier causes a flag page to precede each file for jobs in the printer queue. The /JOB_LIMIT=4 qualifier allows as many as four batch jobs to be initiated concurrently from the batch queue. Both queues are started as soon as they have been initialized.

2

```
$ INITIALIZE/QUEUE/START/BATCH/JOB_LIMIT=3 SYS$BATCH
$ INITIALIZE/QUEUE/START/BATCH/JOB_LIMIT=1/WSEXTENT=2000 BIG_BATCH
$ INITIALIZE/QUEUE/START/DEFAULT=FORM=LN01_PORTRAIT LN01_PRINT
$ INITIALIZE/QUEUE/START/DEFAULT=(FLAG,TRAILER=ONE) LPA0:
$ INITIZLIZE/QUEUE/START/DEFAULT=(FLAG,TRAILER=ONE)/BLOCK_LIMIT=(1000,"") LPB0:
$ INITIALIZE/QUEUE/START/GENERIC=(LPA0,LPB0) SYS$PRINT
$ INITIALIZE/QUEUE/START/FORM_MOUNTED=LETTER/BLOCK_LIMIT=50 LQP /ON=TXA5:
```

The first INITIALIZE/QUEUE command creates a batch queue called SYS$BATCH that can be used for any batch job. The /JOB_LIMIT qualifier allows three jobs to execute concurrently. The second INITIALIZE/QUEUE command creates a second batch queue called BIG_BATCH that is designed for large jobs. Only one job can execute at a time. The working set extent can be as high as 2000.

The remaining INITIALIZE/QUEUE commands set up printer queues. The first creates the printer queue LN01_PRINT with the default form LN01_PORTRAIT. Both queue LPA0 and LPB0 are set to put flag and trailer pages between each file. If a job contains more than one file for printing, flag pages separate each file within the job. In addition, LPB0 has a minimum block size of 1000. Thus only print jobs larger than 1000 blocks can execute on that queue. SYS$PRINT is established as a generic queue that can direct jobs to either LPA0 or LPB0. Jobs that are too small to run on LPB0 will be queued from SYS$PRINT to LPA0.

The last INITIALIZE/QUEUE command sets up a terminal queue on TXA5. This queue is limited to PRINT commands that specify the form type LETTER. LETTER has been established at this site to indicate special letter-head paper. The block size limit is 50, indicating that this queue is reserved for jobs smaller than 51 blocks.

INQUIRE

Requests interactive assignment of a value for a local or global symbol during the execution of a command procedure.

FORMAT

INQUIRE *symbol-name [prompt-string]*

restrictions

None.

PARAMETERS

symbol-name

Specifies a 1- through 255-alphanumeric character symbol to be given a value.

prompt-string

Specifies the prompt to be displayed at the terminal when the INQUIRE command is executed. String values are automatically converted to uppercase. Also, any leading and trailing spaces and tabs are removed, and multiple spaces and tabs between characters are compressed to a single space.

To prohibit conversion to uppercase and retain space and tab characters, place quotation marks around the string. To use quotation marks in a prompt-string, enclose the entire string in quotation marks and use a double set of quotation marks within the string. If the prompt string contains an at sign character (@), enclose the string in quotation marks.

When the system displays the prompt string at the terminal, it generally places a colon (:) and a space at the end of the string. (See the /PUNCTUATION qualifier.)

If you do not specify a prompt string, the command interpreter uses the symbol name to prompt for a value.

DESCRIPTION

The INQUIRE command displays the prompting message to and reads the response from the input stream established when your process was created. This means that when the INQUIRE command is executed in a command procedure executed interactively, the prompting message is always displayed on the terminal, regardless of the level of nesting of command procedures. Note that input to the INQUIRE command in command procedures will be placed in the RECALL buffer.

When you enter a response to the prompt string, the value is assigned as a character string to the specified symbol. Lowercase characters are automatically converted to uppercase. Also, any leading and trailing spaces and tabs are removed, and multiple spaces and tabs between characters are compressed to a single space. To prohibit conversion to uppercase and retain space and tab characters, place quotation marks around the string.

To use symbols or lexical functions when you enter a response to the prompt string, use apostrophes to request symbol substitution.

Note that you can also use the READ command to obtain data interactively from the terminal. The READ command accepts data exactly as the user types it; characters are not automatically converted to uppercase and spaces are not compressed. However, symbols and lexical functions will not be translated even if you use apostrophes to request symbol substitution.

When an INQUIRE command is issued in a batch job, the command reads the response from the next line in the command procedure; if procedures are nested, it reads the response from the first level command procedure. If the next line in the batch job command procedure begins with a dollar sign, the line is interpreted as a command, not as a response to the INQUIRE command. The INQUIRE command then assigns a null string to the specified symbol, and the batch job continues processing with the command on the line following the INQUIRE command.

QUALIFIERS

/GLOBAL

Specifies that the symbol be placed in the global symbol table. If you do not specify the /GLOBAL qualifier, the symbol is placed in the local symbol table.

/LOCAL (default)

Specifies that the symbol be placed in the local symbol table for the current command procedure.

/PUNCTUATION (default)
/NOPUNCTUATION

Controls whether or not a colon (:) and a space follow the prompt when it is displayed on the terminal. If you wish to suppress the colon and space, specify /NOPUNCTUATION.

EXAMPLES

1
```
$ INQUIRE CHECK "Enter Y[ES] to continue"
$ IF .NOT. CHECK THEN EXIT
```

The INQUIRE command displays the following prompting message at the terminal:

```
Enter Y[ES] to continue:
```

The INQUIRE command prompts for a value, which is assigned to the symbol CHECK. The IF command tests the value assigned to the symbol CHECK. If the value assigned to CHECK is true (that is, an odd numeric value, a character string that begins with a T, t, Y, or y, or an odd numeric character string), the procedure continues executing.

If the value assigned to CHECK is false (that is, an even numeric value, a character string that begins with any letter except T, t, Y, or y, or an even numeric character string), the procedure exits.

INQUIRE

```
2   $ INQUIRE COUNT
    $ IF COUNT .GT. 10 THEN GOTO SKIP
      .
      .
      .
    $ SKIP:
```

The INQUIRE command prompts for a count with the message:

```
COUNT:
```

Then the command procedure uses the value of the symbol COUNT to determine whether to execute the next sequence of commands or to transfer control to the line labeled SKIP.

```
3   $ IF P1 .EQS. "" THEN INQUIRE P1 "FILE NAME"
    $ FORTRAN 'P1'
```

The IF command checks whether a parameter was passed to the command procedure by checking if the symbol P1 is null; if it is, it means that no parameter was specified, and the INQUIRE command is issued to prompt for the parameter. If P1 was specified, the INQUIRE command is not executed, and the FORTRAN command compiles the name of the file specified as a parameter.

JOB

Identifies the beginning of a batch job submitted through a card reader.

FORMAT

$ JOB *user-name*

restrictions

JOB cannot be abbreviated.

PARAMETER

user-name

Identifies the user name under which the job is to be run. Specify the user name just as you would enter it during the login procedure.

DESCRIPTION

Each batch job submitted to the system through the system card reader must be preceded by a JOB card.

The JOB card identifies the user submitting the job and is followed by a PASSWORD card giving the password. (Although the PASSWORD card is required, you do not have to use a password on the card if the account has a null password.)

The user name and password are validated by the system authorization file in the same manner as they are validated in the login procedure. The process that executes the batch job is assigned the disk and directory defaults and privileges associated with the user account. If a LOGIN.COM file exists for the specified user name, it is executed at the start of the job.

The end of a batch job is signaled by the EOJ command, by an EOF card (12-11-0-1-6-7-8-9 overpunch), or by another JOB card.

QUALIFIERS

/AFTER=time

Requests that the job be held until after a specific time. If the specified time has already passed, the job is queued for immediate processing.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values.

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

Specifies one or more characteristics that are required for the job. If you specify only one characteristic, you can omit the parentheses. Codes for characteristics are installation-defined. Use the SHOW QUEUE /CHARACTERISTICS command to see which characteristics are available on your system.

When users include the /CHARACTERISTICS qualifier, all the characteristics they specify must also be specified for the queue that will be executing the job. If not, the job will remain pending in the queue until the queue characteristics are changed or they delete the entry with the DELETE/ENTRY command. Users need not specify every characteristic of a queue with the JOB

command as long as the ones they specify are a subset of the characteristics set for that queue. The job will also run if no characteristics are specified.

/CLI=file-name

Enables you to specify a different command language interpreter (CLI) to use in processing the job. The file name specifies that the command language interpreter (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's authorization record.

/CPUTIME=n

Defines a CPU time limit for the batch job. You can specify a delta time (see Section 2.5 of the *VAX/VMS DCL Concepts Manual*), the value 0, or the word NONE or INFINITE for n.

When you need less CPU time than authorized, use the /CPUTIME qualifier to override the base queue value established by the system manager or the value authorized in your user authorization file. Specify 0 or INFINITE to request an infinite amount of time. Specify NONE when you want the CPU time to default to your user authorization file value or the limit specified on the queue. Note that you cannot request more time than permitted by the base queue limits or your user authorization file.

/DELETE (default)
/NODELETE

Controls whether the batch input file is deleted after the job is processed. If you specify /NODELETE, the file is saved in the user's default directory under the default name INPBATCH.COM. If you specify the /NAME qualifier, the file name of the batch input file is the same as the job name you supply with /NAME.

/HOLD
/NOHOLD (default)

Controls whether or not the job is to be made available for immediate processing.

If you specify /HOLD, the job is not released for processing until you specifically release it with the /NOHOLD or /RELEASE qualifier of the SET QUEUE/ENTRY command.

/KEEP
/NOKEEP (default)

Controls whether the log file is deleted after it is printed. /NOKEEP is the default unless the /NOPRINTER qualifier is used.

/LOG_FILE=file-spec
/NOLOG_FILE

Controls whether a log file with the specified name is created for the job or whether a log file is created.

When you use the /LOG_FILE qualifier, the system writes the log file to the file you specify. If you use /NOLOG_FILE, no log file is created. If you use neither form of the qualifier, the log file is written to a file in your default directory that has the same file name as the first command file in the job and

a file type of LOG. Using neither /LOG_FILE nor /NOLOG_FILE is the default.

You can use the /LOG_FILE qualifier to specify that the log file be written to a different device. Logical names that occur in the file specification are translated at the time the job is submitted. 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 and the file type LOG.

/NAME=job-name

Specifies a file name string to be used as the job name and as the file name for both the batch job log file and the command file. The job name value can have from 1 to 39 characters and must be a valid file name. The default log file name is INPBATCH.LOG; the default command file name is INPBATCH.COM.

/NOTIFY
/NONOTIFY (default)

Controls whether a message is broadcast to any terminal at which you are logged in, notifying you that your job has been completed or aborted.

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

Specifies from 1 through 8 optional parameters that can be passed to the command procedure. The parameters define values to be equated to the symbols named P1 through P8 in the batch job. The symbols are local to the initial input stream.

If you specify only one parameter, you can omit the parentheses.

The commas delimit individual parameters. To specify a parameter that contains any special characters or delimiters, enclose the parameter in quotation marks. Individual parameters cannot exceed 255 characters.

/PRINTER=queue-name
/NOPRINTER

Controls whether the job log is queued for printing when the job is completed. The /PRINTER qualifier allows you to specify a particular print queue. The default print queue for the log file is SYS$PRINT.

If you specify /NOPRINTER, the /KEEP qualifier is assumed.

/PRIORITY=n

Requires either operator (OPER) or alter priority (ALTPRI) privilege to raise the priority value above the value of the SYSGEN parameter MAXQUEPRI.

Specifies the job scheduling priority for the specified job. The priority value can be in the range of 0 through 255, where 0 is the lowest priority and 255 is the highest.

The default value for /PRIORITY is the value of the SYSGEN parameter DEFQUEPRI. No privilege is needed to set the priority lower than the MAXQUEPRI value.

The /PRIORITY qualifier has no effect on the process priority. The queue establishes the process priority.

/QUEUE=queue-name[:]

Specifies the name of the particular batch job queue in which the job is to be entered. If you do not specify /QUEUE, the job is placed in the default system batch job queue, SYS$BATCH.

/RESTART
/NORESTART (default)

Specifies whether the job can restart after a crash or a STOP/QUEUE /REQUEUE command.

/TRAILING_BLANKS (default)
/NOTRAILING_BLANKS

Controls whether input cards in the card deck are read in card image form or input records are truncated at the last nonblank character. By default, the system does not remove trailing blanks from the records read through the card reader. Use the /NOTRAILING_BLANKS qualifier to request that input records be truncated.

/WSDEFAULT=n

Defines a working set default for the batch job. The n parameter can be any positive integer from 1 to 65,535, 0, or the keyword NONE.

Use this qualifier to impose a value lower than the base queue value established by the system manager or than the value authorized in your user authorization file. Specify 0 or NONE if you want the working set value defaulted to either your user authorization file or the working set default specified on the queue. You cannot request a value higher than your default.

/WSEXTENT=n

Defines a working set extent for the batch job. The n parameter can be any positive integer from 1 to 65,535, 0, or the keyword NONE.

Use this qualifier to override the base queue value established by the system manager rather than the value authorized in your user authorization file, if you want to impose a lower value. Specify 0 or NONE if you want the working set extent defaulted to either your user authorization file value or the working set extent specified on the queue. However, you cannot request a value higher than your default.

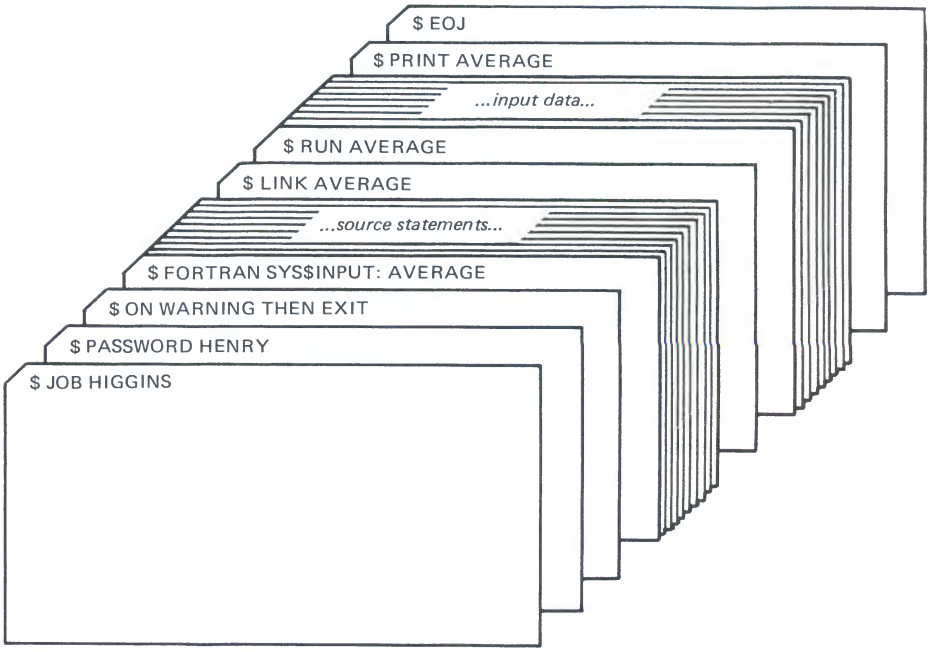
/WSQUOTA=n

Defines the maximum working set size for the batch job. This is the working set quota. The n parameter can be any positive integer from 1 to 65,535, 0, or the keyword NONE.

Use this qualifier to impose a value lower than the base queue value established by the system manager or than the value authorized in your user authorization file. Specify 0 or NONE if you want the working set quota defaulted to either your user authorization file value or the working set quota specified on the queue. You cannot request a value higher than your default.

EXAMPLES

1

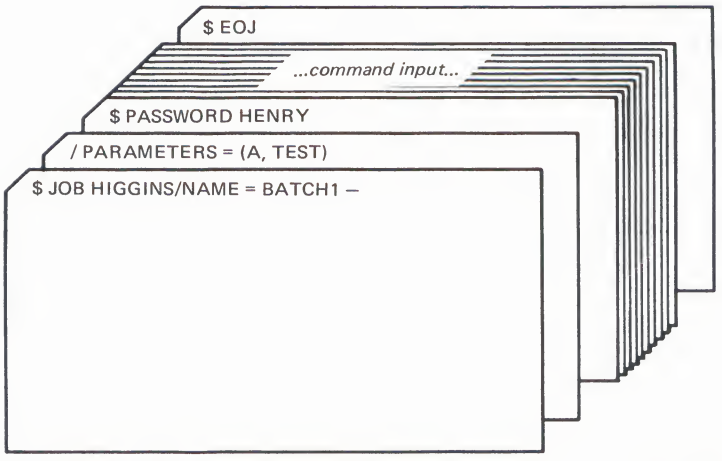


ZK-787-82

The JOB and PASSWORD cards identify and authorize the user HIGGINS to enter batch jobs. The command stream consists of a FORTRAN command and FORTRAN source statements to be compiled. The file name AVERAGE following the device name SYS$INPUT provides the compiler with a file name for the object and listing files. The output files are cataloged in user HIGGINS's default directory.

If the compilation is successful, the LINK command creates an executable image and the RUN command executes it. Input for the program follows the RUN command in the command stream. The last command in the job prints the program listing. The last card in the deck contains the EOJ (end of job) command.

2



ZK-788-82

The /NAME qualifier on the JOB card specifies a name for the batch job. When the job completes, the printed log file will be identified as BATCH1.LOG. The JOB command is continued onto a second card with the continuation character (-). The /PARAMETERS qualifier defines P1 as A and P2 as TEST. The last card in the deck contains the EOJ (end of job) command.

LEXICAL FUNCTIONS

A set of functions that return information about character strings and attributes of the current process.

DESCRIPTION

The command language includes constructs, called lexical functions, that return information about the current process and about arithmetic and string expressions. The functions are called lexical functions because the command interpreter evaluates them during the command input scanning (or lexical processing) phase of command processing.

You can use lexical functions in any context in which you normally use symbols or expressions. In command procedures, you can use lexical functions to translate logical names, perform character string manipulations, and determine the current processing mode of the procedure.

The general format of a lexical function is:

```
F$function-name([args,...])
```

F$
Indicates that what follows is a lexical function.

function-name
Specifies the function to be evaluated. All function names are keywords. You can truncate function names to any unique abbreviation.

()
Enclose function arguments, if any. The parentheses are required for all functions, including functions that do not accept any arguments.

args,...
Specify arguments for the function, if any. You can specify arguments using integer or character string expressions. See Chapter 5 of the *VAX/VMS DCL Concepts Manual* for more information on specifying expressions.

Table DCL–3 lists each lexical function and briefly describes the information that each function returns. A detailed description of each function, including examples, is given in the following pages.

Table DCL–3 Summary of Lexical Functions

Function	Description
F$CVSI	Extracts bit fields from character string data and converts the result, as a signed value, to an integer.
F$CVTIME	Retrieves information about an absolute, combination, or delta time string.
F$CVUI	Extracts bit fields from character string data and converts the result, as an unsigned value, to an integer.
F$DIRECTORY	Returns the current default directory name string.
F$EDIT	Edits a character string based on the edits specified.

Table DCL–3 (Cont.) Summary of Lexical Functions

Function	Description
F$ELEMENT	Extracts an element from a string in which the elements are separated by a specified delimiter.
F$ENVIRONMENT	Obtains information about the DCL command environment.
F$EXTRACT	Extracts a substring from a character string expression.
F$FAO	Invokes the $FAO system service to convert the specified control string to a formatted ASCII output string.
F$FILE_ATTRIBUTES	Returns attribute information for a specified file.
F$GETDVI	Invokes the $GETDVI system service to return a specified item of information for a specified device.
F$GETJPI	Invokes the $GETJPI system service to return accounting, status, and identification information for a process.
F$GETSYI	Invokes the $GETSYI system service to return status and identification information about the local system, or about a node in the local cluster, if your system is part of a cluster.
F$IDENTIFIER	Converts an identifier in named format to its integer equivalent, or vice versa.
F$INTEGER	Returns the integer equivalent of the result of the specified expression.
F$LENGTH	Returns the length of a specified string.
F$LOCATE	Locates a character or character substring within a string and returns its offset within the string.
F$LOGICAL	Translates a logical name and returns the equivalence name string. (Superseded in function by F$TRNLNM.)
F$MESSAGE	Returns the message text associated with a specified system status code value.
F$MODE	Shows the mode in which a process is executing.
F$PARSE	Invokes the $PARSE RMS service to parse a file specification and return either the expanded file specification or the particular file specification field that you request.
F$PID	For each invocation, returns the next process identification number in sequence.
F$PRIVILEGE	Returns a value of "TRUE" or "FALSE" depending on whether your current process privileges match the privileges listed in the argument.
F$PROCESS	Returns the current process name string.
F$SEARCH	Invokes the $SEARCH RMS service to search a directory file, and returns the full file specification for a file you name.

Table DCL–3 (Cont.) Summary of Lexical Functions

Function	Description
F$SETPRV	Sets the specified privileges and returns a list of keywords indicating the previous state of these privileges for the current process.
F$STRING	Returns the string equivalent of the result of the specified expression.
F$TIME	Returns the current date and time of day, in the format dd-mm-yyy hh:mm:ss.cc.
F$TRNLNM	Translates a logical name and returns the equivalence name string or the requested attributes of the logical name.
F$TYPE	Determines the data type of a symbol.
F$USER	Returns the current user identification code (UIC).
F$VERIFY	Returns the integer 1 if command procedure verification is set on; returns the integer 0 if command procedure verification is set off. The F$VERIFY function also can set new verification states.

F$CVSI

Extracts bit fields from character string data and converts the result, as a signed value, to an integer.

FORMAT

F$CVSI*(bit-position,width,string)*

return value

The integer equivalent of the extracted bit field, converted as a signed value.

ARGUMENTS

bit-position

The offset of the first bit to be extracted. The low-order (rightmost) bit of a string is position number 0 for determining the offset. Specify the offset as an integer expression.

If you specify an expression with a negative value, or with a value that exceeds the number of bits in the string, then DCL displays the INVRANGE error message.

width

The number of bits that are to be extracted for conversion to an integer value. Specify the width as an integer expression.

If you specify an expression with a negative value, or with a value that is invalid when added to the bit position offset, then DCL displays the INVRANGE error message.

string

The string from which the bits are taken. Specify the string as a character string expression.

EXAMPLES

1
```
$ A[0,32] = %X2B
$ SHOW SYMBOL A
  A = "+"
$ X = F$CVSI(0,4,A)
$ SHOW SYMBOL X
  X = -5   Hex = FFFFFFFB  Octal = 177773
```

This example uses an arithmetic overlay to assign the hexadecimal value 2B to all 32 bits of the symbol A. See the description of the Assignment Statement for more information on arithmetic overlays.

The symbol A has a string value after the overlay because it was previously undefined. (If a symbol is undefined, it has a string value as a result of an arithmetic overlay. If a symbol was previously defined, it retains the same data type after the overlay.) The hexadecimal value 2B corresponds to the ASCII value of the plus sign (+).

Next, the F$CVSI function extracts the low-order 4 bits from the symbol A; the low order 4 bits contain the binary representation of the hexadecimal value B. These bits are converted, as a signed value, to an integer. The converted value, –5, is assigned to the symbol X.

```
2   $ SYM[0,32] = %X2A
    $ SHOW SYMBOL SYM
      SYM = "*"
    $ Y = F$CVSI(0,33,SYM)
    %DCL-W-INVRANGE, field specification is out of bounds - check sign and size
    $ SHOW SYMBOL Y
    %DCL-W-UNDSYM, undefined symbol - check spelling
```

In this example, the width argument specified with the F$CVSI function is too large. Therefore, DCL issues an error message and the symbol Y is not assigned a value.

F$CVTIME

Converts an absolute or a combination time string to a string of the form "yyyy-mm-dd hh:mm:ss.cc". The F$CVTIME function can also return information about an absolute, combination, or delta time string.

FORMAT

F$CVTIME*([input_time] [,output_time] [,field])*

return value

A character string containing the requested information.

ARGUMENTS

input_time

The time string for which information is requested. Specify the input time string as a character string expression.

You can specify the input time string as an absolute, combination, or delta time. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for information on these time formats. If you omit optional date fields, the system will supply values for the current date. If you omit optional time fields, the fields are filled with zeroes.

If you specify a delta input time, you must also specify DELTA as the output time argument.

Absolute time keywords (TODAY, TOMORROW, and so on) are allowed in the input time specification.

If you omit the time argument or if you specify the time as a null string, the current date and time, in absolute time format, is used.

output_time

The time format for the information you want returned. Specify the output time argument as a character string expression that evaluates to one of the following strings:

ABSOLUTE	The requested information should be returned in absolute time format.
COMPARISON	The requested information should be returned in the form "yyyy-mm-dd hh:mm:ss.cc".
DELTA	The requested information should be returned in delta format. If you specify DELTA as the output time argument, then you must also provide a delta time specification for the input time argument.

If you omit the output time argument, the default is COMPARISON.

field

The type of information to be returned. The information is returned in the time format specified by the output time argument.

Specify the field argument as a character string expression that evaluates to one of the following strings:

DATE	The date field is returned.
DATETIME	The entire date and time string is returned.
DAY	The day field is returned.
HOUR	The hour field is returned.
HUNDREDTH	The hundredth of a second field is returned.
MINUTE	The minute field is returned.
MONTH	The month field is returned. You cannot specify MONTH if you also specify a delta input time and the DELTA output time argument.
SECOND	The second field is returned.
TIME	The time field is returned.
WEEKDAY	The weekday that corresponds with the input time argument is returned. You cannot specify WEEKDAY if you also specify a delta input time and the DELTA output time argument. When the weekday is returned, the first letter is in uppercase, and the following letters are in lowercase.
YEAR	The year field is returned. You cannot specify YEAR if you also specify a delta input time and the DELTA output time argument.

If you do not specify a field argument, the default is DATETIME.

DESCRIPTION

When you use the F$CVTIME function, you can omit optional arguments that can be used to the right of the last argument you specify. However, you must include commas as placeholders if you omit optional arguments to the left of the last argument you specify.

When you specify the input time argument in either absolute or combination time format, you can specify ABSOLUTE or COMPARISON as the output time argument; you cannot specify DELTA.

When you specify the input time argument in delta time format, you must specify DELTA as the output time argument.

EXAMPLES

1
```
$ TIME = F$TIME()
$ SHOW SYMBOL TIME
  TIME = "15-APR-1984 10:56:23.10"
$ TIME = F$CVTIME(TIME)
$ SHOW SYMBOL TIME
  TIME = "1984-04-15 10:56:23.10"
```

This example uses the F$TIME function to return the system time as a character string, and to assign the time to the symbol TIME. Then the F$CVTIME function is used to convert the system time to an alternate time format. Note that you do not need to place quotation marks around the argument TIME because it is a symbol. Symbols are automatically evaluated when they are used as arguments for lexical functions.

You can use the resultant string to compare two dates (using .LTS. and .GTS. operators). For example, you can use F$CVTIME to convert two time strings and store the results in the symbols TIME_1 and TIME_2. You can compare the two values, and branch to a label, based on the results as shown below:

```
$ IF TIME_1 .LTS. TIME_2 THEN GOTO FIRST
```

2
```
$ NEXT = F$CVTIME("TOMORROW",,"WEEKDAY")
$ SHOW SYMBOL NEXT
 NEXT = "Tuesday"
```

In this example, the F$CVTIME returns the weekday that corresponds to the absolute time keyword "TOMORROW". You must enclose the arguments "TOMORROW" and "WEEKDAY" in quotation marks because they are character string expressions. Also, you must include a comma as a placeholder for the output time argument that is omitted.

F$CVUI

Extracts bit fields from character string data and converts the result, as an unsigned value, to an integer.

FORMAT

F$CVUI*(bit-position,width,string)*

return value

The integer equivalent of the extracted bit field, converted as an unsigned value.

ARGUMENTS

bit-position

The offset of the first bit to be extracted. The low-order (rightmost) bit of a string is position number 0 for determining the offset. Specify the offset as an integer expression.

If you specify an expression with a negative value, or with a value that exceeds the number of bits in the string, then DCL displays the INVRANGE error message.

width

The number of bits that are to be extracted for conversion to an integer value. Specify the width as an integer expression.

If you specify an expression with a negative value, or with a value that is invalid when added to the bit position offset, then DCL displays the INVRANGE error message.

string

The string from which the bits are taken. Specify the string as a character string expression.

EXAMPLE

```
$ A[0,32] = %X2B
$ SHOW SYMBOL A
  A = "+"
$ X = F$CVUI(0,4,A)
$ SHOW SYMBOL X
  X = 11    Hex = 0000000B  Octal = 000013
```

This example uses an arithmetic overlay to assign the hexadecimal value 2B to all 32 bits of the symbol A. The symbol A has a string value after the overlay because it was previously undefined. (If a symbol is undefined, it has a string value as a result of an arithmetic overlay. If a symbol was previously defined, it retains the same data type after the overlay.) The hexadecimal value 2B corresponds to the ASCII character "+".

Next, the F$CVUI function extracts the low-order 4 bits from the symbol A; the low-order 4 bits contain the binary representation of the hexadecimal value B. These bits are converted, as a signed value, to an integer. The converted value, 11, is assigned to the symbol X.

F$DIRECTORY

Returns the current default directory name string. The F$DIRECTORY function has no arguments but must be followed by parentheses.

FORMAT F$DIRECTORY()

return value

A character string for the current default directory name, including square brackets ([]). If you use the SET DEFAULT command and specify angle brackets (<>) in a directory specification, the F$DIRECTORY function returns angle brackets in the directory string.

ARGUMENTS

None.

DESCRIPTION

You can use the F$DIRECTORY function to save the name of the current default directory in a command procedure, change the default to another directory to do work, and later restore the original setting.

EXAMPLE

```
$ SAVE_DIR = F$DIRECTORY()
$ SET DEFAULT [MALCOLM.TESTFILES]
   .
   .
   .
$ SET DEFAULT 'SAVE_DIR'
```

This example shows an excerpt from a command procedure that uses the F$DIRECTORY function to save the current default directory setting. The assignment statement equates the symbol SAVE_DIR to the current directory. Then the SET DEFAULT command establishes a new default directory. Later, the symbol SAVE_DIR is used in the SET DEFAULT command that restores the original default directory.

Note that you can use the F$ENVIRONMENT function with the DEFAULT keyword to return the default disk and directory. You should use the F$ENVIRONMENT function rather than the F$DIRECTORY function in situations involving more than one disk.

F$EDIT

Edits a string expression based on the edits specified in the edit-list.

FORMAT

F$EDIT*(string, edit-list)*

return value

A character string containing the specified edits.

ARGUMENTS

string

A string to be edited. Specify the string argument as a character string expression.

edit-list

A keyword or a list of keywords that specify the types of edits to be made to the string. If you use a list of keywords, separate them with commas. You cannot abbreviate these keywords.

Specify the edit-list argument as a character string expression.

You can use one or more of the following keywords in the edit-list:

COLLAPSE	Removes all spaces and tabs from the string.
COMPRESS	Replaces multiple spaces and tabs with a single space.
LOWERCASE	Makes the string lower case.
TRIM	Removes leading and trailing spaces and tabs from the string.
UNCOMMENT	Removes comments from the string.
UPCASE	Makes the string upper case.

Edits are not applied to quoted sections of strings. Therefore, if a string contains quotation marks, the characters within the quotation marks are not affected by the edits specified in the edit list.

EXAMPLES

```
$ LINE = "    THIS    LINE    CONTAINS A ""  QUOTED  "" WORD"
$ SHOW SYMBOL LINE
  LINE =  "    THIS    LINE    CONTAINS A "  QUOTED  " WORD"
$ NEW_LINE = F$EDIT(LINE, "COMPRESS, TRIM")
$ SHOW SYMBOL NEW_LINE
  NEW_LINE = "THIS LINE CONTAINS A "  QUOTED  " WORD"
```

This example uses the F$EDIT function to compress and trim a string by replacing multiple blanks with a single blank, and by removing leading and trailing blanks. The string LINE contains quotation marks around the word QUOTED. (To enter quotation marks into a character string, use double quotations in the assignment statement.)

Note that the F$EDIT function does not compress the spaces in the quoted section of the string; therefore, the spaces are retained around the word QUOTED.

```
$ LOOP:
$       READ/END_OF_FILE = DONE INPUT_FILE RECORD
$       RECORD = F$EDIT(RECORD, "TRIM, UPCASE")
$       WRITE OUTPUT_FILE RECORD
$       GOTO LOOP
        .
        .
        .
```

This example sets up a loop to read records from a file, edit them, and write them to an output file. The edited records have leading and trailing blanks removed, and are converted to uppercase.

F$ELEMENT

Extracts an element from a string in which the elements are separated by a specified delimiter.

FORMAT

F$ELEMENT*(element-number, delimiter, string)*

return value

A character string containing the specified element.

ARGUMENTS

element-number

The number of the element to be retrieved. Element numbers start with zero. Specify the element-number argument as an integer expression.

If the element-number exceeds the number of elements in the string, the delimiter is returned.

delimiter

The delimiter used to separate the elements in the string. The delimiter can be only one character long. Specify the delimiter as a character string expression.

string

A string containing a delimited list of elements. Specify the string as a character string expression.

EXAMPLE

```
$ DAY_LIST = "MON/TUE/WED/THU/FRI/SAT/SUN"
$ INQUIRE DAY "ENTER DAY (MON TUE WED THU FRI SAT SUN)"
$ NUM = 0
$ LOOP:
$       LABEL = F$ELEMENT(NUM,"/",DAY_LIST)
$       IF LABEL .EQS. "/" THEN GOTO ERROR
$       IF DAY .EQS. LABEL THEN GOTO 'LABEL'
$       NUM = NUM +1
$       GOTO LOOP
$
$ MON:
   .
   .
   .
```

This example sets up a loop to test an input value against the elements in a list of values. If the value for DAY matches one of the elements in DAY_LIST, control is passed to the corresponding label. If the value returned by the F$ELEMENT function matches the delimiter, then the value DAY was not present in the DAY_LIST, and control is passed to the label ERROR.

F$ENVIRONMENT

Obtains information on aspects of the DCL command environment.

FORMAT

F$ENVIRONMENT*(item)*

return value

Information that corresponds to the specified item. The return value can be either an integer or a character string, depending on the specified item.

ARGUMENTS

item

A keyword that specifies the type of information to be returned. You cannot abbreviate keywords for the F$ENVIRONMENT lexical function. Specify the item argument as a character string expression.

You can use one of the following keywords to specify the item argument:

CAPTIVE	Returns either the character string "TRUE" or "FALSE" to indicate whether you are logged into a captive account. The system manager can define captive accounts in the user authorization file with the Authorize Utility.
CONTROL	Returns the control characters that are currently enabled with the SET CONTROL command. The control characters are returned as a character string. If more than one control character is enabled, the list is separated by commas. If no control characters are enabled with the SET CONTROL command, the null string ("") is returned.
DEFAULT	Returns the current default device and directory name. This information is returned as a character string; the returned string is the same as the output from the SHOW DEFAULT command.
DEPTH	Returns the current command procedure depth, as an integer. The command procedure depth is 0 when you log in interactively and when you submit a batch job. The command procedure depth is 1 when you execute a command procedure interactively or from within a batch job. A nested command procedure has a depth of 1 greater than the depth of the command procedure from which the nested procedure is executed.
INTERACTIVE	Returns either the character string "TRUE" or "FALSE" to indicate whether the process is interactive.
KEY_STATE	Returns a character string indicating the current locked keypad state. See the description of the DEFINE/KEY command for more information on keypad states.
MAX_DEPTH	Returns an integer specifying the maximum allowable command procedure depth.

MESSAGE	Returns a character string containing the current setting of the the SET MESSAGE command. The string contains slashes (/) to separate qualifier names. Therefore, the output from F$ENVIRONMENT("MESSAGE") can be appended to the SET MESSAGE command to form a valid DCL command line.
NOCONTROL	Returns the control characters that are currently disabled with the SET NOCONTROL command. The control characters are returned as a character string. If more than one character is disabled, the list is separated by commas. If no control characters are disabled with the SET NOCONTROL command, the null string ("") is returned.
ON_CONTROL_Y	If issued from a command procedure, returns either the character string "TRUE" or "FALSE" to indicate whether ON CONTROL_Y is specified. If you use this keyword at command level 0, then the string "FALSE" is always returned.
ON_SEVERITY	If issued from a command procedure, returns a character string indicating the severity level at which the action specified with the ON command is taken. If issued interactively or from within a command procedure where SET NOON is in effect, the string "NONE" is returned.
OUTPUT_RATE	Returns a delta time string containing the default output rate. The output rate indicates how often data is written to a batch job log file while the batch job is executing. If you specify this keyword when you issue the F$ENVIRONMENT function interactively, then a null string is returned.
PROCEDURE	Returns the file specification for the command procedure from which the F$ENVIRONMENT("PROCEDURE") function is issued. The file specification is returned as a character string. If issued interactively, the function returns the null string ("").
PROMPT	Returns a character string containing the current prompt string.
PROMPT_CONTROL	Returns either the character string "TRUE" or "FALSE" to indicate whether a Carriage Return/Line Feed precedes the prompt.
PROTECTION	Returns a character string indicating the current default file protection.
SYMBOL_SCOPE	Returns the character string "[NO]LOCAL, [NO]GLOBAL" to indicate the current symbol scoping state. The string is in a form that can be used with the SET PROTECTION/DEFAULT command to form a valid DCL command line.

VERIFY_IMAGE	Returns either the character string "TRUE" or "FALSE" to indicate whether the SET VERIFY=IMAGE command is in effect. If image verification is in effect, then the command procedure echoes input data read by images.
VERIFY_PROCEDURE	Returns either the character string "TRUE" or "FALSE" to indicate whether the SET VERIFY=PROCEDURE command is in effect. If command verification is in effect, then the command procedure echoes DCL command lines.

EXAMPLES

```
1   $ SAVE_MESSAGE = F$ENVIRONMENT("MESSAGE")
    $ SET MESSAGE/NOFACILITY/NOIDENTIFICATION
      .
      .
      .
    $ SET MESSAGE'SAVE_MESSAGE'
```

This example uses the F$ENVIRONMENT function to save the current message setting before changing the setting. At the end of the command procedure, the original message setting is restored. The apostrophes surrounding the symbol SAVE_MESSAGE indicate that the value for the symbol should be substituted.

```
2   $ MAX = F$ENVIRONMENT("MAX_DEPTH")
    $ SHOW SYMBOL MAX
      MAX = 16   Hex = 00000010  Octal = 000020
```

This example uses the F$ENVIRONMENT function to determine the maximum depth allowable within command procedures.

```
3   $ SAVE_PROT = F$ENVIRONMENT("PROTECTION")
    $ SET PROTECTION = (SYSTEM:RWED, OWNER:RWED, GROUP, WORLD)/DEFAULT
      .
      .
      .
    $ SET PROTECTION = ('SAVE_PROT')/DEFAULT
```

This example uses the F$ENVIRONMENT function to save the current default protection before changing the protection. At the end of the command procedure, the original protection is restored. You must place apostrophes around the symbol SAVE_PROT to request symbol substitution.

F$EXTRACT

Extracts a substring from a character string expression.

FORMAT

F$EXTRACT*(offset,length,string)*

return value

A character string containing the substring that is delimited by the offset and length arguments.

ARGUMENTS

offset

The position, relative to the beginning of the string, that marks the beginning of the substring you want to extract. Specify the offset as an integer expression that is greater than or equal to 0.

The offset is the relative position of a character or a substring with respect to the beginning of the string. Offset positions begin with 0. The string always begins with the leftmost character.

If you specify an offset that is greater than or equal to the length of the string, F$EXTRACT returns a null string ("").

length

The number of characters you want to extract. Specify the length as an integer expression that is greater than or equal to 0.

If you specify a length that exceeds the number of characters from the offset to the end of the string, the F$EXTRACT returns the characters from the offset through the end of the string.

string

The string from which the substring is to be extracted. Specify the string as a character string expression.

EXAMPLES

1
```
$ NAME = "JOE SMITH"
$ FIRST = F$EXTRACT(0,3,NAME)
$ SHOW SYMBOL FIRST
  FIRST = "JOE"
```

This portion of a command procedure uses the F$EXTRACT function to extract the first three characters from the character string assigned to the symbol NAME. The offset and length arguments are integers, and the string argument is a symbol. You do not need to use quotations around integers or symbols when they are used as arguments for lexical functions.

2
```
$ P1 = "MYFILE.DAT"
$ FILENAME = F$EXTRACT(0,F$LOCATE(".",P1),P1)
```

This portion of a command procedure shows how to locate a character within a string, and how to extract a substring ending at that location.

The lexical function F$LOCATE gives the numeric value representing the offset position of a period in the character string value of P1. (The offset position of the period is equal to the length of the substring before the period.)

This F$LOCATE function is used as an argument in the F$EXTRACT function to specify the number of characters to extract from the string. If a procedure is invoked with the parameter MYFILE.DAT, these statements result in the symbol FILENAME being given the value MYFILE.

Note that the F$LOCATE function in the above example assumes that the file specification does not contain a node name or a directory specification containing a subdirectory name. To obtain the file name from a full file specification, use the F$PARSE function.

3
```
$ IF F$EXTRACT(12,2,F$TIME()) .GES. "12" THEN GOTO AFTERNOON
$ MORNING:
$ WRITE SYS$OUTPUT "Good morning!"
$ EXIT
$ AFTERNOON:
$ WRITE SYS$OUTPUT "Good afternoon!"
$ EXIT
```

This example shows a procedure that displays a different message, depending on whether the current time is morning or afternoon. It first obtains the current time of day by using the F$TIME function. The F$TIME function returns a character string, which is the string argument for the F$EXTRACT function. The F$TIME function is automatically evaluated when it is used as an argument, so you do not need to use quotation marks.

Next, the F$EXTRACT function extracts the hours from the date and time string returned by F$TIME. The string returned by F$TIME always contains the hours field beginning at an offset of 12 characters from the start of the string.

The F$EXTRACT function extracts two characters from the string, beginning at this offset, and compares the string value extracted with the string value 12. If the comparison is true, then the procedure writes "Good afternoon!". Otherwise, it writes "Good morning!".

Note that you can also use the F$CVTIME function to extract the hour field from a time specification. This method is easier than the one shown in the above example.

F$FAO

Invokes the $FAO system service to convert the specified control string to a formatted ASCII output string. By specifying arguments for FAO directives in the control string, you can use the F$FAO function to:

- Insert variable character string data into an output string
- Convert integer values into the ASCII representations of their decimal, hexadecimal, and octal equivalents, and substitute the results into the output string

FORMAT

F$FAO*(control-string[,arg1,arg2...arg15])*

return value

A character string containing formatted ASCII output. This output string is created from the fixed text and FAO directives in the control string.

ARGUMENTS

control-string

The fixed text of the output string, plus FAO directives. The control string may be any length and may contain any number of FAO directives. Specify the control string as a character string expression.

The F$FAO function uses FAO directives to modify or insert ASCII data into the fixed text in the control string. Table DCL–4 lists the FAO directives you can specify in a control string.

arg1,arg2...arg15

Arguments that are required by the FAO directives used in the control string. Specify the arguments arg1,arg2...arg15 as integer or character string expressions. Table DCL–4 lists the argument types required by each FAO directive.

If you specify an argument whose type (integer or string) does not match that of the corresponding directive, unpredictable results will be returned. You can use the F$INTEGER and F$STRING lexical functions to convert arguments to the proper type.

FAO directives may require one or more arguments. The order of the arguments must correspond exactly with the order of the directives in the control string. In most cases, an error message is not displayed if you misplace an argument.

If there are not enough arguments listed, F$FAO will continue reading past the end of an argument list. Therefore, always be sure to include enough arguments to satisfy the requirements of all the directives in a control string.

DESCRIPTION

You can specify an FAO directive using any one of the following formats:

Format	Function
!DD	One directive
!n(DD)	A directive that is repeated a specified number of times
!lengthDD	A directive that places its output into a field with a specified length
!n(lengthDD)	A directive that is repeated a specified number of times and generates output fields of a specified length

The exclamation point (!) indicates that the following character or characters are to be interpreted as an FAO directive. DD represents a one- or two-character uppercase code indicating the action that F$FAO is to perform. When specifying repeat counts, n is a decimal value specifying the number of times the directive is to be repeated. The length value is a decimal value that instructs F$FAO to generate an output field of "length" characters.

Repeat counts and output lengths may also be specified by using a number sign (#) in place of an absolute numeric value. If you use a number sign (#), you must specify the numeric value as an integer expression in the corresponding place in the argument list.

When a variable output field is specified with a repeat count, only one length parameter is required, since each output string will have the specified length.

The FAO directives are grouped in the following categories:

- Character string insertion
- Zero-filled numeric conversion
- Blank-filled numeric conversion
- Special formatting
- Parameter interpretation

Table DCL–4 summarizes the FAO directives and shows the required argument types. In addition, the following sections describe output strings from directives that perform character string insertion, zero-filled numeric conversion, and blank-filled numeric conversion.

Table DCL–4 Summary of FAO Directives

Directive	Description	Argument Type
Character string insertion:		
!AS	Inserts a character string as is	String
Zero-filled numeric conversion:		
!OB	Converts a byte to octal notation	Integer
!OW	Converts a word to octal notation	Integer
!OL	Converts a longword to octal notation	Integer
!XB	Converts a byte to hexadecimal notation	Integer

Table DCL–4 (Cont.) Summary of FAO Directives

Directive	Description	Argument Type
!XW	Converts a word to hexadecimal notation	Integer
!XL	Converts a longword to hexadecimal notation	Integer
!ZB	Converts a byte to decimal notation	Integer
!ZW	Converts a word to decimal notation	Integer
!ZL	Converts a longword to decimal notation	Integer
Blank-filled numeric conversion:		
!UB	Converts a byte to decimal notation without adjusting for negative numbers	Integer
!UW	Converts a word to decimal notation without adjusting for negative numbers	Integer
!UL	Converts a longword to decimal notation without adjusting for negative numbers	Integer
!SB	Converts a byte to decimal notation with negative numbers converted properly	Integer
!SW	Converts a word to decimal notation with negative numbers converted properly	Integer
!SL	Converts a longword to decimal notation with negative numbers converted properly	Integer
Special formatting:		
!/	Inserts a carriage return and a line feed	None
!_	Inserts a tab	None
!^	Inserts a form feed	None
!!	Inserts an exclamation mark	None
!%I	Converts a longword integer to a named UIC in the format [group-identifier,member-identifer]	Integer
!%S	Inserts an "s" if the most recently converted number is not 1	None
!%U	Converts a longword integer to a numeric UIC in the format [g,m], where g is the group number and m is the member number. The directive inserts the brackets and the comma.	Integer
!n <...!>	Left-justifies and blank-fills all data represented by the instructions ... in fields n characters wide	None
!n•c	Repeats the character represented by c for n times	None
!%T	Inserts the current time	Integer equal to 0
!%D	Inserts the current date/time	Integer equal to 0

Table DCL–4 (Cont.) Summary of FAO Directives

Directive	Description	Argument Type
Argument interpretation:		
!-	Reuses the last argument	None
!+	Skips the next argument	None

Output Strings from Character String Insertion

The !AS directive inserts a character string (specified as an argument for the directive) into the control string. The field length of the character string when it is inserted into the control string defaults to the length of the character string. If the default length is shorter than an explicitly stated field length, the string is left-justified and blank-filled. If the default length is longer than an explicitly stated field length, the string is truncated on the right.

Output Strings from Zero-Filled Numeric Conversion

Directives for zero-filled numeric conversion convert an integer (specified as an argument for the directive) to decimal, octal, or hexadecimal notation. The ASCII representation of the integer is inserted into the control string. Default output field lengths for the converted argument are determined as follows.

Directives that convert arguments to octal notation return 3 digits for byte conversion, 6 digits for word conversion, and 11 digits for longword conversion. Numbers are right-justified and zero-filled on the left. Explicit-length fields longer than the default are blank-filled on the left. Explicit-length fields shorter than the default are truncated on the left.

Directives that convert arguments to hexadecimal notation return 2 digits for byte conversion, 4 digits for word conversion, and 8 digits for longword conversion. Numbers are right-justified and zero-filled on the left. Explicit-length fields longer than the default are blank-filled on the left. Explicit-length fields shorter than the default are truncated on the left.

Directives that convert arguments to decimal notation return the required number of characters for the decimal number. Explicit-length fields longer than the default are zero-filled on the left. If an explicit-length field is shorter than the number of characters required for the decimal number, the output field is completely filled with asterisks (*).

For byte conversion, only the low-order 8 bits of the binary representation of the argument are used. For word conversion, only the low-order 16 bits of the binary representation of the argument are used. For longword conversion, the entire 32-bit binary representation of the argument is used.

Output Strings from Blank-Filled Numeric Conversion

Directives for blank-filled numeric conversion convert an integer (specified as an argument for the directive) to decimal notation. These directives can convert the integer as a signed or unsigned number. The ASCII representation of the integer is inserted into the control string.

Output field lengths for the converted argument default to the required number of characters. Values shorter than explicit-length fields are right-justified and blank-filled; values longer than explicit-length fields cause the field to be filled with asterisks.

For byte conversion, only the low-order 8 bits of the binary representation of the argument are used. For word conversion, only the low-order 16 bits of the binary representation of the argument are used. For longword conversion, the entire 32-bit binary representation of the argument is used.

EXAMPLES

1
```
$ COUNT = 57
$ REPORT = F$FAO("NUMBER OF FORMS = !SL",COUNT)
$ SHOW SYMBOL REPORT
$ REPORT = "NUMBER OF FORMS = 57"
```

In this command procedure, the FAO directive !SL is used in a control string to convert the number equated to the symbol COUNT to a character string. The converted string is inserted into the control string.

Note that COUNT is assigned an integer value of 57. The F$FAO function returns the ASCII string, "NUMBER OF FORMS = 57", and assigns the string to the symbol REPORT.

2
```
$ A = "ERR"
$ B = "IS"
$ C = "HUM"
$ D = "AN"
$ PHRASE = F$FAO("TO !3(AS)",A,B,C+D)
$ SHOW SYMBOL PHRASE
$ PHRASE = "TO ERRISHUMAN"
```

In this command procedure, the !AS directive is used to insert the values assigned to the symbols A, B, C, and D into the control string.

Since the specified repeat count for the !AS directive is 3, F$FAO looks for three arguments. The arguments in this example include the symbol A ("ERR"), the symbol B ("IS"), and the expression C+D ("HUMAN"). Note that the values of these string arguments are concatenated to form the string "ERRISHUMAN".

3
```
$ A = "ERR"
$ B = "IS"
$ C = "HUMAN"
$ PHRASE = F$FAO("TO !#(#AS)",3,6,A,B,C)
$ SHOW SYMBOL PHRASE
$ PHRASE = "TO ERR   IS    HUMAN "
```

In this command procedure, the F$FAO function is used with the !AS directive to format a character string. The first number sign (#) represents the repeat count given by the first argument, 3. The second number sign (#) represents the field size given by the second argument, 6. The next three arguments (A,B,C) provide the strings that are placed into the control string each time the !AS directive is repeated.

Each argument string is output to a field having a length of six characters. Since each string is less than six characters, each field is left-justified and padded with blank spaces. The resulting string is assigned to the symbol PHRASE.

F$FILE_ATTRIBUTES

Returns attribute information for a specified file.

FORMAT

F$FILE_ATTRIBUTES*(file-spec,item)*

return value

Either an integer or a character string, depending on the item you request. Table DCL–5 shows the data types of the values returned for each item.

ARGUMENTS

file-spec

The name of the file about which you are requesting information. You must specify the file name as a character string expression.

Only one file name may be specified. No wildcard characters are allowed in the file specification.

item

Indicates which attribute of the file is to be returned. The item must be specified as a character string expression, and can be any one of the VAX RMS field names listed in Table DCL–5.

DESCRIPTION

Use the F$FILE_ATTRIBUTES lexical function in DCL assignment statements and expressions to return file attribute information. Table DCL–5 lists the items you can specify with the F$FILE_ATTRIBUTES function, the information returned, and the data type of this information.

Table DCL–5 F$FILE_ATTRIBUTES Items

Item	Information Returned	Data Type
ALQ	Allocation Quantity	Integer
BDT	Backup Date/Time	String
BKS	Bucket Size	Integer
BLS	Block Size	Integer
CBT	True If Contiguous-Best-Try; Returns the values "TRUE","FALSE"	Boolean
CDT	Creation Date/Time	String
CTG	True If Contiguous; Returns the values "TRUE", "FALSE"	Boolean
DEQ	Default Extension Quantity	Integer
DID	Directory ID String	String
DVI	Device Name String	String
EDT	Expiration Date/Time	String
EOF	Number of Blocks Used	Integer

Table DCL–5 (Cont.) F$FILE_ATTRIBUTES Items

Item	Information Returned	Data Type
FID	File ID String	String
FSZ	Fixed Control Area Size	Integer
GRP	Owner Group Number	Integer
KNOWN	Known File; Returns the values "TRUE", "FALSE" to indicate "TRUE", "FALSE" to indicate whether file is installed with the Install Utility	Boolean
MBM	Owner Member Number	Integer
MRN	Maximum Record Number	Integer
MRS	Maximum Record Size	Integer
NOA	Number of Areas	Integer
NOK	Number of Keys	Integer
ORG	File Organization; Returns the values "SEQ","REL","IDX"	String
PRO	File Protection String	String
PVN	Prolog Version Number	Integer
RAT	Record Attributes; Returns the values "CR","PRN","FTN", ""	String
RCK	True If Read Check; Returns the values "TRUE","FALSE"	Boolean
RDT	Revision Date/Time	String
RFM	Record Format String; Returns the values "VAR","FIX","VFC","UDF" "STM","STMLF","STMCR"	String
RVN	Revision Number	Integer
UIC	Owner UIC String	String
WCK	True If Write Check; Returns the values "TRUE","FALSE"	Boolean

File attributes are stored in the file header, which is created from information in VAX RMS control blocks. For more information on VAX RMS control blocks, see the *VAX Record Management Services Reference Manual*.

EXAMPLES

```
1   $ FILE_ORG = F$FILE_ATTRIBUTES("QUEST.DAT","ORG")
    $ SHOW SYMBOL FILE_ORG
      FILE_ORG = "SEQ"
```

This example uses the F$FILE_ATTRIBUTES function to assign the value of the file organization type to the symbol FILE_ORG. The F$FILE_ATTRIBUTES function returns the character string "SEQ" to show that QUEST.DAT is a sequential file.

The QUEST.DAT and ORG arguments for the F$FILE_ATTRIBUTES function are string literals, and must be enclosed in quotation marks when used in expressions.

2
```
$ RFM = F$FILE_ATTRIBUTES("KANSAS::USE$:[CARS]SALES.CMD","RFM")
$ SHOW SYMBOL RFM
  RFM = "VAR"
```

This example uses the F$FILE_ATTRIBUTES function to return information about a file on a remote node. The function returns the record format string VAR, indicating that records are variable length.

F$GETDVI

Invokes the $GETDVI system service to return a specified item of information for a specified device. This function allows a process to obtain information for a device to which the process has not necessarily assigned a channel.

FORMAT

F$GETDVI*(device-name,item)*

return value

Either an integer or a character string, depending on the item you request. Table DCL–6 shows the data types of the values returned for each item.

ARGUMENTS

device-name

A physical device name or a logical name equated to a physical device name. Specify the device name as a character string expression.

After the device-name expression is evaluated, the F$GETDVI function examines the first character of the name. If the first character is an underscore (_), the name is considered a physical device name. Otherwise, a single level of logical name translation is performed and the equivalence name, if any, is used.

item

Specifies the type of device information to be returned. The item must be specified as a character string expression. You can specify any one of the items listed in Table DCL–6.

DESCRIPTION

The F$GETDVI function returns information on all items that can be specified with the $GETDVI system service. In addition to the items that the $GETDVI system service allows, the F$GETDVI function allows you to specify the item EXISTS.

Table DCL–6 lists the items you can specify with the F$GETDVI function, the type of information returned, and the data types of the return values. For more information on the $GETDVI system service and the items you can specify, see the *VAX/VMS System Services Reference Manual*.

Table DCL–6 F$GETDVI Items

Item	Information Returned
ACPPID	Returns the ACP process ID, as a character string.
ACPTYPE	Returns the ACP type code, as one of the following strings: "F11CV1", "F11V2", "JNL", "MTA", "NET", or "REM".
ALL	Returns either the string "TRUE" or "FALSE" to indicate whether the device is allocated.
ALLDEVNAM	Returns the allocation class device name, as a string.

Table DCL–6 (Cont.) F$GETDVI Items

Item	Information Returned
ALLOCLASS	Returns the allocation class of the host, as a longword integer between 0 and 255.
ALT_HOST_AVAIL	Returns either the string "TRUE" or "FALSE" to indicate whether the host serving the alternate path is available.
ALT_HOST_NAME	Returns the name of the host serving the alternate path, as a string.
ALT_HOST_TYPE	Returns the hardware type of the host serving the alternate path, as a string.
AVL	Returns either the string "TRUE" or "FALSE" to indicate whether the device is available for use.
CCL	Returns either the string "TRUE" or "FALSE" to indicate whether the device is a carriage control device.
CLUSTER	Returns the volume cluster size, as an integer.
CONCEALED	Returns either the string "TRUE" or "FALSE" to indicate whether the logical device name translates to a concealed device.
CYLINDERS	Returns the number of cylinders on the volume (disk), as an integer.
DEVBUFSIZ	Returns the device buffer size, as an integer.
DEVCHAR	Returns the device characteristics, as an integer.
DEVCHAR2	Returns additional device characteristics, as an integer.
DEVCLASS	Returns the device class, as an integer. See Table DCL–7 for a list of the values returned.
DEVDEPEND	Returns device-dependent information, as an integer.
DEVDEPEND2	Returns additional device-dependent information, as an integer.
DEVLOCKNAM	Returns a unique lock name for the device, as a string.
DEVNAM	Returns the device name, as a string.
DEVSTS	Returns device dependent status information, as an integer.
DEVTYPE	Returns the device type, as an integer. See Table DCL–8 for a list of the values returned.
DIR	Returns either the string "TRUE" or "FALSE" to indicate whether the device is directory structured.
DMT	Returns either the string "TRUE" or "FALSE" to indicate whether the device is marked for dismount.
DUA	Returns either the string "TRUE" or "FALSE" to indicate whether the device is a generic device.
ELG	Returns either the string "TRUE" or "FALSE" to indicate whether the device has error logging enabled.
ERRCNT	Returns the error count, as an integer.
EXISTS	Returns either the string "TRUE" or "FALSE" to indicate whether the device exists on the system.

Table DCL–6 (Cont.) F$GETDVI Items

Item	Information Returned
FOD	Returns either the string "TRUE" or "FALSE" to indicate whether the device is a files-oriented device.
FOR	Returns either the string "TRUE" or "FALSE" to indicate whether the device is mounted foreign.
FREEBLOCKS	Returns the number of free blocks on the volume (disk), as an integer.
FULLDEVNAM	Returns the fully qualified device name, as a string.
GEN	Returns either the string "TRUE" or "FALSE" to indicate whether the device is a generic device.
HOST_AVAIL	Returns either the string "TRUE" or "FALSE" to indicate whether the host serving the primary path is available.
HOST_COUNT	Returns the number of hosts that make the device available to other nodes in the VAXcluster, as an integer.
HOST_NAME	Returns the name of the host serving the primary path, as a string.
HOST_TYPE	Returns the hardware type of the host serving the primary path, as a string.
IDV	Returns either the string "TRUE" or "FALSE" to indicate whether the device is capable of providing input.
LOCKID	Returns the cluster-wide lock identification, as an integer.
LOGVOLNAM	Returns the logical volume name, as a string.
MAXBLOCK	Returns the number of logical blocks on the volume, as an integer.
MAXFILES	Returns the maximum number of files on the volume, as an integer. This item code is applicable only to disks.
MBX	Returns either the string "TRUE" or "FALSE" to indicate whether the device is a mailbox.
MEDIA_ID	Returns the nondecoded media ID, as a string.
MEDIA_NAME	Returns either the name of the disk or the tape type, as a string.
MEDIA_TYPE	Returns the device name prefix, as a string.
MNT	Returns either the string "TRUE" or "FALSE" to indicate whether the device is mounted.
MOUNTCNT	Returns the mount count, as an integer.
NET	Returns either the string "TRUE" or "FALSE" to indicate whether the device is a network device.
NEXTDEVNAM	Returns the device name of the next volume in a volume set, as a string. This item applies only to disks.
ODV	Returns either the string "TRUE" or "FALSE" to indicate whether the device is capable of providing output.
OPCNT	Returns the operation count, as an integer.
OPR	Returns either the string "TRUE" or "FALSE" to indicate whether the device is an operator.

Table DCL–6 (Cont.) F$GETDVI Items

Item	Information Returned
OWNUIC	Returns the UIC of the device owner, as a string.
PID	Returns the process identification of the device owner, as a string.
RCK	Returns either the string "TRUE" or "FALSE" to indicate whether the device has read checking enabled.
RCT	Returns either the string "TRUE" or "FALSE" to indicate whether the disk contains RCT.
REC	Returns either the string "TRUE" or "FALSE" to indicate whether the device is record oriented.
RECSIZ	Returns the blocked record size, as an integer.
REFCNT	Returns the reference count of processes using the device, as an integer.
REMOTE_DEVICE	Returns either the string "TRUE" or "FALSE" to indicate whether the device is a remote device.
RND	Returns either the string "TRUE" or "FALSE" to indicate whether the device allows random access.
ROOTDEVNAM	Returns the device name of the root volume in a volume set, as a string. This item applies only to disks.
RTM	Returns either the string "TRUE" or "FALSE" to indicate whether the device is real-time.
SDI	Returns either the string "TRUE" or "FALSE" to indicate whether the device is single directory structured.
SECTORS	Returns the number of sectors per track, as an integer. This item applies only to disks.
SERIALNUM	Returns the volume serial number, as an integer. This item applies only to disks.
SERVED_DEVICE	Returns either the string "TRUE" or "FALSE" to indicate whether the device is a served device.
SHR	Returns either the string "TRUE" or "FALSE" to indicate whether the device is shareable.
SPL	Returns either the string "TRUE" or "FALSE" to indicate whether the device is being spooled.
SPLDEVNAM	Returns the name of the device being spooled, as a string.
SQD	Returns either the string "TRUE" or "FALSE" to indicate whether the device is sequential block-oriented (that is, magnetic tape).
STS	Returns status information, as an integer.
SWL	Returns either the string "TRUE" or "FALSE" to indicate whether the device is software write-locked.
TRACKS	Returns the number of tracks per cylinder, as an integer. This item applies only to disks.
TRANSCNT	Returns the volume transaction count, as an integer.
TRM	Returns either the string "TRUE" or "FALSE" to indicate whether the device is a terminal.

Table DCL–6 (Cont.) F$GETDVI Items

Item	Information Returned
TT_ALTYPEAHD	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal has an alternate type-ahead buffer (terminals only).
TT_ANSICRT	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal is an ANSI CRT terminal (terminals only).
TT_APP_KEYPAD	Returns either the string "TRUE" or "FALSE" to indicate whether the keypad is in applications mode (terminals only).
TT_AUTOBAUD	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal has automatic baud rate detection (terminals only).
TT_AVO	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal has a VT100-family terminal display (terminals only).
TT_BLOCK	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal has block mode capability (terminals only).
TT_BRDCSTMBX	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal uses mailbox broadcast messages (terminals only).
TT_CRFILL	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal requires fill after RET (terminals only).
TT_DECCRT	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal is a DIGITAL CRT terminal (terminals only).
TT_DECCRT2	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal is a DIGITAL CRT2 terminal (terminals only).
TT_DIALUP	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal is connected to dialup (terminals only).
TT_DISCONNECT	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal can be disconnected (terminals only).
TT_DMA	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal has DMA mode (terminals only).
TT_DRCS	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal supports loadable character fonts (terminals only).
TT_EDIT	Returns either the string "TRUE" or "FALSE" to indicate whether the edit characteristic is set.
TT_EDITING	Returns either the string "TRUE" or "FALSE" to indicate whether advanced editing is enabled (terminals only).

Table DCL–6 (Cont.) F$GETDVI Items

Item	Information Returned
TT_EIGHTBIT	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal uses the 8-bit ASCII character set (terminals only).
TT_ESCAPE	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal generates escape sequences (terminals only).
TT_FALLBACK	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal uses the multinational fallback option (terminals only).
TT_HALFDUP	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal is in half-duplex mode (terminals only).
TT_HANGUP	Returns either the string "TRUE" or "FALSE" to indicate whether the hangup characteristic is set (terminals only).
TT_HOSTSYNC	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal has host/terminal communication (terminals only).
TT_INSERT	Returns either the string "TRUE" or "FALSE" to indicate whether insert-mode is the default line editing mode (terminals only).
TT_LFFILL	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal requires fill after LF (terminals only).
TT_LOCALECHO	Returns either the string "TRUE" or "FALSE" to indicate whether the local echo characteristic is set (terminals only).
TT_LOWER	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal has the lowercase characters set.
TT_MBXDSABL	Returns either the string "TRUE" or "FALSE" to indicate whether mailboxes associated with the terminal will receive unsolicited input notification or input notification (terminals only).
TT_MECHFORM	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal has mechanical form feed (terminals only).
TT_MECHTAB	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal has mechanical tabs and is capable of tab expansion (terminals only).
TT_MODEM	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal is connected to a modem (terminals only).
TT_MODHANGUP	Returns either the string "TRUE" or "FALSE" to indicate whether the modify hang-up characteristic is set (terminals only).
TT_NOBRDCST	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal will receive broadcast messages (terminals only).

Table DCL–6 (Cont.) F$GETDVI Items

Item	Information Returned
TT_NOECHO	Returns either the string "TRUE" or "FALSE" to indicate whether the input characters are echoed.
TT_NOTYPEAHD	Returns either the string "TRUE" or "FALSE" to indicate whether data must be solicited by a read operation.
TT_OPER	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal is an operator terminal (terminals only).
TT_PAGE	Returns the terminal page length, as an integer (terminals only).
TT_PASTHRU	Returns either the string "TRUE" or "FALSE" to indicate whether there is passall with flow control (terminals only).
TT_PHYDEVNAM	Returns the physical device name associated with a channel number or virtual terminal, as a string.
TT_PRINTER	Returns either the string "TRUE" or "FALSE" to indicate whether there is a printer port available (terminals only).
TT_READSYNC	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal has read synchronization (terminals only).
TT_REGIS	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal has REGIS graphics (terminals only).
TT_REMOTE	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal has established modem control (terminals only).
TT_SCOPE	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal is a video screen display (terminals only).
TT_SECURE	Returns either the string "TRUE" or "FALSE" to indicate whether the terminal can recognize the secure server (terminals only).
TT_SETSPEED	Returns either the string "TRUE" or "FALSE" to indicate whether you can set the speed on the terminal line (terminals only).
TT_SIXEL	Returns either the string "TRUE" or "FALSE" to indicate whether the sixel is supported (terminals only).
TT_TTSYNC	Returns either the string "TRUE" or "FALSE" to indicate whether there is terminal/host synchronization (terminals only).
TT_SYSPWD	Returns either the string "TRUE" or "FALSE" to indicate whether the system password is enabled for a particular terminal.
TT_WRAP	Returns either the string "TRUE" or "FALSE" to indicate whether a new line should be inserted if the cursor moves beyond the right margin.
UNIT	Returns the unit number, as an integer.

Table DCL–6 (Cont.) F$GETDVI Items

Item	Information Returned
VOLCOUNT	Returns the count of volumes in a volume set, as an integer. This item applies only to disks.
VOLNAM	Returns the volume name, as a string.
VOLNUMBER	Returns the number of the current volume in a volume set, as an integer. This item applies only to disks.
VOLSETMEM	Returns the string "TRUE" or "FALSE" to indicate whether the device is a volume set (disks only).
VPROT	Returns the volume protection mask, as a string.
WCK	Returns either the string "TRUE" or "FALSE" to indicate whether the device has write checking enabled.

Table DCL–7 Values Returned by the DEVCLASS Item

Device Class	Value Returned	Symbolic Name Used by $GETDVI
Disk device	1	DC$_DISK
Tape device	2	DC$_TAPE
Synchronous Communications device	32	DC$_SCOM
Card reader	65	DC$_CARD
Terminal	66	DC$_TERM
Line printer	67	DC$_LP
Real-time	96	DC$_REALTIME
Bus	128	DC$_BUS
Mailbox	160	DC$_ MAILBOX
Journal	161	DC$_JOURNAL
Miscellaneous device	200	DC$_MISC

Table DCL–8 Values Returned by the DEVTYPE Item

Device Type	Value	Device Type	Value
RK06	1	VT102	98
RK07	2	VT105	99
RP04	3	VT125	100
RP05	4	VT131	101
RP06	5	VT132	102
RMO3	6	DZ11	66
RP07	7	DZ32	67
RP07HT	8	DZ730	68
RL01	9	DMC11	1
RL02	10	DMR11	2

Table DCL–8 (Cont.) Values Returned by the DEVTYPE Item

Device Type	Value	Device Type	Value
RX02	11	XK_3271	3
RX04	12	XJ_2780	4
RM80	13	NW_X25	5
TU58	14	NV_X29	6
RM05	15	SB_ISB11	7
RX01	16	MX_MUX200	8
ML11	17	DMP11	9
RB02	18	DMF32	10
RB80	19	XV_3271	11
RA80	20	CI	12
RA81	21	NI	13
RA60	22	UNA11	14
RZ01	23	YN_X25	15
RZF01	24	YO_X25	16
RD51	25	YP_ADCCP	17
RX50	26	YQ_3271	18
TE16	1	YR_DDCMP	19
TU45	2	YS_SDLC	20
TU77	3	LP11	1
TS11	4	LA11	2
TU78	5	LA180	3
TA78	6	CR11	1
TU80	7	MBX	1
TU81	8	SHRMBX	2
TA81	9	NULL	3
TTYUNKN	0	LPA11	1
VT105	1	DR780	2
FT1	16	DR750	3
FT2	17	DR11W	4
FT3	18	PCL11R	5
FT4	19	PCL11T	6
FT5	20	DR11C	7
FT6	21	XI_DR11C	8
FT7	22	XP_PCL11B	9
FT8	23	CI780	1
LAX	32	C1750	2
LA36	32	UQPORT	3
LA120	33	UDA50	3
VT5X	64	UDA50A	4

Table DCL–8 (Cont.) Values Returned by the DEVTYPE Item

Device Type	Value	Device Type	Value
VT52	64	RC25	23
VT55	65	TU81P	6
VT100	96	RDRX	7
VK100	2	UNKNJNL	0
VT173	3	RUJNL	1
LA34	34	BIJNL	2
LA38	35	AIJNL	3
LA12	36	ATJNL	4
LA24	37	CLJNL	5
LA100	37	DN11	1
LQP02	38	VT200	110
VT101	97	LA210	40

EXAMPLE

```
$ ERR = F$GETDVI("_DQA0","ERRCNT")
$ SHOW SYMBOL ERR
  ERR = 0  Hex = 00000000 Octal = 000000
```

This example shows how to use the F$GETDVI function to return an error count for the device DQA0. You must place quotation marks around the device-name _DQA0 and the item ERRCNT because they are string literals.

F$GETJPI

Invokes the $GETJPI system service to return accounting, status, and identification information on the specified process. GROUP privilege is required to obtain information on other processes in the same group. WORLD privilege is required to obtain information on any other processes in the system.

FORMAT

F$GETJPI*(pid,item)*

return value

Either an integer or a character string, depending on the item you request. Table DCL–9 shows the data types of the values returned for each item.

ARGUMENTS

pid

The identification number of the process for which information is being reported. Specify the pid argument as a character string expression. When you specify the pid argument, you can omit the leading zeroes.

If you specify a null string (""), the current process identification number is used.

You cannot use a wildcard to specify the pid argument in the F$GETJPI function, as you can with the $GETJPI system service. To get a list of process identification numbers, use the F$PID function.

item

The type of process information to be returned. Specify the item argument as a character string expression. You may specify any one of the items listed in Table DCL–9.

DESCRIPTION

The F$GETJPI function returns information on all items that can be specified with the $GETJPI system service. For more information on the $GETJPI system service, see the *VAX/VMS System Services Reference Manual.*

Table DCL–9 lists the items you can specify with the F$GETJPI function, the information returned, and the data type of this information.

If you use the F$GETJPI function to request information on the null process or the swapper process, you can specify any of the items in Table DCL–9 except: ACCOUNT, BYTLM, ENQCNT, ENQLM, EXCVEC, FILCNT, FILM, FINALEXC, IMAGNAME, LOGINTIM, MSGMASK, PAGFILCNT, PGFLQUOTA, PRCCNT, PRCLM, PROCPRIV, SITESPEC, TQCNT, TQLM, USERNAME, VIRTPEAK, VOLUMES, WSPEAK.

Table DCL–9 F$GETJPI Items

Item	Information Returned
ACCOUNT	Returns the account name, as a string containing 8 characters filled with trailing blanks.
APTCNT	Returns the active page table count, as an integer.
ASTACT	Returns the access modes with active ASTs, as an integer.
ASTCNT	Returns the remaining AST quota, as an integer.
ASTEN	Returns the access modes with ASTs enabled, as an integer.
ASTLM	Returns the AST limit quota, as an integer.
AUTHPRI	Returns the maximum priority that a process without the user privilege ALTPRI can achieve with the $SETPRI system service. This priority is returned as an integer.
AUTHPRIV	Returns the privileges the process is authorized to enable as a string.
BIOCNT	Returns the remaining buffered I/O quota, as an integer.
BIOLM	Returns the buffered I/O limit quota, as an integer.
BUFIO	Returns the count of process buffered I/O operations, as an integer.
BYTCNT	Returns the remaining buffered I/O count quota, as an integer.
BYTLM	Returns the buffered I/O byte count limit quota, as an integer.
CLINAME	Returns the string "DCL" to indicate the name of the command language interpreter the process is using.
CPULIM	Returns the limit on process CPU time, as an integer.
CPUTIM	Returns the CPU time used in hundredths of a second. This value is returned as an integer.
CURPRIV	Returns the process's current privileges, as a string.
DFPFC	Returns the default page fault cluster size, as an integer.
DFWSCNT	Returns the default working set size, as an integer.
DIOCNT	Returns the remaining direct I/O quota, as an integer.
DIOLM	Returns the direct I/O limit quota, as an integer.
DIRIO	Returns the count of direct I/O operations for the process, as an integer.
EFCS	Returns the local event flags 0 through 31 as an integer.
EFCU	Returns the local event flags 32 through 63, as an integer.
EFWM	Returns the event flag wait mask, as an integer.
ENQCNT	Returns the lock request quota remaining, as an integer.
ENQLM	Returns the lock request quota limit, as an integer.
EXCVEC	Returns the address of a list of exception vectors, as an integer.
FILCNT	Returns the remaining open file quota, as an integer.
FILLM	Returns the open file quota, as an integer.
FINALEXC	Returns the address of a list of final exception vectors, as an integer.
FREPOVA	Returns the first free page at end of program region (P0 space), as an integer.

Table DCL–9 (Cont.) F$GETJPI Items

Item	Information Returned
FREP1VA	Returns the first free page at end of control region (P1 space), as an integer.
FREPTECNT	Returns the number of pages available for virtual memory expansion, as an integer.
GPGCNT	Returns the global page count in the working set, as an integer.
GRP	Returns the group number of the UIC, as an integer.
IMAGECOUNT	Returns the number of images that have been run down for the process. This value is returned as an integer.
IMAGNAME	Returns the current image file name, as a string.
IMAGPRIV	Returns the privileges with which the current image was installed. This list is returned as a string.
JOBPRCCNT	Returns the number of subprocesses owned, as an integer.
LOGINTIM	Returns the process creation time, as a string.
MASTER_PID	Returns the process identification of the process at the top of the current job's process tree.
MEM	Returns the member number of the UIC, as an integer.
MODE	Returns the current process mode as one of the following strings: "BATCH", "INTERACTIVE", "NETWORK", or "OTHER".
MSGMASK	Returns the default message mask, as an integer.
OWNER	Returns the process identification number of the owner, as a string.
PAGEFLTS	Returns the count of page faults, as an integer.
PAGFILCNT	Returns the remaining paging file quota, as an integer.
PAGFILLOC	Returns the paging file location, as an integer.
PGFLQUOTA	Returns the paging file quota (maximum virtual page count), as an integer.
PHDFLAGS	Returns the flags word, as an integer.
PID	Returns the process identification number, as a string.
PPGCNT	Returns the process page count, as an integer.
PRCCNT	Returns the count of subprocesses, as an integer.
PRCLM	Returns the subprocess quota, as an integer.
PRCNAM	Returns the process name, as a string.
PRI	Returns the current process priority, as an integer.
PRIB	Returns the process's base priority, as an integer.
PROCPRIV	Returns the process's default privileges, as a string.
SITESPEC	Returns the contents of the per-process site-specific longword, as an integer.
STATE	Returns the process state, as a string.
STS	Returns the process status flags, as an integer.
SWPFILLOC	Returns the swapping file location, as an integer.
TERMINAL	Returns the login terminal name for interactive users, as a string.

Table DCL–9 (Cont.) F$GETJPI Items

Item	Information Returned
TMBU	Returns the termination mailbox unit number, as an integer.
TQCNT	Returns the remaining timer queue entry quota, as an integer.
TQLM	Returns the timer queue entry quota, as an integer.
UIC	Returns the process's UIC, as a string.
USERNAME	Returns the user name string (12 characters filled with trailing blanks), as a string.
VIRTPEAK	Returns the peak virtual address size, as an integer.
VOLUMES	Returns the count of currently mounted volumes, as an integer.
WSAUTH	Returns the maximum authorized working set size, as an integer.
WSAUTHEXT	Returns the maximum authorized working set extent, as an integer.
WSEXTENT	Returns the current working set size extent, as an integer.
WSPEAK	Returns the working set peak, as an integer.
WSQUOTA	Returns the working set size quota, as an integer.
WSSIZE	Returns the process's current working set size, as an integer.

EXAMPLE

```
$ NAME = F$GETJPI("3B0018","USERNAME")
$ SHOW SYMBOL NAME
  NAME = "JANE        "
```

This example shows how to use the F$GETJPI function to return the username for the process number 3B0018. The username is assigned to the symbol NAME.

F$GETSYI

Invokes the $GETSYI system service to return status and identification information about the local system (or about a node in the local VAXcluster, if your system is part of a VAXcluster).

FORMAT

F$GETSYI*(item [,node])*

return value

Either an integer or a character string, depending on the item you request.

ARGUMENTS

item

The type of information to be reported about the local node (or about another node in your VAXcluster, if your system is part of a VAXcluster). Specify the item as a character string expression.

You can specify the items in Table DCL–10 only for your local node; you cannot specify the node argument with these items. You can specify these items whether or not you are in a VAXcluster.

You can specify the items in Table DCL–11 for either your local node or for another node in your VAXcluster. The information in this table is returned for your local node if you do not specify the node argument; the information is returned for the specified node if you include the node argument. Your system must be a member of a VAXcluster in order to specify the items in this table, except for CLUSTER_MEMBER. You can specify CLUSTER_MEMBER whether or not your system is a member of a VAX_CLUSTER.

You can also specify any of the SYSGEN parameters listed in Section A of the *VAX/VMS System Generation Utility Reference Manual*. However, you can specify SYSGEN parameters only for your local node; you cannot specify the node argument with these items.

node

The node in your VAXcluster for which information is to be returned. Specify the node as a character string expression. (This argument can be specified only if your system is part of a VAXcluster.)

You can request information about another node in your VAXcluster only when you specify an item from Table DCL–11. If you do not specify a node, the default is the current node.

You cannot use wildcards to specify the node argument with the F$GETSYI function (as you can with the $GETSYI system service).

DESCRIPTION

The F$GETSYI returns information on the items that can be specified with the $GETSYI system service. See the *VAX/VMS System Services Reference Manual* for more information on the $GETSYI system service.

Table DCL–10 lists the items you can specify with the F$GETSYI function to get information about your local node. Table DCL–11 lists the items you can specify to get information about either your local node, or another node in your VAXcluster.

Table DCL–10 F$GETSYI Items For the Local Node Only

Item	Information Returned
ARCHFLAG	Returns the architecture flags for the system, as a string.
BOOTTIME	Returns the time when the system was booted, as a string.
CHARACTER_EMULATED	Returns either the string "TRUE" or "FALSE" to indicate whether the character string instructions are emulated on the CPU.
CPU	Returns the CPU type, as an integer. The integer 1 represents a VAX–11/780, 2 represents a VAX–11/750, and 3 represents a VAX–11/730.
DECIMAL_EMULATED	Returns either the string "TRUE" or "FALSE" to indicate whether the decimal string instructions are emulated on the CPU.
D_FLOAT_EMULATED	Returns either the string "TRUE" or "FALSE" to indicate whether the D_floating instructions are emulated on the CPU.
F_FLOAT_EMULATED	Returns either the string "TRUE" or "FALSE" to indicate whether the F_floating instructions are emulated on the CPU.
G_FLOAT_EMULATED	Returns either the string "TRUE" or "FALSE" to indicate whether the G_floating instructions are emulated on the CPU.
PAGEFILE_FREE	Returns the number of free pages in the currently installed paging files, as an integer.
PAGEFILE_PAGE	Returns the number of pages in the currently installed paging files, as an integer.
SID	Returns the system identification register, as an integer.
SWAPFILE_FREE	Returns the number of free pages in the currently installed swapping files, as an integer.
SWAPFILE_PAGE	Returns the number of pages in the currently installed swapping files, as an integer.
VERSION	Returns the system version as an 8-character string filled with trailing blanks.

Table DCL–11 F$GETSYI Items for the Local Node or for Other Nodes in the VAXCluster

Item	Information Returned
CLUSTER_FSYSID	Returns the system identification number for first node to boot in the VAXcluster (the founding node). This number is returned as a character string containing a hexadecimal number.
CLUSTER_FTIME	Returns the time when the first node in the VAXcluster booted, as a string.
CLUSTER_MEMBER	Returns either the string "TRUE" or "FALSE" to indicate whether the node is a member of the local VAXcluster.
CLUSTER_NODES	Returns the total number of nodes in the VAXcluster, as an integer.
CLUSTER_QUORUM	Returns the total quorum for the VAXcluster.
CLUSTER_VOTES	Returns the total number of votes in the VAXcluster, as an integer.
NODENAME	Returns the node name, as a string.
NODE_AREA	Returns the VAX DECnet area for the target node, as an integer.
NODE_CSID	Returns the CSID of the specified node, as a string containing a hexadecimal number. The CSID is a form of system identification.
NODE_HWTYPE	Returns the hardware type of the specified node, as a string.
NODE_HWVERS	Returns the hardware version of the specified node, as a string.
NODE_NUMBER	Returns the VAX DECnet number for the specified node, as an integer.
NODE_QUORUM	Returns the quorum that the node has, as an integer.
NODE_SWINCARN	Returns the software incarnation number for the specified node. This number is returned as a string containing a hexadecimal number.
NODE_SWTYPE	Returns the type of operating system software used by the specified node, as a string.
NODE_SWVERS	Returns the software version of the specified node, as a string.
NODE_SYSTEMID	Returns the system identification number for the specified node. This number is returned as a string containing a hexadecimal number.
NODE_VOTES	Returns the number of votes that the node has, as an integer.
SCS_EXISTS	Returns either the string "TRUE" or "FALSE" to indicate whether the system communication subsystem (SCS) is currently loaded on a VAX node.

EXAMPLES

```
1   $ SYSID = F$GETSYI("SID")
    $ SHOW SYMBOL SYSID
      SID = 19923201  Hex = 01300101 Octal = 000401
```

This example shows how to use the F$GETSYI function to return the information in the system identification register. Use quotation marks around the argument SID because it is a string literal. The value returned by the F$GETSYI function is assigned to the symbol SYSID. Because a node is not specified, information about your current node is returned.

```
2   $ MEM = F$GETSYI("CLUSTER_MEMBER", "LONDON")
    $ SHOW SYMBOL MEM
      MEM = "TRUE"
```

This example uses the F$GETSYI function to determine whether the node LONDON is a member of the local VAXcluster. The "TRUE" indicates that the remote node LONDON is a member of the VAXcluster.

```
3   $ LIM = F$GETSYI("BJOBLIM")
    $ SHOW SYMBOL LIM
      LIM = 16    Hex = 00000010  Octal = 00000000020
```

This example uses the SYSGEN parameter BJOBLIM as as argument for the F$GETSYI function. This argument returns the batch job limit for the current system.

F$IDENTIFIER

Converts an identifier in named format to its integer equivalent, or vice versa. An identifier is a name or number that identifies a category of users of a data resource. The system uses identifiers to determine a user's access to a resource.

FORMAT

F$IDENTIFIER*(identifier,conversion-type)*

return value

An integer value if you are converting an identifier from a name to an integer. The F$IDENTIFIER function returns a string if you are converting an identifier from an integer to a name.

If you specify an identifier that is not valid, the F$IDENTIFIER function returns a null string (if you are converting from number to name) or a zero (if you are converting from name to number.)

ARGUMENTS

identifier

The identifier to be converted. Specify the identifier as an integer expression if you are converting an integer to a name. Specify the identifier as a character string expression if you are converting a name to an integer.

The F$IDENTIFIER function does not convert letters in the identifier to uppercase. Therefore, you must specify the identifier the same way it is defined in the "rights database". If the identifier is defined using uppercase letters, you must use uppercase letters when you specify the identifier for the F$IDENTIFIER function.

conversion-type

The type of conversion to be performed. Specify the conversion type as a character string expression that evaluates to one of the following strings: "NAME_TO_NUMBER" or "NUMBER_TO_NAME".

EXAMPLES

1
```
$ UIC_INT= F$IDENTIFIER("SLOANE","NAME_TO_NUMBER")
$ SHOW SYMBOL UIC_INT
  UIC_INT = 15728665   Hex = 00F00019  Octal = 00074000031
$ UIC = F$FAO("!%U",UIC_INT)
$ SHOW SYMBOL UIC
  UIC = [360,031]
```

This example uses the F$IDENTIFIER to convert the member identifier from the UIC [MANAGERS,SLOANE] to an integer. The F$IDENTIFIER function shows that the member identifier SLOANE is equivalent to the integer 15728665. Note that you must specify the identifier SLOANE using uppercase letters.

To convert this octal number to a standard numeric UIC, use the F$FAO function with the !%U directive. (This directive converts a longword to a UIC in named format.) In this example, the member identifier SLOANE is equivalent to the numeric UIC [360,031].

2
```
$ UIC_INT = (%031 + (%X10000 * %0360))
$ UIC_NAME = F$IDENTIFIER(UIC_INT,"NUMBER_TO_NAME")
$ SHOW SYMBOL UIC_NAME
  UIC_NAME = "ODONNELL"
```

This example obtains the alphanumeric identifier associated with the numeric UIC [360,031]. First, you must obtain the longword integer that corresponds to the UIC [360,031]. To do this, place the member number into the low order word, and place the group number into the high order word. Next, use the F$IDENTIFIER function to return the named identifier associated with the integer.

F$INTEGER

Returns the integer equivalent of the result of the specified expression.

FORMAT

F$INTEGER*(expression)*

return value

An integer value which is equivalent to the specified expression.

ARGUMENT

expression

The expression to be evaluated by the F$INTEGER expression. You can specify either an integer or a character string expression.

If you specify an integer expression, the F$INTEGER function evaluates the expression and returns the result. If you specify a string expression, the F$INTEGER function evaluates the expression, converts the resulting string to an integer, and returns the result.

After evaluating a string expression, the F$INTEGER function converts the result to an integer in the following way. If the resulting string contains characters which form a valid integer, the F$INTEGER function returns the integer value. If the string contains characters which do not form a valid integer, the F$INTEGER function returns the integer 1 if the string begins with T, t, Y, or y. The function returns the integer 0 if the string begins with any other character.

EXAMPLE

```
$ A = "23"
$ B = F$INTEGER("-9" + A)
$ SHOW SYMBOL B
  B = -923 Hex=FFFFFC65 Octal=176145
```

This example shows how to use the F$INTEGER function to equate a symbol to the integer value returned by the function.

The F$INTEGER function in the above example returns the integer equivalent of the string expression ("–9" + A). First, the F$INTEGER function evaluates the string expression by concatenating the string literal "–9" with the string literal "23". Note that the value of the symbol A is automatically substituted in a string expression. Also note that the plus sign (+) is a string concatenation operator since both arguments are string literals.

After the string expression is evaluated, the F$INTEGER function converts the resulting character string ("–923") to an integer, and returns the value –923. This integer value is assigned to the symbol B.

F$LENGTH

Returns the length of a specified string.

FORMAT

F$LENGTH*(string)*

return value

An integer value for the length of the string.

ARGUMENT

string

The character string of which you are determining the length. Specify the string argument as a character string expression.

EXAMPLE

```
$ MESSAGE = F$MESSAGE(%X1C)
$ SHOW SYMBOL MESSAGE
  MESSAGE = "%SYSTEM-F-EXQUOTA, exceeded quota"
$ STRING_LENGTH = F$LENGTH(MESSAGE)
$ SHOW SYMBOL STRING_LENGTH
  STRING_LENGTH = 33    Hex = 00000021  Octal = 000041
```

The first assignment statement uses the F$MESSAGE function to return the message which corresponds to the hexadecimal value 1C. The message is returned as a character string and is assigned to the symbol MESSAGE.

The F$LENGTH function is then used to return the length of the character string assigned to the symbol MESSAGE. You do not need to use quotation marks when you use the symbol MESSAGE as an argument for the F$LENGTH function. (Quotation marks are not used around symbols in character string expressions.)

The F$LENGTH function returns the length of the character string and assigns it to the symbol STRING_LENGTH. At the end of the example, the symbol STRING_LENGTH has a value equal to the number of characters in the value of the symbol named MESSAGE, that is, 33.

F$LOCATE

Locates a character or character substring within a string and returns its offset within the string. If the character or character substring is not found, the function returns the length of the string that was searched.

FORMAT

F$LOCATE*(substring,string)*

return value

An integer value representing the offset of the substring argument. An offset is the position of a character or a substring relative to the beginning of the string. The first character in a string is always offset position 0 from the beginning of the string (which always begins at the leftmost character).

If the substring argument is not found, the F$LOCATE function returns an offset of 1 more than the offset of the last character in the string. (This equals the length of the string.)

ARGUMENTS

substring

The string of characters that you want to locate within the string. Specify the substring as a character string expression.

string

The string in which the characters are to be found. Specify the string as a character string expression.

EXAMPLES

```
1   $ FILE_SPEC = "MYFILE.DAT;1"
    $ NAME_LENGTH = F$LOCATE(".",FILE_SPEC)
```

The F$LOCATE function in this example returns the position of the period in the string with respect to the beginning of the string. The period is in offset position 6, so the value 6 is assigned to the symbol NAME_LENGTH. Note that NAME_LENGTH also equals the length of the file name portion of the file specification MYFILE.DAT, that is, 6.

The substring argument, the period, is specified as a string literal and is therefore enclosed in quotation marks. The argument FILE_SPEC is a symbol, so it should not be placed within quotation marks. It is automatically replaced by its current value during the processing of the function.

```
2    $ INQUIRE TIME "Enter time"
     $ IF F$LOCATE(":",TIME) .EQ. F$LENGTH(TIME) THEN -
       GOTO NO_COLON
```

This section of a command procedure compares the results of the F$LOCATE and F$LENGTH functions to see if they are equal. This technique is commonly used to determine whether a character or substring is contained in a string.

In the example, the INQUIRE command prompts for a time value and assigns the user-supplied time to the symbol TIME. The IF command checks for the presence of a colon in the string entered in response to the prompt. If the value returned by the F$LOCATE function equals the value returned by the F$LENGTH function, the colon is not present. You use the .EQ. operator (rather than .EQS.) because the F$LOCATE and F$LENGTH functions return integer values.

Note that quotation marks are used around the substring argument, the colon, because it is a string literal. However, the symbol TIME does not require quotation marks because it is automatically evaluated as a string expression.

F$LOGICAL

Translates a logical name and returns the equivalence name string. The translation is not iterative; the equivalence string is not checked to determine whether it is a logical name.

FORMAT

F$LOGICAL*(logical-name)*

return value

A character string containing the equivalence name of a specified logical name. The F$LOGICAL function searches the process, job, group, and system logical name tables, in that order, and returns the equivalence name for the first match found.

The F$LOGICAL function returns a null string if no match is found.

ARGUMENT

logical-name

The logical name to be translated. You must specify the logical-name argument as a string expression. If you specify a logical name that identifies a search list, then the first name in the list is returned.

If you have defined a logical name to contain both uppercase and lowercase letters, you must type the name using the proper cases. If the logical name is defined using all uppercase or all lowercase letters, you do not need to specify the proper case.

DESCRIPTION

You can use the F$LOGICAL function in command procedures to save the current equivalence of a logical name and later restore it. You can also use it to test whether logical names have been assigned.

Note that the F$LOGICAL function is superseded by the F$TRNLNM function. In addition to being able to return an equivalence string, the F$TRNLNM function offers the full capabilities of the $TRNLNM system service.

EXAMPLES

1

```
$ SAVE_DIR = F$LOGICAL("SYS$DISK")+F$DIRECTORY()
    .
    .
    .
$ SET DEFAULT 'SAVE_DIR'
```

The assignment statement concatenates the values returned by the F$DIRECTORY and F$LOGICAL functions, and assigns the resulting string to the symbol SAVE_DIR. The symbol SAVE_DIR consists of a full device and directory name string.

The argument SYS$DISK is enclosed in quotation marks because it is a character string. (The command interpreter treats all arguments that begin with alphabetic characters as symbols or lexical functions, unless the arguments are enclosed within quotation marks.) None of the optional arguments are specified, so the F$LOGICAL function uses the defaults.

At the end of the command procedure, the original default directory is reset. When you reset the directory, you must place single quotes around the symbol SAVE_DIR to force symbol substitution.

```
2   $ IF F$LOGICAL("INFILE") .EQS. "" THEN GOTO ASSIGN
        .
        .
        .
    $ ASSIGN:
        .
        .
        .
```

This line from a command procedure tests whether the logical name INFILE has been assigned. If there is no current assignment, the F$LOGICAL function returns a null string and the command interpreter starts executing statements at the label ASSIGN.

```
3   $ DEFINE/TABLE=LNM$GROUP TERMINAL 'F$LOGICAL("SYS$COMMAND")'
```

This example shows a line from a command procedure that (1) uses the F$LOGICAL function to determine the name of the current terminal device and (2) creates a group logical name table entry based on the equivalence string.

You must place the argument SYS$COMMAND within quotation marks because it is a character string.

Also, in this example you must place the F$LOGICAL function within single quotes to force the lexical function to be evaluated. Otherwise, the DEFINE command will not automatically evaluate the lexical function.

F$MESSAGE

Returns a character string containing the message text associated with a specific system status code.

FORMAT

F$MESSAGE*(status-code)*

return value

A character string containing the system message which corresponds to the argument you specify.

Note that, although each message in the system message file has a numeric value or range of values associated with it, there are many possible numeric values that do not have corresponding messages. If you specify an argument which has no corresponding message, the F$MESSAGE function returns a string containing the NOMSG error message.

For more information on system error messages, see the *VAX/VMS System Messages and Recovery Procedures Reference Manual.*

ARGUMENT

status-code

The status code for which you are requesting error message text. You must specify the status code as an integer expression.

EXAMPLE

```
$ ERROR_TEXT = F$MESSAGE(%X1C)
$ SHOW SYMBOL ERROR_TEXT
  ERROR_TEXT = "%SYSTEM-F-EXQUOTA, exceeded quota"
```

This example shows how to use the F$MESSAGE function to determine the message associated with the status code %X1C. The F$MESSAGE function returns the message string, which is assigned to the symbol ERROR_TEXT.

F$MODE

Returns a character string showing the mode in which a process is executing. The F$MODE function has no arguments, but must be followed by parentheses.

FORMAT F$MODE()

return value

The string "INTERACTIVE" for interactive processes. If the process is noninteractive, the string "BATCH", "NETWORK" or "OTHER" is returned. Note that the return string always contains uppercase letters.

ARGUMENTS

None.

DESCRIPTION

The F$MODE function is useful in command procedures that must operate differently when executed interactively and noninteractively. You should include either the F$MODE function or the F$ENVIRONMENT function in your login command file to execute different commands for interactive terminal sessions and noninteractive sessions.

If you do not include the F$MODE function to test whether your login command file is being executed from an interactive process, and the login command file is executed from a noninteractive process (such as a batch job), the process may terminate if the login command file contains commands that are appropriate only for interactive processing.

A command procedure can use the F$MODE function to test whether the procedure is being executed during an interactive terminal session. It can direct the flow of execution according to the results of this test.

EXAMPLE

```
$ IF F$MODE() .NES. "INTERACTIVE" THEN GOTO NON_INT_DEF
$ INTDEF:         ! Commands for interactive terminal sessions
   .
   .
   .
$ EXIT
$ NON_INT_DEF:          !Commands for non-interactive processes
   .
   .
   .
```

This example shows the beginning of a login command file. The IF command compares the character string returned by F$MODE with the character string INTERACTIVE; if they are not equal, control branches to the label NON_INT_DEF. Otherwise, the statements following the label INTDEF are executed and the procedure exits before the statements at NON_INT_DEF. Therefore, this login.com file has two sets of initialization commands: one for interactive mode and one for noninteractive mode (including batch and network jobs).

F$PARSE

Invokes the $PARSE RMS service to parse a file specification and return either the expanded file specification or the particular file specification field that you request.

FORMAT

F$PARSE*(file-spec [,default-spec] [,related-spec] [,field] [,parse-type])*

return value

A character string containing the expanded file specification or the field you specify. If you do not provide a complete file specification for the file-spec argument, the F$PARSE function supplies defaults in the return string, as described in the Description section below.

If an error is detected during the parse, the F$PARSE function returns a null string, except when you specify a field name or the SYNTAX_ONLY parse type.

ARGUMENTS

file-spec

The file specification to be parsed. You must specify the file specification as a character string expression.

You can use wildcard characters in the file specification. If you use a wildcard character, the file specification returned by the F$PARSE function will contain the wildcard.

default-spec

A default file specification. You must specify the default file specification as a character string expression.

The fields in the default file specification are substituted in the output string if a particular field in the file-spec argument is missing. You can make further substitutions in the file-spec argument by using the related-spec argument.

related-spec

A related file specification. You must specify the related file specification as a character string expression.

The fields in the related file specification are substituted in the output string if a particular field is missing from both the file-spec and default-spec arguments.

field

The name of a field in a file specification. This argument causes F$PARSE to return a specific portion of a file specification. Specify the field argument as a character string expression.

You can specify one of the following field names:

NODE	Node name
DEVICE	Device name
DIRECTORY	Directory name
NAME	File name
TYPE	File type
VERSION	File version number

The field name cannot be abbreviated.

parse-type

The type of parsing to be performed. By default, the F$PARSE function verifies that the directory in the file specification exists on the device in the file specification. Note that the device and directory can be explicitly given in one of the arguments, or can be provided by default.

Also, by default the F$PARSE function translates logical names if they are provided in any of the arguments. The F$PARSE function stops iterative translation when it encounters a logical name with the CONCEALED attribute.

You can change how the F$PARSE function parses a file specification by using one of the following keywords:

NO_CONCEAL	Logical names are not concealed. Therefore, logical name translation does not end when a concealed logical name is encountered.
SYNTAX_ONLY	The syntax of the file specification is checked without verifying that the specified directory exists on the specified device.

DESCRIPTION

The F$PARSE function allows you to parse file specifications using the $PARSE RMS service. For more information on the $PARSE routine, see the *VAX Record Management Services Reference Manual.*

When you use the F$PARSE function, you can omit optional arguments that can be used to the right of the last argument you specify. However, you must include commas as place holders if you omit optional arguments to the left of the last argument you specify.

If you omit the device and directory names in the file-spec argument, the F$PARSE function supplies defaults, first from the default-spec argument and second from the related-spec argument. If names are not provided by these arguments, the F$PARSE function uses your current default disk and directory.

If you omit the file name, file type, or version number, the F$PARSE function supplies defaults, first from the default-spec argument and second from the related-spec argument. If names are not provided by these arguments, the F$PARSE function returns a null specification for these fields.

EXAMPLES

1
```
$ SET DEF DISK2:[FIRST]
$ SPEC = F$PARSE("JAMES.MAR","[ROOT]",,,"SYNTAX_ONLY")
$ SHOW SYMBOL SPEC
  SPEC = "DISK2:[ROOT]JAMES.MAR;"
```

In this example, the F$PARSE function returns the expanded file specification for the file JAMES.MAR. The example uses the SYNTAX_ONLY keyword to request that F$PARSE should check the syntax, but should not verify that the [ROOT] directory exists on DISK2.

The default device and directory are DISK2:[FIRST]. Since the directory name [ROOT] is specified as the default-spec argument in the assignment statement, it is used as the directory name in the output string. Note that the default device returned in the output string is DISK2 and the default version number for the file is null. You must place quotation marks around the arguments JAMES.MAR and ROOT because they are string literals.

If you had not specified syntax-only parsing, and [ROOT] were not on DISK2, a null string would have been returned.

2
```
$ SET DEFAULT DB1:[VARGO]
$ SPEC = F$PARSE("INFO.COM",,,"DIRECTORY")
$ SHOW SYMBOL SPEC
  SPEC = "[VARGO]"
```

In this example the F$PARSE function returns the directory name of the file INFO.COM. Note that, since the default-spec and related-spec are omitted from the argument list, commas (,) must be inserted in their place.

3
```
$ SPEC = F$PARSE("DENVER::DB1:[PROD]RUN.DAT",,,"TYPE")
$ SHOW SYMBOL SPEC
  SPEC = ".DAT"
```

In this example, the F$PARSE function is used to parse a file specification containing a node name. The F$PARSE function returns the file type DAT for the file RUN.DAT at the remote node DENVER.

F$PID

The F$PID function returns a process identification (PID) number, and updates the context symbol to point to the current position in the system's process list.

FORMAT

F$PID*(context-symbol)*

return value

A character string containing the process identification (PID) number of a process in the system's list of processes.

The PIDs returned by the F$PID function depend on the privilege of your process. If you have GROUP privilege, the F$PID function returns PIDs of processes in your group. If you have WORLD privilege, the F$PID function returns PIDs of all processes on the system. If you lack GROUP or WORLD privileges, the F$PID function returns only your process PID.

After the last PID in the system's process list is returned, the F$PID function returns a null string.

ARGUMENT

context-symbol

A symbol that DCL uses to store a pointer into the system's list of processes. The F$PID function uses this pointer to return a PID.

Specify the context-symbol by using a symbol. The first time you use the F$PID function in a command procedure, you should use a symbol that is either undefined or equated to the null string ("").

If the context-symbol is undefined or equated to a null string (""), the F$PID function returns the first PID in the system's process list that it has the privilege to access. That is, if you have GROUP privilege and if the context-symbol is null or undefined, the F$PID function returns the PID of the first process in your group. If you have WORLD privilege, the F$PID function returns the PID of the first process in the list. If you have neither GROUP nor WORLD privileges, the F$PID returns your process PID.

DESCRIPTION

You can use the F$PID function to obtain the PIDs of all processes in your group (if you have GROUP privilege) or on the system (if you have WORLD privilege.)

The first time you use the F$PID function, use a symbol that is either undefined or equated to the null ("") string. This causes the F$PID function to return the first PID in the system's process list that you have the privilege to access. It also causes the F$PID function to initialize the context-symbol.

Once the context-symbol is initialized, each subsequent F$PID function returns the next PID in sequence, and updates the context symbol. After the last PID in the process list is returned, the F$PID function returns a null string.

EXAMPLE

```
$ CONTEXT = ""
$ START:
$     PID = F$PID(CONTEXT)
$     IF PID .EQS. "" THEN EXIT
$     SHOW SYMBOL PID
$     GOTO START
```

This command procedure uses the F$PID function to display a list of PIDs. The assignment statement declares the symbol CONTEXT, which is used as the context-symbol argument for the F$PID function. Because CONTEXT is equated to a null string, the F$PID function will return the first PID in the process list that it has the privilege to access.

The PIDs displayed by this command procedure depend on the privilege of your process. When run with GROUP privilege, the PIDs of users in your group are displayed. When run with WORLD privilege, the PIDS of all users on the system are displayed. Without GROUP or WORLD privilege, only your PID is displayed.

F$PRIVILEGE

Returns a value of either "TRUE" or "FALSE", depending on whether your current process privileges match the privileges listed in the argument.

FORMAT

F$PRIVILEGE*(priv-states)*

return value

A character string containing the value "TRUE" or "FALSE". The F$PRIVILEGE function returns the string "FALSE" if any one of the privileges in the priv-states list is false.

ARGUMENTS

priv-states

A privilege or a list of privileges separated by commas. Specify the priv-states argument as a character string expression.

Table DCL–1-3 in the *VAX/VMS DCL Concepts Manual* lists the system privileges. You can specify any of these privileges except [NO]ALL.

DESCRIPTION

You can use the F$PRIVILEGE function to identify your current process privileges.

If "NO" precedes the privilege, then the privilege must be disabled in order for the function to return a value of "TRUE". The F$PRIVILEGE function checks each of the keywords in the specified list, and if the result for any one is false, the string "FALSE" is returned.

EXAMPLE

```
$ PROCPRIV = F$PRIVILEGE("OPER,GROUP,TMPMBX,NONETMBX")
$ SHOW SYMBOL PROCPRIV
  PROCPRIV = "FALSE"
```

The F$PRIVILEGE function is used to test whether the process has OPER, USER, TMPMBX, and NETMBX privileges.

The process in this example has OPER, GROUP, TMPMBX, and NETMBX privileges. Therefore, a value of "FALSE" is returned because the process has NETMBX privilege, but NONETMBX was specified in the priv-states list. Although the Boolean result for the other three keywords is true, the entire expression is declared false since the result for NONETMBX was false.

F$PROCESS

Obtains the current process name string. The F$PROCESS function has no arguments, but must be followed by parentheses.

FORMAT

F$PROCESS()

return value

A character string containing the current process name.

ARGUMENTS

None.

EXAMPLE

```
$ NAME = F$PROCESS()
$ SHOW SYMBOL NAME
  NAME = "MARTIN"
```

In this example, the F$PROCESS function returns the current process name and assigns it to the symbol NAME.

F$SEARCH

Invokes the $SEARCH RMS service to search a directory file and return the full file specification for a file you name.

FORMAT

F$SEARCH*(file-spec[,stream-id])*

return value

A character string containing the expanded file specification for the file-spec argument. If the F$SEARCH function does not find the file in the directory, the function returns a null ("") string.

ARGUMENTS

file-spec

The file specification to be searched for. You must specify the file-spec argument as a character string expression.

If you omit the device or directory names, the F$SEARCH function supplies defaults from your current default disk and directory. The F$SEARCH function does not supply defaults for a file name or type. If you omit the version number, the F$SEARCH function returns the specification for the file with the highest version number.

You can use wildcards in the file-spec argument.

stream-id

A search stream identification number. Specify the search stream identification number as a positive integer expression.

A search stream identification number is used to maintain separate search contexts when you use the F$SEARCH function more than once and when you supply different file-spec arguments. If you use the F$SEARCH function more than once in a command procedure and if you also use different file-spec arguments, specify stream-id arguments to identify each search separately.

If you omit the stream-id argument, the F$SEARCH function assumes an implicit single search stream. That is, the F$SEARCH function will start searching at the beginning of the directory file each time you specify a different file-spec argument.

DESCRIPTION

The F$SEARCH function allows you to search for files in a directory using the $SEARCH RMS service. For more information on the $SEARCH routine, see the *VAX Record Management Services Reference Manual*.

You can use the F$SEARCH function in a loop in a command procedure to return file specifications for all files which match a file-spec argument containing a wildcard. Each time the F$SEARCH function is executed, it returns the next file specification that matches the wildcarded file specification. After the last file specification is returned, the next F$SEARCH function returns a null string. When you use the F$SEARCH function in a loop, you must include a wildcard in the file-spec argument. Otherwise, the F$SEARCH will always return the same file specification.

Note that you must maintain the context of the search stream explicitly (by stating a stream-id) or implicitly (by omitting the stream-id and using the same file-spec argument each time you execute the F$SEARCH function).

EXAMPLES

1
```
$ START:
$     FILE = F$SEARCH("SYS$SYSTEM:*.EXE")
$     IF FILE .EQS. "" THEN EXIT
$     SHOW SYMBOL FILE
$     GOTO START
```

This command procedure displays the file-specs of the latest version of all .EXE files in the SYS$SYSTEM directory. (Only the latest version is returned because a wildcard is not used as the version number.) The file-spec argument SYS$SYSTEM:*.EXE is surrounded by quotation marks because it is a character string expression.

Because no stream-id argument is specified, the F$SEARCH uses a single search stream. Each subsequent F$SEARCH function uses the same file-spec argument to return the next file specification of an .EXE file from SYS$SYSTEM:. After the latest version of each .EXE file has been displayed, the F$SEARCH function returns a null string and the procedure exits.

2
```
$ START:
$    COM = F$SEARCH ("*.COM;*",1)
$    DAT = F$SEARCH ("*.DAT;*",2)
$    SHOW SYMBOL COM
$    SHOW SYMBOL DAT
$    IF (COM.EQS. "") .AND. (DAT.EQS. "") THEN EXIT
$    GOTO START
```

This command procedure searches the default disk and directory for both COM and DAT files. Notice that the stream-id is specified for each F$SEARCH function so that the context for each search is maintained.

The first F$SEARCH function starts searching from the top of the directory file for a file with a type of COM. When it finds a COM file, a pointer is set to maintain the search context. When the F$SEARCH function is used the second time, it again starts searching from the top of the directory file for a file with a type of DAT. When the procedure loops back to the label START, the stream-id argument allows each F$SEARCH function to start searching in the correct place in the directory file. After all versions of COM and DAT files are returned, the procedure exits.

3
```
$ FILESPEC = F$SEARCH("TRNTO""SMITH SALLY""::DBA1:[PROD]*.DAT")
$ SHOW SYMBOL FILESPEC
  FILESPEC = "TRNTO"smith password"::DBA1:[PROD]CARS.DAT"
```

This example uses the F$SEARCH function to return a file specification for a file at a remote node. The access control string is enclosed in double quotation marks because it is part of a character string expression when it is an argument for the F$SEARCH function; to include quotation marks in a character string expression you must specify two sets of quotation marks.

Note that, when the F$SEARCH function returns a node name containing an access control string, it substitutes the word "password" for the actual user password.

F$SETPRV

Returns a list of keywords indicating user privileges. In addition, the F$SETPRV function invokes the $SETPRV system service to enable or disable specified user privileges. The return string shows the status of user privileges before changes are made with the F$SETPRV function.

In order to enable or disable privileges, your process must be authorized to set the specified privilege. For detailed information on privilege restrictions, see the description of the $SETPRV system service in the *VAX/VMS System Services Reference Manual.*

FORMAT

F$SETPRV*(priv-states)*

return value

A character string containing keywords for the current process privileges before they were changed by the F$SETPRV function.

ARGUMENT

priv-states

A privilege or a list of privileges separated by commas. Specify the priv-states argument as a character string expression.

Table DCL–1-3 in the *VAX/VMS DCL Concepts Manual* lists the system privileges you can specify in the priv-states argument.

DESCRIPTION

The F$SETPRV function returns keywords for your current privileges, whether or not you are authorized to change the privileges listed in the priv-states argument. The F$SETPRV function will enable or disable only the privileges you are authorized to change.

EXAMPLES

1
```
$ OLDPRIV = F$SETPRV("OPER,NOTMPMBX")
$ SHOW SYMBOL OLDPRIV
  OLDPRIV = "NOOPER,TMPMBX"
```

In this example, the process is authorized to change the OPER and TMPMBX privileges. The F$SETPRV function enables the OPER privilege and disables the TMPMBX privilege. In addition, the F$SETPRV function returns the keywords NOOPER and TMPMBX, showing the state of these privileges before they were changed.

You must place quotation marks around the list of privilege keywords because it is a string literal.

```
2    $ SAVPRIV = F$SETPRV("NOGROUP")
     $ SHOW SYMBOL SAVPRIV
       SAVPRIV = "GROUP"
     $ TEST = F$PRIVILEGE("GROUP")
     $ SHOW SYMBOL TEST
       TEST = "TRUE"
```

In this example, the process is not authorized to change the GROUP privilege. However, the F$SETPRV function still returns the current setting for the GROUP privilege.

The F$PRIVILEGE function is used to see whether the process has GROUP privilege. The return string, TRUE, indicates that the process has GROUP privilege, even though the F$SETPRV function attempted to disable the privilege.

F$STRING

Returns the string equivalent of the result of the specified expression.

FORMAT

F$STRING*(expression)*

return value

A character string which is equivalent to the specified expression.

ARGUMENT

expression

The expression to be evaluated by the F$STRING expression. You can specify either an integer or a character string expression.

If you specify an integer expression, the F$STRING expression evaluates the expression, converts the resulting integer to a string, and returns the result. If you specify a string expression, the F$STRING expression evaluates the expression and returns the result.

When converting an integer to a string, the F$STRING function uses decimal representation. Also, leading zeroes are dropped. When converting a negative number, the F$STRING places a minus sign at the beginning string representation of the number.

EXAMPLE

```
$ A = 5
$ B = F$STRING(-2 + A)
$ SHOW SYMBOL B
  B = "3"
```

The F$STRING function in the above example converts the result of the integer expression (–2 + A) to the numeric string, "3". First, the F$STRING function evaluates the expression (–2 + A). Note that 5, the value of symbol A, is automatically substituted when the integer expression is evaluated.

After the integer expression is evaluated, the F$STRING function converts the resulting integer, 3, to the string "3". This string is assigned to the symbol B.

F$TIME

Returns the current date and time string. The F$TIME function has no arguments, but must be followed by parentheses.

FORMAT F$TIME()

return value

A character string containing the current date and time. The returned string has the following fixed, 23-character format:

```
dd-mmm-yyyy hh:mm:ss.cc
```

When the current day of the month is any of the values 1 through 9, the first character in the returned string is a blank character. Thus, the time portion of the string is always in character position 13, at an offset of 12 characters from the beginning of the string.

Note that you must use the assignment operator (=) to preserve the blank character in the returned string. If you use the string assignment operator (:=), the leading blank is dropped.

ARGUMENTS

None.

EXAMPLE

```
$ OPEN/WRITE OUTFILE DATA.DAT
$ TIME_STAMP = F$TIME()
$ WRITE OUTFILE TIME_STAMP
```

This example shows how to use the F$TIME function to time-stamp a file that you create from a command procedure. OUTFILE is the logical name for the file DATA.DAT, which is opened for writing. The F$TIME function returns the current date and time string, and assigns this string to the symbol TIME_STAMP. The WRITE command writes the date and time string to OUTFILE.

F$TRNLNM

Translates a logical name and returns the equivalence name string, or the requested attributes of the logical name. The translation is not iterative; the equivalence string is not checked to determine whether it is a logical name.

FORMAT

F$TRNLNM*(logical-name [,table] [,index] [,mode] [,case] [,item])*

return value

The equivalence name or attribute of the specified logical name. The return value can be a character string or an integer, depending on the arguments you specify with the F$TRNLNM function.

The F$TRNLNM function returns a null string if no match is found.

ARGUMENTS

logical-name

The logical name to be translated. You must specify the logical-name argument as a string expression.

table

The logical name table or tables that the F$TRNLNM function should search to translate the logical name. The table argument must be a logical name that translates to a logical name table, or to a list of table names. Specify the table argument as a string expression.

If you do not specify a table, the default value is LNM$DCL_LOGICAL. That is, the F$TRNLNM function searches the tables whose names are equated to the logical name LNM$DCL_LOGICAL. Unless LNM$DCL_LOGICAL has been redefined for your process, the F$TRNLNM function searches the process, job, group, and system logical name tables, in that order, and returns the equivalence name for the first match found.

index

The index of the equivalence name to be returned. The index identifies a particular equivalence string when a logical name has more than one equivalence. The index refers to the equivalence strings in the order the names were listed when the logical name was defined.

Specify the index as an integer expression. The index begins with 0; that is, the first name in a list of equivalence names is referenced by the index 0.

If you do not specify an index, the default is 0.

mode

The access mode for the translation. Specify the access mode as a character string expression that evaluates to one of the following strings: USER, SUPERVISOR, EXECUTIVE, or KERNEL.

The F$TRNLNM function starts by searching for a logical name created with the access mode specified in the mode argument. If it does not find a match, the F$TRNLNM function searches for the name created with each inner access mode, and returns the first match found. For example, two logical names can have the same name, but one name can be created with user access mode and the other name with executive access mode. If the mode argument is USER, the F$TRNLNM function will return the equivalence string for the user mode, not the executive mode, logical name.

If you do not specify an access mode, the default mode is USER.

case

The type of case translation to be performed. Specify the case argument as a character string that evaluates to one of the following strings: CASE_BLIND or CASE_SENSITIVE.

If the translation is case blind, then the F$TRNLNM function first searches for a logical name with characters of the same case as the name argument. If no match is found, then the F$TRNLNM function upcases the name argument and the logical names it is searching, and searches again for a match. The result of the first successful translation is returned.

If the translation is case sensitive, the F$TRNLNM function searches only for a logical name with characters of the same case as the name argument. The F$TRNLNM returns a null string if no exact match is found.

If you do not specify the case argument, the default is CASE_BLIND.

item

The type of information that F$TRNLNM should return about the specified logical name. Specify the item argument as a character string expression that evaluates to one of the following strings. If you do not specify an item, the default is VALUE.

ACCESS_MODE	Returns the access mode associated with the logical name. The access mode is returned as one of the following character strings: "USER", "SUPERVISOR", "EXECUTIVE", "KERNEL".
CONCEALED	Returns one of the character strings "TRUE" or "FALSE" to indicate whether the CONCEALED attribute was specified with the /TRANSLATION_ATTRIBUTES qualifier when the logical name was created. The CONCEALED attribute is used to create a concealed logical name.
CONFINE	Returns one of the character strings "TRUE" or "FALSE" to indicate whether the logical name is confined. If the logical name is confined (TRUE), then the name will not be copied to subprocesses. If the logical name is not confined (FALSE), then the name will be copied to subprocesses.
CRELOG	Returns one of the character strings "TRUE" or "FALSE" to indicate whether the logical name was created with the $CRELOG system service or with the $CRELNM system service, using the CRELOG attribute. If the logical name was created with the $CRELOG system service or with the $CRELNM system service, using the CRELOG attribute, then the *string "TRUE" is returned.* Otherwise, the string "FALSE" is returned.

LENGTH	Returns the length of the equivalence name associated with the specified logical name. If the logical name has more than one equivalence name, the F$TRNLNM function returns the length of the name specified by the index argument. The length is returned as an integer.
MAX_INDEX	Returns the largest index defined for the logical name. The index shows how many equivalence names are associated with a logical name. The index is zero based; that is, the index 0 refers to the first name in a list of equivalence names. The index is returned as an integer.
NO_ALIAS	Returns one of the character strings "TRUE" or "FALSE" to indicate whether the logical name has the NO_ALIAS attribute. The NO_ALIAS attribute means that a logical name with the same name cannot be created in an outer access mode.
TABLE	Returns one of the character strings "TRUE" or "FALSE" to indicate whether the logical name is the name of a logical name table.
TABLE_NAME	Returns the name of the table where the logical name was found. The table name is returned as a character string.
TERMINAL	Returns one of the character strings "TRUE" or "FALSE" to indicate whether the TERMINAL attribute was specified with the /TRANSLATION_ATTRIBUTES qualifier when the logical name was created. The TERMINAL attribute indicates that the logical name is not a candidate for iterative translation.
VALUE	Returns the equivalence name associated with the specified logical name. If the logical name has more than one equivalence name, the F$TRNLNM function returns the name specified by the index argument. The equivalence name is returned as a character string. VALUE is the default if you do not specify an item argument.

DESCRIPTION

The F$TRNLNM function offers the capability of the $TRNLNM system service. When you use the F$TRNLNM function, you can omit optional arguments that can be used to the right of the last argument you specify. However, you must include commas as place holders if you omit optional arguments to the left of the last argument that you specify.

You can use the F$TRNLNM function in command procedures to save the current equivalence of a logical name and later restore it. You can also use it to test whether logical names have been assigned.

EXAMPLES

```
$ SAVE_DIR = F$TRNLNM("SYS$DISK")+F$DIRECTORY()
    .
    .
    .
$ SET DEFAULT 'SAVE_DIR'
```

The assignment statement concatenates the values returned by the F$DIRECTORY and F$TRNLNM functions, and assigns the resulting string to the symbol SAVE_DIR. The symbol SAVE_DIR consists of a full device and directory name string.

The argument SYS$DISK is enclosed in quotation marks because it is a character string. (The command interpreter treats all arguments that begin with alphabetic characters as symbols or lexical functions, unless the arguments are enclosed within quotation marks.) None of the optional arguments are specified, so the F$TRNLNM function uses the defaults.

At the end of the command procedure, the original default directory is reset. When you reset the directory, you must place apostrophes around the symbol SAVE_DIR to force symbol substitution.

```
2   $ DEFINE/TABLE=LNM$GROUP TERMINAL 'F$TRNLNM("SYS$OUTPUT")'
```

This example shows a line from a command procedure that (1) uses the F$TRNLNM function to determine the name of the current output device and (2) creates a group logical name table entry based on the equivalence string.

You must place the argument SYS$OUTPUT within quotation marks because it is a character string.

Also, in this example you must place the F$TRNLNM function within single quotes to force the lexical function to be evaluated. Otherwise, the DEFINE command will not automatically evaluate the lexical function.

```
3   $ RESULT = F$TRNLNM("INFILE","LNM$PROCESS",0,"SUPERVISOR",,"NO_ALIAS")
    $ SHOW SYMBOL RESULT
      RESULT = "FALSE"
```

In this example, the F$TRNLNM function searches the process logical name table for the logical name INFILE. The function starts the search by looking for the logical name INFILE created in supervisor mode. If no match is found, the function looks for INFILE created in executive mode.

When a match is found, the F$TRNLNM function determines whether the name INFILE was created with the NO_ALIAS attribute. In this case, the "NO_ALIAS" attribute is not specified.

F$TYPE

Determines the data type of a symbol.

FORMAT

F$TYPE*(symbol-name)*

return value

The string "INTEGER" if the symbol is equated to an integer, or if the symbol is equated to a string in which the characters form a valid integer. The F$TYPE function returns the string "STRING" if the symbol is equated to a character string in which the characters do not form a valid integer. A null string is returned if the symbol is undefined.

ARGUMENT

symbol

The name of symbol that is evaluated.

EXAMPLES

1
```
$ NUM = "52"
$ TYPE = F$TYPE(NUM)
$ SHOW SYMBOL TYPE
  TYPE = "INTEGER"
```

This example uses the F$TYPE function to determine the data type of the symbol NUM. NUM is equated to the character string "52". Because the characters in the string form a valid integer, the F$TYPE function returns the string "INTEGER".

2
```
$ NUM = 52
$ TYPE = F$TYPE(NUM)
$ SHOW SYMBOL TYPE
  TYPE = "INTEGER"
```

In this example, the symbol NUM is equated to the integer 52. The F$TYPE function shows that the symbol has an integer data type.

3
```
$ CHAR = "FIVE"
$ TYPE = F$TYPE(CHAR)
$ SHOW SYMBOL TYPE
  TYPE = "STRING"
```

In this example, the symbol CHAR is equated to the character string "FIVE". Because the characters in this string do not form a valid integer, the F$TYPE function shows that the symbol has a string value.

F$USER

Returns the current user identification code (UIC), in named format. The F$USER function has no arguments, but must be followed by parentheses.

FORMAT

F$USER()

return value

A character string containing the current user identification (UIC), including square brackets. The UIC is returned in the format [group-identifier, member-identifier].

ARGUMENTS

None.

EXAMPLE

```
$ UIC = F$USER()
$ SHOW SYMBOL UIC
  UIC = "[GROUP6,JENNIFER]"
```

In this example the F$USER function returns the current user identification code and assigns it to the symbol UIC.

F$VERIFY

Returns an integer value indicating whether the procedure verification setting is currently on or off. If used with arguments, the F$VERIFY function can turn the procedure and image verification settings on or off. You must include the parentheses after the F$VERIFY function, whether or not you specify arguments.

FORMAT

F$VERIFY *([procedure-value] [,image-value])*

return value

The integer 0 if the procedure verification setting is off, or the integer 1 if the procedure verification setting is on.

ARGUMENTS

procedure-value

An integer expression with a value of 0 to turn procedure verification off, or 1 to turn procedure verification on.

When procedure verification is on, each DCL command line in the command procedure is displayed on the output device. Procedure verification allows you to verify that each command is executing correctly.

If you use the procedure-value argument, the function first returns the current procedure verification setting. Then the command interpreter turns the procedure verification on or off, as specified by the argument.

image-value

An integer expression with a value of 0 to turn image verification off, or 1 to turn image verification on.

When image verification is on, data lines in the command procedure are displayed on the output device.

DESCRIPTION

When you use the F$VERIFY function, you can specify zero, one, or two arguments. If you do not specify any arguments, neither of the verification settings are changed. If you specify only the procedure-value argument, both procedure and image verification are set (if the value is 1) or cleared (if the value is 0).

If you specify both arguments, procedure and image verification are set or cleared independently. If you specify the image-value argument alone, only image verification is set or cleared. However, if you specify the image-value argument alone, you must precede the argument with a comma.

You can use the F$VERIFY function in command procedures to test the current procedure verification setting. For example, a command procedure can save the current procedure verification setting before changing it and then later can restore the setting. In addition, you can construct a procedure that will not display (or print) commands, regardless of what the initial state of verification is.

You can also use the F$ENVIRONMENT function with VERIFY_PROCEDURE or VERIFY_IMAGE as the argument. With the F$ENVIRONMENT function, you can determine either the procedure or image verification setting; the F$VERIFY function determines only the procedure verification setting.

EXAMPLES

1
```
$ SAVE_PROC_VERIFY = F$ENVIRONMENT("VERIFY_PROCEDURE")
$ SAVE_IMAGE_VERIFY = F$ENVIRONMENT("VERIFY_IMAGE")
$ SET NOVERIFY
    .
    .
    .
$ TEMP = F$VERIFY(SAVE_PROC_VERIFY, SAVE_IMAGE_VERIFY)
```

This example shows an excerpt from a command procedure. The first assignment statement assigns the current procedure verification setting to the symbol SAVE_PROC_VERIFY. The second assignment statement assigns the current image verification setting to the symbol SAVE_IMAGE_VERIFY.

Then, the SET NOVERIFY command disables procedure and image verification. Later, the F$VERIFY function resets the verification settings, using the original values (equated to the symbols SAVE_PROC_VERIFY and SAVE_IMAGE_VERIFY). The symbol TEMP contains the procedure verification before it is changed with the F$VERIFY function. (In this example the value of TEMP is not used.)

2
```
$ VERIFY = F$VERIFY(0)
    .
    .
    .
$ IF VERIFY .EQ. 1 THEN SET VERIFY
```

This example shows an excerpt from a command procedure that uses the F$VERIFY function to save the current procedure verification setting and to turn both procedure and image verification off. At the end of the command procedure, if procedure verification was originally on, both the procedure and image verification are turned on.

LIBRARY

Invokes the Librarian Utility to create, modify, or describe an object, macro, help, text, or shareable image library. For a complete description of the Librarian Utility, including information about the LIBRARY command and its qualifiers, see the *VAX/VMS Librarian Reference Manual*.

FORMAT

LIBRARY *library-file-spec [input-file-spec[,...]]*

LINK

Invokes the VAX/VMS Linker to link one or more object modules into a program image, and defines execution characteristics of the image. For a complete description of the linker, including more information about the LINK command, see the *VAX/VMS Linker Reference Manual*.

FORMAT **LINK** *file-spec[,...]*

LOGIN Procedure

Initiates an interactive terminal session.

FORMAT

CTRL/C

CTRL/Y

<RETURN>

restrictions *None.*

PARAMETERS

None.

DESCRIPTION

There is no LOGIN command. You signal your intention to access the system by pressing CTRL/C, CTRL/Y, or <RETURN> on a terminal not currently in use. The system prompts for your user name and your password (and your secondary password, if you have one), and then validates them.

Specify the optional qualifiers immediately after you type your user name; then press RETURN to get the Password: prompts.

The login procedure performs the following functions:

- Validates your right to access the system by checking your user name and passwords against the entries in the system's user authorization file
- Establishes the default characteristics of your terminal session based on your user name entry in the authorization file
- Executes the command procedure file SYS$SYLOGIN.COM if one exists
- Executes either the command procedure file named LOGIN.COM if one exists in your default directory, or the command file defined in the user authorization file, if any

Some systems are set up with a retry facility for users who are accessing the system from remote or dialup locations. With these systems, when you make a mistake typing your user name or password, the system allows you to reenter the information without shutting you off. If you need to reenter your login information, press RETURN to have the system display the Username prompt again. Now retype your user name and press RETURN to send the information to the system and bring up the Password prompt. (There is both a limit to the number of times you can retry to enter your login information and a time limit between tries.)

QUALIFIERS

/CLI=command-language-interpreter

Specifies the name of an alternate command language interpreter (CLI) to override the default CLI listed in the user authorization file. The CLI you specify must be located in SYS$SYSTEM and have the file type EXE.

Note that if you do not specify a command interpreter via the /CLI qualifier, and do not have a default CLI listed in the user authorization file, the system will supply a default of /CLI=DCL.

/COMMAND[=file-spec]
/NOCOMMAND

Controls whether to execute your default login command procedure when you log in. Use the /COMMAND qualifier to specify the name of an alternate login command procedure. If you specify a file name without a file type, the default file type COM is used. If you specify /COMMAND and omit the file specification, your default login command procedure is executed. By default, /COMMAND is assumed.

Use the /NOCOMMAND qualifier if you do not want your default login command procedure to be executed.

/DISK=device-name[:]

Specifies the name of a disk device to be associated with the logical device SYS$DISK for the terminal session. This specification overrides the default SYS$DISK device established in the authorization file.

/TABLES=(command-table[,...])

Specifies the name of an alternate CLI table to override the default listed in the user authorization file (UAF). This table name is considered a file specification. The default device and directory is SYS$SHARE. The default file type is EXE.

If a logical name is used, the table name specification must be defined in the system logical name table.

If the /CLI qualifier is set to DCL or MCR, the /TABLES qualifier defaults to the correct value. If the /TABLES qualifier is specified without the /CLI qualifier, the CLI specified in the user's UAF will be used.

The default is /TABLES=DCLTABLES.

EXAMPLES

1

```
CTRL/Y
Username:  SMITHSON
Password:
```

CTRL/Y accesses the operating system, which immediately prompts for a user name. After validating the user name, the system prompts for the password but does not echo it.

2
```
<RETURN>
Username:  HIGGINS/DISK=USER$
Password:
       Welcome to VAX/VMS Version 4.00
       Last interactive login at 16-AUG-1985 09:16:47.08
       Last non-interactive login at 15-AUG-1985 17:32:34.27
$ SHOW DEFAULT
  USER$:[HIGGINS]
```

The /DISK qualifier requests that the default disk for the terminal session be DISK2. The SHOW DEFAULT command shows that USER$ is the default disk.

3
```
CTRL/C
Username: LIZA/CLI=MCR/COMMAND=ALTLOGIN.COM
Password:
       Welcome to VAX/VMS Version 4.00
       Last interactive login at 16-AUG-1985 09:16:47.08
       Last non-interactive login at 15-AUG-1985 17:32:34.27
>
```

The /CLI qualifier requests the alternate MCR command interpreter. The /COMMAND qualifier indicates that the login command file ALTLOGIN.COM is to be executed instead of the default login command file.

The right angle bracket indicates that MCR is active and expects an MCR command.

4
```
<RETURN>
Username: XENAKIS
Password:
Password:
       Welcome to VAX/VMS Version 4.00
       Last interactive login at 16-AUG-1985 09:16:47.08
       Last non-interactive login at 15-AUG-1985 17:32:34.27
$
```

The second Password: prompt indicates that the user has a secondary password, which must be entered to access the system.

5
```
<RETURN>
Username: JONES
Password:
User authorization failure
<RETURN>
Username: JONES
Password:
        Welcome to VAX/VMS Version 4.00
        Last interactive login at 16-AUG-1985 09:16:47.08
        Last non-interactive login at 15-AUG-1985 17:32:34.27
        1 failure since last successful login.
$
```

The "User authorization failure" message indicates that the password has been incorrectly entered. After successfully logging in, a message is displayed showing the number of login failures since your last successful login. This message is displayed only if one or more login failures have occurred.

```
6    <RETURN>
     Username: JOYCE
     Password:
           Welcome to VAX/VMS Version 4.00
           Last interactive login at 16-AUG-1985 09:16:47.08
           Last non-interactive login at 15-AUG-1985 17:32:34.27
           WARNING - Primary password has expired; update immediately.
     $
```

The WARNING message indicates that your primary password has expired. You must use the SET PASSWORD command to change your password before logging out, or you will be unable to log in.

For more information on changing your password, see the description of the SET PASSWORD command in this manual.

LOGOUT

Terminates an interactive terminal session.

FORMAT

LOGOUT

restrictions *None.*

PARAMETERS

None.

DESCRIPTION

You must use the LOGOUT command to end a terminal session. Under most circumstances, if you turn the power off at your terminal or hang up your telephone connection without using the LOGOUT command, you will remain logged in.

When you use the SET HOST command to log in to a remote processor, you generally need to use the LOGOUT command to end the remote session.

QUALIFIERS

/BRIEF

Requests the brief form of the logout message. The command interpreter displays your user name and the date and time when you logged out. The default for an interactive session is /BRIEF.

/FULL

Requests the long form of the logout message. When you specify /FULL, the command interpreter displays a summary of accounting information for the terminal session. The default for a batch job is /FULL.

/HANGUP
/NOHANGUP

For dialup terminals, determines whether or not the phone will hang up whenever you log out. By default, the /HANGUP setting of your terminal port determines whether the line is disconnected. Your system manager determines whether you are permitted to use this qualifier. For further information, see the *Guide to VAX/VMS System Security*.

EXAMPLES

1
```
$ LOGOUT
  HIGGINS     logged out at 15-APR-1985 17:48:56.73
```

The LOGOUT command uses the default /BRIEF format. No accounting information is displayed.

2

```
$ LOGOUT/FULL
  HIGGINS    logged out at 15-APR-1985 14:23:45.30
Accounting information:
Buffered I/O count:         22     Peak working set size:     90
Direct I/O count:           10     Peak virtual size:         69
Page faults:                68     Mounted volumes:            0
Charged CPU time: 0 00:01:30.50  Elapsed time:      0 04:59:02.63
```

The LOGOUT command with the /FULL qualifier displays a summary of accounting statistics for the terminal session.

MACRO

Invokes the VAX MACRO assembler to assemble one or more assembly language source programs. This description provides a functional overview of the MACRO command, emphasizing DCL syntax and grammar. For a complete functional description of the VAX MACRO assembler directives, see the *VAX MACRO and Instruction Set Reference Volume*.

FORMAT

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

restrictions

See qualifier descriptions.

PARAMETER

file-spec[,...]

Specifies one or more VAX MACRO assembly language source files to be assembled. If you do not specify a file type for an input file, the assembler uses the default file type of MAR.

You can specify more than one input file. If you separate the file specifications with commas, each file is assembled separately. If you separate the file specifications with plus signs, the files are concatenated and assembled as a single input file, producing single object and listing files. If a file already exists with the same name as the object or listing file produced by the command, a new file with a version number one higher than the previously existing file will be created.

No wildcard characters are allowed in the file specifications.

COMMAND QUALIFIERS

/CROSS_REFERENCE[=(function[,...])]
/NOCROSS_REFERENCE (default)

Controls whether a cross-reference listing is included in the listing file. A cross-reference listing is a list of the places in the source file where the item specified is defined or referred to. The /CROSS_REFERENCE qualifier includes a cross-reference listing, and therefore requires that you specify the /LIST qualifier to produce a listing file. The /NOCROSS_REFERENCE qualifier excludes the cross-reference listing.

You can specify one or more of the functions listed below. If you specify /CROSS_REFERENCE without any functions, it is equivalent to /CROSS_REFERENCE=(MACROS,SYMBOLS).

ALL	Cross-references directives, macros, operation codes, registers, and symbols
DIRECTIVES	Cross-references directives
MACROS	Cross-references macros
OPCODES	Cross-references operation codes
REGISTERS	Cross-references registers
SYMBOLS	Cross-references symbols

If you specify more than one function, separate them by commas and enclose the list in parentheses.

/DEBUG[=option]
/NODEBUG (default)

Controls whether local symbols are included in the symbol table, and whether traceback information is included in the object module.

You can use these qualifiers in place of the /ENABLE and /DISABLE qualifiers, when the DEBUG and TRACEBACK functions are used. The /DEBUG or /NODEBUG qualifiers will override debugging characteristics set with the .ENABLE or .DISABLE assembler directives.

You can specify one or more of the functions listed below. If you specify /DEBUG without any functions, it is equivalent to /DEBUG=ALL.

ALL	All local symbols are included in the symbol table, and all traceback information is provided in the object module for the debugger. Specifying this qualifier is equivalent to specifying the qualifier /ENABLE=(DEBUG,TRACEBACK).
NONE	No local symbols and no traceback information are made available to the debugger in the object module. Specifying this qualifier is equivalent to specifying the qualifier /DISABLE=(DEBUG,TRACEBACK).
SYMBOLS	All local symbols are made available to the debugger in the object module. No traceback information is made available to the debugger. Specifying this qualifier is equivalent to specifying the qualifiers /ENABLE=DEBUG and /DISABLE=TRACEBACK.
TRACEBACK	Traceback information is made available to the debugger in the object module. No local symbols are made available to the debugger. Specifying this qualifier is equivalent to specifying the qualifiers /ENABLE=TRACEBACK and /DISABLE=DEBUG.

/DISABLE=(function[,...])
/NODISABLE

Provides initial settings for the functions controlled by the assembler directive .DISABLE. See /ENABLE for a list of these functions.

The default is /DISABLE=(ABSOLUTE,DEBUG,TRUNCATION).

If you specify more than one function, separate them by commas and enclose the list in parentheses.

The /NODISABLE qualifier has the same effect as not specifying the /DISABLE qualifier, or negates the effects of any /DISABLE qualifiers specified earlier on the command line.

/ENABLE=(function[,...])
/NOENABLE

Provides initial settings for the functions controlled by the assembler directive .ENABLE. You must specify at least one of the functions listed below. You can enable or disable:

ABSOLUTE	Assembly of relative addresses as absolute addresses
DEBUG	Inclusion of local symbol table information in the object file for use with the debugger
TRUNCATION	Truncation of floating-point numbers (if truncation is disabled, numbers are rounded)
GLOBAL	Assumption that undefined symbols in the assembly are external symbols
SUPPRESSION	Suppression of the listing of unreferenced symbols in the symbol table
TRACEBACK	Provision of information to the debugger traceback mechanism

The default is /ENABLE=(GLOBAL,TRACEBACK,SUPPRESSION).

If you specify more than one function, separate them by commas and enclose the list in parentheses.

The /NOENABLE qualifier has the same effect as not specifying the /ENABLE qualifier, or negates the effects of any /ENABLE qualifiers specified earlier on the command line.

/LIBRARY
/NOLIBRARY

Positional qualifier.

The /LIBRARY qualifier cannot be used with the /UPDATE qualifier.

The /LIBRARY qualifier requires that the associated input file be a macro library. If you do not specify a file type, the assembler uses the default file type of MLB. The maximum number of libraries that can be searched in any given assembly is 16, one of which is always STARLET.MLB. If you specify more than one macro library in an assembly, the libraries are searched in reverse order of their specification.

Note that the restriction on the number of libraries that can be searched applies to a particular assembly, not necessarily to a particular MACRO command. If you issue the MACRO command so that more than one source file is assembled, but the source files are assembled **separately**, you are allowed to specify the maximum number of macro libraries for each separate assembly.

When a macro call is issued in a source program, any libraries specified with the .LIBRARY directive are searched first, in the reverse order in which they were declared; if the macro definition is not found in any of the libraries specified with the .LIBRARY directive, any libraries specified in the DCL MACRO command line are then searched (again, in the reverse order in which they were specified); finally, if the macro definition is not found in the libraries specified in the command line, STARLET.MLB is searched.

The /NOLIBRARY qualifier has the same effect as not specifying the /LIBRARY qualifier, or negates the effects of any /LIBRARY qualifiers specified earlier on the command line.

/LIST[=file-spec]
/NOLIST

Controls whether an output listing is created, and optionally provides an output file specification for the listing file.

If you issue the MACRO command interactively, the assembler, by default, does not create a listing file. When /NOLIST is present, either explicitly or by default, errors are reported on the current output device.

If you execute the MACRO command in a batch job, /LIST is the default. When you specify /LIST, you can control the defaults applied to the output file specification by the placement of the qualifier in the command. See Section 2.4 of the *VAX/VMS DCL Concepts Manual* for more information on entering output file qualifiers.

The default file type provided for listing files is LIS.

No wildcard characters are allowed in the file specification.

/OBJECT[=file-spec]
/NOOBJECT

Controls whether an object module is created by the assembler. It also defines the file specification for the file.

By default, the assembler creates an object module with the same file name as the first input file. The default file type for object files is OBJ. When you specify /OBJECT, you can control the defaults applied to the output file specification by the placement of the qualifier in the command. See Section 2.4 of the *VAX/VMS DCL Concepts Manual* for more information on entering output file qualifiers.

No wildcard characters are allowed in the file specification.

/SHOW[=(function[,...])
/NOSHOW[=(function[,...])]

Provides initial settings for the functions controlled by the assembler directives .SHOW and .NOSHOW. You can specify one or more of the functions listed below. If you specify /SHOW without any functions, the listing level count is incremented. If you specify /NOSHOW without any functions, the listing level count is decremented.

Since the /SHOW qualifier affects the listing, you must use the /LIST qualifier to produce a listing in order to use the /SHOW qualifier.

CONDITIONALS	Lists unsatisfied conditional code associated with .IF and .ENDC directives
CALLS	Lists macro calls and repeat range expansions
DEFINITIONS	Lists macro definitions
EXPANSIONS	Lists macro expansions
BINARY	Lists binary code generated by the expansion of macro calls

If you omit the /SHOW qualifier, the default is equivalent to /SHOW=(CONDITIONALS,CALLS,DEFINITIONS).

If you specify more than one function, separate them by commas and enclose the list in parentheses.

For additional details on these functions, see the *VAX MACRO and Instruction Set Reference Manual*.

/UPDATE[=(update-file-spec[,...])]
/NOUPDATE

Positional Qualifier.

The /UPDATE qualifier cannot be used with the /LIBRARY qualifier.

The /UPDATE qualifier requires usage with an associated input file that is to be updated with the specified update file or files. No wildcard characters are allowed in the update file specifications.

The update process uses the batch editor, SLP, to update the input file with the update file or files.

By default, the assembler uses update files with the same name as the input source file and a file type of UPD.

When multiple update files are specified with the /UPDATE qualifier, the assembler merges the contents into a single list of updates before applying the updates to the source file. You must separate multiple update files with commas and enclose the list in parentheses.

The input source file and update files are not changed by the update operation. The effects of the update appear in the compiled output. The listing also provides an audit trail of the changes if you specify /LIST with /UPDATE.

If an update file is not found, the assembler prints an informational message but continues the assembly.

The /NOUPDATE qualifier has the same effect as not specifying the /UPDATE qualifier, or negates the effects of any /UPDATE qualifiers specified earlier on the command line.

EXAMPLES

1
```
$ MACRO  ORION
```

The MACRO assembler assembles the file ORION.MAR and creates an object file named ORION.OBJ. If this command is executed in a batch job, the assembler also creates a listing file named ORION.LIS.

2
```
$ MACRO/LIST  CYGNUS, LYRA/OBJECT=LYRAN
```

This MACRO command requests two separate assemblies. The MACRO command assembles CYGNUS.MAR to produce CYGNUS.LIS and CYGNUS.OBJ. Then it assembles LYRA.MAR and creates a listing file named LYRA.LIS and an object module named LYRAN.OBJ.

```
3  $ MACRO  ALPHA/LIST+MYLIB/LIBRARY
   $_ + [TEST]OLDLIB/LIBRARY + []BETA
   $ PRINT ALPHA
```

The MACRO command concatenates the files ALPHA.MAR and BETA.MAR to produce an object file named ALPHA.OBJ and a listing file named ALPHA.LIS. MYLIB.MLB (in the current default directory) and OLDLIB.MLB (in the directory [TEST]) are specified as libraries to be searched for macro definitions. When macro calls are found in BETA.MAR, OLDLIB, MYLIB, and the system library STARLET.MLB are searched, in that order, for the definitions.

The PRINT command prints the listing file ALPHA.LIS.

```
4  $ MACRO  DELTA+TESTLIB/LIBRARY, ALPHA+MYLIB/LIBRARY
```

This MACRO command requests two separate assemblies. TESTLIB.MLB and the system library STARLET.MLB are searched for macro definitions when macro calls are found in DELTA.MAR, and MYLIB.MLB and the system library STARLET.MLB are searched for macro definitions when macro calls are found in ALPHA.MAR.

MAIL

Invokes the VAX/VMS Personal Mail Utility (MAIL), which is used to send messages to other users of the system. For a complete description of the VAX/VMS Personal Mail Utility, including information about the MAIL command and its qualifiers, see the *VAX/VMS Mail Utility Reference Manual*.

FORMAT

MAIL *[file-spec] [recipient-name]*

MERGE

Invokes the VAX Sort Utility to combine two through ten similarly sorted input files and create a single output file. Note that input files to be merged must be in sorted order. For a complete functional description of the Sort Utility, including more information about MERGE command and its qualifiers, see the *VAX/VMS Sort/Merge Utility Reference Manual*.

FORMAT **MERGE** *input-file-spec1,input-file-spec2[,...] output-file-spec*

MESSAGE

Invokes the VAX Message Utility (MESSAGE) to compile one or more files of message definitions. For a complete functional description of the Message Utility, including more information about the MESSAGE command and its qualifiers, see the *VAX/VMS Message Utility Reference Manual*.

FORMAT MESSAGE *file-spec[,...]*

MONITOR

Invokes the VAX/VMS Monitor Utility (MONITOR) to monitor classes of systemwide performance data at a specified interval. For a complete description of the Monitor Utility, including information about the MONITOR command, refer to the *VAX/VMS Monitor Utility Reference Manual*.

FORMAT **MONITOR** *[class-name[,...]]*

MOUNT

Invokes the Mount Utility (MOUNT) to make a disk or magnetic tape volume available for processing. For a complete description of the Mount Utility, including information about the MOUNT command and its qualifiers, see the *VAX/VMS Mount Utility Reference Manual*.

FORMAT **MOUNT** *device-name[:][,...] [volume-label[,...]] [logical-name[:]]*

ON

Defines the default courses of action when a command or program executed within a command procedure (1) encounters an error condition or (2) is interrupted by CTRL/Y. The specified actions are taken only if the command interpreter is enabled for error checking or CTRL/Y interrupts; these are the default conditions. You can use the ON command only within a command procedure.

FORMAT

ON *condition THEN [$] command*

restrictions

None.

PARAMETERS

condition

Specifies either the severity level of an error or a CTRL/Y interrupt. To specify the severity level of an error, use one of the following keywords:

WARNING
ERROR
SEVERE_ERROR

You can truncate any of these keywords to one or more characters. The default error condition is ON ERROR THEN EXIT.

To specify a CTRL/Y interrupt, use the following keyword:

CONTROL_Y

command

Specifies the action to be taken. You can specify any valid command line after the keyword THEN; you can optionally 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 three low-order bits of $STATUS. This severity level is also represented by the global symbol $SEVERITY. See the description of the EXIT command for information on severity level values.

If an ON command action specifies the severity level of an error, the command interpreter will execute the ON command action for errors at the specified severity level or greater. For example, the command:

```
$ ON WARNING THEN EXIT
```

causes a procedure to exit on warnings, errors, or severe errors.

The default action is:

```
$ 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. Note, however, that there is an exception to the default ON action. If you use the GOTO command 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 issuing 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. (See Table DCL–2-1 in the *VAX/VMS DCL Concepts Manual* for a list of commands that are executed within the command interpreter.)

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 interrupted the command procedure while it was executing an image that contained an exit handler, the exit handler will be allowed to execute before the new command (image) is executed. (However, if you issue the STOP command after you interrupt a command procedure with 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 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. It is not recommended, 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
```

After this statement is executed in a command procedure, when a warning or an error occurs the procedure continues normally. When a severe error occurs, the ON statement instructs the procedure to continue with the next statement anyway. Once the statement has been executed as a result of a fatal 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, reenables 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 clean-up operations (in this example, closing files and exiting).

OPEN

Opens a file for reading and/or writing. When opening a file, the OPEN command assigns a logical name to a file and places the name in the process logical name table.

FORMAT

OPEN *logical-name[:] file-spec*

restrictions

See qualifier descriptions.

PARAMETERS

logical-name[:]

Specifies a logical name to be assigned to the file.

file-spec

Specifies the name of the file or device to be opened for input or output. If the file specification does not include a file type, the system uses the default file type of DAT.

If you specify a file that does not exist, you can use the /WRITE qualifier to create a new, sequential file. See the description of the /WRITE qualifier for more information.

No wildcard characters are allowed in the file specification.

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 explicitly 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 VAX–11 RMS to open files, and is thus subject to RMS restrictions on using process permanent files. You can use the OPEN command to open sequential, relative, or indexed sequential access method (ISAM) files.

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

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 reference the file to which the logical name was originally assigned.

You can issue 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 using the /SHARE=READ or /SHARE=WRITE qualifier, other users may access the file with the TYPE or SEARCH command.

QUALIFIERS

/APPEND

Requests that an existing file be opened for writing and that the record pointer be positioned at the end-of-file. Any new records are added to the end of the file.

You can use the /APPEND qualifier only to add records to an existing file. The /APPEND and the /WRITE qualifiers are mutually exclusive.

/ERROR=label

Specifies a label on a line in the command procedure to receive control if the open request results in an error. The error routine specified for the /ERROR qualifier takes precedence over any action statement indicated in an ON command. If /ERROR 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

Requests that the file be opened for reading. This is the default if you do not specify either /READ or /WRITE. If you specify the /READ qualifier without the /WRITE qualifier, you must specify an existing file.

/SHARE[=option]

Requests that the specified file be opened as a shareable file to allow other users read or write access. If you specify /SHARE=READ, users are allowed read access to the file. If you specify /SHARE=WRITE or omit the option, users are allowed read and write access to the specified file.

/WRITE

Requests that the file be opened for writing. The following restrictions apply to the /WRITE qualifier:

- You can use the /WRITE qualifier to open and create a sequential file if you specify a file that does not exist and if you do not also use the /READ qualifier on the command line. 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, the OPEN /WRITE command creates a new file with a version number one greater than the existing file.
- You can use the /READ qualifier with the /WRITE qualifier to open an existing file. When the file is 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.
- If you specify both the /WRITE and /SHARE qualifiers with a sequential file, the file must contain fixed-length 512-byte records.

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::DBAO:[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. At this point, the CLOSE command closes the file and deassigns the logical name INPUT_FILE.

PASSWORD

Provides the password associated with the user name specified with the JOB card for a batch job submitted through a card reader. Although the PASSWORD card is required, the password on the card is optional if the account has a null password.

FORMAT

$ PASSWORD *[password]*

restrictions

- Valid only in a batch job submitted through a card reader.
- Requires a dollar sign preceding the PASSWORD command on the card.

PARAMETER

password

Specifies the password associated with the user name specified with the JOB command. The password parameter can have from 1 to 31 characters.

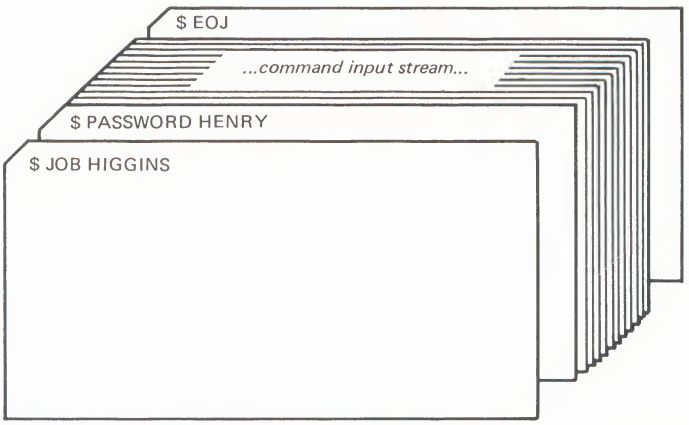
If you are submitting the job from an account with a null password, omit the password specifier on the PASSWORD card.

DESCRIPTION

The PASSWORD command is used in conjunction with the JOB command. The JOB command notifies the system of the user name that applies to the batch job being submitted through a card reader. The PASSWORD command notifies the system of the password associated with that user name. The password is checked by the system to be sure it matches the password associated with the user name specified 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 keypunch the PASSWORD card to prohibit other users from seeing the password when the PASSWORD card is in use.

EXAMPLE



ZK-786-82

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

PATCH

Invokes the VAX/VMS Image File Patch Utility (PATCH) to patch an executable image, shareable image, or device driver image. For a complete functional description of the Patch Utility, including more information about the PATCH command and its qualifiers, see the *VAX/VMS Patch Utility Reference Manual*.

FORMAT

PATCH *file-spec*

PHONE

Invokes the VAX/VMS Phone Utility that allows you to communicate with other users on your system or any other VAX/VMS system connected to your system by DECnet–VAX. For a complete description of the Phone Utility, including information on the PHONE command and its qualifiers, see the *VAX/VMS Phone Utility Reference Manual.*

FORMAT **PHONE** *[phone-command]*

PRINT

Queues one or more files for printing, either to the default system printer queue or to a specified queue.

FORMAT

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

restrictions

- Requires the /REMOTE qualifier if you include a node name in your file specification.
- Requires operator (OPER) privilege, execute (E) access to the queue, or write (W) access to the queue.

PARAMETER

file-spec[,...]

Specifies one or more files to be printed. If you specify two or more files, separate the file specifications with either commas or plus signs. The PRINT command concatenates the files into a single print job and by default, gives the job the name of the first file specified.

You can use wildcard characters in the directory specification, file name, file type, or version number fields. Node names are not allowed in the file specification unless the /REMOTE qualifier is specified.

If you do not specify a file type for the first input file, the PRINT command uses the default file type LIS.

DESCRIPTION

The PRINT command places the specified files in a printer or terminal queue for printing. All files queued by a single PRINT command are considered one job. The system assigns a unique job identification number—the "job entry number"—to each job in the queue and displays this entry number when the PRINT command completes execution. (Completion of a PRINT command means that the job has been queued to the appropriate printer queue. It does not imply that the actual printing process is finished.)

Once a print job has been queued, the version of the file submitted is printed, even if a newer version of the file is created before the print job runs.

Printer queues are identified by name. If you do not specify a queue name with the /QUEUE qualifier, the system queues the job to SYS$PRINT. The PRINT command, by default, displays the name of the queue on which it entered the job.

QUALIFIERS

/AFTER=time
/NOAFTER

Requests that the job not be printed until a specific time of day.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values.

If the specified time has already passed, the job is queued for printing immediately.

/BACKUP
/NOBACKUP

Selects files according to the dates of their most recent backup. This qualifier is only relevant when used with the /BEFORE or /SINCE qualifier. Use of the /BACKUP qualifier is incompatible with /CREATED, /EXPIRED, and /MODIFIED. The default is /CREATED.

/BEFORE[=time]
/NOBEFORE

Selects only those files that are dated before the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/BURST[=keyword]
/NOBURST

Positional qualifier.

Controls whether a burst page is printed preceding a file. A burst page precedes a flag page and contains the same information. However, it is printed over the perforation between the burst page and the flag page. The printing on the perforation makes it easy to see where individual print jobs or individual files within a single print job begin.

When you specify /BURST, you need not specify /FLAG; a flag page automatically follows a burst page.

If the /BURST qualifier is positioned between the PRINT command and the file specifications, it can take either of two keywords: ALL or ONE. The ALL keyword indicates that each file in the job will be preceded by a burst page and flag page. The ONE keyword indicates that a burst page will apply only to the first copy of the first file in the job.

To have the /BURST qualifier apply to individual files in a multi-file job, place the qualifier directly after each file that you want to have a burst page.

Use the /[NO]BURST qualifier to override the installation-defined defaults that have been set for the printer queue you are using.

/BY_OWNER[=uic]
/NOBY_OWNER

Selects one or more files only if their owner user identification code (UIC) matches the specified owner UIC.

Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual.*

If the /BY_OWNER qualifier is specified without a UIC, the UIC of the current process is assumed.

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

Specifies one or more characteristics desired for printing the files. If you specify only one characteristic, you can omit the parentheses. Characteristics can refer to such things as color of ink. Codes for characteristics can be either names or values from 0 to 127 and are installation-defined. Use the SHOW QUEUE/CHARACTERISTICS command to see which characteristics have been defined for your system. Use the SHOW QUEUE/FULL command to see which characteristics are available on a particular queue.

A print job can execute on a printer queue only if each characteristic specified with the PRINT command is also specified for that particular printer queue. If you specify a characteristic that has not been specified for that particular printer queue, the job remains in a pending status. (In order for your job to print, the system manager should stop the queue, physically change the characteristics of the printer, and restart the queue, specifying the new values listed in the /CHARACTERISTICS qualifier.)

Specification of a characteristic for a printer queue does not prevent jobs that do not specify that characteristic from being executed.

/CONFIRM
/NOCONFIRM (default)

Controls whether a request is issued before each individual PRINT operation to confirm that the operation should be performed on that file.

When the system issues the prompt, you can issue any of the following responses:

YES	NO	QUIT
TRUE	FALSE	CTRL/Z
1	0	ALL
	<RET>	

You can use any combination of upper- and lowercase letters for word responses. Word responses can be abbreviated to one or more letters (for example, T, TR, or TRU for TRUE). Affirmative answers are YES, TRUE, and 1. Negative answers are NO, FALSE, 0, and <RET> . QUIT or CTRL/Z indicates that you want to stop processing the command at that point. When you respond with 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, the prompt will be reissued.

/COPIES=n

Positional qualifier.

Specifies the number of copies to print. Any number of copies, from 1 through 255, may be specified. By default, the PRINT command prints a single copy of the specified files.

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 /COPIES following a file specification, only that file is printed the specified number of times.

/CREATED (default)
/NOCREATED

Selects files based on their dates of creation. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /CREATED qualifier is incompatible with /BACKUP, /EXPIRED, and /MODIFIED.

/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 /DELETE after a file specification, only that file is deleted after it is printed.

The protection applied to the file must allow delete access to the current UIC.

/DEVICE=queue-name[:]

Synonymous with the /QUEUE qualifier with the following exception: the /DEVICE qualifier is reserved for special use by DIGITAL. Its useage is therefore, not recommended.

/EXCLUDE=(file-spec[,...])
/NOEXCLUDE

Any files that match the listed file specifications are excluded from the PRINT operation. If you specify only one file, you can omit the parentheses.

The file specification can contain a directory specification, but you cannot include the device in the file specifications you supply with the /EXCLUDE qualifier.

Wildcard characters are supported for file specifications. However, you cannot use relative version numbers to exclude a specific version.

/EXPIRED
/NOEXPIRED

Selects files according to the dates on which they will expire. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /EXPIRED qualifier is incompatible with /BACKUP, /CREATED, and /MODIFIED. The default is /CREATED.

/FEED (default)
/NOFEED

Positional qualifier.

Controls whether the PRINT command automatically inserts form feeds when it nears the end of a page. Use of the /FEED qualifier causes the PRINT command to insert a form feed when the printer reaches the bottom margin of the form. The number of lines per form can be reset by the /FORM qualifier. 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 may be used to override the installation-defined defaults that have been set for the printer queue you are using.

/FLAG[=keyword]
/NOFLAG

Positional qualifier.

Controls whether a 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 two keywords: ALL or ONE. The ALL keyword indicates that each file in the job will be preceded by a flag page. The ONE keyword indicates that a flag page will apply only to the first copy of the first file in the job.

To have the /FLAG qualifier 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 installation-defined defaults that have been set for the printer queue you are using.

/FORM=type

Specifies the name or number of the form that you want for the print job.

Specify the forms type using a numeric value or alphanumeric name. Form types can refer to the print image width and length or the type of paper. Codes for form types are installation-defined. You can use the SHOW QUEUE/FORM command to find out the form types available for your system. Use the SHOW QUEUE/FULL command to find out which form is available for a particular queue.

If you specify a form type different from that of the queue, your job remains pending until the form type of the queue is set equal to the form type of the job. (In order to have your job print, the system manager should stop the queue, physically change the form type of the printer, and restart the queue specifying the new form type.)

/HEADER
/NOHEADER (default)

File-spec qualifier.

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

/HOLD
/NOHOLD (default)

Controls whether the file is available for printing immediately. When you specify the /HOLD qualifier, the file is not released for actual printing until you use the SET QUEUE/RELEASE/ENTRY command to release it.

/IDENTIFY (default)
/NOIDENTIFY

Controls whether the PRINT command displays a message on the current output device indicating the job number of the print job and the name of the queue in which it is entered whenever a job is successfully queued.

/JOB_COUNT=n

Requests that the entire job be printed n times, where n is a decimal integer from 1 to 255.

By default the job is printed once.

/LOWERCASE
/NOLOWERCASE (default)

Indicates whether the job must be printed on a printer that can print both uppercase and lowercase letters.

By default, files can be printed on printers that have only uppercase letters.

/MODIFIED
/NOMODIFIED

Selects files according to the dates on which they were last modified. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /MODIFIED qualifier is incompatible with /BACKUP, /CREATED, and /EXPIRED. The default is /CREATED.

/NAME=job-name

Defines a name string to identify the job. The name string can have from 1 to 39 characters. The job name is used in the SHOW QUEUE command display and is printed on the flag page for the job.

If you do not specify /NAME, the name string defaults to the file name of the first, or only, file in the job.

/NOTE=string

Allows you to specify a message to appear on the flag page for the job. The string can contain up to 255 characters.

/NOTIFY
/NONOTIFY (default)

Controls whether a message is broadcast to any terminal at which you are logged in, notifying you when your print job has been completed or aborted.

/OPERATOR=string

Allows you to specify a message to be sent to the operator. The string can contain up to 255 characters. When the job begins execution, the printer queue pauses and the message is transmitted to the operator.

/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 will start on the first page of the file.

The uplim specifier refers to the last page of the file that you want printed. When you want to print to the end of the file, but do not know how many pages that will be, you can use "" 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 five through 10; /PAGES=(5,"") starts printing at page 5 in the file 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 include up to 255 characters.

If you specify only one parameter, you can omit the parentheses.

The commas delimit individual parameters. To specify a parameter that contains any special characters or delimiters, enclose the parameter in quotation marks.

/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 /PAGE_SETUP qualifiers, will be ignored.

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

/PRIORITY=n

Specifies the priority of the print job. The priority value must be in the range of 0 through 255, where 0 is the lowest priority and 255 is the highest.

The default value for /PRIORITY is the value of the SYSGEN parameter DEFQUEPRI. You must have either OPER (operator) or ALTPRI (alter priority) privilege to raise the priority value above the value of the SYSGEN parameter MAXQUEPRI. No privilege is needed to set the priority lower than the MAXQUEPRI value.

/QUEUE=queue-name[:]

Requests that the print job be placed on the specified queue. If no queue is specified, files are queued to SYS$PRINT.

/REMOTE

Indicates that the specified files are to be printed on a remote node. Since the file to be printed must exist on the remote node, the file specification must contain the name of the remote node. If this qualifier is specified, the file is queued for printing in the default queue of the node on which the file exists.

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

Not all PRINT qualifiers are compatible with /REMOTE. Only the following qualifiers may be specified with /REMOTE: /BACKUP, /BEFORE, /BY_OWNER, /CONFIRM, /CREATED, /EXCLUDE, /EXPIRED, /MODIFIED, and /SINCE.

/RESTART (default)
/NORESTART

Specifies whether the job can restart after a crash or a STOP/QUEUE /REQUEUE command.

/SETUP=module[,...]

Calls for the specified modules to be extracted from the device control library and copied 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, and is recommended only for experimental setups.

For production setups, see DEFINE/FORM/SETUP.

/SINCE[=time]
/NOSINCE

Selects only those files that are dated after the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/SPACE
/NOSPACE (default)

Positional qualifier.

Controls whether output is to be double-spaced. The default is /NOSPACE, which results in single-spaced output.

/TRAILER[=keyword]
/NOTRAILER

Positional qualifier.

Controls whether a 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 two keywords: ALL or ONE. The ALL keyword indicates that each file in the job will be preceded by a trailer page. The ONE keyword indicates that a trailer page will apply only to the last copy of the last file in the job.

To have the /TRAILER qualifier 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 installation-defined defaults that have been set for the printer queue you are using.

/USER=username

Requires the change mode to kernel (CMKRNL) privilege and read (R) access to the system authorization file.

Allows you to submit a print job for another user. The print job will run exactly as if that user had submitted it. The job runs under that user's UIC. Accounting information is logged to that user's account. By default, the user identification comes from the requesting process.

The specifier can be any username that is validated on your system.

EXAMPLES

1
```
$ PRINT AVERAGE
Job AVERAGE (queue SYS$PRINT, entry 236) started on LPA0
```

The PRINT command queues the file AVERAGE.LIS to SYS$PRINT. The system displays the job name, the queue to which it was submitted, the job entry number, and the status. The job is already executing on the queue LPA0.

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

The PRINT command submits the files ALPHA.TXT, BETA.TXT, and GAMMA.TXT 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/QUEUE=LPB0: *.TXT
Job SPEECH1 (queue LPB0, entry 238) pending
```

The PRINT command queues to the execution printer queue LPB0 the highest versions of all files with the file type TXT. The system assigns the name of the first file as the job name.

4
```
$ PRINT/COPIES=10/AFTER=20   ALPHA.TXT
Job ALPHA (queue SYS$PRINT, entry 239) holding until 15-APR-84 20:00
```

The PRINT command queues 10 copies of the file ALPHA.TXT to the printer, but requests that the copies not be printed until after 8:00 P.M.

5
```
$ PRINT/LOWERCASE   ALPHA.TXT/COPIES=2, -
_$BETA.DOC/COPIES=3
  Job ALPHA (queue SYS$PRINT, entry 240) pending
```

The print job queued by this PRINT command consists of two copies of ALPHA.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.

6
```
$ PRINT/JOB_COUNT=3   ALPHA.TXT,BETA/NOIDENTIFY
```

This PRINT command concatenates the files ALPHA.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.

7
```
$ PRINT/REMOTE BOSTON::WORK$:[SMITH.MEMO]JUNE10.MEM
  Job JUNE10 (queue SYS$PRINT, entry 476) started on LPA0
```

The PRINT command, which is entered on a node other than BOSTON, queues the file JUNE10.MEM that resides on the BOSTON node. The file is entered on the printer queue at node BOSTON.

8
```
$ COPY REPORT.MEM BOSTON::
$ PRINT/REMOTE BOSTON::REPORT.MEM
  Job REPORT (queue SYS$PRINT, entry 342) started on LPA0
```

The two commands are entered at a node other than BOSTON. The COPY command copies the file REPORT.MEM from the current node to the BOSTON node. The PRINT command queues the file REPORT.MEM located on the BOSTON node for printing at the BOSTON node.

9
```
$ PRINT/HOLD   MASTER.DOC
  Job MASTER (queue SYS$PRINT, entry 540) holding
    .
    .
    .
$ SET QUEUE/ENTRY=540/RELEASE
```

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

PURGE

Deletes all but the highest-numbered versions of the specified files.

FORMAT

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

restrictions

None.

PARAMETER

file-spec[,...]

Specifies one or more files to be purged. If you specify two or more files, separate them with either commas or plus signs. If you do not provide a file specification, the PURGE command purges all files in the current default directory.

There are no file name or file type defaults with the PURGE command. You cannot specify a version number with any file spec. However, you can use wildcard characters in the directory specification, file name field, or file type field.

DESCRIPTION

You use the PURGE command to delete earlier versions of files. PURGE never deletes all versions of any file. By default, PURGE 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

Selects files according to the dates of their most recent backup. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /BACKUP qualifier is incompatible with /CREATED, /EXPIRED, and /MODIFIED. /CREATED is the default.

/BEFORE[=time]

Selects only those files that are dated before the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/BY_OWNER[=uic]

Selects one or more files only if their owner user identification code (UIC) matches the specified owner UIC.

Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

If the /BY_OWNER qualifier is specified without a UIC, the UIC of the current process is assumed.

/CONFIRM
/NOCONFIRM (default)

Controls whether a request is issued before each individual PURGE operation to confirm that the operation should be performed on that file.

When the system issues the prompt, you can issue any of the following responses:

YES	NO	QUIT
TRUE	FALSE	CTRL/Z
1	0	ALL
	<RET>	

You can use any combination of upper- and lowercase letters for word responses. Word responses can be abbreviated to one or more letters (for example, T, TR, or TRU for TRUE). Affirmative answers are YES, TRUE, and 1. Negative answers are NO, FALSE, 0, and <RET> . QUIT or CTRL/Z indicates that you want to stop processing the command at that point. When you respond with 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, the prompt will be reissued.

/CREATED (default)

Selects files based on their dates of creation. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /CREATED qualifier is incompatible with /BACKUP, /EXPIRED, and /MODIFIED.

/ERASE
/NOERASE (default)

When you simply 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. With the /ERASE qualifier, the storage location is overwritten with a system specified pattern so that the data no longer exists.

/EXCLUDE=(file-spec[,...])

Any files that match the listed file specifications are excluded from the PURGE operation. If you specify only one file, you can omit the parentheses. The file specification can contain a directory specification.

The file specification can contain a directory specification, but you cannot include the device in the file specifications you supply with the /EXCLUDE qualifier.

Wildcard characters are supported for file specifications. However, you cannot use relative version numbers to exclude a specific version.

/EXPIRED

Selects files according to the dates on which they will expire. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /EXPIRED qualifier is incompatible with /BACKUP, /CREATED, and /MODIFIED. /CREATED is the default.

/KEEP=n

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 the PURGE command displays the file specifications of files as it deletes them.

/MODIFIED

Selects files according to the dates on which they were last modified. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /MODIFIED qualifier is incompatible with /BACKUP, /CREATED, and /EXPIRED. /CREATED is the default.

/SINCE[=time]

Selects only those files that are dated after the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

EXAMPLES

1
```
$ PURGE
```

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

2
```
$ PURGE/KEEP=2
```

The PURGE command deletes all but the two highest-numbered versions of all the files in the default directory.

3
```
$ PURGE *.COM
```

The PURGE command deletes all but the highest-numbered version of each file with a file type of COM.

4
```
$ PURGE/KEEP=3  [WILDER.JOB308]ACCOUNT.COB
```

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

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

The PURGE command purges all files cataloged in the subdirectory named [MAL.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/KEEP=2 TAMPA::DISK1:[EXAMPLE]*.LIS`

This PURGE command 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

Reads a single record from a specified input file and assigns the contents of the record to a specified symbol name.

FORMAT

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

restrictions

None.

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 a 1- through 255-alphanumeric character symbol name to be equated to the contents of the record being read. A symbol name must start with an alphabetic letter, underscore, or 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 is already defined, the READ command redefines it to the new value being read.

DESCRIPTION

The READ command can read data from sequential, relative, or indexed sequential access method (ISAM) 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 ISAM 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.

In order to read a file, the file must be opened 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 explicitly opened 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

Specifies that a record be deleted from an ISAM file after it has been read. An ISAM file must be opened with the /READ and /WRITE qualifiers in order to use READ/DELETE.

/END_OF_FILE=label

Specifies the label on a line in the current command procedure where control is transferred after the last record in a file is read. When the last record in the file is read, the VAX Record Management Services (VAX RMS) return an error condition indicating the end-of-file. If the /END_OF_FILE qualifier is specified, the command interpreter transfers control to the command line at the specified label.

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

/ERROR=label

Specifies a label on a line in the command procedure to receive control if the read request results in an error. If no error routine is specified and an error occurs during the reading of the file, the current ON condition action is taken.

The error routine specified for this qualifier takes precedence over any action statement indicated in an ON command.

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

/INDEX=n

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

If you omit the /INDEX qualifier, the primary index (0) is assumed.

/KEY=string

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

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

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

/MATCH=option

Specifies the ISAM key match algorithm that is to be used when reading a record. You can specify the following options:

EQ	Selects keys that are equal to the specified key.
GE	Selects keys that are greater than or equal to the specified key.
GT	Selects keys that are greater than the specified key.

If you use the /MATCH qualifier, you must specify one of the options. If you are reading ISAM files and you do not use the /MATCH qualifier, the default is /MATCH=EQ.

/NOLOCK

Specifies that the record to be read is not to be locked, and also 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 read.

/PROMPT=string

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

/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, and the value must be from 0 through 255.

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

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 and on error conditions. At the end-of-file, 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

Displays previously entered commands so that you can reprocess them.

FORMAT

RECALL *[command-specifier]*

restrictions

None.

PARAMETER

command-specifier

Specifies the number or leading substring of the command you wish to recall.

Command numbers can range from 1 to 20. The most recently entered command is number 1. If you omit the command specifier, RECALL recalls the most recently entered command. Command 2 is the next to last command entered. The RECALL command itself is never assigned a number.

The term "leading substring" refers to the first several characters in a command string. The substring can be as short as a single character. If the substring is not unique, RECALL recalls the most recently issued command line that matches the substring. For example, suppose you issue a SHOW STATUS command and later issue 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.

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 will also be placed in the RECALL buffer. The RECALL command itself is never stored in the RECALL buffer.

The RECALL buffer can hold up to 20 commands or 1025 characters. You can use continuation characters with the commands to be recalled, but only 255 characters can be read at a time.

When you use the RECALL command, the system displays the command, but does not process it. If you want it processed as it appears, simply 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.

QUALIFIER

/ALL

Specifies that all the commands currently stored in the RECALL buffer be displayed, along with their command numbers. Remember that the RECALL command is never stored in the RECALL buffer.

EXAMPLE

```
$ SHOW DEFAULT
  DISK3:[SMITH]
$> DIRECTORY SEPT*
    .
    .
    .
$ SET DEFAULT [SMITH.LETTERS]
$ RECALL/ALL
1 SET DEFAULT [SMITH.LETTERS]
2 DIRECTORY SEPT*
3 SHOW DEFAULT
$ RECALL 2
$ DIRECTORY SEPT* <RETURN>
    .
    .
    .
$ RECALL 2
$ SET DEFAULT [SMITH.LETTERS]
<DELETE> <DELETE> <DELETE> <DELETE> <DELETE> <DELETE> <DELETE> <DELETE>
$ SET DEFAULT [SMITH.MEMOS] <RETURN>
$ RECALL 2
$ DIRECTORY SEPT* <RETURN>
```

This example starts with a SHOW DEFAULT and a DIRECTORY command. Not finding the file you want, you issue the SET DEFAULT command to move to the LETTERS subdirectory. You then use the RECALL/ALL command to see the list of commands you have entered. When you issue the RECALL 2 command, the system repeats the DIRECTORY command in the LETTERS subdirectory. Since you have still not found the file you want, you issue the RECALL command again to recall the SET DEFAULT command. Because you issued the DIRECTORY command again, the SET DEFAULT command becomes command 2 in the RECALL list. Using the DELETE key, you can edit the command line so that the system will set the default to the MEMOS subdirectory. Finally, you recall the DIRECTORY command to try once more to find the file. (At this point there are two identical DIRECTORY commands in the list, numbered 2 and 4. Press RETURN to process the recalled command.)

RENAME

Changes the directory specification, file name, file type, or file version of an existing disk file or disk directory.

FORMAT

RENAME *input-file-spec[,...] output-file-spec*

restrictions

None.

PARAMETERS

input-file-spec[,...]

Specifies the names of one or more files whose specifications are to be changed.

You can use wildcard characters in the 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.

output-file-spec

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 specification to provide defaults for fields in the output file that are either not specified, or indicated by a wildcard character. 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 according to the first description below that applies:

1. If the output file specification contains an explicit version number, the RENAME command uses that version number.
2. If the input file specification or output file specification contains an asterisk in the version number field, the RENAME command uses the version number of each input file to name the corresponding output file.
3. If no file currently exists with the same file name and file type as that specified for the output file, the RENAME command gives the new file a version number of 1.
4. If a file currently exists with the same file name and file type as that specified for the output file, the RENAME command gives the output file a version number one greater than the highest existing version, unless the /NONEW_VERSION qualifier is specified.

DESCRIPTION

The RENAME command enables you to change 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.

QUALIFIERS

/BACKUP

Selects files according to the dates of their most recent backup. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /BACKUP qualifier is incompatible with /CREATED, /EXPIRED, and /MODIFIED. /CREATED is the default.

/BEFORE[=time]

Selects only those files that are dated before the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/BY_OWNER[=uic]

Selects one or more files only if their owner user identification code (UIC) matches the specified owner UIC.

Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual.*

If the /BY_OWNER qualifier is specified without a UIC, the UIC of the current process is assumed.

/CONFIRM
/NOCONFIRM (default)

Controls whether a request is issued before each individual RENAME operation to confirm that the operation should be performed on that file.

When the system issues the prompt, you can issue any of the following responses:

YES	NO	QUIT
TRUE	FALSE	CTRL/Z
1	0	ALL
	<RET>	

You can use any combination of upper- and lowercase letters for word responses. Word responses can be abbreviated to one or more letters (for example, T, TR, or TRU for TRUE). Affirmative answers are YES, TRUE, and 1. Negative answers are NO, FALSE, 0, and <RET> . QUIT or CTRL/Z indicates that you want to stop processing the command at that point. When you respond with 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, the prompt will be reissued.

/CREATED (default)

Selects files based on their dates of creation. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /CREATED qualifier is incompatible with /BACKUP, /EXPIRED, and /MODIFIED.

/EXCLUDE=(file-spec[,...])

Any files that match the listed file specifications are excluded from the RENAME operation. If you specify only one file, you can omit the parentheses. The file specification can contain a directory specification.

Wildcard characters are supported for file specifications. However, you cannot use relative version numbers to exclude a specific version. The file specifications cannot include a device name.

/EXPIRED

Selects files according to the dates on which they will expire. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /EXPIRED qualifier is incompatible with /BACKUP, /CREATED, and /MODIFIED. /CREATED is the default.

/LOG
/NOLOG (default)

Controls whether the RENAME command displays the file specification of each file that it renames.

/MODIFIED

Selects files according to the dates on which they were last modified. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /MODIFIED qualifier is incompatible with /BACKUP, /CREATED, and /EXPIRED. /CREATED is the default.

/NEW_VERSION (default)
/NONEW_VERSION

Controls whether the RENAME command automatically assigns a new version number to the output file, if a file with the same file name and file type already exists.

With /NEW_VERSION, if a file with the same file name and file type exists, the RENAME command assigns a new version number to the new file. If you specify /NONEW_VERSION and if there is a file with the same file name and file type as the output file already in the directory, the system displays an error message.

/SINCE[=time]

Selects only those files that are dated after the specified time. You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

EXAMPLES

1
```
$ RENAME  AVERAGE.OBJ  OLDAVERAGE
```

The RENAME command changes the file name of 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 of 1.

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

The RENAME command 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 will not rename the file, but instead will report the error.

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

The RENAME command 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 [HIGGINS]COMPLIB.OLB  [HIGGINS.LIBRARY]
```

The RENAME command changes the directory name of the object module library COMPLIB.OLB. The library is now cataloged in the subdirectory [HIGGINS.LIBRARY].

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

The RENAME command renames the file SAVE.DAT in the directory MALCOLM.TESTFILES to TEST.DAT. The new file is cataloged in the current default directory.

6
```
$ RENAME/LOG
$_From:      DATA.*,INFO.*
$_To:        NEW
%RENAME-I-RENAMED, _DISKO:[SYSTEM]DATA.AAA;1 renamed to _DISKO:[SYSTEM]NEW.AAA;1
%RENAME-I-RENAMED, _DISKO:[SYSTEM]DATA.BBB;1 renamed to _DISKO:[SYSTEM]NEW.BBB;1
%RENAME-I-RENAMED, _DISKO:[SYSTEM]DATA.CCC;1 renamed to _DISKO:[SYSTEM]NEW.CCC;1
%RENAME-I-RENAMED, _DISKO:[SYSTEM]INFO.001;1 renamed to _DISKO:[SYSTEM]NEW.001;1
%RENAME-I-RENAMED, _DISKO:[SYSTEM]INFO.002;1 renamed to _DISKO:[SYSTEM]NEW.002;1
%RENAME-I-RENAMED, _DISKO:[SYSTEM]INFO.003;1 renamed to _DISKO:[SYSTEM]NEW.003;1
$
```

Three files exist with the file name of DATA and three files have the file name of INFO. This RENAME command illustrates wildcard characters 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.

7
```
$ RENAME NODE1::DISK2:[SMITH]ASSEMSHT.EXE NODE1::DISK3:[JONES]ASSEMBLYSHEET.EXE
```

This RENAME command 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. Note that you can rename a file on another node and disk providing the new file resides on that same node and disk.

REPLY

Enables an operator to communicate with system users.

FORMAT

REPLY *["message-text"]*

restrictions

See qualifier descriptions.

PARAMETER

message-text

Specifies the message to be displayed at one or more users' terminals. The maximum length of a message is 128 characters. If this limit is exceeded, an error occurs.

When a message consists of more than one word, enclose the message in quotation marks.

DESCRIPTION

The operator uses 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
- Closes the operator's log file and opens a new one

Displaying Messages at Users' Terminals

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

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

The /ALL qualifier sends a message to all terminals that are online and connected to the VAX/VMS system or VAXcluster. 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 VAXcluster.

The /USERNAME qualifier sends a message to terminals at which one or more system or VAXcluster 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 (operator) 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 issues the REQUEST/REPLY command, the process associated with the requesting user's terminal is put in a wait state until the operator responds 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 cancelled.

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 issues the REQUEST/REPLY command, the message appears at the system console terminal. For example:

```
%OPCOM, 15-AUG-1984 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 /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 five-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 multivolume tape volume reaches the end-of-tape 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 issues the REPLY/TO command. However, if the file system requests a new volume, the operator can mount a scratch tape and issue the REPLY/INITIALIZE_TAPE or REPLY/BLANK_TAPE command with the message "label". The double 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 cancelled.

Enabling and Disabling Operator Status on a Terminal

Any terminal connected to the VAX/VMS operating system can be established as an operator's terminal. When an operator who is logged in to an account with 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 will retain 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 /TEMPORARY).

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

When the operator issues the REPLY/ENABLE command, the operations command procedure (OPCOM) confirms that the terminal has been enabled. For example:

```
$ REPLY/ENABLE
%OPCOM, 15-AUG-1984 10:22:19.75, operator status for operator OPA0
CENTRAL, PRINTER, TAPES, DISKS, DEVICES, CARDS, NETWORK, CLUSTER,
OPER1, OPER2, OPER3, OPER4, OPER5, OPER6, OPER7, OPER8, OPER9, OPER10,
OPER11, OPER12
```

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

```
%OPCOM, 15-AUG-1984 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 types the following:

```
$ REPLY/ENABLE=(DISKS,TAPES)
%OPCOM, 15-AUG-1984 10:04:00.18, operator enabled, operator OPA0
$
%OPCOM, 15-AUG-1984 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 communication with a particular facility, 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 issues the following command:

```
$ REPLY/DISABLE=DISKS
%OPCOM, 15-AUG-1984 10:04:30.83, operator status for operator OPA0
TAPES
```

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 user requests in the operator log file and responds to these requests with the following message:

```
%OPCOM-S-OPRNOTIF, operator notified, waiting. 10:06:03.25
%OPCOM-S-OPREPLY, %OPCOM 15-AUG-1984 10:06:03:25, no operator coverage
```

To determine the operator status and obtain a list of pending requests for a particular terminal, the operator issues 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 a new one, the operator issues the REPLY/LOG command. All subsequent messages are recorded in the new log file. To close the current log file without opening a new one, the operator issues the REPLY/NOLOG command. All subsequent messages are not recorded until the operator issues 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 aborts the request.

/ALL

Requires operator (OPER) privilege.

Broadcasts a message to all terminals that are attached to the system or VAXcluster, are turned on, and have broadcast message reception enabled.

/BELL

Rings a bell at the terminal which is receiving the message. This qualifier can be used only with the following qualifiers:

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

/BLANK_TAPE=identification-number

Requires volume protection (VOLPRO) 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.

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

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

Restores to normal status (that is, nonoperator status) the terminal at which the command is issued. The REPLY/DISABLE command cannot be issued from a batch job.

To restrict the types of messages displayed on an operator's terminal, specify one or more of the keywords in Table DCL–12 at the end of this section. If no keywords are specified, all operator characteristics are disabled. If only one keyword is specified, you can omit the parentheses.

When an operator logs out, the operator terminal is automatically disabled.

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

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

Designates the terminal at which the command is issued to be an operator's terminal. The REPLY/ENABLE command cannot be issued from a batch job.

To designate a terminal to receive messages from a particular facility, specify one or more of the keywords in Table DCL–12 at the end of this section. If only one keyword is specified, you can omit the parentheses.

/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.

/LOG
/NOLOG

Requires operator (OPER) privilege.

Closes the current operator's log file and controls whether a new operator's log file is opened. The /LOG qualifier closes the current log file and opens a new one. The operator can then examine the contents of the previous log file.

The /NOLOG qualifier closes the current log file but does not open a new one. Hence, no messages are recorded until the operator opens another log file.

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

Sends a message to the local VAXcluster node only. The optional parameter list allows you to specify which nodes will receive the message. The default is to send the message to all the nodes on the cluster.

/NOTIFY (default)
/NONOTIFY

Requests that the originating terminal be notified when the message is successfully received.

/PENDING=identification-number

Requires operator (OPER) privilege.

Sends a message to the user indicated by the unique identification number, and keeps the process from which the request was issued in a wait state. The user cannot enter other commands until the operator fulfills or aborts the request.

/SHUTDOWN

Sends a message which starts with "*SHUTDOWN* message ...". If the /BELL qualifier was also specified, the bell will ring three times.

/STATUS

Requires operator (OPER) privilege.

Reports the current operator status and all outstanding user requests for the terminal from which this command was entered.

/TEMPORARY

Designates the terminal at which the command is issued 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 operator (OPER) privilege.

Broadcasts a message to the specified terminals. If only one terminal is specified, you can omit the parentheses.

/TO=identification-number

Requires operator (OPER) privilege.

Sends a message to the user or file system indicated by the identification number and completes the request.

/URGENT

Sends a message which starts with "*URGENT* message ...". If the /BELL qualifier was also specified, the bell will ring twice.

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

Requires operator (OPER) privilege.

Broadcasts a message to all terminals at which users are logged in to the system or VAXcluster. The optional parameter list allows you to specify a list of users, who will receive your message.

/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 VAXcluster, the message will only be sent to the local node.

Table DCL–12 REPLY/ENABLE and REPLY/DISABLE Keywords

Keyword	Function
CARDS	Inhibits/allows messages pertaining to card readers.
CENTRAL	Inhibits/allows messages sent to the operator identified as the central system operator.
CLUSTER	Inhibits/allows messages from the connection manager pertaining to changes in the state of the cluster.

Table DCL–12 (Cont.) REPLY/ENABLE and REPLY/DISABLE Keywords

Keyword	Function
DEVICES	Inhibits/allows messages pertaining to mounting and dismounting disks and tape volumes.
DISKS	Inhibits/allows messages pertaining to mounting and dismounting disk volumes.
NETWORK	Inhibits/allows messages pertaining to networks. The CENTRAL keyword must also be specified to inhibit network messages.
OPER1 through OPER12	Inhibits/allows messages sent to operators identified as OPER1, OPER2 ,..., OPER12.
PRINTER	Inhibits/allows messages pertaining to print requests.
SECURITY	Inhibits/allows messages pertaining to security events. Requires SECURITY privilege.
TAPES	Inhibits/allows messages pertaining to mounting and dismounting tape volumes.

EXAMPLES

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

This command broadcasts a message to all terminals on the system. When the message appears at the user's terminal, it is prefixed with terminal name, the user name of the sender, and, when DECnet–VAX is installed, the node name. The bell sounds at the terminal as the message is displayed.

2
```
$ REPLY/ENABLE=DISKS
%OPCOM, 15-AUG-1984, 10:17:09.02, operator enabled, operator OPA0
$
%OPCOM, 15-AUG-1984 10:17:10.30, operator status for operator OPA0
DISKS
```

The REPLY/ENABLE command 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, 15-AUG-1984 10:19:33.21, request 5, from user SYSTEM
OPA0, Please mount OPGUIDE on DBA3:
$ REPLY/PENDING=5 "YOU'LL HAVE TO WAIT-THERE ARE SEVERAL REQUESTS BEFORE YOURS"
   .
   .
   .
REPLY/TO=5
 15-AUG-1984 10:20:25.50, request 5 completed by operator OPA0
```

The OPCOM message indicates that a user wants the operator to place the disk volume labeled OPGUIDE on the disk drive DBA3 and ready the device. The REPLY/PENDING command indicates that the operator can perform the task but not immediately; the /PENDING qualifier keeps the process associated with the user in a wait state. Therefore, the user cannot enter other commands until the operator fulfills or aborts the request. The operator later mounts the specified disk on the specified drive and sends a message to the user indicating that the request has been fulfilled. When no message

accompanies the REPLY/TO command, OPCOM sends a standard message indicating that the task has been performed.

```
4  $ REPLY/STATUS
   %OPCOM, 15-AUG-1984 10:20:50.39, operator status for operator OPA0
   DISKS
```

The REPLY/STATUS command 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.

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

This REPLY command sends a message to the user logged in at terminal TTC1. When the message is displayed, a bell rings at that terminal.

```
6  $ REPLY/ENABLE
   $
   $ REPLY/ENABLE
   %OPCOM, 15-AUG-1984 10:22:19.75, operator status for operator OPA0
   CENTRAL, PRINTER, TAPES, DISKS, DEVICES, CARDS, NETWORK, CLUSTER,
   OPER1, OPER2, OPER3, OPER4, OPER5, OPER6, OPER7, OPER8, OPER9, OPER10,
   OPER11, OPER12
      .
      .
      .
   $ REPLY/DISABLE=(PRINTER, TAPES)
   %OPCOM, 15-AUG-1984 10:22:26.07, operator disabled, operator OPA0
```

The REPLY/ENABLE command 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

Displays a message at a system operator's terminal, and optionally requests a reply. System operators are identified by the functions they perform. If more than one operator is designated for a particular function, all receive the specified message.

FORMAT

REQUEST *"message-text"*

restrictions

None.

PARAMETER

"message-text"

Specifies the text of a message to be displayed at the specified operator terminals.

The message text can have a maximum of 128 characters. If you type more than one word, enclose the text in double quotation marks.

DESCRIPTION

When you use the REQUEST command to send a message to an operator, the message is displayed at the specified operator terminals, according to the keywords specified with the /TO qualifier. If you specify /REPLY, the message is assigned an identification number, so the operator can respond to the message.

If you specify /REPLY, you receive the message:

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

When the operator responds to your request, you receive a message in the format:

```
%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, you receive the message:

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

At this time, you can enter either a message to be displayed at the specified operator terminals, or you can press CTRL/Z to cancel the request. If you enter a 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 specified message.

If you request a reply, the message is assigned a unique identification number that the operator uses when he issues his REPLY command.

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

Specifies one or more operators to whom you wish to send the message. By default, the message is sent to all terminals currently designated as operator terminals. However, you can specify one or more of the keywords to limit the number of operators receiving the message. If you specify only one keyword, you can omit the parentheses.

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 designated to mount and dismount disk and tape volumes.
DISKS	Sends the message to operators designated to mount and dismount disk volumes.
NETWORK	Sends the message to the network operator.
OPER1 through OPER12	Sends the message to installation-specified operators identified as OPER1, OPER2, and so on.
PRINTER	Sends the message to operators designated to respond to printer 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
 15-APR-1984 10:25:32.40, request 3 completed by operator OPA0
```

The PRINT command requests that multiple copies of a file be printed using a special type of paper (/FORM=LETTER). After queueing the job to the printer, the REQUEST command sends a message to the system operator designated to handle users' requests about print jobs. The message asks the operator if the job can be printed using the specified form.

The reply indicates that the request has been taken care of.

```
2   $ REQUEST/REPLY  "ARE YOU THERE"
    %OPCOM-S-OPRNOTIF, operator notified, waiting...14:54:30.33
    ^C
    REQUEST-Enter message or cancel request with ^Z
    REQUEST-Message?^Z
    %OPCOM-S-OPRNOTIF, operator notified, waiting... 14:59:01.38
    %OPCOM-F-RQSTCAN, request was cancelled
```

The REQUEST command issues a message to see if any operators are at their terminals to respond to the message. When no operator sends a reply to the question, CTRL/C is used to interrupt the request and then CTRL/Z is used to cancel it.

RETURN

Terminates a GOSUB subroutine procedure. Returns control to the command following the calling GOSUB command.

FORMAT

RETURN *[status-code]*

restrictions

None.

PARAMETER

status-code

Defines a numeric value for the reserved global symbol $STATUS. You can specify the status-code as an integer or an expression equivalent to an integer value. The value can be tested by the next outer command level. low-order three bits of the longword integer value change the value of the reserved global symbol $SEVERITY.

If you specify a status-code, DCL will interpret 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

Use the RETURN command to terminate the GOSUB subroutine and return 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 three 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 zero through four:

Value	Severity
0	Warning
1	Success
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
  26-SEP-1985 14:25:42
$ GOSUB SYMBOL
$ EXIT
$ SYMBOL:
$     SHOW SYMBOL RED
      RED = "SET DEFAULT [MALIN.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)

Places an image into execution in the process. You can truncate the RUN command to a single letter, **R**.

FORMAT

RUN *file-spec*

restrictions

If you specify an image name in the command line with an explicit version number (or a semicolon), the image will run with current process privileges.

However, if you specify the image name in the command line without an explicit version number (or a semicolon), the image will run with any privileges with which it was installed.

PARAMETER

file-spec

Specifies an executable image to be executed. If you do not specify a file type, the RUN command uses the default file type of EXE.

No wildcard characters are allowed in the file specification.

QUALIFIER

/DEBUG
/NODEBUG

Controls, for native VAX/VMS images, whether the image is to be run with the debugger. If the image was linked with the /DEBUG qualifier and if you do not want the debugger to prompt, use the /NODEBUG qualifier. If the image was linked with /TRACEBACK, traceback reporting is performed when an error occurs.

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

EXAMPLES

1
```
$ RUN LIBRA
```

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 will still run with its amplified privileges, if you issue the RUN command as follows:

```
RUN LIBRA.EXE
```

```
2  $ MACRO/ENABLE=DEBUG ORION
   $ LINK/DEBUG ORION
   $ RUN ORION
        VAX DEBUG Version 4.0
   %DEBUG-I-INITIAL, language is MACRO, module set to '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 has been installed with amplified privileges, it does not run with those privileges because you have explicitly specified a version number. Instead, the image runs with only current process privileges. When you explicitly 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 issue the RUN command as follows:

```
RUN AQUARIUS.EXE;
```

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

RUN (Process)

Creates a subprocess or a detached process to execute the specified image. If you specify any of the qualifiers except /UIC or /DETACHED, the RUN command creates a subprocess. A detached process is created if the /UIC qualifier is specified and you have the DETACH user privilege.

FORMAT

RUN *file-spec*

restrictions

See qualifier descriptions.

PARAMETER

file-spec

Specifies an executable image to be executed in a separate process. If the file specification does not include a file type, the RUN command assumes the default file type EXE.

No wildcard characters are allowed in the file specification.

DESCRIPTION

The RUN command creates a process to execute the specified image. If you specify the /UIC qualifier, RUN 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 code (PID) 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 which was running that image.

By default, the RUN command creates a subprocess with the same UIC, current disk and directory defaults, privileges, and priority as the current process.

Both the /DETACHED and the /UIC qualifiers request the RUN command to create a detached process. You must have the user privilege DETACH to create a detached process with a different UIC (user identification code). When you create a detached process, the resource quotas are the same as those of the current process. However, if you have DETACH 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 with 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 issue 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, issue the SHOW PROCESS/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 Wakeups

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 a state of hibernation. The process cannot execute the image until it is awakened. Time values specified with these three qualifiers control when the process will be awakened to execute the specified image.

You can schedule wakeups for a specified delta time (/DELAY qualifier) or an absolute time (/SCHEDULE qualifier). You can also schedule wakeups 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. When the image completes normally, the process is returned to a state of hibernation. The image is reactivated at the next scheduled wakeup time. Note that if the image completes abnormally (for example, if it calls the $EXIT system service) the process does not return to hibernation.

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 accounting (ACNT) privilege to disable accounting.

Controls whether accounting records are to be logged in the system accounting file for the created process.

/AST_LIMIT=quota

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

If you do not specify an AST limit quota, the default value established at system generation time is used. The minimum required for any process to execute is 2. A value of 10 is typical.

The AST limit quota is nondeductible.

/AUTHORIZE
/NOAUTHORIZE (default)

Requires DETACH privilege.

When the image to be executed is the system login image (LOGINOUT.EXE), this qualifier controls whether login searches the user authorization file to validate a detached process.

By default, the login image creates a detached process running under the control of the command interpreter. Specify /AUTHORIZE to have the login image check the user authorization file 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.

Any nonspecified attributes of the created process default to those of the current process.

/BUFFER_LIMIT=quota

Specifies the maximum amount of memory, in bytes, that the process can use for buffered I/O operations or 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. A value of 10,240 is typical.

The buffer limit quota is pooled.

/DELAY=delta-time

Requests that the created process be placed in hibernation and then awakened after a specified time has elapsed.

Specify the delta time according to the rules for entering delta times given in Section 2.5 of the *VAX/VMS DCL Concepts Manual*.

If you specify both the /INTERVAL and /DELAY qualifiers, the first wakeup request occurs at the time specified by /DELAY, and all subsequent wakeups occur at the interval specified by /INTERVAL.

/DETACHED
/NODETACHED

Directs that the created process be a detached process with the same user identification code (UIC) as the current process. By default, the created process is not a detached process.

The detached process will have the same resource quotas as the current process, unless the current process has DETACH privilege. With DETACH privilege, you can specify any quotas you need.

Unless you have DETACH privilege, the maximum number of detached processes that you can create is limited to the quota defined by MAX_DETACH in your user authorization file (UAF).

Use the /UIC qualifier to create a detached process with a different UIC.

/DUMP
/NODUMP (default)

If an image terminates due to a condition that was not handled, the /DUMP qualifier causes the contents of the address space to be written to a file in your current default directory. The file name used will be that of the running image, and the file type will be DMP. 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.

If you do not specify an enqueue limit quota, the default value established at system generation time is used. The minimum required for any process to operate is 2. A value of 6 is typical.

/ERROR=file-spec

Defines an equivalence name string for the logical device name SYS$ERROR. The equivalence name can have from 1 to 63 alphanumeric characters. The logical name and equivalence name are placed in the process logical name table for the created process.

Note that 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.

If you do not specify an extent quota, the system uses the default value established at system generation time. The minimum value required for any process to execute is 10 pages. A value in the range of 400 to 2000 is typical.

The extent quota is nondeductible.

/FILE_LIMIT=quota

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

If you do not specify an open file quota for a created process, the system uses the default value established at system generation time. The minimum amount required for any process to execute is 2. A value of 20 is typical.

The file limit quota is pooled.

/INPUT=file-spec

Defines an equivalence name string for the logical device name SYS$INPUT. The equivalence name can have from 1 to 63 alphanumeric characters. 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 awakened at regularly scheduled intervals.

Specify the delta time according to the rules for entering delta times given in Section 2.5 of the *VAX/VMS DCL Concepts Manual*.

If you specify the /DELAY or /SCHEDULE qualifier with the /INTERVAL qualifier, the first wakeup occurs at the time specified by /DELAY or /SCHEDULE; all subsequent wakeups occur at intervals specified by /INTERVAL. If you do not specify /DELAY or /SCHEDULE, but do specify /INTERVAL, the first wakeup 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.

If you do not specify a buffered I/O quota, the default value established at system generation time is used. The minimum required for any process to execute is 2. A value of 6 is typical.

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.

If you do not specify a direct I/O quota, the default value established at system generation time is used. The minimum required for any process to execute is 2. A value of 6 is typical.

The direct I/O quota is nondeductible.

/JOB_TABLE_QUOTA=quota

Allows you to specify a quota for a detached process's job-wide 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 only has meaning for detached processes. If the /JOB_TABLE_QUOTA qualifier is specified in a RUN command which results in the creation of subprocess, it will be ignored.

/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 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 Adjust Working Set Limit system service.)

If you do not specify a working set quota, the system uses the default value established at system generation time. The minimum value required for any process to execute is 10 pages. A value of 200 is typical.

The maximum working set quota is nondeductible.

/OUTPUT=file-spec

Defines an equivalence name string for the logical device name SYS$OUTPUT. The equivalence name can have from 1 to 63 alphanumeric characters. 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 pages 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.

If you do not specify a paging file quota, the system uses the default value established at system generation time. The minimum value required for a process to execute is 256 pages. A value of 10000 pages is typical.

The paging file quota is pooled.

/PRIORITY=n

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

Specifies the base priority at which the created process will execute.

The priority value is a decimal number from 0 through 31, where 31 is the highest priority and 0 is the lowest. Normal priorities range from 0 through 15; real-time priorities range from 16 through 31.

If you try to specify a higher value when you do not have the ALTPRI privilege, or if you do not specify a priority, the priority defaults to the base priority of the current process.

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

Requires SETPRIV 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.

If you specify only one privilege, you can omit the parentheses. For the list of privileges, see Table 1-3 in the *VAX/VMS DCL Concepts Manual*.

You can also use the keyword NOSAME as the privilege parameter. If you do not use the /PRIVILEGES qualifier, the created process has the same privileges as your current process. If you specify /PRIVILEGES=NOSAME, the created process has no privileges.

Note that if you use an explicit version number with the file specification parameter (or simply include a semicolon), the current process privileges are used, regardless of any privilege specified with the /PRIVILEGES qualifier.

/PROCESS_NAME=process-name

Defines a name for the created process. The process name can have from 1 to 15 alphanumeric characters. The process name is implicitly qualified by the group number of the process's user identification code (UIC).

If you do not specify a process name, the created process has a null name by default.

/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.

If you do not specify a timer queue entry quota, the system uses the default value established at system generation time. A process does not require any timer queue quota in order to execute. A value of 8 is typical.

The timer queue entry quota is pooled.

/RESOURCE_WAIT (default)
/NORESOURCE_WAIT

Enables or disables resource wait mode for the created process. By default, the system places a process in a wait state when a resource required for a particular function is not available.

If you specify /NORESOURCE_WAIT, the process will receive an error status code when a resource is unavailable.

/SCHEDULE=absolute-time

Requests that the created process be placed in hibernation and awakened at a specific time of day.

Specify the absolute time value according to the rules for entering absolute time values given in Section 2.5 of the *VAX/VMS DCL Concepts Manual*.

/SERVICE_FAILURE
/NOSERVICE_FAILURE (default)

Enables or disables system service failure exception mode for the created process. By default, for errors occurring when a process calls a system service, a status code indicating an error is returned.

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

/SUBPROCESS_LIMIT=quota

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

If you do not specify a subprocess limit, the system uses the default value established at system generation time. A process does not require any subprocess quota in order to execute. A value of 8 is typical.

The subprocess limit quota is pooled.

/SWAPPING (default)
/NOSWAPPING

Requires PSWAPM privilege to disable process swapping.

Enables or disables process swap mode for the created process. By default, a process is 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 is 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) allocated to the created process. The resolution is to ten milliseconds. When the time expires, the process is deleted. The default value is established at system generation time. A CPU time limit of 0 indicates that CPU time is not restricted; this is a typical value.

If you restrict CPU time for a process, specify the time limit according to the rules for specifying delta time values, as given in Section 2.5 of the *VAX/VMS DCL Concepts Manual*.

The time limit quota is deductible.

/UIC=uic

Directs that the created process is to be a detached process with the specified user identification code (UIC). Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

/WORKING_SET=default

Specifies the default working set size for the created process, that is, the number of pages in the working set for the image that is executing.

If you do not specify a default working set size, the system uses the default value 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 working set quota (specified with the /MAXIMUM_WORKING_SET qualifier). A value of 200 pages is typical.

The maximum working set quota is nondeductible.

EXAMPLES

1
```
$ RUN/PROCESS_NAME=SUBA   SCANLINE
%RUN-S-PROC_ID, identification of created process is 00010044.
```

The RUN command creates a subprocess named SUBA to execute the image SCANLINE.EXE. The system gives the subprocess an identification number of 00010044.

2
```
$ RUN/INTERVAL=1:40/PROCESS_NAME=STAT   STATCHK
%RUN-S-PROC_ID, identification of created process is 00050023
    .
    .
    .
$ CANCEL STAT
```

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 /SCHEDULE qualifier is specified, the first wakeup 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.

3
```
$ RUN/PROCESS_NAME=LYRA  LYRA -
$_/OUTPUT=_TTB3: -
$_/ERROR=_TTB3:
%RUN-S-PROC_ID, identification of created process is 000A002F
```

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.

4
```
$ RUN/UIC=[100,4]/PRIVILEGES=(SAME,NOPSWAPM) -
$_/NORESOURCE_WAIT   OVERSEER
%RUN-S-PROC_ID, identification of created process is 0001C02C
```

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.

5
```
$ DEFINE/GROUP  TEST [MALCOLM.TESTFILES]
$ RUN/PROCESS=SUB  WATCH -
$_/INPUT=TEST:OUT1 -
$_/OUTPUT=F$LOGICAL("SYS$OUTPUT")
%RUN-S-PROC_ID, identification of created process is 0001002E
```

The DEFINE command creates an entry in the group logical name table for the logical name TEST. The RUN command creates a subprocess to execute the image WATCH.EXE.

The /INPUT qualifier defines SYS$INPUT for the subprocesses. The logical name TEST defines the directory for the file OUT1.DAT. Because the logical name TEST is in the group logical name table, the logical name can be translated and referred to by the image WATCH.EXE.

The /OUTPUT qualifier uses the lexical function F$LOGICAL to translate the logical name of the current process's SYS$OUTPUT device. The equivalence name string is equated to the device SYS$OUTPUT for the subprocess.

RUNOFF

Invokes the DIGITAL Standard Runoff (DSR) text formatter to format one or more ASCII files. This description provides a functional overview of the RUNOFF command, emphasizing DCL syntax and grammar. For a complete functional description of the DSR formatter, including more information about the RUNOFF command, see the *VAX DIGITAL Standard Runoff (DSR) Reference Manual*. For information about the RUNOFF/CONTENTS and RUNOFF/INDEX commands, see the separate descriptions of those commands.

FORMAT

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

restrictions

None.

PARAMETER

file-spec[,...]

Specifies one or more ASCII files (containing text and DSR commands) to be formatted by the RUNOFF command. If you omit the file type, DSR assumes a file type of RNO. Separate multiple files with commas.

DSR produces an output file having the same file name as the input file. The output file type depends on the input file type. For a list of input file types and the associated output file types, see the *VAX DIGITAL Standard Runoff (DSR) Reference Manual*. The default input file type is RNO and the default output file type is MEM.

Wildcard characters are not allowed in the file specification.

DESCRIPTION

The RUNOFF command allows you to:

- Adjust the amount of text on a page
- Control the position of text on a page
- Control underlining, overwriting, and bolding of text
- Override some DSR commands and flags in your input file
- Process all or part of your input file
- Create an intermediate file for indexes or tables of contents

QUALIFIERS

/BACKSPACE

Positional qualifier.

Controls whether DSR uses the ASCII backspace character to perform character-by-character overprinting.

The default is for DSR to perform line-by-line overprinting.

/BOLD[=n]
/NOBOLD

Positional qualifier.

Controls whether characters flagged for bolding are overprinted. You can specify the number of times DSR overprints flagged text by stating a value for the variable n. If you specify /BOLD without specifying a number, the flagged text is overprinted once (/BOLD=1). If you specify /BOLD=0, or specify /NOBOLD, the text is not overprinted.

/CHANGE_BARS[="character"]
/NOCHANGE_BARS

Positional qualifier.

Controls whether DSR outputs change bars in the formatted file. The default change-bar character is the vertical bar (|). The change bars appear 3 spaces to the left of the lines of text that you have marked for change bars. See the .BEGIN BAR and .END BAR commands in *VAX DIGITAL Standard Runoff (DSR) Reference Manual*.

You can replace the default change-bar character by supplying a substitute character for the /CHANGE_BARS[="character"] qualifier. You must specify the replacement character as either a character enclosed in quotation marks or as an octal, decimal, or hexadecimal value for the desired character.

The /CHANGE_BARS qualifier without a value uses the default change-bar character (|). The /NOCHANGE_BARS qualifier overrides any change-bar commands in the input file and disables the output of change bars.

/DEBUG[=(option[,...])]
/NODEBUG (default)

Positional qualifier.

Controls whether DSR outputs the information specified by the /DEBUG options. You can use one or more of the following command options:

ALL	Specifies all five options (CONDITIONALS, CONTENTS, FILES, INDEX, and SAVE_RESTORE).
CONDITIONALS	Causes DSR to ignore all conditional processing commands (.IF, .IFNOT, .ELSE, .ENDIF) in the input file. DSR includes both "true" and "false" conditional information in the output file along with formatted text. See the *VAX DIGITAL Standard Runoff (DSR) Reference Manual* for further details on the .IF, .IFNOT, .ELSE, .ENDIF, and .VARIABLE commands and the /VARIANT qualifier.
CONTENTS	Causes DSR to output all .SEND TOC commands along with the text being sent to the table of contents.

FILES	Causes DSR to output all .REQUIRE commands as well as the text of the require files.
INDEX	Causes DSR to output the indexing commands, .INDEX and .ENTRY, in addition to the text to which they refer.
SAVE_RESTORE	Causes DSR to output all .SAVE and .RESTORE commands.

If you specify more than one option, separate them with commas and enclose the list in parentheses. If you specify /DEBUG without specifying any options, ALL is assumed.

/DEVICE=(option[,...])

Positional qualifier.

Controls whether DSR generates an output file (LNI) that is suitable for printing on an LN01, LN01E, or an LN03 laser printer. If you do not get the output that you expect when you print an LNI file on an LN01 or an LN01E, check with your system manager. Appendix B in the *VAX DIGITAL Standard Runoff (DSR) Reference Manual* contains information for system managers about setting LN01 and LN01E laser printers to print LNI files.

You can choose options from the following list to indicate output device, page orientation, and type of emphasis for flagged characters in your LNI file:

Device Options	You Must Choose One of These.
LN01	Produces an output file that is suitable for printing on an LN01 laser printer. The output file name is the same as the input file name and the default file type is LNI. A paper size of 8 1/2 by 11 inches is the default.
LN01E	Produces an output file that is suitable for printing on an LN01E laser printer using the standard European paper size (A4). The output file name is the same as the input file name and the default file type is LNI.
LN03	Produces an output file that is suitable for printing on an LN03 laser printer. The output file name is the same as the input file name, and the default file type is LNI.

Emphasis Options	The Default Is ITALIC.
ITALIC	This keyword causes the italic and bold-italic fonts to be loaded into the LN01 printer. The specific font used depends on the type of emphasis you specify in your input file. Characters that you have marked with the DSR underline flag will be italicized by default or if you specify this option. The LN03 requires no loading of fonts since default fonts are present. Text flagged for emphasis is printed italic if the current font has the ITALIC attribute; otherwise the flagged text is underlined.

Emphasis Options	The Default Is ITALIC.
UNDERLINE	This keyword causes the text and bold fonts to be loaded into the LN01. The characters you have flagged with the DSR underline flag will be underlined. The LN01 allows only 63 underline **segments** per line. You create an underline segment each time you start and stop an underlining sequence. For example, if you want to underline individual words and not underline the spaces between them, you will only be able to underline 63 words (segments) per line. DSR does not report an error if the user exceeds this limit of the hardware. On an LN03, if you have specified the underlining option the flagged text is underlined. The printer does not default to italic even if the current font has the ITALIC attribute.

Orientation Options	The Default Is PORTRAIT.
PORTRAIT	This keyword causes the appropriate fonts for portrait mode to be loaded into the LN01. The characters are printed as they are in a standard business letter. The page will be 8 1/2 inches in width and 11 inches in length. PORTRAIT mode is the default when you specify /DEVICE=LN01, /DEVICE=LN01E, or /DEVICE=LN03.
LANDSCAPE	This keyword causes the appropriate fonts for landscape mode to be loaded into the LN01. The page is 11 inches in width and 8 1/2 inches in length. You must set the margins and page size in your input file to appropriate values for LANDSCAPE mode. You must specify the LANDSCAPE option to override PORTRAIT mode, which is the default.

/DOWN[=n]
/NODOWN (default)

Positional qualifier.

Controls whether DSR inserts a specified number of blank lines at the top of each page. These blank lines precede any header information. The number of blank lines you specify (n) does not affect in any way the number of text lines on a page.

For example, if you specify /DOWN=10 with a .PAGE SIZE of 58 lines, up to 58 lines of text will be output after 10 blank lines.

If you specify the /DOWN qualifier without a value, five blank lines are inserted. If you specify /DOWN=0 or omit the qualifier, no blank lines are inserted, except those associated with the print device or header layout.

/FORM_SIZE=n

Helps to control the maximum number of lines that can be output on a page. Lines for header information, footnotes, and page numbers must be included in this maximum number. When used with /SIMULATE, /FORM_SIZE controls the physical size of the page by putting out line feeds to match the number specified by n. When used with /NOSIMULATE, /FORM_SIZE=n causes DSR to suppress the form feed that DSR would normally insert at the line number specified by the value n. If the number of lines that DSR is going to put on any given page does not match the value of n, a formfeed character will be written into the output file.

The default value for n is derived from the VAX Run-Time Library (RTL) routine LIB$LP_LINES. This will default to 66 unless the logical SYS$LP_LINES is defined, in which case, the assigned value will be used. You can change the default value by specifying a different value for /FORM_SIZE=n.

/INTERMEDIATE[=file-spec]
/NOINTERMEDIATE (default)

Positional qualifier.

Controls whether DSR generates an intermediate output file that can be used as input to the DSR table of contents utility and the DSR indexing utility. See RUNOFF/CONTENTS and RUNOFF/INDEX for further information on producing tables of contents and indexes.

If you specify /INTERMEDIATE, DSR creates an output file that has the same file name as the input file and a file type of BRN. You can rename the output file by supplying a file specification that is different from the default values.

/LOG
/NOLOG (default)

Controls whether DSR displays the following information at the terminal on completion of processing:

- DSR version number
- number of diagnostic messages reported
- number of output pages generated
- output file specification

If there are errors in processing, DSR outputs a message to the terminal even if the /NOLOG qualifier is specified.

/MESSAGES=(option[,...])

Positional qualifier.

Specifies the destination of all DSR error messages. You can indicate a specific destination by using one or both of the following options:

OUTPUT	Messages are sent to the output MEM file
USER	Messages are displayed on the terminal (SYS$ERROR)

If you specify both options, separate them with commas and enclose the list in parentheses.

The default, /MESSAGES=(OUTPUT,USER), sends messages to the output MEM file and displays them on the terminal.

/OUTPUT[=file-spec]
/NOOUTPUT

Positional qualifier.

Specifies where DSR sends the output. If you specify /OUTPUT without a file specification, or if you omit the qualifier, the output file name is the same as the input file name. The output file type depends on the input file type. For a list of input file types and the associated output file types, see the *VAX DIGITAL Standard Runoff (DSR) Reference Manual*. The default input file type is RNO and the default output file type is MEM.

You can change the name of the output file by supplying a file specification for the value file-spec.

/OUTPUT=SYS$OUTPUT causes output to be sent to the terminal rather than to a disk file. You can use the value SYS$OUTPUT when you are logged in to a hard-copy terminal device of the "daisy wheel" type.

The /NOOUTPUT qualifier suppresses the creation of an output file. Using the /NOOUTPUT qualifier with the /INTERMEDIATE qualifier causes DSR to produce only an intermediate BRN file and not a formatted output file.

You can also use /NOOUTPUT to check an input file for errors without using system resources to generate a formatted output file.

/PAGES=string

Positional qualifier.

Tells DSR to output only the pages within a specified range. You indicate a range of page numbers by specifying starting and ending page numbers that are separated by a colon (/PAGES=20:40).

When specifying more than one range of page numbers, separate each range with a comma and enclose the list in quotation marks. The maximum number of ranges you can specify is five (/PAGES="2-9:2-12, 4-1:4-10, 5-9:5-9, A-1:A-3, Index-1:Index-5").

If you specify only a starting page number, output begins at the specified page and continues to the end of the file. To output a single page, the start range and end range must be the same (/PAGES=5:5).

For an entire appendix, only a letter is required (for example, /PAGES="A"). For an entire index, only the word "INDEX" is required (/PAGES="INDEX"). You can refer to specific appendix or index pages with a numeric suffix such as INDEX-10.

Note that the /PAGES qualifier does not recognize any display modes. You must specify the default form of page numbers (5-15) rather than any special form you may have specified with the .DISPLAY NUMBER command (V-15). For details on .DISPLAY NUMBER, see the *VAX DIGITAL Standard Runoff (DSR) Reference Manual*.

By default, DSR outputs all pages.

/PAUSE
/NOPAUSE (default)

Controls whether DSR pauses after printing each page of output.

You can use the /PAUSE qualifier to insert single sheets of paper or reproduction masters into hardcopy output devices. When output is halted, the terminal bell rings to remind you to insert a new form. Press the space bar to resume processing.

Do not use this qualifier in a batch job.

/REVERSE_EMPHASIS

Positional qualifier.

Directs DSR to change the order in which flagged text is underlined on an output device. If you use this qualifier, the printer first prints the characters to be underlined, then issues a carriage return without a line feed, and prints the underscores to underline the flagged text. If you view your file on the terminal, the flagged text is overwritten by the underline character.

/RIGHT[=n]
/NORIGHT (default except for LN01)

Positional qualifier.

Causes the text on each page (including header information) to be shifted to the right the number of columns specified by the value n. Note that these columns are not deducted from the page width specified in the input file.

If you specify /RIGHT without specifying a number, text is shifted to the right five spaces. If you specify a value of zero or omit the qualifier, no shift occurs.

/SEPARATE_UNDERLINE["character"]

Positional qualifier.

Causes text marked with the DSR underline flag to be underlined by hyphens on the next line instead of by underscores on the same line. You can specify a character to replace the hyphen (the default character for separate underlining). You must specify the replacement character as either a character enclosed in quotation marks or as an octal, decimal, or hexadecimal value for the desired character.

Do not use this qualifier with the /[NO]UNDERLINE_CHARACTER qualifiers. The functions of /SEPARATE_UNDERLINE and /[NO]UNDERLINE_CHARACTER are mutually exclusive.

/SEQUENCE
/NOSEQUENCE (default)

Positional qualifier.

Controls whether DSR generates input file sequence numbers in the output file.

For editors that generate line numbers in the input file, the /SEQUENCE qualifier causes similar numbering to appear in the output file. The line numbers appear in the left margin at the beginning of each line of output.

If the text editor does not generate sequential numbers in the input file, sequential numbers are still generated in the output file, but without leading zeros.

/SIMULATE
/NOSIMULATE (default)

Controls whether DSR uses line feeds or form feeds to advance to the top of each page.

For devices that do not have a formfeed capability, use /SIMULATE to put out enough blank lines to cause a skip to the top of each new page. The /SIMULATE qualifier also causes a pause before the first page of output. To continue after the pause, press the space bar.

/UNDERLINE_CHARACTER[="character"]
/NOUNDERLINE_CHARACTER

Positional qualifier.

Causes text marked with the DSR underline flag to be underlined before a new line is processed. The default underline character is an underscore (_). You can specify a character to replace the default underline character. You must specify the replacement character as either a character enclosed in quotation marks or as an octal, decimal, or hexadecimal value for the desired character.

Do not use this qualifier with the /SEPARATE_UNDERLINE qualifier. The functions of /[NO]UNDERLINE_CHARACTER and /SEPARATE_UNDERLINE are mutually exclusive.

/VARIANT=string

Positional qualifier.

Controls the processing of the conditional commands (.IF, .IFNOT, .ELSE, and .ENDIF) by specifying the names of the segments to be processed. See the *VAX DIGITAL Standard Runoff (DSR) Reference Manual* for descriptions of the conditional commands.

If you specify multiple names in a string, you must separate them by commas and enclose the string in quotation marks.

EXAMPLES

```
1  $ RUNOFF CHAPT1.RNO
```

The RUNOFF command takes the input file, CHAPT1.RNO, and writes formatted output to the file CHAPT1.MEM.

2
```
$ RUNOFF CHAPT1/RIGHT=10,CHAPT2
```

This RUNOFF command produces a CHAPT1.MEM file with margins ten spaces to the right of the margins specified in the input file CHAPT1.RNO. It also generates a CHAPT2.MEM file whose margins are not affected by the /RIGHT=10 qualifier.

3
```
$ RUNOFF/OUTPUT=SYS$OUTPUT TEXT.DAT,INTRO
```

This command sends output to the terminal rather than to a disk file. The qualifier applies to both the input files, TEXT.DAT and INTRO.RNO.

4
```
$ RUNOFF/NOOUTPUT/INTERMEDIATE -
$_CHAPT1,CHAPT2,CHAPT3,CHAPT4,CHAPT5/LOG
```

This RUNOFF command generates intermediate BRN files for each of the input files. The BRN files are used as input for the DSR table of contents program, and for the DSR indexing program. The /NOOUTPUT qualifier suppresses the generation of formatted text files for each input file. The /LOG qualifier produces a termination message after RUNOFF processes each file.

RUNOFF/CONTENTS

Invokes the DIGITAL Standard Runoff (DSR) table of contents utility to create an RNT file that can be processed by DSR to make a table of contents. The input file for this command is an intermediate binary file (BRN) that is produced with the RUNOFF command and the /INTERMEDIATE qualifier (see the RUNOFF command). This description provides a functional overview of the RUNOFF /CONTENTS command, emphasizing DCL syntax and grammar. For a complete functional description of the DSR table of contents utility, see the *VAX DIGITAL Standard Runoff (DSR) Reference Manual*.

FORMAT

RUNOFF/CONTENTS *file-spec[,...] or file-spec[+...]*

restrictions

None.

PARAMETER

file-spec[,...] or file-spec[+...]

Specifies one or more intermediate binary files (BRN) that contain information (chapter titles, header levels, sections, and so on) for making a table of contents. To create a BRN file, use the RUNOFF command with the /INTERMEDIATE qualifier. See the RUNOFF command for more information on the /INTERMEDIATE qualifier.

If you omit the input file type, the DSR table of contents utility uses a default file type of BRN. RUNOFF/CONTENTS will also process BTC files that the previous version of DSR produced.

For single input files, the DSR table of contents utility produces an output file having the same file name as the input file. The output file type is RNT.

If you separate multiple input files with commas, the table of contents utility produces a separate RNT file for each input file. If you separate multiple input files with plus signs (+), the table of contents utility produces a single RNT file that contains table of contents information for all of the input files. The default output file name is the same as the first input file name, and the default file type is RNT.

Wildcard characters are not allowed in the file specification.

DESCRIPTION

The RUNOFF/CONTENTS command allows you to create an RNT file that can be processed by DSR to produce a table of contents. Qualifiers to this command allow you to specify the following characteristics for table of contents entries:

- Bolding or underlining of entries
- Deepest header level to be included
- Running page numbers or chapter-oriented page numbers
- Section numbers displayed or not displayed

QUALIFIERS

/BOLD
/NOBOLD (default)

Controls whether the bolding specified in chapter and header titles in the input file appears in the table of contents.

If you specify /BOLD, the text flagged for bolding in the body of the document is marked for overprinting in the finished table of contents.

If you specify /NOBOLD, the text flagged for bolding in the document is not overprinted in the table of contents.

/DEEPEST_HEADER=n

Controls how many levels of header levels are output in the table of contents. You can specify any number of header levels to be displayed by changing the value of n.

The default is /DEEPEST_HEADER=6.

/IDENTIFICATION
/NOIDENTIFICATION (default)

Controls whether the current version number of the DSR table of contents utility is reported.

/INDENT
/NOINDENT (default)

Controls how many spaces the header levels after level 1 are indented in the table of contents.

If you omit this qualifier, or if you specify /NOINDENT, all header levels after header level 1 will be indented 2 spaces.

If you specify /INDENT, each header level after header level 1 will be indented 2 spaces beyond the preceding header level.

/LOG
/NOLOG (default)

Controls whether the DSR table of contents utility displays the following information at the terminal:

- The name of each input file as it is being processed
- The name of each input file after it is processed
- The name of the output file that is created

If there are any errors in processing, the DSR table of contents utility will send messages to the terminal even if /NOLOG is in effect.

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the DSR table of contents utility sends the output.

If you specify the /OUTPUT qualifier without a file specification, or if you omit the qualifier entirely, the DSR table of contents utility creates a file with the same file name as the input file. The default file type is RNT.

You can change the name of the output file by supplying a file specification for the value file-spec.

The /NOOUTPUT qualifier suppresses the creation of an output file. You can use /NOOUTPUT to check an input file for errors without using system resources to generate an output file.

/PAGE_NUMBERS=(option[,...])

Controls whether the table of contents contains running page numbers or chapter-oriented page numbers; also controls how many levels of headers have page references listed in the table of contents. To specify these options, select from the following list:

Option	Purpose
LEVEL=n	Specifies that header levels up to and including header level n have page numbers listed in the table of contents. The default is to display page numbers for 6 levels of headers.
NORUNNING	Specifies chapter-oriented page numbers (such as 1–3, 10–42). You can specify chapter-oriented numbers for the table of contents even if the document does not have chapter-oriented numbers. NORUNNING is the default.
RUNNING	Specifies running page numbers (such as 3, 42). You can specify running page numbers for the table of contents even if the document does not have running page numbers.

If you supply more than one option, separate them with commas and enclose the list in parentheses.

/REQUIRE=file-spec
/NOREQUIRE (default)

Allows you to change or delete the heading on the first page of a table of contents. The default heading is the word CONTENTS, centered on the page, followed by one blank line. You can substitute another word as a heading, or have no heading.

To change the heading, do one of the following:

1 If you do not want any heading, merely specify a null file as the file-spec for /REQUIRE.

```
$ RUNOFF/CONTENTS/REQUIRE=nl:
```

2 If you want to use a different heading, create or edit a file that specifies the heading that you want. Use the file that you create as the file-spec for /REQUIRE.

When you use the /REQUIRE qualifier, the default heading for the first page of the contents is not generated. The file that you are "requiring" must provide the heading. The file can contain both DSR commands that change the format of the first page, and the text that you want to appear at the top of the page. Or the file can contain only DSR commands to format the first page of the contents. For example, you can put the command .FIGURE 10 in the file. This command generates 10 blank lines at the top of the first page of the table of contents. You can use these blank lines for later pasteup.

/SECTION_NUMBERS (default)
/NOSECTION_NUMBERS

Controls whether the DSR table of contents utility displays section numbers in the table of contents. The /SECTION_NUMBERS qualifier displays sections numbers for all header levels in the table of contents. /NOSECTION_NUMBERS suppresses the display of section numbers for all header levels.

/UNDERLINE
/NOUNDERLINE (default)

Controls whether the underlining specified in chapter and header titles in the input file appears in the table of contents.

If you specify /UNDERLINE, the text flagged for underlining in the body of the document is underlined in the table of contents.

If you specify /NOUNDERLINE, the text flagged for underlining in the body of the document is not underlined in the table of contents.

EXAMPLES

1
```
$ RUNOFF/INTERMEDIATE CHPT1,CHPT2,CHPT3
```

Before using RUNOFF/CONTENTS, you must use RUNOFF /INTERMEDIATE to create a BRN file as input for the DSR table of contents utility. The above command line creates three separate files: CHPT1.BRN, CHPT2.BRN, and CHPT3.BRN.

2
```
$ RUNOFF/CONTENTS CHPT1.BRN
```

The RUNOFF/CONTENTS command takes the file CHPT1.BRN as input and creates CHPT1.RNT, which can be processed by DSR to produce a final table of contents for Chapter 1.

3
```
$ RUNOFF/CONTENTS/INDENT/NOSECTION_NUMBERS CHPT2
```

This command takes the file CHPT2.BRN as input and creates CHPT2.RNT. When processed with the RUNOFF command, the RNT file will produce a table of contents in which each header level after header level 1 is indented 2 spaces beyond the preceding header level. The table of contents will not have section numbers listed. See the next example for a sample command line for processing RNT files.

4
```
$ RUNOFF/LOG CHPT2.RNT
```

This command produces CHPT2.MEC, which is a formatted table of contents. You can type or print this file to view the table of contents.

RUNOFF/INDEX

Invokes the DIGITAL Standard Runoff (DSR) indexing utility to create an RNX file that can be processed by DSR to create an index. The input file for this command is an intermediate binary file (BRN) that is produced with the RUNOFF command and the /INTERMEDIATE qualifier (see the RUNOFF command). This description provides a functional overview of the RUNOFF/INDEX command, emphasizing DCL syntax and grammar. For a complete functional description of the DSR indexing utility, see the *VAX DIGITAL Standard Runoff (DSR) Reference Manual*.

FORMAT

RUNOFF/INDEX *file-spec[,...] or file-spec[+...]*

restrictions

None.

PARAMETER

file-spec[,...] or file-spec[+...]

Specifies one or more intermediate binary files (BRN) that contain information (index entries, page number references, and so on) for making an index. To create a BRN file, use the RUNOFF command with the /INTERMEDIATE qualifier. See the RUNOFF command for more information on the /INTERMEDIATE qualifier.

If you omit the input file type, the DSR indexing utility uses a default file type of BRN. RUNOFF/INDEX will also process BIX files that the previous version of DSR produced.

For single input files, the DSR indexing utility produces an output file having the same file name as the input file. The output file type is RNX.

If you separate multiple input files with commas, the indexing utility produces a separate RNX file for each input file. If you separate multiple input files with plus signs (+), the indexing utility produces a single RNX file that contains indexing information for all of the input files. The default output file name is the same as the first input file name and the default file type is RNX.

Wildcard characters are not allowed in the file specification.

DESCRIPTION

The RUNOFF/INDEX command creates an RNX file that can be processed by DSR to produce an index. The formatted index is a 2-column index with balanced columns on each page. This index can be used for draft copies or for final production. Qualifiers to this command allow you to specify the following characteristics for index entries:

- Running page numbers or chapter-oriented page numbers
- The number of lines of index entries per page
- Special text and heading on the first page of the index

QUALIFIERS

/IDENTIFICATION
/NOIDENTIFICATION (default)

Reports the current version number of the DSR indexing utility.

/LINES_PER_PAGE=n

The value n specifies the number of lines of index entries on each page of the finished index. This number does not include the number of lines required for headings and footings.

The default is 55 lines. This value is designed to work properly in the default formatting environment of DSR. You must calculate the value n if you change the default environment in any of the following ways:

- If you use subtitles in the document that requires the RNX file
- If you make the page length for the document anything other than 58 lines per page
- If you use any .LAYOUT other than zero (0)

To calculate the correct value for /LINES_PER_PAGE use the following formula:

```
/LINES_PER_PAGE=n
n = .PAGE SIZE ( the first parameter is length value)
    minus 4 if subtitles are used, minus 3 if no subtitles
    minus the number of lines reserved for .LAYOUT 1,
          .LAYOUT 2, or .LAYOUT 3.
```

/LOG
/NOLOG (default)

Controls whether the DSR indexing utility displays the following information at the terminal upon completion of processing:

- The name of each input file as it is being processed
- The name of each input file after it is processed
- The name of the output file that is created

If there are any errors in processing, INDEX will send messages to the terminal even if /NOLOG is in effect.

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the DSR indexing utility sends the output.

If you specify the /OUTPUT qualifier without a file specification, or if you omit the qualifier entirely, the output file name is the same as the input file name. The default file type is RNX.

You can change the name of the output file by supplying a file specification for the value file-spec.

The /NOOUTPUT qualifier suppresses the creation of an output file. You can use /NOOUTPUT to check an input file for errors without generating an output file.

/PAGE_NUMBERS=option

Controls whether the page number references in the index are running page numbers or chapter-oriented page numbers. To specify the type of page numbers you want, select from the following options:

Option	Purpose
NORUNNING	Specifies chapter-oriented page numbers (such as 1–3, 10–42). You can specify chapter-oriented numbers for an index even if they do not appear in the document. NORUNNING is the default.
RUNNING	Specifies running page numbers (such as 1, 50, 230). You can specify running page numbers for an index even if the document does not display running page numbers.

/REQUIRE=file-spec
/NOREQUIRE (default)

Allows you to change the heading on the first page of an index. The default heading is the word INDEX, centered on the page, followed by three blank lines.

To change the heading:

1. Create or edit a file that specifies the format and the text that you want as the heading on the first index page.
2. Use the file you create as the file-spec for /REQUIRE.

When you use the /REQUIRE qualifier, the default heading for the first page of the index is not generated. Your file must provide the heading. The file can contain both DSR commands and text that you want to appear at the top of the first page of the index, or it can contain solely DSR commands. For example, you can put the DSR command .FIGURE 10 in the file. This command generates 10 lines of white space at the top of the first page of the index. You can use these blank lines for later pasteup. See the *VAX DIGITAL Standard Runoff (DSR) Reference Manual* for a sample file that changes the index heading.

If you are adding lines of text or white space to the heading on the first page of the index, you must allow space for this addition. Use the /RESERVE=n qualifier to provide the space you need. See the /RESERVE qualifier for more information.

/RESERVE=n
/NORESERVE (default)

Allows you to reserve space at the top of the first page of the index for text or white space that you want to include with the /REQUIRE=file-spec qualifier. Determine how many lines of text or white space you are adding to the top of the first page of the index and use this number as the value n for the /RESERVE qualifier.

EXAMPLES

1
```
$ RUNOFF/INTERMEDIATE CHPT1,CHPT2,CHPT3
```

Before using RUNOFF/INDEX, you must create a BRN file as input for the DSR indexing utility. This command creates three separate files: CHPT1.BRN, CHPT2.BRN, and CHPT3.BRN.

2
```
$ RUNOFF/INDEX CHPT1.BRN
```

The RUNOFF/INDEX command takes the file CHPT1.BRN as input and creates CHPT1.RNX, which can be processed by DSR to produce an index for Chapter 1.

3
```
$ RUNOFF/INDEX/LINE_PER_PAGE=52 CHPT2
```

This command takes the file CHPT2.BRN as input and creates CHPT2.RNX. The RNX file will produce an index with 52 lines of index entries per page. It was necessary to adjust the lines per page because the writer used a page layout with the page numbers centered at the bottom of the page (.LAYOUT 1, .LAYOUT 2, .LAYOUT 3) and this takes up three more spaces than .LAYOUT 0, which is the default for DSR. To produce the final index, you must use the RNX file as input to DSR. See the next example.

4
```
$ RUNOFF CHPT2.RNX
```

This command produces CHPT2.MEX, which is a formatted index. You can type or print this file to view the index.

SEARCH

Searches one or more files for the specified string or strings and lists all the lines containing occurrences of the strings.

FORMAT

SEARCH *file-spec[,...] search-string[,...]*

restrictions

None.

PARAMETERS

file-spec[,...]

Specifies the names of one or more files to be searched. You must specify at least one file name. If you specify two or more file names, separate them with commas.

Wildcard characters are allowed in the file specification.

search-string[,...]

Specifies one or more strings to search for in the specified files. If the search string contains any lowercase letters or nonalphanumeric characters (including spaces), enclose it in quotation marks.

You can use the /MATCH and /EXACT qualifiers to alter the way that SEARCH matches search strings.

DESCRIPTION

The SEARCH command enables you to search through files for specific character strings. The system displays all lines containing occurrences of the strings. You can use SEARCH qualifiers to tailor the search operation to your specific needs.

The SEARCH command opens the file with shared read and write access. Therefore, any file that has its attributes set to shared write will be searched even if it is currently opened by other users.

QUALIFIERS

/EXACT
/NOEXACT (default)

Controls whether the SEARCH command matches the search string exactly, or treats uppercase and lowercase letters as equivalents. The default qualifier, /NOEXACT, causes SEARCH to ignore 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 /EXACT.

/EXCLUDE=(file-spec[,...])

Causes the SEARCH command to exclude the listed file specifications from the search. Do not include a directory or device name in this file specification.

Wildcard characters are supported for file specifications. However, you cannot use relative version numbers to exclude a specific version.

If you use this qualifier, you must include at least one file specification. If you specify only one file, you can omit the parentheses.

If you omit the /EXCLUDE qualifier, SEARCH will examine all files in the input file list.

/FORMAT=option

Controls reformatting of records during output. You can specify one of the following formatting options:

TEXT	Replaces control characters in the 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.
PASSALL	Moves characters to the output device with no translation whatsoever. 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). The terminal driver cannot send eight-bit characters to the terminal unless either SET TERMINAL/PASSALL or SET TERMINAL/EIGHT_BIT is already in effect.
DUMP	DUMP format is very similar to TEXT format. However, with DUMP format, all control characters (including <HT>, <CR>, and <LF>) are displayed as ANSI mnemonics.
NONULLS	NONULLS is identical to the DUMP option, except that all null characters are removed 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.

/HEADING (default)
/NOHEADING

Controls whether file names and window separators are printed in the output. With the default heading format, file names are printed only when more than one file is specified or when wildcard characters are used. The separator, a line of 30 asterisks, is displayed between groups of lines that belong to different files. If you specify the /WINDOW qualifier, a line of 15 asterisks separates each window within a file.

/LOG
/NOLOG (default)

Controls whether the SEARCH command produces a line containing the file name and the number of records and matches for each file searched. The log information is output to the current SYS$OUTPUT device.

/MATCH=option

Indicates how the SEARCH command matches multiple search strings. The /MATCH qualifier has four options:

OR	A match occurs if a record contains any of the search strings.
AND	A match occurs if and only if all of the search strings are present in the single record.
NOR	The negation of AND. A match occurs only if the record does not contain any of the search strings.
NAND	The negation of OR. A match occurs only if the record does not contain all of the search strings.

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.

/NUMBERS
/NONUMBERS (default)

Controls whether the source line number is displayed at the left margin of each line.

/OUTPUT[=file-spec]
/NOOUTPUT

Controls whether the results of the search are output to a specified file. The output will be 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.

/REMAINING
/NOREMAINING (default)

Controls whether the SEARCH command displays all records from the first matched record to the end of the file. The /REMAINING qualifier overrides the /WINDOW=n2 qualifier. However, you can still use the /WINDOW=n1 qualifier to specify the number of lines to be printed above the line containing the matched record.

/STATISTICS
/NOSTATISTICS (default)

Controls whether SEARCH displays statistics about the search. The statistics displayed are:

- 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

/WINDOW[=(n1,n2)]
/NOWINDOW (default)

Controls the number of lines that are listed along with the line containing the matching string.

If you specify the /WINDOW qualifier with a single number n, n–1 additional lines are displayed with each line containing the search string. Half of the additional lines are listed above the line containing the match, and half are listed below. If n-1 is an odd number, the extra line is listed below the 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, making a total of ten lines.

If you specify /WINDOW without specifying a number, the default number of five lines—two above, one containing the search string, and two below—is used.

If the form /WINDOW=(n1,n2) is used, n1 refers to the number of lines above the matched line and n2 refers to the number of lines below. Either of these numbers can be zero.

If /WINDOW=0 is specified, SEARCH will display the file name of each file containing a match, but no records. You can use this specification to create 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 entirely, only the line in which the match is satisfied is displayed.

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 will effectively copy 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 will be ready to use, since it will not contain headings and window separators.

5
```
$ SEARCH/OUTPUT=SWAP.LIS/FORMAT=PASSALL/NUMBERS/EXACT -
$_ /WINDOW=10000 SWAP.PAS SWAP
```

The effect of this SEARCH command is to produce a listing file with the line numbers at the left margin. The /FORMAT=PASSALL qualifier is specified so that formfeed characters in the source will be passed through. The /EXACT qualifier is specified for efficiency (since it is known that the name SWAP in the program statement is always in uppercase). The /WINDOW qualifier is issued so that the entire file will be 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 VAX
```

The SEARCH command searches through the files SUB.DAT and DATA.LIS at remote node OMAHA for all occurrences of the string VAX. The list of all records containing the string VAX is displayed at the local terminal.

SET

Defines or changes, for the current terminal session or batch job, characteristics associated with files and devices owned by the process.

FORMAT

SET *option*

restrictions

Restrictions are noted in the command descriptions for each SET option.

DESCRIPTION

The SET command options are described individually in this manual. Table DCL–13 lists all the SET command options, including those that are generally reserved for use by system operators and managers.

Table DCL–13 SET Command Options

Option	Function
ACCOUNTING	Initializes the accounting log file
ACL	Associates an Access Control List with one or more system objects
AUDIT	Enables or disables forms of security auditing
BROADCAST	Determines which messages will be broadcast to SYS$OUTPUT
CARD_READER	Defines the default ASCII translation mode for a card reader
CLUSTER/QUORUM	Sets the cluster quorum to a value that you specify, or if no value is specified, sets the cluster quorum 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
[NO]CONTROL	Disables/enables interrupts caused by CTRL/T or CTRL/Y
DAY	Overrides the default day type specified in the user authorization file (UAF)
DEFAULT	Establishes a device and/or directory as the current default for file specifications
DEVICE	Defines device characteristics
DEVICE/SERVED	Allows you to make a disk on a local node available to all the nodes on a VAXcluster
DIRECTORY	Modifies the characteristics of one or more directories
FILE	Modifies the characteristics of one or more files
FILE/ACL	Modifies the access control list of one or more files

Table DCL–13 (Cont.) SET Command Options

Option	Function
HOST	Connects the user's terminal through the current host processor to a remote VAX processor
HOST/DTE	Connects your system to a remote system via an outgoing terminal line
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
[NO]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	Allows users to change their own passwords; enables system managers to change the system password
PRINTER	Defines characteristics of a printer
PROCESS	Defines execution characteristics of the current process
PROMPT	Defines a prompt string
PROTECTION	Defines the protection status of a file or group of files
PROTECTION/DEFAULT	Establishes the default protection to be applied to all files subsequently created during the job
PROTECTION/DEVICE	Establishes the protection to be applied to a specific non-file-structured device
QUEUE	Changes the current status or attributes of the specified queue
QUEUE/ENTRY	Changes the current status or attributes of a job that is not currently executing in a queue
RESTART_VALUE	Established a test value for restarting portions of batch jobs
RIGHTS_LIST	Allows users to modify the process rights list; allows privileged users to modify the system rights list
RMS_DEFAULT	Provides default multiblock and multibuffer count values to be used by VAX RMS for file operations
SYMBOL	Controls access to local and global symbols in command procedures
TERMINAL	Defines operational characteristics of a terminal
TIME	Resets the system clock to the specified value
UIC	Changes the UIC of the current process

Table DCL–13 (Cont.) SET Command Options

Option	Function
[NO]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

Disables or enables the logging of various activities in the accounting log file SYS$MANAGER:ACCOUNTING.DAT. You can also use SET ACCOUNTING to close the current accounting log file and to open a new one with a version number incremented by 1.

FORMAT

SET ACCOUNTING

restrictions

Requires operator (OPER) privilege.

PARAMETERS

None.

DESCRIPTION

Disables or enables the logging of the specified activities recorded in the accounting log file. If you specify only /DISABLE, the logging of all activities is disabled. If you specify only /ENABLE, the logging of all activities is enabled.

For a detailed description of the accounting log file records, see the discussion of the Send Message to Accounting Manager (_$SNDACC) system service in the *VAX/VMS System Services Reference Manual.*

QUALIFIERS

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

Disables the logging of all activities in the accounting log file. To disable specific activities selectively, you include one or more keywords with /DISABLE. When you specify more than one keyword, separate them with commas and enclose the entire list in parentheses. Table DCL–14 lists and describes the activities that can be disabled.

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

Enables the logging of all activities in the accounting file. To enable specific activities selectively, you include one or more keywords with /ENABLE. When you specify more than one keyword, separate them with commas and enclose the entire list within parentheses. Table DCL–14 lists and describes the activities that can be enabled.

/NEW_FILE

Closes the current accounting file and opens a new version of that file.

Table DCL–14 SET ACCOUNTING/DISABLE and SET ACCOUNTING/ENABLE Keywords

Keyword	Function
BATCH	Inhibits/allows the recording of batch job termination
DETACHED	Inhibits/allows the recording of detached process termination
IMAGE	Inhibits/allows the recording of image activation
INTERACTIVE	Inhibits/allows the recording of interactive job termination
LOGIN_FAILURE	Inhibits/allows the recording of login failures
MESSAGE	Inhibits/allows the recording of user messages
NETWORK	Inhibits/allows the recording of network job termination
PRINT	Inhibits/allows the recording of all print jobs
PROCESS	Inhibits/allows the recording of all process termination
SUBPROCESS	Inhibits/allows the recording of all subprocess termination

EXAMPLES

1
```
$ SET ACCOUNTING/ENABLE=(BATCH,INTERACTIVE)
```

This command requests that all batch and interactive jobs be recorded in the accounting file at job termination.

2
```
$ SET ACCOUNTING/NEW_FILE
```

This command closes the current accounting file and creates a new version of it.

SET ACL

Allows you to modify the access control list (ACL) of an object.

FORMAT

SET ACL *object-name*

restrictions

None.

PARAMETERS

object-name

Specifies the object whose access control list (ACL) is being modified. Wildcard characters are not allowed in the object names. Each file must be a disk file on a Files–11 Structure Level 2 formatted volume.

DESCRIPTION

The SET ACL command enables you to manipulate an entire access control list (ACL) of an object, or to create, modify, or delete access control entries (ACEs) in the ACL of an object. (For information on the format of ACEs and ACLs, see Section 7.2 of the *VAX/VMS DCL Concepts Manual*.) To use the SET ACL command, you specify the object name of the object whose ACL you want to manipulate.

The SET ACL command is used to add ACEs to an ACL by specifying the ACEs with the /ACL qualifier. For example, the following command adds an ACE to the ACL of the file SALARY85.DAT so that all users associated with the identifier PERSONNEL are allowed read access to the file:

```
$ SET ACL/ACL=(IDENTIFIER=PERSONNEL,ACCESS=READ) SALARY85.DAT
```

If the object specified with the SET ACL command does not have an ACL, one is created.

The SET ACL command provides the following qualifiers to manipulate ACEs and ACLs in various ways:

/AFTER
/DELETE
/LIKE
/NEW
/REPLACE

You can delete ACEs from an ACL by including the /DELETE qualifier and specifying the ACEs with /ACL. To delete all the ACEs (except those with the PROTECTED option), include the /DELETE qualifier and specify the /ACL qualifier without specifying any ACEs.

You can copy an ACL from one object to another by using the /LIKE qualifier. When using the /LIKE qualifier, you must specify the object type and object name. The ACL of the object specified with /LIKE replaces the ACL of the object given with the command.

You can replace existing ACEs in the ACL of the object specified with the command by using the /REPLACE qualifier. Any ACEs specified with /ACL are deleted and replaced by those specified with /REPLACE.

The /NEW qualifier is used to delete all ACEs (except those with the PROTECTED option) before adding any ACEs specified by /ACL, /LIKE, or /REPLACE.

When referring to existing ACEs with /DELETE, /REPLACE, or/AFTER, the existing ACE may be abbreviated.

By default, any ACEs, except security alarm ACEs, added to an ACL are placed at the top of the ACL. Security alarm ACEs are always positioned at the top of the ACL, regardless of positioning qualifiers. Whenever the system receives a request for access to an object that has an ACL, the system searches each entry in the ACL from the first to the last for the first match it can find, and then stops searching. If another match occurs further down in the ACL, it will have no effect. Since the position of an ACE in an ACL is so important, you can use the /AFTER qualifier to correctly position an ACE. When you use the /AFTER qualifier, any additional ACEs will be added after the specified ACE.

The SET ACL command can also be used with the /EDIT qualifier to invoke the ACL editor. When the /EDIT qualifier is specified, only one object name is allowed. The following qualifiers can be used only when the /EDIT qualifier has been specified.

/JOURNAL
/KEEP
/MODE
/RECOVER

For more information on these qualifiers and the ACL editor in general, see the *VAX/VMS Access Control List Editor Reference Manual.*

QUALIFIERS

/ACL[=(ace[,...])]

Specifies one or more access control entries (ACEs) to be modified. When no ACE is specified, the entire access control list is affected. Separate multiple ACEs with commas. The specified ACEs are inserted at the top of the ACL unless the /AFTER qualifier is given.

/AFTER=ace

Indicates that all access control entries (ACEs) specified with the /ACL qualifier will be added after the ACE specified with the /AFTER qualifier. By default, any ACEs added to the ACL are always placed at the top of the list.

This qualifier cannot be used with the /EDIT qualifier.

/BEFORE[=time]

Selects a file that is dated before the specified time. You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

This qualifier cannot be used with the /EDIT qualifier and can be used only with an object that is a file.

/BY_OWNER[=uic]

Selects a file whose owner user identification code (UIC) matches the specified owner UIC. If the /BY_OWNER qualifier is specified without a UIC, the UIC of the current process is assumed.

Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

This qualifier cannot be used with the /EDIT qualifier and can be used only with an object that is a file.

/CONFIRM
/NOCONFIRM (default)

Controls whether a request is issued before each individual SET ACL operation to confirm that the operation should be performed on that object.

When the system issues the prompt, you can issue any of the following responses:

YES	NO	QUIT
TRUE	FALSE	CTRL/Z
1	0	ALL
	<RET>	

You can use any combination of upper- and lowercase letters for word responses. Word responses can be abbreviated to one or more letters (for example, T, TR, or TRU for TRUE). Affirmative answers are YES, TRUE, and 1. Negative answers are NO, FALSE, 0, and <RET> . QUIT or CTRL/Z indicates that you want to stop processing the command at that point. When you respond with 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, the prompt will be reissued.

This qualifier cannot be used with the /EDIT qualifier and can be used only with an object that is a file.

/CREATED

Selects a file based on its creation date. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier.

This qualifier cannot be used with the /EDIT qualifier and can be used only with an object that is a file.

/DEFAULT

Creates an ACL for the specified files as if the files were newly created. For a directory file, the /DEFAULT qualifier propagates the entire ACL (except ACEs with the NOPROPAGATE option) so that a particular access protection can be propagated throughout a directory tree. For all other files, the /DEFAULT qualifier propagates the DEFAULT option ACEs in the ACL of the parent directory to the ACL of the specified files.

The /DEFAULT qualifier uses the ACL of the parent directory of the specified file, not the current default directory.

This qualifier cannot be used with the /EDIT qualifier and can be used only with an object that is a file.

/DELETE

Indicates that the access control entries (ACEs) specified with the /ACL qualifier are to be deleted. If no ACEs are specified with /ACL, the entire ACL is deleted (except those with the PROTECTED option). If you specify an ACE that was not specified with the /ACL qualifier, you will be notified that the ACE does not exist, and the delete operation will continue.

This qualifier cannot be used with the /EDIT qualifier.

/EDIT

Invokes the ACL Editor and allows you to use the /JOURNAL, /KEEP, /MODE, or /RECOVER qualifiers. Any other qualifiers specified with /EDIT are ignored.

For more information on the ACL Editor, see the *VAX/VMS Access Control List Editor Reference Manual*.

/EXCLUDE=(file-spec[,...])

Excludes any files that match the listed file specifications from the SET ACL operation. Wildcard characters are allowed in the file specification. However, you cannot use relative version numbers to exclude a specific version. The file specification can contain a directory specification, but it cannot contain a device name.

This qualifier cannot be used with the /EDIT qualifier and can be used only with an object that is a file.

/JOURNAL[=file-spec]
/NOJOURNAL

Controls whether a journal file is created from the editing session. By default, a journal file is created if the editing session ends abnormally.

If you omit the file specification, the journal file has the same name as the input file and a file type of JOU. You can use the /JOURNAL qualifier to specify a journal file name that is different from the default. No wildcard characters are allowed in the /JOURNAL file-spec parameter.

You must specify /EDIT in order to use this qualifier.

/KEEP=(option[,...])

Determines whether the journal file or the recovery file will be deleted when the editing session ends. The options are:

- JOURNAL—saves the journal file for current editing session
- RECOVER—saves the journal file used for restoring the ACL

You can shorten the keywords JOURNAL and RECOVER to J and R, respectively. If you specify only one option, you can omit the parentheses.

You must specify /EDIT in order to use this qualifier.

/LIKE=(OBJECT_TYPE=type,OBJECT_NAME=name)

Indicates that the ACL of the object-name given with the /LIKE qualifier is to replace the ACL of the objects specified with SET ACL. Any existing ACE (except those with the PROTECTED option) will be deleted before the ACL

specified by /LIKE is copied. The type of the source and destination objects do not have to be the same.

You can specify the following keywords for OBJECT_TYPE: DEVICE, FILE, SYSTEM_GLOBAL_SECTION, GROUP_GLOBAL_SECTION, or LOGICAL_NAME_TABLE. The object-name is specified as it is specified for the command. No wildcard characters are allowed in the /LIKE parameters.

This qualifier cannot be used with the /EDIT qualifier.

/LOG
/NOLOG (default)

Controls whether the SET ACL command displays the object name of the object that has been affected by the command.

This qualifier cannot be used with the /EDIT qualifier.

/MODE=[NO]PROMPT

Determines whether the ACL editor prompts for field values. By default, the ACL editor selects prompt mode.

You must specify the /EDIT qualifier to use this qualifier.

/NEW

Indicates that any existing ACE in the ACL of the object specified with SET ACL (except those with the PROTECTED option) is to be deleted. In order to use the /NEW qualifier, you must specify a new ACL or ACE with the /ACL, /LIKE, or /REPLACE qualifier.

This qualifier cannot be used with the /EDIT qualifier.

/OBJECT_TYPE=type

Specifies the type of the object whose ACL is being edited. By default, the ACL editor assumes that the object whose ACL is being edited is a file. If the object is not a file, the /OBJECT qualifier is required. The following keywords may be specified with /OBJECT:

FILE	Specifies that the object type is a file or a directory file.
DEVICE	Specifies that the object type is a device.
SYSTEM_GLOBAL_SECTION	Specifies that the object type is a system global section.
GROUP_GLOBAL_SECTION	Specifies that the object type is a group global section.
LOGICAL_NAME_TABLE	Specifies that the object type is a system logical name table.

By default, a file object type is assumed, and therefore the /OBJECT_TYPE=file need not be specified.

/RECOVER[=file-spec]
/NORECOVER (default)

Specifies the name of the journal file to be used in a recovery operation. If the file specification is omitted with /RECOVER, the journal is assumed to have the same name as the input file and a file type of JOU. No wildcard characters are allowed with the /RECOVER file-spec parameter.

You must specify /EDIT in order to use this qualifier.

/REPLACE=(ace[,...])

Deletes the access control entries (ACEs) specified with the /ACL qualifier and replaces them with those specified with /REPLACE. Any ACEs specified with the /ACL qualifier must exist and must be specified in the order in which they appear in the ACL.

This qualifier cannot be used with the /EDIT qualifier.

/SINCE[=time]

Selects a file that is dated after the specified time. You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

This qualifier cannot be used with the /EDIT qualifier and can be used only with an object that is a file.

EXAMPLE

```
$ SET ACL/LIKE=(OBJECT_TYPE=FILE,OBJECT_NAME=USER.LIS) ACCOUNTS.LIS
```

This example replaces the ACL of the file ACCOUNTS.LIS with the ACL for the file USER.LIS.

SET AUDIT

Enables or disables security auditing on a VAX/VMS system. (Note that you must specify the /ALARM qualifier.)

FORMAT

SET AUDIT

restrictions

Requires the SECURITY system privilege.

PARAMETERS

None.

DESCRIPTION

The SET AUDIT/ALARM command enables security auditing to send alarms to terminals that have been enabled as security operators whenever specified events are detected by the system. For example, you can use a SET AUDIT command to cause an alarm message to occur if breakin attempts are observed or if the system or network UAF file is modified.

For details on how to enable a terminal for security alarms, see the REPLY /ENABLE command.

The results of a SET AUDIT command can be displayed with the DCL command SHOW AUDIT.

Since the security auditing features entail a certain amount of system overhead, you should apply some discretion in selecting the features that will provide the most benefit in your work environment. In particular, be aware that enabling alarms for all events can result in a large number of alarm messages being sent to a security terminal. This situation will affect the use of the security terminal, because alarm messages have priority over any other I/O from a terminal.

QUALIFIERS

/ALARM

Causes alarm messages to be sent to all terminals enabled as security operators. See the description of the DCL command REPLY/ENABLE for details on how to enable terminals as security operators. Both /ALARM and either /ENABLE or /DISABLE are required.

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

Indicates that one or more events should not result in any alarm.

You can disable alarms for all events by specifying the keyword ALL, or you can specify the appropriate keywords to selectively disable alarms for from one to all events that are currently enabled. You must specify at least one keyword. See the /ENABLE qualifier description for a list of the keywords to use with the /DISABLE qualifier.

In processing the SET AUDIT command, the system processes the /DISABLE qualifier last. Thus, if you accidentally specify both /ENABLE and /DISABLE in the same command, the /DISABLE qualifier prevails.

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

Indicates that the particular event specified in the keyword list should result in an alarm.

You can enable alarms for all events by specifying the keyword ALL, or you can specify the appropriate keywords to selectively enable alarms for from one to all events that are currently enabled. You must specify at least one keyword.

The possible events that may be specified in the keyword list of either the /ENABLE or /DISABLE qualifier are:

ACL	An event requested by an access control list (ACL) item, including ACLs on files and global sections. Access control lists are described in Section 7.2 of the *VAX/VMS DCL Concepts Manual*.
ALL	All possible events.
AUDIT	An event resulting from the execution of a SET AUDIT command.
AUTHORIZATION	The modification of any portion of the system or network user authorization file (UAF), including any password changes; the modification of any portion of the rights database.
BREAKIN=(keyword[,...])	The occurrence of one or more of the following classes of breakin attempts, as specified by one or more of the keywords:

ALL	All possible sources of breakins, as defined by the remaining keywords
DETACHED	Detached process breakin attempt
DIALUP	Dialup breakin attempt
LOCAL	Local breakin attempt
NETWORK	Network server breakin attempt
REMOTE	Remote breakin attempt

FILE_ACCESS=(keyword[,...])

The occurrence of file and global section access events (regardless of the value specified in the file's access control list, if any). You can specify one or more of the following keywords to describe the file access event to be noted.

ALL — All types of file access events, as defined by the remaining keywords.

BYPASS [:access [,access...]] — Successful file access due to the use of the BYPASS privilege

FAILURE [:access [,access...]] — Unsuccessful file access

GRPPRV [:access [,access...]] — Successful file access due to the use of the GRPPRV privilege

READALL [:access [,access...]] — Successful file access due to the use of the READALL privilege

SUCCESS [:access [,access...]] — Successful file access

SYSPRV [:access [,access...]] — Successful file access due to the use of the SYSPRV privilege

Most of the keywords permit you to define the type of file access that was obtained with the following keywords:

ALL — All types of file access events, as defined by the remaining keywords. If no access types are specified, ALL is assumed by the system.

READ — Read access

WRITE — Write access

EXECUTE — Execute access

DELETE — Delete access

CONTROL — Owner access

INSTALL

The occurrence of any INSTALL operations.

LOGFAILURE=(keyword[,...])

The occurrence of one or more of the following classes of login failure, as specified by one or more of the keywords:

ALL	All possible types of login failures, as defined by the remaining keywords
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
SUBPROCESS	Subprocess login failure

LOGIN=(keyword[,...])

The occurrence of one or more of the following classes of login attempts, as specified by one or more of the keywords:

ALL	All possible sources of logins, as defined by the remaining keywords
BATCH	Batch process login
DETACHED	Detached process login
DIALUP	Dialup interactive login
LOCAL	Local interactive login
NETWORK	Network server task login
REMOTE	Interactive login from another network node, for example, with a SET HOST command
SUBPROCESS	Subprocess login

LOGOUT=(keyword[,...])	The occurrence of one or more of the following classes of logouts, as specified by one or more of the keywords:	
	ALL	All possible sources of logouts, as defined by the remaining keywords
	BATCH	Batch process logout
	DETACHED	Detached process logout
	DIALUP	Dialup interactive process logout
	LOCAL	Local interactive process logout
	NETWORK	Logout by a network server task
	PROCESS	Subprocess or detached process logout
	REMOTE	Logout of a process that logged in interactively from another network node
MOUNT	The issuing of a MOUNT or DISMOUNT request	

EXAMPLES

1

```
$ SET AUDIT/ALARM/ENABLE=ALL
$ SHOW AUDIT
Security alarms currently enabled for:
    ACL
    MOUNT
    AUTHORIZATION
    BREAKIN:      (DIALUP,LOCAL,REMOTE,NETWORK,DETACHED)
    LOGIN:        (BATCH,DIALUP,LOCAL,REMOTE,NETWORK,SUBPROCESS,DETACHED)
    LOGFAILURE:   (BATCH,DIALUP,LOCAL,REMOTE,NETWORK,SUBPROCESS,DETACHED)
    LOGOUT:       (BATCH,DIALUP,LOCAL,REMOTE,NETWORK,SUBPROCESS,DETACHED)
    FILE_ACCESS:
        FAILURE:  (READ,WRITE,EXECUTE,DELETE,CONTROL)
        SUCCESS:  (READ,WRITE,EXECUTE,DELETE,CONTROL)
        SYSPRV:   (READ,WRITE,EXECUTE,DELETE,CONTROL)
        BYPASS:   (READ,WRITE,EXECUTE,DELETE,CONTROL)
        GRPPRV:   (READ,WRITE,EXECUTE,DELETE,CONTROL)
        READALL:  (READ,WRITE,EXECUTE,DELETE,CONTROL)
```

This SET AUDIT command enables alarms for all possible events; the display from the SHOW AUDIT command identifies the possible events.

2

```
$ SET AUDIT/ALARM/ENABLE=(AUTHORIZATION,BREAKIN)
```

The SET AUDIT command enables alarms at all terminals established as security operators for any change in the system or network user authorization file and for any breakin attempts.

3

```
$ SET AUDIT/ALARM/DISABLE=ALL
```

This command disables all alarms to security operators.

SET BROADCAST

Enables you to selectively screen out various kinds of messages from being broadcast to your terminal.

FORMAT

SET BROADCAST=*(class-name[,...])*

restrictions

SYS$COMMAND must be a terminal.

PARAMETERS

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:

ALL	All message classes enabled
[NO]DCL	CTRL/T and SPAWN/NOTIFY messages
[NO]GENERAL	All normal REPLY messages or messages from $BRDCST
[NO]MAIL	Notification of mail
NONE	All message classes disabled
[NO]OPCOM	Messages issued by OPCOM
[NO]PHONE	Messages from the Phone Utility
[NO]QUEUE	Messages referring to print or batch jobs issued by the queue manager
[NO]SHUTDOWN	Messages issued from REPLY/SHUTDOWN
[NO]URGENT	Messages issued from REPLY/URGENT
[NO]USER1—[NO]USER16	Messages from the specified user groups

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.

EXAMPLES

1

```
$ SET BROADCAST=(NOMAIL, NOPHONE)
.
.
.
$ SET BROADCAST=MAIL
```

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)
```

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 CARD_READER

Defines the default translation mode for cards read from a card reader. All subsequent input read from the specified card reader will be converted using the specified mode.

FORMAT

SET CARD_READER *device-name[:]*

restrictions

None.

PARAMETER

device-name[:]

Specifies the name of the card reader for which the translation mode is to be set. 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.

EXAMPLES

1
```
$ ALLOCATE CR:
  _CRA0: ALLOCATED
$ SET CARD_READER CRA0:/029
$ COPY  CRA0: [MALCOLM.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_READER 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 [MALCOLM.DATAFILES].

SET CLUSTER/QUORUM

Sets the cluster quorum to a value that you specify or, if no value is specified, sets the cluster quorum to a value determined by the system. The /QUORUM qualifier is required.

FORMAT

SET CLUSTER/QUORUM[=quorum-value]

restrictions

- Requires operator (OPER) privilege.

PARAMETERS

None.

DESCRIPTION

The SET CLUSTER/QUORUM command enables you to manually adjust the cluster quorum value. The cluster quorum is based upon an estimate of the total number of votes available in the cluster. It is automatically adjusted upward as new systems join.

You can specify the desired quorum value as part of the SET CLUSTER /QUORUM command string. If you issue the SET CLUSTER/QUORUM command without specifying a value for quorum, the system calculates the value for you, using the formula:

```
(V+2)/2
```

where V is the sum of the votes of all nodes that are currently in the cluster. The system will not allow you to set the quorum to a value less than or equal to the value calculated by the system formula or to a value greater than the number of votes present.

When you issue the SET CLUSTER/QUORUM command without specifying a quorum 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/QUORUM 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, since it is assumed that the node may be rebooted and rejoin the cluster. If a node is removed from the cluster 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 until that node rejoins.

The purpose of the quorum is to eliminate any possibility of the cluster partitioning into separate clusters and simultaneously accessing the same resources (such as HSC50 disks). If the sum of the votes of all members of the cluster is smaller than the cluster quorum, all nodes in the cluster will block activity until new nodes join to increase the vote total. Lowering the quorum value when one or more nodes leave the cluster, reduces the possibility of this happening.

SET CLUSTER/QUORUM

When you issue the SET CLUSTER/QUORUM command, either with or without a quorum value specified, the system will respond with a message indicating the new quorum value that was actually set. Note that you need only issue this command on one node in the cluster, since the new value for the quorum will be propogated through the cluster. This new quorum value should then be stored in the SYSGEN parameter QUORUM on each cluster 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 SYSGEN parameter QUORUM to its original value before bringing the node back to the cluster. Note that you do not need to use the SET CLUSTER/QUORUM command to increase the cluster quorum, since the quorum value will be increased automatically when the node rejoins the cluster.

EXAMPLES

1
```
$ SET CLUSTER/QUORUM
```

This command instructs the system to calculate the cluster quorum value for you, since no value is specified as part of the command string. The system will use the formula: (V+2)/2, described above.

2
```
$ SET CLUSTER/QUORUM=9
```

This command sets the cluster quorum to 9, which is the value specified in the command string.

SET COMMAND

Invokes the Command Definition Utility to add commands to your process command table or to a specified command table file. For a complete description of the Command Definition Utility, including information about the SET COMMAND command, see the *VAX/VMS Command Definition Utility Reference Manual.*

FORMAT

SET COMMAND *[file-spec[,...]]*

SET CONTROL

Determines whether control passes to the command interpreter when CTRL/Y is pressed and whether process statistics are displayed when CTRL/T is pressed.

FORMAT

SET [NO]CONTROL*[=(T,Y)]*

restrictions

SET CONTROL=T requires that SET TERMINAL/BROADCAST be set in order for the information to be displayed at your terminal.

PARAMETER

(T,Y)

Specifies that one or both control functions, T (CTRL/T) or Y (CTRL/Y), be enabled or disabled by the SET CONTROL command. If you specify both control characters, separate them with a comma and enclose the list in parentheses. By default, if you do not specify either control character (T or Y), only Y is assumed.

DESCRIPTION

The CTRL/Y function provides a general-purpose escape from the current operation. CTRL/Y 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 SET NOCONTROL=Y is in effect, the INTERRUPT message is displayed, but no interruption takes place.

The effect of SET NOCONTROL=Y also disables the CTRL/C cancel function for all commands and programs that do not have special action routines responding to CTRL/C.

The CTRL/T function displays a single line of statistical information about the current process. When you use CTLR/T during an interactive terminal session, it momentarily interrupts the current command, command procedure, or image in order 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, CTRL/T does not cause any statistics to be displayed.

EXAMPLES

1
```
$ SET NOCONTROL=Y
```

The SET NOCONTROL command disables the CTRL/Y function as well as most CTRL/C functions.

2
```
$ SET CONTROL=T
```

The SET CONTROL command enables the CTRL/T function.

3
```
$ SET NOCONTROL=(T,Y)
```

The SET NOCONTROL command 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
```

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 appears.

SET DAY

Allows you to reset the default day type specified in the user authorization file (UAF) for the current day.

FORMAT

SET DAY

restrictions

Requires operator (OPER) privilege.

PARAMETERS

None.

DESCRIPTION

The SET DAY command allows you to override 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
```

This command overrides the current default day type and sets the today until midnight to a primary day.

2
```
$ SET DAY/DEFAULT
```

This command overrides the previous SET DAY command and sets today's day type to the UAF-defined default.

SET DEFAULT

Changes the default device and/or directory name for the current process. 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[:]*

restrictions

None.

PARAMETER

device-name[:]

Specifies a device and/or a directory name to be used as the default device or directory in file specifications.

If you specify a physical device name, terminate the device name with a colon. If you specify a directory name, you must enclose it in square brackets.

When you use a search list logical name as the first element in the SET DEFAULT parameter, the logical name is not translated by SET DEFAULT. Instead, SET DEFAULT retains the logical name so that RMS will be able to access the entire search list. If you issue the SHOW DEFAULT command, the search list logical name will be displayed as the default device, along with its equivalence strings. See Section 4.8 of the *VAX/VMS DCL Concepts Manual* for more information.

You can use the minus sign as a directory-searching wildcard character in the directory specification.

EXAMPLES

1

```
$ SET DEFAULT [CARPENTER]
$ COPY A.* B.*
```

The SET DEFAULT command changes the default directory to [CARPENTER]. The default disk device does not change. The directory name [CARPENTER] is assumed to be the default directory for subsequent file operations, such as the COPY command shown.

2

```
$ SET DEFAULT DISK2:
```

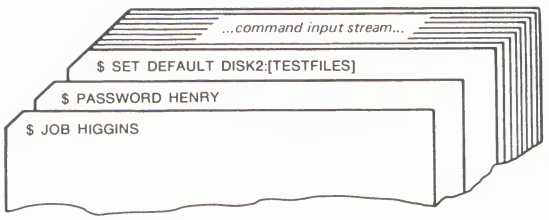
The SET DEFAULT command changes the default disk device to DISK2. The default directory name does not change.

3

```
$ SET DEFAULT DISK0:[CLARK]
```

The SET DEFAULT command changes the default disk to DISK0 and the default directory name to CLARK.

4



ZK-789-82

A batch user submits a job using the card reader. The first command in the batch job is a SET DEFAULT command, which causes all file specifications to default to the directory [TESTFILES] on the disk DISK2.

5

```
$ SAVEDEF = F$ENVIRONMENT("DEFAULT")
$ SET DEFAULT [122001.JONES.APP10]
      .
      .
      .
$ SET DEFAULT  'SAVEDEF'
```

This command procedure 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.JONES.APP10. Later, the symbol SAVEDEF is used to restore the original default directory.

6

```
$ SHOW DEFAULT
$ 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]
FOO.TMP;1
Total of 1 file.
Directory WORK:[TOP.SUB2]
FOO.TMP;1
Total of 1 file.
Grand total of 2 directories, 2 files.
$ DIRECTORY []
Directory WORK:[TOP]
FOO.TMP;1           NETSERVER.LOG;2
Total of 2 files.
```

The default disk and directory is WORK:[TOP]. X is then defined to be a search list consisting of two subdirectories. When the SET DEFAULT X command is entered, the search list (X) is equated with the logical name SYS$DISK and 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 issued, the directories searched will be those complete directory specifications contained in the logical name X. However, if the current default directory specification ([]) is explicitly entered, the current default directory, rather than SYS$DISK, will be searched.

SET DEVICE

Establishes a print device or terminal as a spooled device or establishes the status of error-logging for a device.

FORMAT

SET DEVICE *device-name[:]*

restrictions

Requires operator (OPER) privilege.

PARAMETER

device-name[:]

Specifies the name of the device whose spooling or error-logging status is to change. The device must be a print device or a terminal if its spooling status is to change; the device must be a disk or magnetic tape if its error-logging status is to change.

DESCRIPTION

When you specify the /SPOOLED qualifier, program output that uses VAX RMS or FCS and specifies the print device name is written onto an intermediate disk rather than written directly to the print device or terminal.

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.

QUALIFIERS

/ACL

Modifies an access control list (ACL) associated with one or more devices. For more information, see the description of the SET DEVICE/ACL command.

/AVAILABLE
/NOAVAILABLE

Controls whether the specified disk is to be considered available. This command can be issued only after the specified disk has been dismounted. If you specify /NOAVAILABLE, any attempt to mount the specified disk is prevented.

/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.

/ERROR_LOGGING
/NOERROR_LOGGING

Controls whether device errors are logged in the error log file. Use the SHOW DEVICE/FULL command to find out the current status.

/LOG
/NOLOG (default)

Controls whether log information will be displayed at the terminal.

/SPOOLED[=(queue-name[:], intermediate-disk-name[:])]
/NOSPOOLED

Controls whether files will be 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.

EXAMPLES

1
```
$ SET DEVICE/SPOOLED=(LPA0)  LPA0:
```

In this command, 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  DBB2:
```

This command requests that all device errors reported on DBB2 be logged in the error log file.

3
```
$ SET DEVICE/NOAVAILABLE DRA0:
```

This command prevents any attempt to mount a disk on DRA0.

4
```
$ SET DEVICE/DUAL_PORT DRA0:
```

This command enables the dual port seize logic in DRA0.

SET DEVICE/ACL

Allows you to modify the access control list (ACL) of a device. The /ACL qualifier is required.

FORMAT

SET DEVICE/ACL*[=(ace[,...])] device-name*

restrictions

None.

PARAMETERS

device-name

Specifies a device whose access control list (ACL) is being modified. Wildcard characters are not allowed in the device name.

(ace[,...])

Specifies one or more access control entries (ACEs) to be modified. When no ACE is specified, the entire access control list is affected. Separate multiple ACEs with commas. The specified ACEs are inserted at the top of the ACL unless the /AFTER qualifier is specified.

DESCRIPTION

The SET DEVICE/ACL command enables you to manipulate the entire access control list (ACL) of a device, or to create, modify, or delete access control entries (ACEs) in the ACL of a device. (For information on the format of ACEs and ACLs, see Section 7.2 of the *VAX/VMS DCL Concepts Manual*.) The only valid access to be specified in an ACE for a device is READ access, which means that the device can be allocated.

To use the SET DEVICE/ACL command, you specify the name of the device whose ACL you want to manipulate.

The SET DEVICE/ACL command can be used to add ACEs to an ACL. For example, the following command adds an ACE to the ACL of the terminal device TTA3 so that no users associated with the identifier SALES have access to that terminal:

```
$ SET DEVICE/ACL=(IDENTIFIER=SALES,ACCESS=NONE) TTA3
```

If the device specified with the SET DEVICE/ACL command does not have an ACL, one is created.

The SET DEVICE/ACL command provides the following qualifiers to manipulate ACEs and ACLs:

/AFTER
/DELETE
/LIKE
/NEW
/REPLACE

You can delete ACEs from an ACL by including the /DELETE qualifier and specifying the ACEs with the /ACL qualifier. To delete all the ACEs (except those with the PROTECTED option), include the /DELETE qualifier and specify /ACL without specifying any ACEs.

You can copy an ACL from one object to another by using the /LIKE qualifier. The ACL of the object specified with /LIKE replaces the ACL of the device given with the command.

You can replace existing ACEs in the ACL of the device specified with the command by using the /REPLACE qualifier. Any ACEs specified with /ACL are deleted and replaced by those specified with /REPLACE.

The /NEW qualifier is used to delete all ACEs (except those with the PROTECTED option) before adding any ACEs specified by /ACL, /LIKE, or /REPLACE.

When referring to existing ACEs with /DELETE, /REPLACE, or /AFTER, the existing ACE may be abbreviated.

By default, any ACEs (except security alarm ACEs) added to an ACL are placed at the top of the ACL. Security alarm ACEs are always positioned at the top of the ACL, regardless of positioning qualifiers. Whenever the system receives a request for access to a device that has an ACL, the system searches each entry in the ACL from the first to the last for the first match it can find, and then stops searching. If another match occurs further down in the ACL, it will have no effect. Since the position of an ACE in an ACL is so important, you can use the /AFTER qualifier to correctly position an ACE. When you use the /AFTER qualifier, any ACEs added will be added after the ACE specified with /AFTER.

The SET DEVICE/ACL command can also be used with the /EDIT qualifier to invoke the ACL editor. The following qualifiers can be used only when the /EDIT qualifier has been specified.

/JOURNAL
/KEEP
/MODE
/RECOVER

For more information on these qualifiers and the ACL editor in general, see the *VAX/VMS Access Control List Editor Reference Manual*.

QUALIFIERS

/AFTER=ace

Causes all access control entries (ACEs) specified with the /ACL qualifier to be added after the ACE specified with the /AFTER qualifier. By default, any ACEs added to the ACL are always placed at the top of the list.

This qualifier cannot be used with the /EDIT qualifier.

/DELETE

Indicates that the access control entries (ACEs) specified with the /ACL qualifier are to be deleted. If no ACEs are specified with /ACL, the entire ACL is deleted (except for ACEs with the PROTECTED option). If you specify an ACE that does not exist, you will be notified that the ACE does not exist, and the delete operation will continue.

This qualifier cannot be used with the /EDIT qualifier.

/EDIT

Invokes the ACL Editor and allows you to use the /JOURNAL, /KEEP, /MODE, or /RECOVER qualifiers. Any other qualifiers specified with /EDIT are ignored.

For more information on the ACL Editor, see the ACL Editor description in the *VAX/VMS Access Control List Editor Reference Manual.*

/JOURNAL[=file-spec]
/NOJOURNAL

Controls whether a journal file is created from the editing session. By default, a journal file is created if the editing session ends abnormally.

If you omit the file specification, the journal file has the same name as the input file and a file type of JOU. You can use the /JOURNAL qualifier to specify a journal file name that is different from the default. No wildcard characters are allowed in the /JOURNAL file-spec parameter.

You must specify /EDIT in order to use this qualifier.

/KEEP=(option[,...])

Determines whether the journal file or the recovery file will be deleted when the editing session ends. The options are:

- JOURNAL—saves the journal file for current editing session
- RECOVER—saves the journal file used for restoring the ACL

You can shorten the keywords JOURNAL and RECOVER to J and R, respectively. If you specify only one option, you can omit the parentheses.

You must specify /EDIT in order to use this qualifier.

/LIKE=object-spec

Indicates that the ACL of the object given with the /LIKE qualifier is to replace the ACL of the device specified with SET DEVICE/ACL. Any existing ACE (except those with the PROTECTED option) will be deleted before the ACL specified by /LIKE is copied.

No wildcard characters are allowed in the /LIKE device-name parameter.

This qualifier cannot be used with the /EDIT qualifier.

/LOG
/NOLOG (default)

Controls whether the SET DEVICE/ACL command displays the device name of the device that has been affected by the command.

This qualifier cannot be used with the /EDIT qualifier.

/MODE=[NO]PROMPT

Determines whether the ACL editor prompts for field values. By default, the ACL editor selects prompt mode.

You must specify the /EDIT qualifier to use this qualifier.

/NEW

Indicates that any existing ACE in the ACL of the device specified with SET DEVICE/ACL (except those with the PROTECTED option) is to be deleted. In order to use the /NEW qualifier, you must specify a new ACL or ACE with the /ACL, /LIKE, or /REPLACE qualifier.

This qualifier cannot be used with the /EDIT qualifier.

/RECOVER[=file-spec]
/NORECOVER (default)

Specifies the name of the journal file to be used in a recovery operation. If the file specification is omitted with /RECOVER, the journal file is assumed to have the same name as the input file and a file type of JOU. No wildcard characters are allowed with the /RECOVER file-spec parameter.

You must specify /EDIT in order to use this qualifier.

/REPLACE=(ace[,...])

Deletes the access control entries (ACEs) specified with the /ACL qualifier and replaces them with those specified with /REPLACE. Any ACEs specified with the /ACL qualifier must exist and must be specified in the order in which they appear in the ACL.

This qualifier cannot be used with the /EDIT qualifier.

EXAMPLES

1
```
$ SET DEVICE/ACL/LIKE=DOCD$ WRKD$
```

This example replaces the ACL of WRKD$ with the ACL for the device DOCD$.

2
```
$ SET DEVICE/ACL/EDIT/JOURNAL=ACL.JOB WORK3:
```

This SET DEVICE/ACL command invokes the interactive ACL editor and creates a journal file, named ACL.JOB, if the editing session ends abnormally. The /EDIT qualifier enables use of the /JOURNAL qualifier.

SET DEVICE/SERVED

Allows you to make a disk on a local node available to all the nodes in a cluster. The /SERVED qualifier is required.

FORMAT

SET DEVICE/SERVED *node-name$DDcu:*

restrictions

Applies only to VAXcluster environments.

PARAMETERS

node-name$DDcu:

Specifies the device name of the device that you wish 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 a VAXCluster, and the local MSCP server must have been invoked by the SYSGEN utility.

Note: Unless the disk device that you intend to make available to the cluster is a system disk, it must not already be mounted when you issue the SET DEVICE/SERVED command.

The SET DEVICE/SERVED command string can be included as part of the local startup command file, and issued before the MOUNT utility mounts the disk to be served (made available to the entire cluster).

EXAMPLE

```
$ SET DEVICE/SERVED DRA4:
```

This command instructs the MSCP server to make the disk device DRA4: on your local node available to all other processors on your cluster.

SET DIRECTORY

Modifies the characteristics of one or more directories.

FORMAT

SET DIRECTORY *directory-spec[,...]*

restrictions

See qualifier descriptions.

PARAMETER

directory-spec[,...]

Specifies one or more directories to be modified. You must supply a complete directory specification enclosed in square brackets. If you specify two or more directories, separate them with commas.

Wildcard characters are allowed in directory specifications.

QUALIFIERS

/ACL

Associates an Access Control List (ACL) with one or more directory files. For more information, see the description of the SET DIRECTORY/ACL command.

/BACKUP

Selects directories according to the dates of their most recent backup. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /BACKUP qualifier is incompatible with /CREATED, /EXPIRED, and /MODIFIED. The default is /CREATED.

/BEFORE[=time]

Selects only those directories that are dated before the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/BY_OWNER[=uic]

Specifies that only those directories which are owned by the specified user identification code (UIC) will be modified.

Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

If the /BY_OWNER qualifier is specified without a UIC, the UIC of the current process is assumed.

/CONFIRM
/NOCONFIRM (default)

Controls whether the SET DIRECTORY command displays the directory specification of each directory before modifying it, and requests you to confirm that the operation should be performed on that directory.

When the system issues the prompt, you can issue any of the following responses:

YES	NO	QUIT
TRUE	FALSE	CTRL/Z
1	0	ALL
	<RET>	

You can use any combination of upper- and lowercase letters for word responses. Word responses can be abbreviated to one or more letters (for example, T, TR, or TRU for TRUE). Affirmative answers are YES, TRUE, and 1. Negative answers are NO, FALSE, 0, and <RET> . QUIT or CTRL/Z indicates that you want to stop processing the command at that point. When you respond with 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, the prompt will be reissued.

/CREATED (default)

Selects directories based on their dates of creation. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /CREATED qualifier is incompatible with /BACKUP, /EXPIRED, and /MODIFIED.

/EXCLUDE=(directory-spec[,...])

Any directories that match the listed directory specifications are excluded from the SET DIRECTORY operation.

Wildcard characters are supported for directory specifications. If you specify only one directory, you can omit the parentheses. You cannot include the device name in the directory specifications you supply with the /EXCLUDE qualifier.

/EXPIRED

Selects directories according to the dates on which they will expire. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /EXPIRED qualifier is incompatible with /BACKUP, /CREATED, and /MODIFIED. The default is /CREATED.

/LOG
/NOLOG (default)

Controls whether the SET DIRECTORY command displays the directory specification of each directory that is modified.

/MODIFIED

Selects directories according to the dates on which they were last modified. This qualifier is only relevant when used with the /BEFORE or /SINCE

qualifier. Use of the /MODIFIED qualifier is incompatible with /BACKUP, /CREATED, and /EXPIRED. The default is /CREATED.

/OWNER_UIC[=uic]

Requires SYSPRV privilege to specify a UIC other than your own.

Sets the owner user identification code (UIC) of the directory to the specified UIC. Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

/SINCE[=time]

Selects only those directories that are dated after the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/VERSION_LIMIT[=n]

Specifies the total number of versions that a file in the specified directory can have. If you do not specify a version limit, a value of 0 is used, indicating that the number of versions of a file is limited only to the Files–11 architectural limit—32,767.

If you change the version limit for the directory, the new value applies only to files created after the change has been made. The new limit does *not* apply to new versions of files that existed before the change. The limits that apply to those files are the same as the limits for the initial versions of the files.

The SET DIRECTORY version limit value refers to the number of files with the same file name and type that can exist in the directory at one time. It has no effect on the version number field of a particular file specification. Use the SET FILE command to set limits on file version numbers.

To find out what the current version limit is for a directory, you must use the DUMP/HEADER command. Specify the /FORMATTED qualifier to format the output and the /BLOCKS=COUNT:0 qualifier to avoid dumping the entire directory contents. For example,

```
DUMP/HEADER/FORMATTED/BLOCKS=COUNT:0 directory-spec
```

EXAMPLES

1
```
$ SET DIRECTORY/VERSION_LIMIT=5/CONFIRM [SMITH.FORTRAN]
```

The SET DIRECTORY command limits to five the number of versions for files created after the command is issued. The /CONFIRM qualifier requests that you confirm whether or not the specified directory should actually be modified.

2
```
$ SET DIRECTORY/OWNER_UIC=[360,020] [DAVIDSON],[USERS]
```

The SET DIRECTORY command modifies both the [DAVIDSON] and [USERS] directories, changing their owner UICs. Use of the OWNER_UIC qualifier requires SYSPRV (system privilege).

SET DIRECTORY/ACL

Allows you to modify the access control list (ACL) of one or more directories. The /ACL qualifier is required.

FORMAT

SET DIRECTORY/ACL*[=(ace[,...])] directory-spec[,...]*

restrictions

None.

PARAMETERS

directory-spec[,...]

Specifies one or more directories whose access control list (ACL) is being modified. Separate multiple directory specifications with commas. Device name and colon are optional. Wildcard characters are allowed in the directory specifications. Each directory must be a disk directory on a Files–11 Structure Level 2 formatted volume.

When the /EDIT qualifier is used, only one directory specification can be given, and it cannot include any wildcard characters.

(ace[,...])

Specifies one or more access control entries (ACEs) to be modified. When no ACE is specified, the entire access control list is affected. Separate multiple ACEs with commas. The specified ACEs are inserted at the top of the ACL unless the /AFTER qualifier is specified.

DESCRIPTION

The SET DIRECTORY/ACL command enables you to manipulate an entire access control list (ACL) of one or more directories, or to create, modify, or delete access control entries (ACEs) in the ACL of one or more directories. (For information on the format of ACEs and ACLs, see Section 7.2 of the *VAX/VMS DCL Concepts Manual*.) To use the SET DIRECTORY/ACL command, you specify the directory specification of the directory whose ACL you want to manipulate.

By using wildcard characters in the directory specification to the command, you can manipulate the ACLs of multiple directories with a single command. The following qualifiers can be used with wildcard characters to select a subset of the specified directories:

/BEFORE
/BY_OWNER
/CREATED
/EXCLUDE
/SINCE

You can also use the /CONFIRM qualifier to verify the directory selection.

The SET DIRECTORY/ACL command can be used to add ACEs to an ACL. For example, the following command adds an ACE to the ACL of the directory [.CONFIDENTIAL] so that all users associated with the identifier PERSONNEL are allowed read access to that directory:

```
$ SET DIRECTORY/ACL=(IDENTIFIER=PERSONNEL,ACCESS=READ) [.CONFIDENTIAL]
```

If the directory specified with the SET DIRECTORY/ACL command does not have an ACL, one is created.

The SET DIRECTORY/ACL command provides the following qualifiers to manipulate ACEs and ACLs:

/AFTER
/DELETE
/LIKE
/NEW
/REPLACE

You can delete ACEs from an ACL by including the /DELETE qualifier and specifying the ACEs with the /ACL qualifier. To delete all the ACEs (except those with the PROTECTED option), include the /DELETE qualifier and specify /ACL without specifying any ACEs.

You can copy an ACL from one object to a directory by using the /LIKE qualifier. The ACL of the object specified with /LIKE replaces the ACL of the directory given with the command.

You can replace existing ACEs in the ACL of the directory specified with the command by using the /REPLACE qualifier. Any ACEs specified with /ACL are deleted and replaced by those specified with /REPLACE.

The /NEW qualifier is used to delete all ACEs (except those with the PROTECTED option) before adding any ACEs specified by /ACL, /LIKE, or /REPLACE.

When referring to existing ACEs with /DELETE, /REPLACE, or /AFTER, the existing ACE may be abbreviated.

By default, any ACEs (except security alarm ACEs) added to an ACL are placed at the top of the ACL. Security alarm ACEs are always positioned at the top of the ACL, regardless of positioning qualifiers. Whenever the system receives a request for access to a directory that has an ACL, the system searches each entry in the ACL from the first to the last for the first match it can find and then stops searching. If another match occurs further down in the ACL, it will have no effect. Since the position of an ACE in an ACL is so important, you can use the /AFTER qualifier to correctly position an ACE. When you use the /AFTER qualifier, any ACEs added will be added after the ACE specified with /AFTER.

The SET DIRECTORY/ACL command can also be used with the /EDIT qualifier to invoke the ACL editor. When the /EDIT qualifier is specified, only one directory specification is allowed. When specifying the directory with /EDIT, use the syntax directory.DIR. The following qualifiers can be used only when the /EDIT qualifier has been specified.

/JOURNAL
/KEEP
/MODE
/RECOVER

For more information on these qualifiers and the ACL editor in general, see the *VAX/VMS Access Control List Editor Reference Manual*.

QUALIFIERS

/AFTER=ace

Causes all access control entries (ACEs) specified with the /ACL qualifier to be added after the ACE specified with the /AFTER qualifier. By default any ACEs added to the ACL are always placed at the top of the list.

This qualifier cannot be used with the /EDIT qualifier.

/BEFORE[=time]

Selects only those directories that are dated before the specified time. You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

This qualifier cannot be used with the /EDIT qualifier.

/BY_OWNER[=uic]

Selects one or more directories whose owner user identification code (UIC) matches the specified owner UIC. If the /BY_OWNER qualifier is specified without a UIC, the UIC of the current process is assumed.

Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

This qualifier cannot be used with the /EDIT qualifier.

/CONFIRM
/NOCONFIRM (default)

Controls whether a request is issued before each individual SET DIRECTORY /ACL operation to confirm that the operation should be performed on that directory.

When the system issues the prompt, you can issue any of the following responses:

YES	NO	QUIT
TRUE	FALSE	CTRL/Z
1	0	ALL
	<RET>	

You can use any combination of upper- and lowercase letters for word responses. Word responses can be abbreviated to one or more letters (for example, T, TR, or TRU for TRUE). Affirmative answers are YES, TRUE, and 1. Negative answers are NO, FALSE, 0, and <RET> . QUIT or CTRL/Z indicates that you want to stop processing the command at that point. When you respond with 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, the prompt will be reissued.

This qualifier cannot be used with the /EDIT qualifier.

/CREATED

Selects directories based on their dates of creation. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier.

This qualifier cannot be used with the /EDIT qualifier.

/DEFAULT

Creates an ACL for the specified files as if the files were newly created. For a directory file, the /DEFAULT qualifier propagates the entire ACL (except ACEs with the NOPROPAGATE option) so that a particular access protection can be propagated throughout a directory tree. For all other files, the /DEFAULT qualifier propagates the DEFAULT option ACEs in the ACL of the parent directory to the ACL of the specified files.

The /DEFAULT qualifier uses the ACL of the parent directory of the specified file, not the current default directory. This qualifier cannot be used with the /EDIT qualifier.

/DELETE

Indicates that the access control entries (ACEs) specified with the /ACL qualifier are to be deleted. If no ACEs are specified with /ACL, the entire ACL is deleted (except for ACEs with the PROTECTED option). If you specify an ACE that does not exist, you will be notified that the ACE does not exist and the delete operation will continue.

This qualifier cannot be used with the /EDIT qualifier.

/EDIT

Invokes the ACL Editor and allows you to use the /JOURNAL, /KEEP, /MODE, or /RECOVER qualifiers. Any other qualifiers specified with /EDIT are ignored. You can only supply one directory file specification with SET DIRECTORY/ACL/EDIT.

For more information on the ACL Editor, see the ACL Editor description in the *VAX/VMS Access Control List Editor Reference Manual*.

/EXCLUDE=(directory-spec[,...])

Excludes any directories that match the listed directory specifications from the SET DIRECTORY/ACL operation. If you specify only one directory, you can omit the parentheses. Wildcard characters are allowed in the directory specifications. The directory specification cannot contain a device name.

This qualifier cannot be used with the /EDIT qualifier.

/JOURNAL[=file-spec]
/NOJOURNAL

Controls whether a journal file is created from the editing session. By default, a journal file is created if the editing session ends abnormally.

If you omit the file specification, the journal file has the same name as the input file and a file type of JOU. You can use the /JOURNAL qualifier to specify a journal file name that is different from the default. No wildcard characters are allowed in the /JOURNAL file-spec parameter.

You must specify /EDIT in order to use this qualifier.

/KEEP=(option[,...])

Determines whether the journal file or the recovery file will be deleted when the editing session ends. The options are:

- JOURNAL—saves the journal file for the current editing session
- RECOVER—saves the journal file used for restoring the ACL

You can shorten the keywords JOURNAL and RECOVER to J and R, respectively. If you specify only one option, you can omit the parentheses.

You must specify /EDIT in order to use this qualifier.

/LIKE=object-spec

Indicates that the ACL of the object given with the /LIKE qualifier is to replace the ACL of the directories specified with SET DIRECTORY/ACL. Any existing ACE (except those with the PROTECTED option) will be deleted before the ACL specified by /LIKE is copied.

No wildcard characters are allowed in the /LIKE object-spec parameter.

This qualifier cannot be used with the /EDIT qualifier.

/LOG
/NOLOG (default)

Controls whether the SET DIRECTORY/ACL command displays the directory specification of each directory that has been affected by the command. By default, no log information is displayed.

This qualifier cannot be used with the /EDIT qualifier.

/MODE=[NO]PROMPT

Determines whether the ACL editor prompts for field values. By default, the ACL editor selects prompt mode.

You must specify the /EDIT qualifier to use this qualifier.

/NEW

Indicates that any existing ACE in the ACL of the directory specified with SET DIRECTORY/ACL (except those with the PROTECTED option) is to be deleted. In order to use the /NEW qualifier, you must specify a new ACL or ACE with the /ACL, /LIKE, or /REPLACE qualifier.

This qualifier cannot be used with the /EDIT qualifier.

/RECOVER[=file-spec]
/NORECOVER (default)

Specifies the name of the journal file to be used in a recovery operation. If the file specification is omitted with /RECOVER, the journal file is assumed to have the same name as the input file and a file type of JOU. No wildcard characters are allowed with the /RECOVER file-spec parameter.

You must specify /EDIT in order to use this qualifier.

/REPLACE=(ace[,...])

Deletes the access control entries (ACEs) specified with the /ACL qualifier and replaces them with those specified with /REPLACE. Any ACEs specified with the /ACL qualifier must exist and must be specified in the order in which they appear in the ACL.

This qualifier cannot be used with the /EDIT qualifier.

/SINCE[=time]

Selects only those directories that are dated after the specified time. You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

This qualifier cannot be used with the /EDIT qualifier.

EXAMPLES

1
```
$ SET DIRECTORY/ACL/LIKE=[.USER] [.USER.CAPTIVE_ACCOUNTS]
```

This example replaces the ACL of the directory CAPTIVE_ACCOUNTS with the ACL for the directory USER.LIS.

2
```
$ SET DIRECTORY/ACL=(IDENTIFIER=[123,321]+NETWORK,ACCESS=NONE) [.CONFIDENTIAL]
```

This command adds an ACE that specifies that NETWORK access for user [123,321] is not allowed for directory CONFIDENTIAL.

3
```
$ SET DIRECTORY/ACL/EDIT/JOURNAL=ACL.JOB [.PAYROLL]
```

This SET DIRECTORY/ACL command invokes the interactive ACL editor and creates a journal file, named ACL.JOB, if the editing session ends abnormally. The /EDIT qualifier enables use of the /JOURNAL qualifier.

4
```
$ SET DIRECTORY/ACL/EDIT/RECOVER=ACL.JOB [.PAYROLL]
```

This command uses the /RECOVER qualifier to restore the editing session from the previous example after the session ended abnormally.

SET FILE

Modifies the characteristics of one or more files.

FORMAT

SET FILE *file-spec[,...]*

restrictions

See qualifier descriptions.

PARAMETER

file-spec[,...]

Specifies one or more files to be modified. If you specify two or more files, separate them with commas.

Wildcard characters are allowed in the file specifications.

DESCRIPTION

The SET FILE command enables you to modify a number of file characteristics. Use the SET FILE/ACL command to modify the access control list of one or more files.

The common qualifiers /BACKUP, /EXPIRED, and /MODIFIED have not been implemented for SET FILE.

QUALIFIERS

/ACL

Modifies an access control list (ACL) associated with one or more files. For more information, see the description of the SET FILE/ACL command.

/BACKUP
/NOBACKUP (default)

Controls whether the BACKUP command will back up the contents of the specified files. This qualifier is valid only for Files–11 Structure Level 2 files.

If you specify /NOBACKUP, BACKUP will record the attributes of the file but not the data in the file. The /NOBACKUP qualifier is useful for saving files that contain unimportant data, such as SWAPFILES.

/BEFORE[=time]

Selects only those files that are dated before the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/BY_OWNER[=uic]

Selects one or more files only if their owner user identification code (UIC) matches the specified owner UIC.

Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

If the /BY_OWNER qualifier is specified without a UIC, the UIC of the current process is assumed.

/CONFIRM
/NOCONFIRM (default)

Controls whether a request is issued before each individual SET FILE operation to confirm that the operation should be performed on that file.

When the system issues the prompt, you can issue any of the following responses:

YES	NO	QUIT
TRUE	FALSE	CTRL/Z
1	0	ALL
	<RET>	

You can use any combination of upper- and lowercase letters for word responses. Word responses can be abbreviated to one or more letters (for example, T, TR, or TRU for TRUE). Affirmative answers are YES, TRUE, and 1. Negative answers are NO, FALSE, 0, and <RET> . QUIT or CTRL/Z indicates that you want to stop processing the command at that point. When you respond with 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, the prompt will be reissued.

/CREATED

Selects files based on their dates of creation. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier.

/DATA_CHECK[=([NO]READ,[NO]WRITE)]

Specifies whether a READ data check, a WRITE data check, or a combination of the two is performed on the file during transfers. If you do not specify either READ or WRITE, a WRITE data check is performed on the file by default.

/END_OF_FILE

Resets the end-of-file mark to the highest block allocated.

/ENTER=new-file-spec

Enables you to assign an additional name to a file. The file now has a second name, or alias, but both the original name and the alias reference the same file. For this reason, care should be taken when deleting files which have aliases. In order to keep the file but remove one of its names, use the /REMOVE qualifier with SET FILE.

No wildcards are allowed in the file specification.

Use of the /ENTER qualifier is discouraged.

/ERASE_ON_DELETE

Specifies that the specified files will be erased from the disk (not just merely written over) when a command such as DELETE or PURGE is issued for the files. See DELETE/ERASE for more information.

/EXCLUDE=(file-spec[,...])

Any files that match the listed file specifications are excluded from the SET FILE operation. If you specify only one file, you can omit the parentheses.

Wildcard characters are supported for file specifications. However, you cannot use relative version numbers to exclude a specific version. The file specification can contain a directory specification, but not a device specification.

/EXPIRATION_DATE=date
/NOEXPIRATION_DATE

Controls whether an expiration date is assigned to the specified files. Specify the date according to the rules described in Section 2.5 of the *VAX/VMS DCL Concepts Manual*. Absolute date keywords are allowed. If you specify 0 as the date, today's date is used. Use of this qualifier requires ownership of the file or access control.

/EXTENSION[=n]

Sets the default extend quantity for the file. You can specify a value from 0 to 65,535. If you omit the value specification or specify a value of 0, VAX RMS calculates its own /EXTENSION value.

See the SET RMS_DEFAULT command for a description of the /EXTEND_QUANTITY qualifier.

/GLOBAL_BUFFER=n

Specifies the VAX RMS global buffer count for the specified files. Specify a number (n) in the range of 0 through 32,767 to indicate the number of buffers that can be shared by processes accessing the file.

A value of 0 disables buffer sharing.

/LOG
/NOLOG (default)

Controls whether the SET FILE command displays the file specification of each file after the modification is made.

/NODIRECTORY

Removes the directory attributes of a file. This qualifier 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 become 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 the Files-11 Structure Level-2 files. For more information about the Verify Utility, see the *VAX/VMS Verify Utility Reference Manual*.

/OWNER_UIC[=uic]

Requires GRPPRV (group privilege) privilege to set the owner to another member of the same group. Requires SYSPRV (system privilege) privilege to set the owner to any UIC outside your group.

Sets the owner user identification code (UIC) of the file to the specified UIC.

Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual.*

/PROTECTION[=(code)]

Enables you to change or reset the protection for one or more of your files. If no protection code is specified, the access of the specified files is set to the current default protection.

For more information on specifying the protection code, see Section 7.1.2 of the *VAX/VMS DCL Concepts Manual.*

/REMOVE

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 will not be able to access that file with most DCL commands; you will have to use the ANALYZE/DISK_STRUCTURE utility to retrieve the file. Use of the /REMOVE qualifier is therefore discouraged.

/SINCE[=time]

Selects only those files that are dated after the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/UNLOCK

Requests that one or more files be unlocked. A locked file is a file which has become inaccessible as a result of having been improperly closed.

/TRUNCATE

Requests that the file is to be truncated at the end of the block containing the end-of-file (EOF) mark.

/VERSION_LIMIT[=n]

Specifies the maximum number of versions a file can have. If you do not specify a version limit, a value of 0 is used, indicating that the number of versions of a file is limited only to the Files–11 architectural limit, 32,767. When you create a new version of a file that would cause the version limit to be exceeded, the earliest version is deleted from the directory.

When the /VERSION_LIMIT qualifier is in effect, only one version of a file is deleted each time a new file is created. For example, if you set the version limit to 3 when there are already five versions of that file in your directory, there will continue to be five versions of the file unless you specifically delete some or purge the directory. Once the number of versions is equal to or less than the current version limit, the version limit will be maintained.

EXAMPLES

1
```
$ SET FILE/EXPIRATION_DATE=15-APR-1984:11:00 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., April 15, 1984.

2
```
$ SET FILE/BEFORE=15-APR/ERASE_ON_DELETE PERSONNEL*.SAL
```

This SET FILE command calls for all files that match the file specification PERSONNEL*.SAL and that are dated before April 15 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 UIC and assigning a file version limit of 100. Note that you must have system privilege (SYSPRV) in order to change the owner UIC.

SET FILE/ACL

Allows you to modify the access control list (ACL) of one or more files. The /ACL qualifier is required.

FORMAT

SET FILE/ACL*[=(ace[,...])] file-spec[,...]*

restrictions

None.

PARAMETERS

file-spec[,...]

Specifies one or more files whose access control list (ACL) is being modified. Separate multiple file specifications with commas. Wildcard characters are allowed in the file specifications. Each file must be a disk file on a Files–11 Structure Level 2 formatted volume.

When the /EDIT qualifier is used, only one file specification can be given, and it cannot include any wildcard characters.

(ace[,...])

Specifies one or more access control entries (ACEs) to be modified. When no ACE is specified, the entire access control list is affected. Separate multiple ACEs with commas. The specified ACEs are inserted at the top of the ACL unless the /AFTER qualifier is specified.

DESCRIPTION

The SET FILE/ACL command enables you to manipulate an entire access control list (ACL) of one or more files, or to create, modify, or delete access control entries (ACEs) in the ACL of one or more files. (For information on the format of ACEs and ACLs, see Section 7.2 of the *VAX/VMS DCL Concepts Manual*.) To use the SET FILE/ACL command, you specify the file specification of the file whose ACL you want to manipulate.

By using wildcard characters in the file specification to the command, you can manipulate the ACLs of multiple files with a single command. The following qualifiers can be used with wildcard characters to select a subset of the specified files:

/BEFORE
/BY_OWNER
/CREATED
/EXCLUDE
/SINCE

You can also use the /CONFIRM qualifier to verify the file selection.

The SET FILE/ACL command is used to add ACEs to an ACL. For example, the following command adds an ACE to the ACL of the file SALARY85.DAT so that all users associated with the identifier PERSONNEL are allowed read access to the file:

```
$ SET FILE/ACL=(IDENTIFIER=PERSONNEL,ACCESS=READ) SALARY85.DAT
```

If the file specified with the SET FILE/ACL command does not have an ACL, one is created.

The SET FILE/ACL command provides the following qualifiers to manipulate ACEs and ACLs:

/AFTER
/DELETE
/LIKE
/NEW
/REPLACE

You can delete ACEs from an ACL by including the /DELETE qualifier and specifying the ACEs with /ACL. To delete all the ACEs (except those with the PROTECTED option), include the /DELETE qualifier and specify /ACL without specifying any ACEs.

You can copy an ACL from one file to another by using the /LIKE qualifier. The ACL of the file specified with /LIKE replaces the ACL of the file given with the command.

You can replace existing ACEs in the ACL of the file specified with the command by using the /REPLACE qualifier. Any ACEs specified with /ACL are deleted and replaced by those specified with /REPLACE.

The /NEW qualifier is used to delete all ACEs (except those with the PROTECTED option) before adding any ACEs specified by /ACL, /LIKE, or /REPLACE.

When referring to existing ACEs with /DELETE, /REPLACE, or/AFTER, the existing ACE may be abbreviated.

By default, any ACEs (except security alarm ACE), added to an ACL are placed at the top of the ACL. Security alarm ACEs are always positioned at the top of the ACL, regardless of positioning qualifiers. Whenever the system receives a request for access to a file that has an ACL, the system searches each entry in the ACL from the first to the last for the first match it can find and then stops searching. If another match occurs further down in the ACL, it will have no effect. Since the position of an ACE in an ACL is so important, you can use the /AFTER qualifier to correctly position an ACE. When you use the /AFTER qualifier, any ACEs added will be added after the ACE specified with /AFTER.

The SET FILE command can also be used with the /EDIT qualifier to invoke the ACL editor. When the /EDIT qualifier is specified, only one file specification is allowed. The following qualifiers can be used only when the /EDIT qualifier has been specified.

/JOURNAL
/KEEP
/MODE
/RECOVER

For more information on these qualifiers and the ACL editor in general, see *VAX/VMS Access Control List Editor Reference Manual.*

QUALIFIERS

/AFTER=ace

Causes all access control entries (ACEs) specified with the /ACL qualifier to be added after the ACE specified with the /AFTER qualifier. By default, any ACEs added to the ACL are always placed at the top of the list.

This qualifier cannot be used with the /EDIT qualifier.

/BEFORE[=time]

Selects only those files that are dated before the specified time. You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

This qualifier cannot be used with the /EDIT qualifier.

/BY_OWNER[=uic]

Selects one or more files whose owner user identification code (UIC) matches the specified owner UIC. If the /BY_OWNER qualifier is specified without a UIC, the UIC of the current process is assumed.

Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

This qualifier cannot be used with the /EDIT qualifier.

/CONFIRM
/NOCONFIRM (default)

Controls whether a request is issued before each individual SET FILE/ACL operation to confirm that the operation should be performed on that file.

When the system issues the prompt, you can issue any of the following responses:

YES	NO	QUIT
TRUE	FALSE	CTRL/Z
1	0	ALL
	<RET>	

You can use any combination of upper- and lowercase letters for word responses. Word responses can be abbreviated to one or more letters (for example, T, TR, or TRU for TRUE). Affirmative answers are YES, TRUE, and 1. Negative answers are NO, FALSE, 0, and <RET> . QUIT or CTRL/Z indicates that you want to stop processing the command at that point. When you respond with 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, the prompt will be reissued.

This qualifier cannot be used with the /EDIT qualifier.

/CREATED

Selects files based on their dates of creation. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier.

This qualifier cannot be used with the /EDIT qualifier.

/DEFAULT

Creates an ACL for the specified files as if the files were newly created. For a directory file, the /DEFAULT qualifier propagates the entire ACL (except ACEs with the NOPROPAGATE option) so that a particular access protection can be propagated throughout a directory tree. For all other files, the /DEFAULT qualifier propagates the DEFAULT option ACEs in the ACL of the parent directory to the ACL of the specified files.

The /DEFAULT qualifier uses the ACL of the parent directory of the specified file, not the current default directory. This qualifier cannot be used with the /EDIT qualifier.

/DELETE

Indicates that the access control entries (ACEs) specified with the /ACL qualifier are to be deleted. If no ACEs are specified with /ACL, the entire ACL is deleted (except for ACEs with the PROTECTED option). If you specify an ACE that does not exist with the /ACL qualifier, you will be notified that the ACE does not exist and the delete operation will continue.

This qualifier cannot be used with the /EDIT qualifier.

/EDIT

Invokes the ACL Editor and allows you to use the /JOURNAL, /KEEP, /MODE, or /RECOVER qualifiers. Any other qualifiers specified with /EDIT are ignored. You can only supply one file specification with SET FILE/ACL /EDIT.

For more information on the ACL Editor, see the *VAX/VMS Access Control List Editor Reference Manual.*

/EXCLUDE=(file-spec[,...])

Excludes any files that match the listed file specifications from the SET FILE /ACL operation. If you specify only one file, you can omit the parentheses. Wildcard characters are allowed in the file specifications. However, you cannot use relative version numbers to exclude a specific version. The file specification can contain a directory specification; however, the file specification cannot contain a device name.

This qualifier cannot be used with the /EDIT qualifier.

/JOURNAL[=file-spec]
/NOJOURNAL

Controls whether a journal file is created from the editing session. By default, a journal file is created if the editing session ends abnormally.

If you omit the file specification, the journal file has the same name as the input file and a file type of JOU. You can use the /JOURNAL qualifier to specify a journal file name that is different from the default. No wildcard characters are allowed in the /JOURNAL file-spec parameter.

You must specify /EDIT in order to use this qualifier.

/KEEP=(option[,...])

Determines whether the journal file or the recovery file will be deleted when the editing session ends. The options are:

- JOURNAL—saves the journal file for current editing session
- RECOVER—saves the journal file used for restoring the ACL

You can shorten the keywords JOURNAL and RECOVER to J and R, respectively. If you specify only one option, you can omit the parentheses.

You must specify /EDIT in order to use this qualifier.

/LIKE=file-spec

Indicates that the ACL of the file given with the /LIKE qualifier is to replace the ACL of the files specified with SET FILE/ACL. Any existing ACE (except those with the PROTECTED option) will be deleted before the ACL specified by /LIKE is copied.

No wildcard characters are allowed in the /LIKE file-spec parameter.

This qualifier cannot be used with the /EDIT qualifier.

/LOG
/NOLOG (default)

Controls whether the SET FILE/ACL command displays the file specification of each file that has been affected by the command.

This qualifier cannot be used with the /EDIT qualifier.

/MODE=[NO]PROMPT

Determines whether the ACL editor prompts for field values. By default, the ACL editor selects prompt mode.

You must specify the /EDIT qualifier to use this qualifier.

/NEW

Indicates that any existing ACE in the ACL of a file specified with SET FILE /ACL (except those with the PROTECTED option) is to be deleted. In order to use the /NEW qualifier, you must specify a new ACL or ACE with the /ACL, /LIKE, or /REPLACE qualifier.

This qualifier cannot be used with the /EDIT qualifier.

/RECOVER[=file-spec]
/NORECOVER (default)

Specifies the name of the journal file to be used in a recovery operation. If the file specification is omitted with /RECOVER, the journal file is assumed to have the same name as the input file and a file type of JOU. No wildcard characters are allowed with the /RECOVER file-spec parameter.

You must specify /EDIT in order to use this qualifier.

/REPLACE=(ace[,...])

Deletes the access control entries (ACEs) specified with the /ACL qualifier and replaces them with those specified with /REPLACE. Any ACEs specified with the /ACL qualifier must exist and must be specified in the order in which they appear in the ACL.

This qualifier cannot be used with the /EDIT qualifier.

/SINCE[=time]

Selects only those files that are dated after the specified time. You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

This qualifier cannot be used with the /EDIT qualifier.

EXAMPLES

1
```
$ SET FILE/ACL/LIKE=USER.LIS CAPTIVE_ACCOUNTS.LIS
```

This example replaces the ACL of the file CAPTIVE_ACCOUNTS.LIS with the ACL for the file USER.LIS.

2
```
$ SET FILE/ACL=(IDENTIFIER=[123,321]+NETWORK,ACCESS=NONE) *.*
```

This command adds an ACE that specifies that NETWORK access for user [123,321] is not allowed for each file in the default directory.

3
```
$ SET FILE/ACL=(IDENTIFIER=[SALES,FRANK],ACCESS=READ)/DELETE *.*
```

This SET FILE/ACL command deletes the specified ACE from all files in the default directory.

4
```
$ SET FILE/ACL/EDIT/JOURNAL=ACL.JOB PASSWORD_2.DAT
```

This SET FILE/ACL command invokes the interactive ACL editor and creates a journal file, ACL.JOB, if the editing session ends abnormally. The /EDIT qualifier enables use of the /JOURNAL qualifier.

5
```
$ SET FILE/ACL/EDIT/RECOVER=ACL.JOB PASSWORD_2.DAT
```

This command uses the /RECOVER qualifier to restore the editing session from the previous example after the session ended abnormally.

SET HOST

Connects your terminal (through the current host processor) to another processor, called the remote processor. Both processors must be running DECnet.[1]

FORMAT

SET HOST *node-name*

restrictions

- You can use the SET HOST command only if your system is connected by DECnet to another system.
- You must have an account on the remote system in order to log in after the SET HOST command has made the connection.
- The SET HOST command requires the network mailbox privilege NETMBX.

PARAMETER

node-name

Specifies the node name of the remote processor.

DESCRIPTION

Use the SET HOST command to connect to another processor on a network. (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 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.

Use the LOGOUT command to log off the last processor you have logged in to. If you have connected to and logged in to more than one processor, the LOGOUT command leaves you logged in to the next previous 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.

When you use the LOGOUT command, you have logged off (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 off ALBANY, and disconnects you from it. A third LOGOUT command logs you off 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. You will be prompted:

```
Are you repeating ^Y to abort the remote session?
```

[1] Available under separate license.

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 issuing a series of LOGOUT commands, or when part of the network becomes disconnected and you want to return to the host.

QUALIFIER

/LOG[=file-spec]
/NOLOG (default)

Controls whether a log file of the entire session is kept. If you use /LOG without the file specification, the log information is stored in the file SETHOST.LOG.

EXAMPLE

```
$ SET HOST ALBANY
Username:  SMITH
Password:
```

This SET HOST command connects the user terminal to the processor at the network node named ALBANY. The remote processor then prompts for user name and password. Use the normal login procedure to log in to the remote processor.

SET HOST/DTE

Connects your system to a remote system through an out-going terminal line.

FORMAT

SET HOST/DTE *terminal-name*

restrictions

- You must have an account on the remote system in order to log in after the SET HOST/DTE command has made the connection.
- Use of the SET HOST/DTE command requires the ability to assign a channel to the terminal port specified. By default, the privilege required is BYPASS. This can be changed by setting the device protection for the terminal port.

PARAMETER

terminal-name

Specifies the name of an out-going terminal line, which connects your system either directly to another system, or to a modem.

DESCRIPTION

Allows you to logically 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 connected to the remote system and logged in, you can use DCL commands just as you would on your local system.

To exit from the remote system, type CTRL/\; that is, type a backslash (\), while holding down the CTRL key.

In order to log in on lines that expect a break rather than a carriage return, type CTRL/] to generate the break.

When connecting directly to another system, the out-going port should be set to NOTYPEAHEAD. This will avoid the possibility of an endless loop, wherein noise on the line causes each of the ports to attempt to initiate a login. (Note that the terminal line to which you have connected via SET HOST/DTE, must be set to TYPEAHEAD in order to allow the login.)

QUALIFIERS

/DIAL=(NUMBER:number[,MODEM_ TYPE:modem-type])

Allows a modem attached to the out-going terminal line to be autodialed using the autodial protocol of that modem. The NUMBER: parameter is required and must be followed by the telephone number that is to be autodialed.

The optional MODEM_TYPE: parameter may be used to specify a modem-type of either DF03 or DF112. By default, a modem-type of DF03 is assumed.

In addition, MODEM_TYPE: may be used to specify a modem-type other than a DF03 or DF112. A template is provided for users interested in supporting other modems with autodial capabilities (see SYS$EXAMPLES:DTE_DF03.MAR).

/LOG[=file-spec]
/NOLOG

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.

When used to log a modem session, the log file will contain any noise which 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 the /LOG qualifier is not recommended for the purpose of file transfers.

Asynchronous DECnet is the recommended way to transfer files. For additional information, see the *VAX/VMS Networking Manual*.

EXAMPLE

```
$ SET HOST/DTE TTA2:/DIAL=NUMBER:5551234
Username:  SMITH
Password:
```

This SET HOST/DTE command connects the user terminal to the out-going 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.

```
$ 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. Note that the number sign (#) is required to activate the autodialer in the DF112.

SET HOST/HSC

Connects your terminal to a remote HSC50 through the Computer Interconnect bus.

FORMAT

SET HOST/HSC *node-name*

restrictions

- Used only with remote HSC50s.
- Requires the DIAGNOSE privilege.

PARAMETER

node-name

Specifies the node name of the remote HSC50.

DESCRIPTION

Use the SET HOST/HSC command to connect to an HSC50 via the CI bus. (The SHOW CLUSTER command lists the names of HSC50s that are accessible to your node.) No password is required to access the HSC50; however, actions other than SHOW commands are prevented when the HSC50's Secure/Enable switch is in the SECURE position.

Once the connection is made to the HSC50, operations may be performed as if you are attached to the local terminal of the HSC with the exception of access to ODT (Octal Debugging Tool) and offline diagnostics.

Press CTRL/C to obtain a prompt from the HSC50 before entering commands. To exit from the HSC50 and return to the local system, press CTRL/\; that is, type a backslash (\) while holding down the CTRL key.

A description of HSC50 commands and utilities may be obtained from the *HSC User's Guide*.

Preparing for Use

To use the SET HOST/HSC facility you must first install FYDRIVER, 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:SYSGEN
CONNECT FYA0/NOADAPTER
```

This operation requires the privilege CMKRNL.

QUALIFIER

/LOG[=file-spec]
/NOLOG (default)

Controls whether a log file of the entire session is kept. If you use /LOG without the file specification, the log information is stored in the file HSCPAD.LOG.

SET HOST/HSC

EXAMPLE

```
$ SET HOST/HSC HSC001
%HSCPAD-I-LOCPROGEXE, Local program executing - type ^\ to exit, ^Y for prompt
HSC50>
```

This SET HOST/HSC command connects the user terminal to the HSC named HSC001.

SET KEY

Changes the current key definition state. Keys are defined by the DEFINE/KEY command.

FORMAT

SET KEY

restrictions

None.

PARAMETERS

None.

DESCRIPTION

When you define keypad keys using the DEFINE/KEY command, you can assign a specific state name to the key definition. If that state is not set when you press the key, the definition will not be 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 state for the system to set. The state name can be any alphanumeric string. If you omit the /STATE qualifier or use /NOSTATE, the current state is left unchanged. The default state is DEFAULT.

EXAMPLE

```
$ SET KEY /STATE=EDITING
```

The SET KEY command changes the key state to the EDITING state. You can now use the key definitions that are associated with this state.

SET LOGINS

Sets a limit on the number of users who can gain access to the operating system. This command also displays the current interactive level as described below.

FORMAT

SET LOGINS

restrictions

Requires operator (OPER) privilege.

PARAMETERS

None.

DESCRIPTION

The SET LOGINS command is not retroactive. Thus, all users logged in to the system before you issue 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 value for n in 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. When you do not supply a value for the n, SET LOGINS displays the current status of the login quotas.

EXAMPLES

1

```
$ SET LOGINS/INTERACTIVE=5
%SET-T-INTSET, login interactive limit=5, current interactive value=3
```

This command specifies that only five interactive users can be logged in to the system.

2

```
$ SET LOGINS/INTERACTIVE
%SET-T-INTSET, login interactive limit=15, current interactive value=6
```

When the SET LOGINS command is issued without a parameter, 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 15 and that the number of interactive users currently logged in is 6. No change is made.

SET MAGTAPE

Defines the default characteristics associated with a specific magnetic tape device for subsequent file operations.

FORMAT

SET MAGTAPE *device-name[:]*

restrictions

The SET MAGTAPE command is valid for magnetic tape devices that have foreign volumes mounted on them.

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

Specifies the default density, in bits per inch (bpi), for all write operations on the magnetic tape device when the volume is mounted as a foreign tape or as an unlabeled tape. The density can be specified as 800, 1600, or 6250, if supported by the magnetic tape drive.

/END_OF_FILE

Writes a tape mark at the current position on the magnetic tape volume.

/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 that are corrected by the hardware without software intervention. This qualifier only affects devices that support hardware error correction, such as the TU78 magnetic tape drive. When used with other devices, this qualifier has no effect.

/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:

BLOCK:n	Directs the SET MAGTAPE command to skip the specified number of blocks
END_OF_TAPE	Directs the SET MAGTAPE command to position the volume at the end-of-tape mark
FILES:n	Directs the SET MAGTAPE command to skip the specified number of files
RECORD:n	Directs the SET MAGTAPE command to skip the specified number of records

/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 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 mounts a foreign tape on the device MTA0; the SET MAGTAPE command directs the magnetic tape position to skip four files.

SET MESSAGE

Permits you to specify the display format of messages, or to override or supplement the system messages.

FORMAT

SET MESSAGE *[file-spec]*

restrictions

None.

PARAMETER

file-spec

Specifies an optional message file. If the file specification does not contain a file type, the default type is EXE.

No wildcard characters are allowed in the file specification.

DESCRIPTION

The SET MESSAGE command allows you to specify which message fields VMS displays. The message format is:

```
%FACILITY-L-IDENT, text
```

When a process is initially created, the default is to display all four message fields: facility, severity, identification, and text. You can control which fields of a message are displayed by specifying the appropriate qualifiers. For example, if you want the message displays to omit the FACILITY field, include the /NOFACILITY qualifier with the SET MESSAGE command.

By including the SET MESSAGE command in your login command file, you can select specific portions of the messages for your process.

The SET MESSAGE command is also used 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, then process-permanent message files, and then the system message file. Thus, when you specify a message file with the SET MESSAGE command, you can introduce messages earlier in the searching order; you can also either override or supplement the system messages. (Note that the new message definitions affect only your process.)

If a process-permanent message file is already in effect when you specify the SET MESSAGE command with a file specification, the old file is removed and the new file added.

The message definitions you specify must result from a successful compilation with the MESSAGE command. For full details on the use of the Message Utility to create your own messages, see the *VAX/VMS Message Utility Reference Manual*.

QUALIFIERS

/DELETE

Removes the currently selected process message file from your process. Do not include a file specification when you issue the /DELETE qualifier.

/FACILITY
/NOFACILITY

Controls whether the facility name prefix is displayed for all messages that are returned for your process.

/IDENTIFICATION
/NOIDENTIFICATION

Controls whether the identification prefix (an abbreviation that identifies the message) is included in all messages that are returned for your process.

/SEVERITY
/NOSEVERITY

Controls whether the severity level is displayed for all messages that are returned for your process.

/TEXT
/NOTEXT

Controls whether the message text is displayed for all messages that are returned for your process.

EXAMPLES

1
```
$ TYPE XXX
%TYPE-W-OPENIN, error opening DB1:[MALCOLM]XXX.LIS; as input
-RMS-E-FNF, file not found
    .
    .
    .
$  SET MESSAGE/NOIDENTIFICATION
    .
    .
    .
$ TYPE XXX
%TYPE-W, error opening DB1:[MALCOLM]XXX.LIS; as input
-RMS-E, file not found
```

When the first TYPE command is issued, 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. Note the absence of the IDENT component in the two subsequent messages that result from attempting to type a file that does not exist.

2
```
$ SET MESSAGE NEWMSG
```

The SET MESSAGE command specifies that the message text in NEWMSG.EXE supplements the existing system messages.

SET ON

Controls whether the command interpreter performs error checking following the execution of each command in a command procedure.

FORMAT

SET [NO]ON

restrictions

None.

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 three bits of this value are also saved in the reserved symbol $SEVERITY.

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 will continue to execute no matter how many errors are returned.

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 will continue execution 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

Sets the rate at which output is written to a batch job log file.

FORMAT

SET OUTPUT_RATE*[=delta-time]*

restrictions

This command is used only within command procedures that are submitted as batch jobs.

PARAMETER

delta-time

Specifies how often output will be written from the output buffer to the batch job log file. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for information on how to specify a delta time.

If you do not specify a delta time, then DCL writes the information in the output buffer to the log file but does not change the default output rate.

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

Allows users to change their own passwords. Also allows system managers to change the system password.

FORMAT

SET PASSWORD

restrictions

See qualifier descriptions.

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. In addition, a system manager can establish a mimimum acceptable password length, and the maximum period of time that a password may remain unchanged. Once a password has expired, you must use the SET PASSWORD command to change your password.

Systems can also have passwords. The system manager uses the SET PASSWORD/SYSTEM command to change the system password from time to time.

Passwords can contain from 1 to 31 characters. The valid characters are:

A through Z
a through z
0 through 9
$ (dollar sign)
_ (underscore)

Note that all lowercase characters are converted to uppercase before the password is encrypted. (Thus, **EAGLE** is the same as **eagle**.)

When the old and new passwords are entered, the user input is not echoed at the terminal (to help ensure secrecy). To protect against typing errors that are not seen when entering the new password, you must enter the new password twice. If the two entries of the new password do not match, the password remains unchanged.

The following guidelines are recommended to minimize the chances that passwords can be discovered by the trial-and-error method or by exhaustive search:

- Passwords should be at least six characters long.
- Names or words that are readily associated with any user should be avoided.
- Users should change passwords at least once every month.

One means to ensure that the above guidelines are met is to use the /GENERATE[=value] qualifier. This will generate random passwords of up to 12 characters in length. It is possible for the system manager to require individual users to use the /GENERATE qualifier. For more information about this, see the description of the AUTHORIZE utility in the *VAX/VMS Authorize Utility Reference Manual*.

QUALIFIERS

/GENERATE[=value]

Requests that passwords be generated for you. If the /GENERATE qualifier is used, the system will display five random passwords and ask you to select one of them. If you do not like any of the passwords displayed, pressing RETURN will cause five new passwords to be displayed.

The optional value determines the number of letters in the passwords generated. For any value, n, SET PASSWORD will generate passwords of from n to (n+2) characters in length. Values of 1 through 10 may be specified.

If no value is specified, SET PASSWORD will use a default value of 6, and will generate passwords of from 6 to 8 characters in length. If a value larger than 10 is specified, it will be ignored and a value of 10 (the largest acceptable value) will be used instead.

Note that if your system manager has established a minimum password length for your account, SET PASSWORD/GENERATE=n will compare that length with the length of the optional value, and use the larger of the two values.

/SECONDARY

Allows you to change your secondary password, or to create one if you do not currently have one. The procedure is the same as for 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.

You may remove your secondary password by specifying a RETURN, when SET PASSWORD/SECONDARY prompts you for a new password and verification. Once this is done, you will only receive a single PASSWORD: prompt when logging in.

Secondary passwords make it possible to set up an account, such that two different people, each knowing one of the two passwords, would be required to access that account.

The /SECONDARY and /SYSTEM qualifiers are incompatible.

/SYSTEM

Requires both SECURITY and CMKRNL privileges.

Indicates that you are changing the system password, rather than your user password.

If a terminal line has the system password (SYSPWD) characteristic set, no terminal prompts will be sent to that terminal until the system password is entered.

A system password is valid only for the node it is set on. In a VAXcluster, each node can have a different system password.

The /SYSTEM and /SECONDARY qualifiers are incompatible.

EXAMPLE

```
$ SET PASSWORD
Old password: HONCHO
New password: BIG_ENCHILADA
Verification: BIG_ENCHILADA
```

In response to the SET PASSWORD command, the system prompts for the old password, next for the new password. Then the system asks for the new password again for verification purposes. 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, the password is changed. Otherwise, an error message appears and the password remains unchanged.

Note that in a real session, neither the old password nor the new password and its verification appear on the screen or paper.

SET PRINTER

Establishes the characteristics of a specific line printer. The defaults listed below are the defaults for an initially bootstrapped system.

FORMAT

SET PRINTER *printer-name[:]*

restrictions

- Use of this command requires operator (OPER) privilege.
- If the printer is a spooled device, the logical I/O privilege (LOG_IO) is required to modify its characteristics.

PARAMETER

printer-name[:]

Specifies the name of a line printer that will have its characteristics set or modified. If the printer has been set to /SPOOLED, the logical I/O privilege (LOG_IO) is required to 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 /NOCR 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 or not 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

Allows the operator to set the appropriate printer type when the printer is an LA11 line printer. 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

Allows the operator to set the appropriate printer type when the printer is an LA180 line printer. 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.

/LOWERCASE
/NOLOWERCASE (default)

Indicates whether the printer prints both upper- 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, /LOWERCASE is equivalent to /NOUPPERCASE, and /NOLOWERCASE is equivalent to /UPPERCASE.

/LOG
/NOLOG (default)

Determines whether information confirming the printer setting is displayed at the terminal from which the SET PRINTER command was issued.

/LP11 (default)

Allows the operator to set the appropriate printer type when the printer is an LP11 line printer. 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=n

Establishes the number of lines per page on the currently installed form; the number of lines can range from 1 through 255. The printer driver uses this value to determine the number of line feeds that must be issued to simulate a form feed. (See the /FF\/NOFF qualifier description for SET PRINTER.)

The default value is 64 lines per page.

/PASSALL
/NOPASSALL (default)

Controls whether the system interprets special characters or passes them as 8-bit binary data.

If you specify /PASSALL, the printer does not expand tab characters to spaces, fill carriage return or line feed characters, or recognize control characters.

/PRINTALL
/NOPRINTALL (default)

Controls whether the line printer driver outputs printable 8-bit multinational characters.

/TAB
/NOTAB (default)

Controls how the printer handles TAB characters. The /NOTAB qualifier expands all tab characters to spaces and assumes tab stops at eight 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 VAX/VMS operating system requires that printers expand tabs at eight-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

Allows the operator to set the appropriate printer type when the printer is an unknown type. 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 (default)
/NOUPPERCASE

Indicates whether the printer prints both uppercase and lowercase letters or only uppercase ones. When you specify /UPPERCASE, all letters are translated to uppercase.

The /[NO]UPPERCASE and /[NO]LOWERCASE qualifiers are complementary; that is, /UPPERCASE is equivalent to /NOLOWERCASE, and /NOUPPERCASE is equivalent to /LOWERCASE.

/WIDTH=n

Establishes the number of characters per output line on currently installed forms. The width, n, can range from 0 through 65535 for LP11 controllers, and from 0 through 255 for DMF32 controllers.

The default value is 132 characters per line.

/WRAP
/NOWRAP (default)

Controls whether the printer generates a carriage return/line feed when it reaches the end of a line.

If the /NOWRAP qualifier is specified, the printer will write characters out in the last position on the line.

If the /WRAP qualifier is specified, the terminal generates a carriage return /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 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 establishes the line printer LPB0 as an LA11 printer.

3 `$ SET PRINTER/LOWERCASE LPA0:`

The SET PRINTER command requests that lowercase printing be enabled on line printer LPA0.

SET PROCESS

Changes the execution characteristics associated with the specified process for the current terminal session or job. If no process is specified, changes are made to the current process.

FORMAT

SET PROCESS *[process-name]*

restrictions

- Requires GROUP privilege to change other processes in the same group.
- Requires WORLD privilege to change processes outside your group.

PARAMETER

process-name

Specifies the name of the process for which the characteristics are to be changed. The process name can contain from 1 to 15 alphanumeric characters. Process names are linked to group numbers. The specified process must have the same group number in its user identification code (UIC) as the current process. 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.

If you specify the /IDENTIFICATION qualifier, the process name parameter is ignored. If you include neither the process name nor the /IDENTIFICATION qualifier, the current process is assumed.

The process name parameter is limited to use only with the /PRIORITY, /RESUME, and /SUSPEND qualifiers.

QUALIFIERS

/CPU=[NO]ATTACHED

Specifies whether the process is eligible or ineligible for scheduling on the attached CPU of any multiprocessor (MP) system (VAX–11/782, VAX 8300, VAX 8800). Since all kernel mode activities on an MP system run on the primary process, a process that frequently requests kernel mode service (I/O, pagefaulting, most other system services) is a good candidate for running /CPU=NOATTACHED.

/DUMP
/NODUMP (default)

If an image terminates due to an unhandled condition, the /DUMP qualifier causes the contents of the address space to be written to a file in your current default directory. The file name will be the same as the image which terminated. The file type will be DMP.

You can then use the ANALYZE/PROCESS_DUMP utility to analyze the dump.

/IDENTIFICATION=pid

Specifies the process identification value (PID) of the process for which characteristics are to be changed. 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.

The /IDENTIFICATION qualifier can be used only with the /PRIORITY, /RESUME, and /SUSPEND qualifiers.

/NAME=string

Changes the name of the current process to the specified name. The string parameter can have from 1 to 15 alphanumeric characters.

/PRIORITY=n

Requires GROUP or WORLD privilege to change the priority of other processes on the system.
Requires alter priority (ALTPRI) privilege to set the priority higher than the base priority of the current process.

Changes the priority for the specified process. 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.

GROUP or WORLD privilege is required to change the priority of other processes on the system.

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

Requires SETPRV to enable a privilege that you are not authorized to have.

Enables or disables the user privileges for the current process. If you specify only one privilege, you can omit the parentheses. See Table DCL–1-3 in the *VAX/VMS DCL Concepts Manual* for a list of the privilege keywords. The table also contains a description of each privilege.

If you do not have SETPRV, the privilege is not enabled, and a warning message is issued. Use the SHOW PROCESS/PRIVILEGES command to determine what privileges are currently enabled.

/RESOURCE_WAIT
/NORESOURCE_WAIT

Enables or disables resource wait mode for the current process.

If you specify /NORESOURCE_WAIT, the process will receive 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 limit
Buffered I/O limit
Buffered I/O byte count (buffer space) quota

/RESUME

Specifies that a process suspended by a previous SET PROCESS command is to be resumed.

/SUSPEND
/NOSUSPEND

Requires GROUP or WORLD privilege to use this qualifier.

Controls whether to suspend or resume the process.

If you specify /SUSPEND, the process is placed in a suspended state. You can specify /NOSUSPEND to have a previously suspended process resume operation.

/SWAPPING (default)
/NOSWAPPING

Requires the user privilege process swap privilege (PSWAPM) to disable swapping for your process.

Enables or disables process swap mode for the current process. By default, a process that is not currently executing can be removed from physical memory so that other processes can execute. If you specify /NOSWAPPING, the process is not swapped out of the balance set when it is in a wait state.

EXAMPLES

1
```
$ SET PROCESS/NORESOURCE_WAIT
```

The SET PROCESS command disables resource wait mode for the current process.

2
```
$ RUN/PROCESS_NAME=TESTER  CALC
%RUN-S-PROC_ID, identification of created process is 0005002F
$ SET PROCESS/PRIORITY=10  TESTER
```

The RUN command creates a subprocess and gives it the name TESTER. Subsequently, the SET PROCESS/PRIORITY command assigns the subprocess a priority of 10.

SET PROMPT

Enables you to have DCL use a different prompt string.

FORMAT

SET PROMPT *[=string]*

restrictions

None.

PARAMETER

string

Specifies the string to replace the default DCL prompt string ($). The string can consist of more than one character. All valid ASCII characters can be used in the string.

In order to include spaces or lowercase letters in your string, you must enclose the string in quotation marks. Otherwise, letters are automatically converted to uppercase, and leading and trailing spaces are removed.

If no string is specified with the SET PROMPT command, the DCL default prompt string ($) is restored.

DESCRIPTION

The SET PROMPT command enables you to customize prompts for your main process or a subprocess.

When the system prompts you for parameters that are missing from a command line, the prompt string is replaced with the prompt string found in the command tables for that process.

When a continued command is read from the terminal or an indirect command is read from a command procedure, an underscore is placed in front of the prompt string by DCL.

QUALIFIER

/CARRIAGE_CONTROL (default)
/NOCARRIAGE_CONTROL

Determines whether carriage return and line feed characters are inserted before the prompt string.

EXAMPLE

```
$ SET PROMPT ="DCL ---> "
DCL ---> SHOW TIME
 15-APR-1985 14:08:58
```

The dollar sign prompt is replaced with the string "DCL —> ". When you see the prompt on your screen, you can enter any DCL command. This example uses the SHOW TIME command.

SET PROTECTION

Establishes the protection to be applied to a particular file or a group of files. The protection of a file limits the type of access available to system users.

FORMAT

SET PROTECTION*[=(code)] file-spec[,...]*

restrictions

None.

PARAMETERS

code

Defines the protection to be applied to the specified files. If no code is included, the access of the specified files is set to the current default protection.

The format for specifying the code is described in Section 7.1.2 of the *VAX/VMS DCL Concepts Manual*.

file-spec[,...]

Specifies one or more files for which the protection is to be changed.

A file name and file type are required; if you omit a version number, the protection is changed for only the highest existing version of the file.

You can specify wildcard characters in the directory, file name, file type, and version fields.

DESCRIPTION

You can use the SET PROTECTION command to change or reset the access for one or more files. If you include a protection code, the file access is changed to that code. When you omit the protection code and do not use the /PROTECTION qualifier, the file access changes to the default file access established by the SET PROTECTION/DEFAULT command. See the SET PROTECTION/DEFAULT command for information on how to change the default file accesses.

All disk and tape volumes have protection codes that restrict access to the volume. The protection codes for disk and tape volumes are assigned with the INITIALIZE and MOUNT commands. They cannot be changed by the SET PROTECTION command.

For disk volumes, each file on the volume, including a directory file, can have a different protection associated with it. The SET PROTECTION command and other file manipulating commands allow you to define the protection for individual files.

If you omit both the code and the /PROTECTION file qualifier, your current default protection (established by the SET PROTECTION/DEFAULT command) is applied to the files.

QUALIFIERS

/CONFIRM
/NOCONFIRM (default)

Controls whether the SET PROTECTION command displays the file specification of each file before applying the new protection, and requests you to confirm that the file's protection should be changed. If you specify /CONFIRM, you must respond to the prompt with a Y (YES) or a T (TRUE), and then press RETURN before the SET PROTECTION command will change the file protection. If you enter anything else, such as N or NO, the requested file protection is not applied.

/LOG
/NOLOG (default)

Controls whether the SET PROTECTION command displays the file specification of each file after it has reset the protection.

/PROTECTION=(code)

File-spec qualifier.

Defines the protection code to be applied to the associated file specification. Use this qualifier to assign different protection codes to several files with a single SET PROTECTION command.

If you specify the command's code parameter in addition to using the /PROTECTION qualifier with a file specification, the attributes specified with the command's code parameter are applied first. Any attributes specified with the /PROTECTION qualifier override the command's code parameter attributes.

Specify the protection code using the format described in Section 7.1.2 of the *VAX/VMS DCL Concepts Manual*.

EXAMPLES

1

```
$ DELETE INCOME.DAT;3
%DELETE-W-FILNOTDEL, error deleting DISK1:[SMITH]INCOME.DAT;3
-RMS-E-PRV, insufficient privilege or file protection violation
$ SET PROTECTION=OWNER:D INCOME.DAT;3
$ DELETE INCOME.DAT;3
```

The file INCOME.DAT;3 has been protected against deletion. This SET PROTECTION command changes the owner's access to delete for the file INCOME.DAT;3 only. Now the file can be deleted.

2

```
$ SET PROTECTION -
$_PAYROLL.LIS/PROTECTION=(SYSTEM:R,OWNER:RWED,GROUP:RW),-
$_PAYROLL.OUT/PROTECTION=(SYSTEM:RWED,GROUP:RWED,W)
```

The SET PROTECTION command changes the protection codes applied to two files. To the file PAYROLL.LIS, it gives the system read-only access; the owner read, write, execute, and delete access; and users in the owner's group read and write access. To the file PAYROLL.OUT, it gives the system and group all types of access; the current access for the owner does not change, but the world is denied all types of access.

3
```
$ SET PROTECTION A.DAT, B.DAT/PROTECTION=OWNER:RWED, C.DAT
```

The SET PROTECTION command specifies that the file A.DAT receive the default protection established for your files. The existing protection for the file B.DAT is overridden, only for the owner category, to provide read, write, execute, and delete access. Note that no protection is specified for the file C.DAT at either the command or file level. Thus, like A.DAT, C.DAT receives the default protection.

Since no version numbers are specified, the protection settings affect only the highest versions of the three files.

4
```
$ SET PROTECTION=OWNER:D -
$_[MALCOLM.SUB1]SUB2.DIR/PROTECTION=GROUP:D
```

The SET PROTECTION command changes the protection for the owner and group categories of the subdirectory [MALCOLM.SUB1.SUB2] to permit deletion. However, the protection for the world and system categories is not changed.

5
```
$ DIR/PROTECTION INCOME.DAT
Directory DBA0:[SMITH]
INCOME.DAT;2          (RWED,RWED,RWED,RWED)
INCOME.DAT;1          (RWED,RWED,RWED,RWED)
Total of 2 files.
$ SET PROTECTION=(OWNER:RWE) INCOME.DAT;1
$ PURGE
```

The file INCOME.DAT;1 has been protected against deletion by the owner. However, since the owner is also a member of the group and world categories, the file is still vulnerable to deletion. The subsequent PURGE command will delete INCOME.DAT;1.

In order to protect the file against deletion by you (the owner), you also need to protect the file against deletion by all outer access categories. The following command shows the proper way to do this.

```
$ SET PROTECTION=(OWNER:RWE,GROUP:RWE,WORLD:RWE) INCOME.DAT;1
```

SET PROTECTION/DEFAULT

Establishes the default protection for all files subsequently created during the terminal session or batch job. The protection for a file limits the type of access available to system users. The /DEFAULT qualifier is required.

FORMAT

SET PROTECTION*[=(code)]/DEFAULT*

restrictions

None.

PARAMETER

code

Defines the protection to be applied to all files subsequently created in cases where a different protection is not specified with the SET PROTECTION or CREATE commands. The format for specifying the protection code is described in Section 7.1.2 of the *VAX/VMS DCL Concepts Manual*.

If you fail to specify a protection code, the current default protection remains unchanged.

EXAMPLE

```
$ SET PROTECTION=(GROUP:RWED,WORLD:R)/DEFAULT
```

The SET PROTECTION/DEFAULT command sets the default protection applied to all files subsequently created in this terminal session or batch job, allowing other users in the same group unlimited access and all users read access. The default protections for system and owner are not changed.

SET PROTECTION/DEVICE

Establishes the protection to be applied to a specific non-file-structured device. The protection for a device limits the type of access available to users. The /DEVICE qualifier is required.

FORMAT

SET PROTECTION*[=code]/DEVICE device-name[:]*

restrictions

Requires operator (OPER) privilege.

PARAMETERS

code

Establishes the protection code for a device. The protection code defines the user and type of access allowed the user. The code should be specified according to the syntax rules explained in the Description section below.

Only those protection code categories specified for the code parameter will be changed. Any protection code category that the operator does not specify will remain unchanged.

device-name[:]

Specifies the name of the device whose protection is to be set or modified. The device must be a non-file-structured device.

DESCRIPTION

The same four user categories that apply to files also apply to devices: SYSTEM, OWNER, GROUP, and WORLD.

- System—all users who have group numbers of 0 through 10 octal and users with physical or logical I/O privilege (generally, system managers, system programmers, and operators). The octal group numbers 0 through 10 for system users are the default group numbers. The group number parameter can be changed at system generation time to any octal value from 0 through 377.
- Owner—the user identification code (UIC) of the user who "owns" the device and is issuing the SET PROTECTION command.
- Group—all users who have the same group number in their UICs as the owner of the device.
- World—all users who do not fall into the categories above.

For shareable devices (such as the LPA11-K), each user category can be allowed or denied one of the following types of access:

- Read—the right to issue read requests to the device
- Write—the right to issue write requests to the device
- Logical I/O—the right to issue logical I/O requests to the device
- Physical I/O—the right to issue physical I/O requests to the device

For nonshareable devices, such as terminals and card readers, each category of user can either be allowed or denied access to allocate and assign channels to the device. The Read category controls whether a user can allocate and assign channels to the device. All other categories are not relevant for nonshareable devices.

Any combination of access types can be specified for any category of user.

When the operator specifies a user access code, the code must be abbreviated to one character. The abbreviations are:

Read	R
Write	W
Logical I/O	L
Physical I/O	P

The user access codes and user categories can be specified in any order. If you specify a user category without including any user access code, that category of user is denied all types of access. When you specify one or more access codes with a user category, that user category receives only those specified types of access. If you omit a user category, the access for that category is unchanged.

To specify a protection code, separate the user category from the access type with a colon. To specify more than one user category, separate each category by a comma and enclose the entire protection code specification in parentheses.

QUALIFIER

/OWNER_UIC=uic

Requests that the specified user identification code (UIC) be assigned ownership of the device for the purpose of access checks. The default owner is the UIC of the process issuing the SET PROTECTION command.

Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

EXAMPLES

1
```
$ SET PROTECTION=(S:RWLP,O:RWLP,G,W)/DEVICE  LAA0:
```

This command requests that the protection for device LAA0 be set to allow all types of access to system processes and processes with the UIC of the current process, and to deny access to anyone else.

2
```
$ SET PROTECTION=(S,O:RWLP,G,W)/DEVICE/OWNER_UIC=[103,4] LAB0:
```

This command requests that the protection for device LAB0 be set to allow all types of access to processes with a UIC of [103,4], and to deny access to anyone else.

3
```
$ SET PROTECTION=(S:R,O,G,W)/DEVICE/OWNER_UIC=[1,4] TTA1:
```

This command requests that the protection for the terminal TTA1 be set to allow only system processes to allocate the device, and denies access to anyone else. This type of protection is recommended for interactive terminals if system security is necessary. Note that the above protection code restricts which users can allocate the device, but does not restrict users from logging in to the device.

SET QUEUE

Changes the current status or attributes of the specified queue.

FORMAT

SET QUEUE *queue-name[:]*

restrictions

Requires operator (OPER) privilege or execute (E) access to the specified queue.

PARAMETERS

queue-name[:]

Specifies the name of an execution queue or a generic queue.

DESCRIPTION

After a printer or batch queue has been created by the INITIALIZE/QUEUE command, you can use the SET QUEUE command to change the attributes or status of the queue without having to stop the queue.

The defaults for the SET QUEUE parameters depend on the parameters 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 did not specify the /JOB_LIMIT qualifier with the SET QUEUE command, the job limit would remain at 3 for that queue.

QUALIFIERS

/BASE_PRIORITY=n

Specifies the base process priority at which jobs are initiated from a batch queue. (You must stop and restart symbiont queues to change the symbiont priority for printer, terminal, or server queues.) The n specifier can be any decimal value from 0 through 15.

/BLOCK_LIMIT=([lowlim,]uplim)
/NOBLOCK_LIMIT

Limits the size of print jobs that can be executed on a printer or terminal 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 will be 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 job limit for the queue is changed, enabling it to execute.

The uplim parameter is a decimal number referring to the maximum number of blocks that will be accepted by the queue for a print job. If a print job is submitted that exceeds this value, the job remains until the job limit for the queue is changed, enabling it to execute.

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 will execute in the queue. In order to specify only a lower job limit, you must use two sets of double quotation marks to indicate the upper specifier. For example, /BLOCK_LIMIT=(500,"") means any job with 500 or more

blocks will execute 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 will not run in the queue.

The /NOBLOCK_LIMIT qualifier cancels the /BLOCK_LIMIT setting previously established for that queue.

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

Specifies one or more characteristics for processing jobs on the queue. If only one characteristic is specified, you can omit the parentheses.

Each time you specify /CHARACTERISTICS, all previously set characteristics are erased. Only the ones specified with the qualifier are now established for the queue.

Queue characteristics are installation-specific. The characteristic parameter can be either a value from 0 through 127 or a characteristic name that has been defined by the DEFINE/CHARACTERISTIC command.

When users include the /CHARACTERISTICS qualifier with a PRINT or SUBMIT command, all the characteristics they specify must also be specified for the queue that will be executing the job. If not, the job will remain pending in the queue until the queue characteristics are changed or they delete the entry with the DELETE/ENTRY command. Users need not specify every characteristic of a queue with a PRINT or SUBMIT command as long as the ones they specify are a subset of the characteristics set for that queue. The job will also run if no characteristics are specified.

The /NOCHARACTERISTICS qualifier cancels any /CHARACTERISTIC settings previously established for that queue.

/CPUDEFAULT=time

Defines the default CPU time limit for batch jobs.

Specify the CPU default time as a delta time value, the numeric value 0, or the keyword NONE or INFINITE. The value 0 and the keyword INFINITE allow a job unlimited CPU time, subject to restrictions imposed by the /CPUMAXIMUM qualifier or the user authorization file (UAF). Specify NONE when a default CPU time limit is not needed.

The time cannot exceed the CPU time limit set by the /CPUMAXIMUM qualifier. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for information on specifying delta times.

/CPUMAXIMUM=time

Defines the maximum CPU time limit for batch jobs. Use this qualifier to override the CPU time limit specified in the user authorization file (UAF).

Specify CPU maximum time as a delta time value, the numeric value 0, or the word NONE or INFINITE. Specify NONE when a maximum CPU time limit is not desired. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for information on specifying delta times.

A CPU time limit for processes is specified by each user record in the system (UAF). You can also specify the following: a default CPU time limit for all jobs in a given queue and a maximum CPU time limit for all jobs in a given queue. Refer to Table DCL–1 for more information on specifying CPU time limits.

/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. Once an option is set for the queue by the /DEFAULT qualifier, users do not have to specify that option in their PRINT commands.

The options are:

[NO]BURST[=keyword]	Specifies whether file burst pages will be printed. If the keyword is ALL (the default), burst pages are placed before each file in the print job. If the keyword is ONE, a burst page is placed before the first copy of the first file in the job.
[NO]FEED	Specifies whether a form-feed is automatically inserted at the end of a page.
[NO]FLAG[=keyword]	Specifies whether file flag pages will be printed. If the keyword is ALL (the default), flag pages are placed before each file in the print job. If the keyword is ONE, a flag page is placed before the first copy of the first file in the job.
FORM=type	Specifies the default form for a printer, terminal, or server queue. If a job is not submitted with an explicit form definition, then this form will be used to process the job. The systemwide default form, form=0, is the default value for this keyword. See also /FORM_MOUNTED.
[NO]TRAILER[=keyword]	Specifies whether file trailer pages will be printed. If the keyword is ALL (the default), trailer pages are placed at the end of each file in the print job. If the keyword is ONE, a trailer page is placed after the last copy of the last file in the job.

If you specify any of the keywords BURST, FLAG, TRAILER without specifying a value, the value ALL is used by default.

/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 have specific targets can be placed in this execution queue for processing.

/FORM_MOUNTED=type

Specifies the form type for a printer, terminal, or server queue. If the stock of the mounted form is not identical to the stock of the default form, as indicated by the DCL command qualifier /DEFAULT=FORM=type, then all jobs submitted to this queue without an explicit form definition will enter a pending state. 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, then the job will enter a pending state. In both cases, the pending state will be maintained

until the stock of the mounted form of the queue is identical to the stock of the form associated with the job.

Specify the form type using a numeric value or a form name that has been defined by the DEFINE/FORM command. Form types are installation-specific.

/JOB_LIMIT=n

Indicates the number of batch jobs that can be executed concurrently from the queue.

/OWNER_UIC=uic

Requires OPER privilege.

Enables you to change the user identification code UIC of the queue. Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

/PROTECTION=(ownership[:access],...)

Requires OPER privilege.

Specifies the protection of the queue. By default, the queue protection is (SYSTEM:E, OWNER:D, GROUP:R, WORLD:W). If you include only one protection code, you can omit the parentheses.

/RECORD_BLOCKING
/NORECORD_BLOCKING

Determines whether the symbiont can concatenate (or block together) output records for transmission to the output device. If you specify /NORECORD_BLOCKING, the symbiont is directed to send each formatted record in a separate I/O request to the output device. For the standard VAX/VMS print symbiont, record blocking can have a significant performance advantage over single-record mode.

/RETAIN[=option]
/NORETAIN

Specifies that jobs be retained in the queue in a completed status after they have executed.

The option parameter can be the keyword ALL or ERROR. If ALL is specified, all jobs are retained in the queue after execution. If ERROR is specified, only jobs that have completed unsuccessfully are retained in the queue. If you omit the option parameter, ALL is assumed.

The /NORETAIN qualifier enables you to reset the queue to the default.

/SCHEDULE=[NO]SIZE

Specifies whether pending jobs in a printer or terminal queue are scheduled for printing based on the size of the job. When /SCHEDULE=SIZE is in effect, shorter jobs will print before longer ones. With /SCHEDULE=NOSIZE, jobs are printed in the order they were submitted, regardless of size.

If you issue this command while there are pending jobs in any queue, its effect on future jobs is unpredictable.

/SEPARATE=(option[,...])
/NOSEPARATE

Specifies the job separation defaults for a printer or terminal queue. The job separation options are:

[NO]BURST	Specifies whether a burst page will be printed at the beginning of every job. Specifying BURST also results in a flag page being printed.
[NO]FLAG	Specifies whether a flag page will be printed at the beginning of every job.
[NO]TRAILER	Specifies whether a trailer page will be printed at the end of every job.
[NO]RESET=(module[,...])	Specifies a job reset sequence for the queue. The specified modules from the device control library are used to reset the device each time a job reset occurs.

/WSDEFAULT=n

Defines a working set default for a batch 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 a positive integer in the range 1 through 65,535, 0, or the word NONE as the value for n. If 0 or NONE is specified for n, the working set default value defaults to the value specified either in the UAF or by the SUBMIT command (if specified). For more information refer to Table DCL–2.

/WSEXTENT=n

Defines a working set extent for the batch 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 a positive integer in the range 1 through 65,535, 0, or the word NONE as the value for n. If 0 or NONE is specified for n, the working set extent value defaults to the value specified either in the UAF or by the SUBMIT command (if specified). For more information refer to Table DCL–2.

/WSQUOTA=n

Defines the working set page size (working set quota) for a batch 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 a positive integer in the range 1 through 65,535, 0, or the word NONE as the value for n. If 0 or NONE is specified for n, the working set quota value defaults to the value specified either in the UAF or by the SUBMIT command (if specified). For more information refer to Table DCL–2.

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), and can be specified for individual jobs and/or for all jobs in a given queue. The decision table (Table DCL–2) shows the action taken for different combinations of specifications that involve working set size and working set quota values.

EXAMPLE

```
$ INITIALIZE/QUEUE/DEFAULT=BURST/FORM_MOUNTED=LETTER/START SYS$PRINT
    .
    .
    .
$ STOP/QUEUE/NEXT SYS$PRINT
$ SET QUEUE /DEFAULT=BURST/FORM_MOUNTED=MEMO SYS$PRINT
```

The queue is initialized with the INITIALIZE/QUEUE command to have the following attributes: a burst page preceding each file in the job and the form type LETTER. Later the queue is stopped with the STOP/QUEUE/NEXT command so that the current job finishes processing before the queue stops. Then the SET QUEUE command is issued to change the form type to MEMO.

```
$ SET QUEUE/DEFAULT=FORM=LN01_PORTRAIT LN01_PRINT
```

The SET QUEUE command changes the default form to LN01_PORTRAIT for the LN01_PRINT queue.

SET QUEUE/ENTRY

Changes the current status or attributes of a job that is not currently executing in a queue. The /ENTRY qualifier is required.

FORMAT

SET QUEUE/ENTRY=*entry-number queue-name[:]*

restrictions

Requires operator (OPER) privilege or execute (E) access to the specified queue. If you have D access to the specified job, you can alter the attributes for that job.

PARAMETERS

entry-number

Specifies the entry number of the job you want to change.

queue-name[:]

Specifies the name of the queue in which the specified job is entered.

DESCRIPTION

The SET QUEUE/ENTRY command allows you to change the status or attributes of a job that has been submitted to a printer or batch queue, as long as the job is not currently executing. (You cannot affect individual files within a multifile job with the SET/QUEUE/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 QUEUE/ENTRY qualifiers are the attributes and status that the job has before you issue the SET QUEUE/ENTRY command.

The system assigns a unique entry number to each queued print or batch job in the system. The PRINT and SUBMIT commands display the job number when they successfully queue a job for processing. You can issue the SHOW QUEUE command to refresh your memory about a job's entry number. Use the job entry number to specify which job you want to change.

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 queued for immediate processing.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values.

/BURST[=keyword]
/NOBURST

Controls whether a burst page is included at the beginning of a print job. A burst page precedes a flag page and contains the same information. However, it is printed over the perforation between the burst page and the flag page.

The printing on the perforation makes it easy to separate individual print jobs.

When you specify /BURST, you need not specify /FLAG; a flag page will automatically follow the burst page.

You can specify one of the following keywords:

ALL	All printed files contain a burst page.
ONE	The first printed file contains a burst page.

Use the /[NO]BURST qualifier to override the installation-defined defaults that have been set for the printer queue you are using.

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

Enables you to change the characteristics desired for the job. If you specify only one characteristic, you can omit the parentheses. Codes for characteristics can be either names or values from 0 to 127 and are installation-defined. Use the SHOW QUEUE/CHARACTERISTICS command to see which characteristics have been defined for your system. Use the SHOW QUEUE/FULL command to see which characteristics are available on a particular queue.

When you include the /CHARACTERISTICS qualifier with the SET QUEUE /ENTRY command, all the characteristics you specify must also be specified for the queue that will be executing the job. If not, the job will remain pending in the queue until the queue characteristics are changed or you delete the entry with the DELETE/ENTRY command. You need not specify every characteristic of a queue with the SET QUEUE/ENTRY command as long as the ones you specify are a subset of the characteristics set for that queue. The job will also run if no characteristics are specified.

Specification of a characteristic for a queue does not prevent jobs that do not specify that characteristic from being executed.

/CLI=filename

Enables you to specify a different command language interpreter (CLI) to use in processing the 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's authorization record, or whatever CLI was specified when the job was originally submitted to the queue.

/COPIES=n

Specifies the number of copies to print. The n parameter can be any number from 1 to 255.

When you use the /COPIES qualifier with the SET QUEUE/ENTRY command, the number of copies can apply only to the entire job. You cannot use this qualifier to specify different numbers of copies for individual files within a multifile job.

/CPUTIME=option

Defines a CPU time limit for the batch job. You can specify a delta time (see Section 2.5 of the *VAX/VMS DCL Concepts Manual*), the value 0, or the keyword NONE or INFINITE for n.

When you need less CPU time than authorized, use the /CPUTIME qualifier to override the base queue value established by the system manager or the value authorized in your user authorization file. Specify 0 or INFINITE to request an infinite amount of time. Specify NONE when you want the CPU time to default to your user authorization file (UAF) value or the limit specified on the queue. Note that you cannot request more time than permitted by the base queue limits or your own UAF.

/FEED
/NOFEED

Controls whether form feeds are inserted into print jobs when the printer nears the end of a page. The number of lines per form can be reset by the /FORM qualifier. 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 QUEUE/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 multi-file job.

/FLAG[=keyword]
/NOFLAG

Controls whether a flag page is printed preceding a print job. The flag page contains the name of the user submitting the job, the job entry number, and other information about the job. You can specify one of the following keywords:

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

Use the /[NO]FLAG qualifier to override the installation-defined defaults that have been set for the printer queue you are using.

/FORM=type

Specifies the name of the form that you want for the print job.

Specify the form type using a numeric value or alphanumeric code. Form types can refer to the width, length, or type of paper. Codes for form types are installation-defined. You can use the SHOW QUEUE/FORM command to find out the form types available for your system. The SHOW QUEUE/FULL command tells you which form is set for a specific queue.

If you specify a form type different from that of the queue, your job remains pending until the form type of the queue is set equal to the form type of the job or you delete the job with the DELETE/ENTRY command. You can use the SET QUEUE/ENTRY to change the form type of your job to match that of the queue so your job can be printed. In order to have the form type for the queue changed, request that the system manager stop the queue, physically change the form type of the printer, and restart the queue specifying the new form type.

/HEADER
/NOHEADER

Controls whether a heading line is printed at the top of each output page in a print job.

/HOLD
/NOHOLD

Controls whether or not the job is to be made available for immediate processing or held for processing later.

If you specify /HOLD, the job is not released for processing until you specifically release it with the /NOHOLD or /RELEASE qualifier. You can use the SET QUEUE/ENTRY command to release a job that was previously submitted with a /HOLD qualifier or you can place a job on hold so that it will run later.

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 held in a queue that has /RETAIN specified.
- A user-written symbiont has refused a job.

/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 specified or defaulted with the PRINT command.

/KEEP
/NOKEEP

Controls whether the batch job log file is deleted after it is printed.

/LOG_FILE=file-spec
/NOLOG_FILE

Controls whether a log file with the specified name is created for the batch job or whether a log file is created.

When you use the /LOG_FILE qualifier, the system writes the log file to the file you specify. If you use /NOLOG_FILE, no log file is created. If neither form of the qualifier has been used for the job, the log file is written to a file in the default directory that has the same file name as the first command file and a file type of LOG.

You can use the /LOG_FILE qualifier to specify that the log file be written to a different device. Logical names in the file specification are translated in the context of the process that executes the SET QUEUE/ENTRY command. 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; the file type is LOG. When you omit the /LOG_FILE qualifier, the job-name value used with /NAME must be a valid file name.

/LOWERCASE
/NOLOWERCASE

Indicates whether the files 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 supporting only uppercase letters. If all

available printers can print both uppercase and lowercase letters, you do not need to specify /LOWERCASE.

/NAME=job-name

Defines a name string to identify the job. The name string can have from 1 to 39 characters. The job name is used in the SHOW QUEUE command display. 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.

If the /NAME qualifier has not been specified for the job, the name string defaults to the file name of the first, or only, file in the job; the file type is LOG.

/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 no /DELETE qualifier was specified when the job was originally submitted to the queue, you cannot use the SET QUEUE/ENTRY to establish file deletion at a later time.

You cannot use the /NODELETE qualifier to specify that individual files in a multi-file job not be deleted.

/NOTE=string

Allows you to specify a message to appear on the flag page for the print job. The string can contain up to 255 characters.

/NOTIFY
/NONOTIFY

Controls whether a message is broadcast to any terminal at which you are logged in, notifying you when your job has been completed or aborted.

/OPERATOR=string

Allows you to specify a message to be sent to the operator. The string can contain up to 255 characters.

When the job begins execution, the queue pauses and the message is transmitted to the operator.

/PAGES=([l,]u)

Specifies the number of pages to print for the specified job. You can use the /PAGES qualifier to print portions of a long file.

When you use the /PAGES qualifier with the SET QUEUE/ENTRY command, the qualifier can only apply to an entire job. You cannot use this qualifier to specify different numbers of pages to be printed for individual files within a multi-file job.

The l (lower) specifier refers to the first page in the group of pages that you want printed for that job. If you omit the l specifier, the printing starts on the first page of the job. The u (upper) specifier refers to the last page of the file that you want printed. When you want to print to the end of the file but do not know how many pages that will be, you can use "" as the u specifier. You can omit the parentheses when you specify only a value for u. For example, /PAGES=10 prints the first 10 pages of the job; /PAGES=(5,10) prints pages 5 through 10; /PAGES=(5,"") starts printing at page 5 and continues until the end of the job is reached.

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

Specifies from 1 to 8 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. The commas delimit individual parameters. 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 through 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 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, will be ignored.

When you use the /PASSALL qualifier with the SET QUEUE/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

Controls whether the batch job log is queued for printing when your job is completed. The /PRINTER qualifier allows you to specify a particular printer queue.

If you specify /NOPRINTER, /KEEP is assumed.

/PRIORITY=n

Specifies the priority of the job. The priority value must be in the range of 0 through 255, where 0 is the lowest priority and 255 is the highest.

The default value for /PRIORITY is the value of the SYSGEN parameter DEFQUEPRI. You must have either OPER (operator) or ALTPRI (alter priority) privilege to raise the priority value above the value of the SYSGEN parameter MAXQUEPRI. No privilege is needed to set the priority lower than the MAXQUEPRI value.

/RELEASE

Releases a previously held job for processing. You can use this qualifier to release jobs that have been held for the following reasons:

- A job was submitted with the /HOLD qualifier.
- A job was submitted with the /AFTER qualifier.
- A completed job is being held in a queue that has /RETAIN specified.
- A user-written symbiont has refused a job.

/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 will be restarted after a system crash or a STOP/QUEUE/REQUEUE command.

/SETUP=module[,...]

Calls for the specified modules to be extracted from the device control library and copied to the printer before a job is printed.

When you use the /SETUP qualifier with the SET QUEUE/ENTRY command, the qualifier applies to the entire job. You cannot use this qualifier to specify different setup modules for individual files within a multi-file job.

/SPACE
/NOSPACE

Controls whether output is to be double-spaced.

When you use the /SPACE qualifier with the SET QUEUE/ENTRY command, the qualifier applies to the entire job. You cannot use this qualifier to specify different spacing for individual files within a multi-file job.

/TRAILER[=keyword]
/NOTRAILER

Controls whether a trailer page is printed at the end of a job. The trailer page displays the job entry number, as well as information about the user submitting the job.

When you use the /TRAILER qualifier with the SET QUEUE/ENTRY command, trailer pages are placed at the end of each file in a multi-file job. You can specify one of the following keywords:

ALL	All printed files contain a trailer page.
ONE	The last printed file contains a trailer page.

Use the /[NO]TRAILER qualifier to override the installation-defined defaults that have been set for the printer queue you are using.

/WSDEFAULT=n

Defines a working set default for a batch job. You can specify a positive integer in the range 1 through 65,535, 0, or the word NONE for n.

Use this qualifier to override the base queue value established by the system manager or the value authorized in the user authorization file (UAF), provided you want to impose a lower value. Specify 0 or NONE if you want the working set value defaulted to either the UAF value or the working set quota specified on the queue. You cannot request a value higher than the default.

/WSEXTENT=n

Defines a working set extent for a batch job. You can specify a positive integer in the range 1 through 65,535, 0, or the word NONE for n.

Use this qualifier to override the base queue value established by the system manager or the value authorized in the user authorization file (UAF), provided you want to impose a lower value. Specify 0 or NONE if you want the working set extent defaulted to either the UAF or the working set extent specified on the queue. You cannot request a value higher than the default.

/WSQUOTA=n

Defines the maximum working set size for a batch job. This is the working set quota. You can specify a positive integer in the range 1 through 65,535, 0, or the word NONE for n.

Use this qualifier to override the base queue value established by the system manager or the value authorized in the user authorization file (UAF), provided you want to impose a lower value. Specify 0 or NONE if you want the working set quota defaulted to either the user authorization file value or the working set quota specified on the queue. You cannot request a value higher than the default.

EXAMPLES

1

```
$ PRINT/HOLD   MYFILE.DAT
  Job MYFILE (queue SYS$PRINT, entry 112) holding
    .
    .
    .
$ SET QUEUE/ENTRY=112/RELEASE/JOB_COUNT=3
```

The PRINT command requests that the file MYFILE.DAT be queued to the system printer, but placed in a hold status. The SET QUEUE/ENTRY command releases the file for printing and changes the number of copies of the job to three.

2

```
$ SUBMIT  WEATHER
  Job WEATHER (queue SYS$BATCH, entry 210) pending
$ SUBMIT  CLIMATE
  Job CLIMATE (queue SYS$BATCH, entry 211) pending
$ SET  QUEUE/ENTRY=211/HOLD/NAME=TEMP  SYS$BATCH
```

The two SUBMIT commands queue command procedures for batch processing. The system assigns them job numbers of 210 and 211, respectively. The SET QUEUE/ENTRY command places the second job in a hold state and changes the job name to TEMP, assuming that job 211 had not yet begun.

```
3   $ 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 QUEUE/ENTRY=172 /BURST/NOSPACE/HEADER SYS$PRINT
```

The PRINT command 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 QUEUE/ENTRY command is issued. This command calls for a burst page at the beginning of each file and requests that all files in the job be single-spaced. Headers are printed on each page of each file in the job.

SET RESTART_VALUE

Establishes a test value for restarting portions of batch jobs. If the command is encountered by the system interactively, no action is taken. Use the SET RESTART_VALUE in command procedures.

FORMAT

SET RESTART_VALUE=*string*

restrictions

None.

PARAMETER

string

Specifies the test value to be used in the batch job. The string can contain up to 255 characters.

DESCRIPTION

The SET RESTART_VALUE command is used in command procedures to restart a job in the middle after a system interruption has occurred. The command relies on the values of two DCL symbols: $RESTART and BATCH$RESTART. $RESTART is set to false if a batch job is executing for the first time. If the batch job is rerunning, because the system failed or because the job was requeued, the value for $RESTART is set to true.

BATCH$RESTART has no definition if no SET RESTART_VALUE command has ever been executed by the batch job. Otherwise, the BATCH$RESTART definition has the value of the string parameter that was used with the most recently executed SET RESTART_VALUE command.

If a batch job has SET RESTART_VALUE commands in it, but you want the job to run in its entirety, issue the SET QUEUE/ENTRY/NOCHECKPOINT command to clear the BATCH$RESTART.

EXAMPLE

```
$ SET RESTART_VALUE = FIRSTPART
$ IF $RESTART THEN GOTO 'BATCH$RESTART
   .
   .
   .
$ FIRSTPART:
$ RUN PART1
   .
   .
   .
$ SET RESTART_VALUE = SECONDPART
$ SECONDPART:
$ RUN PART2
   .
   .
   .
```

The first command in the procedure sets the BATCH$RESTART value to FIRSTPART. The second command states that, if $RESTART is true, proceed to the value contained in BATCH$RESTART. ($RESTART will be 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 $FIRSTPART: line marks a target for the GOTO statement. The following line contains the command to run PART1.EXE.

The second SET RESTART_VALUE command sets the BATCH$RESTART value to SECONDPART. The following line marks another target for the GOTO statement. The last line shown contains the command to run PART2.EXE.

When the job is first submitted, the $RESTART value is false, so the IF—GOTO command line is ignored. If the job is stopped during the run of PART1.EXE, the value of BATCH$RESTART is FIRSTPART. When the job is reinitiated, the value of $RESTART is true. Thus, the IF—GOTO command is processed and transfers to the $FIRSTPART: line 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 reinitiated, the value of $RESTART is true. In this instance, the IF—GOTO command transfers to the $ SECONDPART: line in the procedure so that PART2.EXE can be run. PART1.EXE is not rerun.

SET RIGHTS_LIST

Allows users to modify the process or system rights list. You must specify either /DISABLE or /ENABLE with the SET RIGHTS_LIST command.

FORMAT

SET RIGHTS_LIST *id-name[,...]*

restrictions

See the Description section for an explanation of the privileges required to use the SET RIGHTS_LIST command.

PARAMETER

id-name[,...]

Identifiers to be added to or removed from the process or system rights list. **Id-name** is a string of 1 to 31 alphanumeric characters, underscores, and dollar signs; each name must contain at least one nonnumeric character.

DESCRIPTION

Use the SET RIGHTS_LIST command to modify identifiers in your current process rights list, the rights list of another process on the system, or the system rights list. The following guidelines can be used 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 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 privileges.

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. Valid keywords are:

[NO]DYNAMIC	Indicates whether or not unprivileged holders of the identifiers may add or remove them from the process rights list. The default is NODYNAMIC.
[NO]RESOURCE	Indicates whether or not holders of the identifiers may charge resources to them. The default is NORESOURCE.

/DISABLE

Removes the identifiers from the process or system rights list. You cannot use /DISABLE with the /ENABLE qualifier.

/ENABLE

Adds the identifiers to the process or system rights list. You cannot use /ENABLE with the /DISABLE qualifier.

/IDENTIFICATION=pid

Specifies the process identification value (PID) 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 /IDENTIFICATION 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 /PROCESS 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 /SYSTEM with /PROCESS or /IDENTIFICATION.

EXAMPLES

1
```
$ SET RIGHTS_LIST/ENABLE/ATTRIBUTES=RESOURCE MARKETING
```

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-NOCMKRNL, operation requires CMKRNL privilege
$ SET PROCESS/PRIVILEGES=(CMKRNL,SYSNAM)
$ SET RIGHTS_LIST/ENABLE/SYSTEM PHYSICS101
```

Adds the PHYSICS101 identifier to the system rights list. You must have both the CMKRNL and SYSNAM privilege to modify the system rights list.

SET RMS_DEFAULT

Defines default values for the multiblock and multibuffer counts, network transfer sizes, prologue level, and extend quantity used by VAX RMS for file operations. Defaults are set for sequential, indexed sequential, or relative file organizations on a process-only basis, unless a systemwide basis is requested.

FORMAT

SET RMS_DEFAULT

restrictions *None.*

PARAMETERS

None.

DESCRIPTION

Multiblocking and multibuffering of file operations can enhance the speed of input/output operations with VAX 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 *VAX/VMS System Services Reference Manual.*

For indexed-sequential files, SET RMS_DEFAULT defines default prologue level options.

For sequential files, SET RMS_DEFAULT defines default extensions. If a default extension is not specified in your program, the process or system default is used.

For network operations, SET RMS_DEFAULT defines network buffer sizes for transfer.

QUALIFIERS

/BLOCK_COUNT=count

Specifies a default multiblock count for sequential file operations to and from a disk. The specified count, representing the number of blocks to be allocated for each I/O buffer, can range from 0 through 127.

If you specify 0, RMS uses the process default value. If this value is 0, RMS then uses the system default value. If the system default value is also 0, then RMS uses a value of 1.

The /BLOCK_COUNT qualifier applies only to record I/O operations, not block I/O operations.

For more information on multiblock count, see the description of the RAB$B_MBC in the *VAX Record Management Services Reference Manual.*

/BUFFER_COUNT=count

Specifies a default multibuffer count for file operations. The specified count, representing the number of buffers to be allocated, can range from 0 through 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 /BUFFER_COUNT is specified without any of these qualifiers, /SEQUENTIAL is assumed.

If you specify 0, VAX RMS uses the process default value. If this value is 0, RMS then uses the system default value. If the system default value is also 0, then RMS uses a value of 1.

For more information on multibuffer count, see the description of the RAB$B_MBF in the *VAX Record Management Services Reference Manual*.

/DISK

Indicates that the specified defaults are to be applied to file operations on disk devices. If /SEQUENTIAL is specified, /DISK is assumed together with /MAGTAPE and /UNIT_RECORD.

/EXTEND_QUANTITY=n

Specifies the number of blocks (n) to extend a sequential file. You can specify a value from 0 to 65,535.

The /EXTEND_QUANTITY qualifier is used when the program does not specify an extent quantity.

If you omit the value specification or if you specify a value of 0, VAX RMS calculates its own /EXTEND_QUANTITY value.

/INDEXED

Indicates that the specified multibuffer default is to be applied to indexed file operations.

/MAGTAPE

Indicates that the specified multibuffer default is to be applied to operations on magnetic tape volumes. If /SEQUENTIAL is specified, /MAGTAPE is assumed together with /DISK and /UNIT_RECORD.

/NETWORK_BLOCK_COUNT=count

Specifies a default block count for network access to remote sequential, indexed sequential, and relative files. You can specify a value in the range of 0 to 127.

The network block count value represents the number of blocks that VAX RMS is prepared to allocate for the I/O buffers used to transmit and receive data. For remote file access, the buffer size is negotiated between between VAX 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 will be used. It also places an upper limit on the largest record that may be transferred to or from a remote file. In other words, the largest record that can be transferred must be less than or equal to this value.

If you omit the value or specify a value of 0, VAX RMS uses the systemwide block count value. If this value is also 0, VAX RMS uses a size of one block.

/PROLOG=n

Specifies a default prologue level for indexed sequential files where n is a value of 0, 2, or 3. A value of 1 is not allowed. If 0 is specified, VAX RMS sets an appropriate prologue level. By default, 0 is assumed.

/RELATIVE

Indicates that the specified multibuffer default is to be applied to file operations on relative files.

/SEQUENTIAL (default)

Indicates that the specified multibuffer default is to be applied to all sequential file operations, including operations on disk, magnetic tape, and unit record devices.

The /SEQUENTIAL qualifier is the default if you do not specify either /RELATIVE or /INDEXED.

/SYSTEM

Requires change-mode-to-kernel (CMKRNL) privilege.

Indicates that the specified defaults are to be applied on a systemwide basis to file operations performed by all processes.

/UNIT_RECORD

Indicates that the multibuffer default is to be applied to file operations on unit record devices. If /SEQUENTIAL is specified, /UNIT_RECORD is assumed together with /DISK and /MAGTAPE.

EXAMPLES

1

```
$ SET RMS_DEFAULT/BLOCK_COUNT=24
$ SHOW RMS
                  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

                  Prologue     Extend Quantity
        Process      0               0
        System       0               0
```

The SET RMS_DEFAULT command defines the multiblock count for disk file input/output operations as 16 blocks. This default is defined only for the current process, and will be used for disk file operations in user programs that do not explicitly set the multiblock count.

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    24      |    0         0           0       8         0       |    0
System     16      |    0         0           0       0         0       |    8

          Prologue    Extend Quantity
Process       0              0
System        0              0
```

The SET RMS_DEFAULT command defines the default multibuffer count for input/output operations on magnetic tapes as eight buffers.

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    24      |    0         0           0       8         0       |    0
System     16      |    0         0           7       7         7       |   16

          Prologue    Extend Quantity
Process       0              0
System        0              0
```

The SET RMS_DEFAULT command defines the systemwide default multibuffer count for all sequential file operations as seven buffers. Since neither /RELATIVE nor /INDEXED was specified, the multibuffer count is defined for sequential file operations on disk, magnetic tape, and unit record devices. The network block count is set to sixteen blocks.

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    24      |    5         0           0       8         0       |    0
System     16      |    0         0           7       7         7       |   16

          Prologue    Extend Quantity
Process       0             50
System        0              0
```

The SET RMS_DEFAULT command defines the default multibuffer count for I/O operations on indexed files as 90 buffers. It also defines the default multiblock count for sequential I/O operations as 10 blocks. This default is defined only for the current process and will be used for disk file operations in user programs that do not explicitly set the multiblock count.

5
```
$ SET RMS_DEFAULT/PROLOG=2
$ SHOW RMS_DEFAULT
          MULTI-   |                MULTIBUFFER COUNTS                | NETWORK
          BLOCK    | Indexed  Relative           Sequential           |  BLOCK
          COUNT    |                        Disk   Magtape  Unit Record |  COUNT
Process    24      |    5         0           0       8         0       |    0
System     16      |    0         0           7       7         7       |   16

          Prologue   Extend Quantity
Process       2             50
System        0              0
```

The SET RMS_DEFAULT command defines the process default prologue level for indexed sequential files as Prolog 2.

SET SYMBOL

Controls access to local and global symbols in command procedures.

FORMAT

SET SYMBOL

restrictions

None.

PARAMETERS

None.

DESCRIPTION

Since all global and local symbols defined in an outer procedure level are accessible to inner procedure levels, the need to mask these symbols without deleting them often exists. The SET SYMBOL command provides this ability by treating symbols as undefined.

The symbol scoping context is different for local and global symbols. For local symbols, it indicates a procedure dependent translation limit. This means that specifying SET SYMBOL/SCOPE=NOLOCAL causes all 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 access only level 2 local symbols. Level 3 can access levels 2 and 3 local symbols, and level 4 can access only level 4 local symbols.

Global symbols are procedure level independent. The global symbol scoping context that is in effect at the time a new procedure level is invoked is propagated to the new procedure level. Specifying /SCOPE=NOGLOBAL causes all global symbols to become inaccessible for all subsequent commands until either /SCOPE=GLOBAL is specified or the procedure exits to a previous level at which global symbols were accessible.

When you exit a procedure level back to a previous procedure, the symbol scoping context from the previous level is restored for both local and global symbols.

The F$ENVIRONMENT(symbol_scope) lexical function returns the string "[NO]LOCAL,[NO]GLOBAL" as the current symbol scoping state. The state of local symbols always precedes the state of global symbols in the returned string.

QUALIFIER

/SCOPE=(keyword,...)

Controls access to local and global symbols. Allows the user to treat symbols as being undefined. Possible keywords are:

[NO]LOCAL	Specifying the NOLOCAL keyword causes all local symbols defined in outer procedure levels to be treated as being undefined by the current procedure and all inner procedure levels. Specifying LOCAL removes any symbol translation limit set by the current procedure level.
[NO]GLOBAL	Specifying the NOGLOBAL keyword causes all global symbols to be inaccessible to the current procedure level and all inner procedure levels unless otherwise changed. Specifying GLOBAL restores access to all global symbols.

EXAMPLES

1
```
$ SET SYMBOL/SCOPE=NOLOCAL
```

All local symbols defined in outer procedure levels are now undefined by the current procedure and all inner procedure levels.

2
```
$ SET SYMBOL/SCOPE=NOGLOBAL
```

All global symbols are now inaccessible to the current procedure level and all inner procedure levels unless otherwise changed.

SET TERMINAL

Changes the system's interpretation of the terminal characteristics.

FORMAT

SET TERMINAL *[device-name[:]]*

restrictions

See qualifier descriptions.

PARAMETER

device-name[:]

Specifies the name of the terminal whose characteristics are to be changed.

If you do not specify a device name, the qualifiers change the characteristics of the current SYS$COMMAND device, if SYS$COMMAND is a terminal. If the current SYS$COMMAND device is not a terminal, the system returns an error message.

DESCRIPTION

The SET TERMINAL command allows you to modify specific terminal characteristics for a particular application or to override system default characteristics. (These defaults are defined at each installation, based on the most common type of terminal in use.)

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 a VAX, 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 VAX/VMS system includes a set of VT100-family terminals that support special DEC ANSI characteristics and escape sequences. For a description of these special characteristics and escape sequences, see the *VAX/VMS I/O User's Reference Manual: Part I*. The VT100-family includes the following terminals: VT100, VT101, VT102, VT105, VT125, VT131, VT132, and VT173.

QUALIFIERS

/ADVANCED_VIDEO
/NOADVANCED_VIDEO

Specifies whether the terminal has advanced video attributes and is capable of 132-column mode operation.

If the terminal width is set to 132 columns and /ADVANCED_VIDEO is enabled, the terminal page limit is set to 24 lines. If /NOADVANCED_VIDEO is enabled on a terminal set to 132 columns, the terminal page limit is set to 12 lines.

/ALTYPEAHD
/NOALTYPEAHD

Used with the /PERMANENT qualifier to control the size of the type-ahead buffer. You should specify SET TERMINAL/PERMANENT/ALTYPEAHD in

the SYS$SYSTEM:STARTUP.COM for those communication lines that require this capability.

In order to use this feature interactively, specify SET TERMINAL /PERMANENT/ALTYPEAHD. This specification will take effect at your next login.

/ANSI_CRT
/NOANSI_CRT

Specifies whether the terminal conforms to ANSI CRT programming standards. The /ANSI_CRT qualifier indicates that the terminal is capable of processing the ANSI sequences listed in the *VAX/VMS I/O User's Reference Manual: Part I.*

Since 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 so that you can use the DEFINE/KEY facility. By default, the terminal is set to /NUMERIC_KEYPAD mode.

/AUTOBAUD
/NOAUTOBAUD

When used with the /PERMANENT qualifier, controls whether to enable automatic baud rate detection for a terminal line. Also sets the default terminal speed to 9600 baud.

If you specify /AUTOBAUD, automatic baud rate detection is enabled allowing the terminal baud rate to be set at login. You can set the baud rate at login by pressing the RETURN key two or more times separated by an interval of at least one second. The valid baud rates are:

110	1200	4800
150	1800	9600
300	2400	19200
600	3600	

Note: If you press a key other than RETURN, /AUTOBAUD might detect the wrong baud rate. If this happens, wait for the login procedure to time out before continuing.

/BLOCK_MODE
/NOBLOCK_MODE

Specifies whether the terminal is capable of performing block mode transmission, local editing, and field protection. Terminals that support all these functions include the VT131 and VT132.

/BRDCSTMBX
/NOBRDCSTMBX

Controls whether broadcast messages are sent to an associated mailbox if one exists. For a description of message formats, see the *VAX/VMS I/O User's Reference Manual: Part I.*

/BROADCAST (default)
/NOBROADCAST

Controls whether the terminal can receive broadcast messages such as MAIL notifications and REPLY messages. By default, a terminal receives any messages the system operator or another privileged user sends.

Use 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. You can use the SET BROADCAST command to exclude only certain types of messages from being broadcast to your terminal, rather than eliminating all messages.

/CRFILL[=formula]

Specifies whether the system must generate fill characters following the pressing of the RETURN key on the terminal.

The recommended formula is a number in the range of 0 through 9 indicating the number of null fill characters required to ensure that the RETURN completes successfully before the next meaningful character is sent. You might need to use this qualifier if you are using a non-DIGITAL terminal or a video terminal. This qualifier prevents the system from sending out data before the terminal is ready to accept it.

The default is /CRFILL=0.

/DEC_CRT[=(value1,value2)]
/NODEC_CRT[=(value1,value2)]

Specifies whether the terminal conforms to DEC VT100-family standards and supports the minimum VT100 standards including DEC escape sequences.

Two optional values may be specified. A value of 1 requests that the DEC_CRT terminal characteristic be set. This is the default. A value of 2 requests that the DEC_CRT2 terminal characteristic be set. This determines whether the terminal conforms to DEC VT200-family standards and supports the minimum VT200 standards, including additional DEC escape sequences.

Note that DEC_CRT2 is a superset of DEC_CRT. Clearing DEC_CRT will cause DEC_CRT2 to be cleared. Similarly, setting DEC_CRT2 will cause DEC_CRT (and ANSI_CRT) to be set.

For more information, see the *VAX/VMS I/O User's Reference Manual: Part I.*

/DEVICE_TYPE=terminal-type

Informs the system of the terminal type. When you specify this qualifier, the terminal sets the default characteristics for the specified terminal. The default characteristics for each type of terminal are listed in Table DCL–15. That table also shows which characteristics are valid for each type of terminal.

You can specify any of the following terminal types:

```
UNKNOWN          LA34
FT1 - FT8        LA38
LA12             LA100
LA36             LQP02
LA120            VT125
VT05             VT131
VT52             VT132
VT55             VT173
VT100            VT200
VT101            PRO_SERIES
VT102            LA210
VT105
```

For compatibility reasons, terminal types listed in the first column can be specified as qualifiers by themselves (for example, /FT8). However, you are encouraged to use the /DEVICE_TYPE qualifier.

Terminal types FT1 through FT8 permit up to eight different categories of terminals that are not supported by VMS to be identified as foreign terminals. See the *VAX/VMS I/O User's Reference Manual: Part I* to find out how to obtain the terminal type from a user program. When you specify a foreign terminal type (FTn), you do not change the default characteristics established for the terminal.

/DIALUP
/NODIALUP (default)

Specifies that the terminal is a dialup terminal.

/DISCONNECT
/NODISCONNECT (default)

Specifies that the process connected to this terminal not be discontinued if the line detects a hangup. The /DISCONNECT qualifier is valid only when /PERMANENT is specified. For more information on process connection, see the CONNECT and DISCONNECT commands.

/DISMISS
/NODISMISS (default)

Determines the terminal driver's treatment of parity errors. If you specify /DISMISS, the terminal driver will ignore the data that caused the error. When /NODISMISS, the default, is in effect, the terminal driver terminates the currently outstanding I/O with an error status.

/DMA
/NODMA

Controls the use of direct memory access (DMA) mode on a controller that supports this feature. The default is /DMA for DMA controllers, and /NODMA for non-DMA controllers.

/ECHO (default)
/NOECHO

Controls whether the terminal echoes (that is, displays) the input lines that it receives.

When /NOECHO is set, the terminal displays only data that a system or user application program writes to it.

/EDIT_MODE
/NOEDIT_MODE
Specifies whether the terminal is capable of performing ANSI defined advanced editing functions.

/EIGHT_BIT
/NOEIGHT_BIT
Indicates whether the terminal uses an 8-bit ASCII character code.

The default is /NOEIGHT_BIT for all terminals, except the VT200 Series; the terminal interprets characters using 7-bit ASCII code.

/ESCAPE
/NOESCAPE (default)
Controls whether ANSI standard escape sequences transmitted from the terminal are handled as a single multiple-character terminator.

If you specify the /ESCAPE qualifier, the terminal driver checks the escape sequences for syntax before passing them to the program. For information on escape sequences, see the *VAX/VMS I/O User's Reference Manual: Part I.*

/FALLBACK
/NOFALLBACK
Specifies that 8-bit DEC Multinational Character Set characters be displayed on the terminal in their 7-bit representation. The default depends on the /EIGHTBIT setting of the terminal.

/FRAME=n
Controls the number of data bits that the terminal driver expects for every character that is input or output. The default depends on the /PARITY and /EIGHTBIT settings of the terminal. The n parameter can take a value from 5 through 8.

/FORM
/NOFORM
Controls whether the terminal driver translates form-feed characters into one or more line feeds or merely outputs the untranslated form-feed character.

/FULLDUP
/NOFULLDUP (default)
Specifies whether the terminal's mode of operation is full-duplex or half-duplex.

For a description of these modes of operation, see the *VAX/VMS I/O User's Reference Manual: Part I.*

This qualifier is complementary to the /HALFDUP qualifier, that is, /FULLDUP is equivalent to /NOHALFDUP.

/HALFDUP (default)
/NOHALFDUP
Specifies whether the terminal's mode of operation is full-duplex or half-duplex.

For a description of these modes of operation, see the *VAX/VMS I/O User's Reference Manual: Part I.*

This qualifier is complementary to the /FULLDUP qualifier, that is, /HALFDUP is equivalent to /NOFULLDUP.

/HANGUP
/NOHANGUP (default)

Controls whether the terminal modem is hung up when you log out. In order to specify /HANGUP, you might require LOG_IO or PHY_IO privilege depending on parameters specified during system generation.

/HARDCOPY
/NOHARDCOPY

Indicates whether the terminal is a hard-copy or video terminal. It also affects how the terminal interprets certain input keys. The /HARDCOPY qualifier establishes the terminal as a hard-copy device. Thus, the DELETE (or RUBOUT) key cannot accomplish backspace deletions. Instead, a backslash character is output each time the DELETE key is pressed.

This qualifier is complementary to the /SCOPE qualifier; that is, /HARDCOPY is equivalent to /NOSCOPE.

/HOSTSYNC
/NOHOSTSYNC (default)

Controls whether the system can synchronize the flow of input from the terminal.

When you specify the /HOSTSYNC qualifier, the system generates a CTRL/S or a CTRL/Q to enable or disable the reception of input. When the type-ahead buffer is full, the system sends a CTRL/S to temporarily stop input; when the buffer is empty, the system sends a CTRL/Q so that more input can be entered. The size of the type-ahead buffer is established using system generation parameters.

/INQUIRE

Requests a response sequence from DIGITAL terminals and sets the device type according to the response. Some VT100-family terminals, including the VT101 and VT105, return a VT100-type response. LA38 terminals respond as LA34 terminals.

You can include the SET TERMINAL/INQUIRE command in your LOGIN.COM file to automatically detect the terminal type.

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 you are editing command lines. The default mode is /OVERSTRIKE, which allows you to type over the current character when you are editing a command line.

/LFFILL[=formula]

Specifies whether the system must generate fill characters following a line feed on the terminal.

The recommended formula value is a number in the range from 0 through 9 indicating the number of null fill characters required to ensure that the line feed completes successfully before the next meaningful character is read. You might need to use this qualifier if you are using a non-DIGITAL terminal or a video terminal.

This qualifier prevents the system from sending out data before the terminal is ready to accept it.

The default is installation-dependent (see Table DCL–15).

/LINE_EDITING
/NOLINE_EDITING

Enables the advanced line editing features for editing command lines.

If you specify /LINE_EDITING, both RETURN and CTRL/Z are recognized as line terminators. Escape sequences are also enabled as line terminators. The default is installation-dependent (see Table DCL–15).

/LOCAL_ECHO
/NOLOCAL_ECHO (default)

When used with the /ECHO qualifier, controls whether the terminal echoes characters locally rather than relying on the host to echo them.

Local echo mode is supported only for command level terminal functions. Certain utilities that require control over echoing, such as the EDT editor in screen mode, should not be used with /LOCAL_ECHO set.

CAUTION: **When logging in to terminals with /LOCAL_ECHO set, VMS has no control over the echoing of passwords.**

/LOWERCASE
/NOLOWERCASE

Indicates whether the terminal has uppercase and lowercase characters.

If you specify /NOLOWERCASE, all letters that are input are translated to uppercase. If you specify /LOWERCASE, lowercase characters are not converted to uppercase.

This qualifier is complementary to the /UPPERCASE qualifier, that is, /LOWERCASE is equivalent to /NOUPPERCASE.

/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

Indicates whether the terminal is connected to a modem or a cable that supplies standard EIA modem control signals. The VAX/VMS operating

system automatically interprets modem signals. If the terminal is set to /MODEM and you enter SET TERMINAL /NOMODEM, you are automatically logged out.

/NUMERIC_KEYPAD (default)

Specifies that the keypad is to be set to /NUMERIC_KEYPAD mode so that you can use the keys on the numeric keypad to type numbers and punctuation marks. In order to use the DEFINE/KEY facility, you must set the terminal to /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. You can set your terminal to /INSERT if you want to be able to insert characters when editing command lines.

/PAGE[=n]

Specifies the page length of the terminal. For hard-copy terminals, the page size (n) equals the number of print lines between perforations on the paper. When the terminal reads a form feed character, it advances the paper to the next perforation. A page size of 0 indicates that the terminal treats each formfeed as a line feed.

You can specify values of 0 through 255 for the page size. The default size is installation-dependent. However, if you specify /PAGE without a value, the default value for n is 0.

/PARITY[=option]
/NOPARITY (default)

Defines the parity for the terminal. You can specify either of the following options:

```
EVEN
ODD
```

If you use the /PARITY qualifier without an option, the SET TERMINAL command assumes /PARITY=EVEN.

/PASTHRU
/NOPASTHRU (default)

Controls whether the system interprets special characters or passes all data to an application program as binary data.

A terminal operating in PASTHRU mode does not (1) expand tab characters to spaces, (2) fill carriage return or line feed characters, or (3) recognize other control characters. The setting of /TTSYNC is honored.

/PERMANENT

Requires LOG_IO or PHY_IO privilege.

Controls whether the characteristics that you specify are established permanently or only for the current terminal session. By default the characteristics that you set during a session are in effect only for that session.

The permanent characteristics are restored when the current user logs out. If you use the /PERMANENT qualifier to override the system default characteristics established at system generation time, remember that if the system is halted, the permanent characteristics revert to those defined at system generation.

You can use the /PERMANENT qualifier with the SET TERMINAL command in a system start-up file to establish the characteristics for all terminals on the system.

/PRINTER_PORT
/NOPRINTER_PORT

Specifies that the terminal has a printer port available. This terminal attribute is not set automatically by SET TERMINAL/INQUIRE. The default is installation-dependent (see Table DCL–15).

/PROTOCOL=DDCMP
/PROTOCOL=NONE (default)

Controls whether the terminal port specified is changed into an asynchronous DDCMP line. Use the /PROTOCOL=NONE qualifier to change an asynchronous DDCMP line back into a terminal line.

Note that the /PROTOCOL=DDCMP qualifier is a permanent characteristic; therefore, the /PERMANENT qualifier need not be specified for this characteristic.

For more information on how to use these lines with DECNET, refer to the *VAX/VMS Networking Manual*.

/READSYNC
/NOREADSYNC (default)

Controls whether the system solicits read data from a terminal using CTRL/Q and terminates the read using CTRL/S.

The default is /NOREADSYNC; the system does not use CTRL/S and CTRL/Q 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. This qualifier is set for VT125 terminals, which have graphics capabilities.

/SCOPE
/NOSCOPE

Indicates whether the terminal is a video terminal, and thus how it reacts when certain keys are pressed. The /SCOPE qualifier establishes the terminal as a video terminal. Thus, when you press the DELETE key, the printing position is moved left one column and the character displayed in that position is erased.

This qualifier is complementary to the /HARDCOPY qualifier, that is, /SCOPE is equivalent to /NOHARDCOPY.

/SET_SPEED
/NOSET_SPEED

Requires either LOG_IO or PHY_IO privilege.

Controls whether the /SPEED qualifier can be used to change the terminal speed. To reset the speed, use the /SET_SPEED qualifier *and* the /SPEED qualifier.

/SECURE_SERVER
/NOSECURE_SERVER (default)

Determines whether the BREAK key on the terminal logs out the current process. With /SECURE_SERVER in effect, pressing the BREAK key when there is no current process initiates the login sequence. By default, the BREAK key has no effect.

The /SECURE_SERVER qualifier has no effect on terminals that are set with /AUTOBAUD.

/SIXEL_GRAPHICS
/NOSIXEL_GRAPHICS

Controls whether the terminal is capable of displaying graphics using the REGIS-defined SIXEL graphics protocol. The default is device-dependent, see Table DCL–15.

/SOFT_CHARACTERS
/NOSOFT_CHARACTERS

Controls whether the terminal has the capability to load a user-defined character set. The default is device-dependent, see Table DCL–15.

/SPEED=rate

Specifies the rate at which the terminal sends and receives data.

If you specify the rate as a single value, the input and output baud rates are set to the same speed. To specify different baud rates for input and output, specify the rate in the format (n,m). The values n and m indicate the input (terminal to host) and output (host to terminal) baud rates, respectively. 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:

50	150	1800	4800
75	300	2000	7200
110	600	2400	9600
134	1200	3600	19200

Note: Some speeds are not supported on certain terminals. For a list of speeds supported on a particular terminal, see the appropriate hardware documentation.

/SWITCH=DECNET

Causes the terminal lines at each node to be switched to dynamic asynchronous DDCMP lines, when specified with the /PROTOCOL=DDCMP qualifier.

Note that the /SWITCH=DECNET qualifier is a permanent characteristic; therefore, the /PERMANENT qualifier need not be specified for this characteristic.

/SYSPASSWORD
/NOSYSPASSWORD (default)

Determines whether the terminal requires that a system password be entered before the USERNAME: prompt is displayed. This qualifier requires LOG_IO privilege and should be used in conjunction with the /PERMANENT qualifier.

For more information, see the *Guide to VAX/VMS System Security*.

/TAB
/NOTAB

Controls how the terminal handles tab characters. The /NOTAB qualifier expands all tab characters to spaces and assumes tab stops at eight-character intervals.

Use the /TAB qualifier when you do not want the system to convert tabs to spaces, but want the terminal to process the tab characters. VMS requires that terminals expand tabs at eight-character intervals. The default is device-dependent, see Table DCL–15.

/TTSYNC (default)
/NOTTSYNC

Controls whether the system responds to a CTRL/S or CTRL/Q issued from the terminal in order to synchronize output.

The default is /TTSYNC; the system stops sending output when CTRL/S is sent by the terminal and resumes output when a CTRL/Q is sent.

/TYPE_AHEAD (default)
/NOTYPE_AHEAD

Controls whether the terminal accepts unsolicited input, that is, input that you type when there is no outstanding read.

When you specify /NOTYPE_AHEAD, the terminal is dedicated, and will accept input only when a program or the system issues a read to the terminal.

Use the /NOTYPE_AHEAD qualifier to ensure that a specific terminal remains dedicated to a particular application. Logins are disabled on a terminal with /NOTYPE_AHEAD set.

When you specify /TYPE_AHEAD, 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

Indicates that the system does not recognize the terminal type. When you specify this qualifier, the system uses the default terminal characteristics for unknown terminals. For a summary of the settings, see Table DCL–15.

/UPPERCASE
/NOUPPERCASE

Specifies whether the terminal should translate all input lowercase letters to uppercase.

This qualifier is complementary to the /LOWERCASE qualifier, that is, /UPPERCASE is equivalent to /NOLOWERCASE.

/WIDTH=n

Specifies the number of characters on each input or output line. The width value must be in the range 1 through 511.

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 12 lines.

When the /WRAP qualifier is in effect, the terminal generates a carriage return/line feed when a line reaches the column position specified by the width value.

/WRAP (default)
/NOWRAP

Controls whether or not the terminal generates a carriage return/line feed when it reaches the end of the line. The end of a line is determined by the setting of the terminal width.

If you specify /NOWRAP and the terminal is accepting input, the terminal does not generate a carriage return/line feed when it reaches the end of a line, but continues to accept input at the last physical character position on the terminal line. If you specify /NOWRAP and the terminal is writing output, it continues to write characters out until it reaches the last position on the line.

When you specify the /WRAP qualifier, the terminal generates a carriage return/line feed whenever the end of the line is reached. /WRAP is the default.

EXAMPLES

1
```
$ SET TERMINAL/DEVICE=VT102
```

This 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
```

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.

Table DCL–15 Default Characteristics for Terminals

Name	Unk.	For.	LA12	LA34 LA38 LA100 LQP02	LA36	LA120	LA210 LN03 LN01K	VT05	VT52 VT55	VT101	VT102	VT100 VT105	VT125	VT131 VT132	VT173	VT200	PRO
ADVANCED_VIDEO	no	•	no	no	no	no	no	no	no	no	yes	@	@	@	yes	yes	yes
ALTYPEAHD	no	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ANSI_CRT	no	•	no	no	no	no	no	no	no	yes	yes	yes	yes	yes	yes	yes	yes
APPLICATION_KEYPAD /NUMERIC_KEYPAD	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
AUTOBAUD	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
BLOCK_MODE	no	•	no	no	no	no	no	no	no	no	no	no	no	yes	yes	no	no
BRDCSTMBX	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
BROADCAST	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
CRFILL	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
DEC_CRT	no	•	no	no	no	no	no	no	no	yes	yes	yes	yes	yes	no	yes	yes
DEC_CRT2	no	•	no	no	no	no	no	no	no	no	no	no	no	no	no	yes	no
DIALUP	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
DISCONNECT	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
DMA	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ECHO	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
EDIT_MODE	no	•	no	no	no	no	no	no	no	no	yes	@	@	@	yes	yes	yes
EIGHT_BIT	•	•	no	no	no	no	yes	no	no	no	no	no	no	no	no	yes	no
ESCAPE	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
FALLBACK	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	no	no
FRAME	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
FORM	•	•	yes	yes	no	yes	yes	no	no	no	no	no	no	no	no	no	no
FULLDUP/HALFDUP	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
HANGUP	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
HARDCOPY/SCOPE	•	•	hard	hard	hard	hard	hard	scope	scope	scope	scope	scope	scope	scope	scope	scope	scope
HOSTSYNC	•	•	•	•	•	•	•	•	•	•	•	•	•	yes	yes	•	•
INSERT/OVERSTRIKE	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
LFFILL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LINE_EDITING	no	•	no	no	no	no	no	no	no	•	•	•	•	•	no	•	•
LOCAL_ECHO	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
LOWERCASE/UPPERCASE	•	•	low	low	low	low	low	up	low	low	low	low	low	low	low	low	low
MODEM	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
PAGE	•	•	66	66	66	66	66	20	24	24	24	24	24	24	16	24	24
PARITY	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
PRINTER_PORT	no	•	no	no	no	no	no	no	no	no	yes	@	yes	no	no	@	yes

• Indicates that the current setting is not affected by terminal type.
@ Optional terminal feature.

ZK-5170-86

Table DCL–15 (Cont.) Default Characteristics for Terminals

Name	Unk.	For.	LA12	LA34 LA38 LA100 LQP02	LA36	LA120	LA210 LN03 LN01K	VT05	VT52 VT55	VT101	VT102	VT100 VT105	VT125	VT131 VT132	VT173	VT200	PRO
READSYNC	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
REGIS	no	•	no	no	no	no	no	no	no	no	no	no	yes	no	no	@	@
REMOTE	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
SECURE_SERVER	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
SET_SPEED	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
SIXEL_GRAPHICS	no	•	no	no	no	no	yes	no	no	no	no	no	yes	no	no	@	no
SOFT_CHARACTERS	no	•	no	no	no	no	no	no	no	no	no	no	no	no	no	@	no
SPEED	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
SYSPASSWORD	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
TAB	•	•	yes	no	no	yes	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
TTSYNC	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
TYPE_AHEAD	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
WIDTH	•	•	80	132	132	132	80	72	80	80	80	80	80	80	80	80	80
WRAP	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

• Indicates that the current setting is not affected by terminal type.
@ Optional terminal feature.

ZK-5171-86

SET TIME

Resets the system time. The system time is used for all time-dependent activities in the VAX/VMS operating system.

FORMAT

SET TIME*[=time]*

restrictions

Requires operator (OPER) and logical I/O (LOG_IO) privileges.

PARAMETER

time

The VAX/VMS system contains an interval clock, which is used both as a timer to record intervals between various internal events and as the source clock for displaying the time of day.

The SET TIME command allows you to reset the system time. However, both operator (OPER) and logical I/O (LOG_IO) privileges are required.

To reset the system time, type the SET TIME command, optionally followed by an explicit time value expressed in either standard absolute time format or combination time format, described below. If the explicit time value is specified, the interval system clock is reset to the time specified.

The date, the time, or both can be specified for absolute time. The following format is used to specify the absolute time parameter:

```
[dd-mmm-yyyy[:]] [hh:mm:ss.cc]
```

Combination time consists of an absolute time value plus or minus a delta time value. (A delta time is an offset from the current time to a time in the future.)

For further information on absolute time format or the use of combination time format, see Section 2.5 of the *VAX/VMS DCL Concepts Manual*.

If the explicit time value is not specified, the interval system clock is automatically reset according to the time-of-year clock.

Note that the time-of-year clock is optional for some processors. For further information about the time-of-year clock, see the *VAX Hardware Handbook*.

EXAMPLES

1

```
$ SET TIME = 03:21:24.03
$ SHOW TIME
19-JUN-1985 03:22:19.53
```

The SET TIME command sets the system time to the specified time. The SHOW TIME command requests a display of the current time.

2
```
$ SET TIME
$ SHOW TIME
19-JUN-1985 03:21:27.53
```

The SET TIME command sets the system time according to the Time-of-year clock. The SHOW TIME command requests a display of the current time.

SET UIC

Establishes a new user identification code (UIC) as the default. Use the SET UIC command to gain access to a restricted file, that is, a file contained in a directory whose protection restricts access to the owner of that directory.

FORMAT

SET UIC *uic*

restrictions

Requires CMKRNL (change-mode-to-kernel) privilege.

PARAMETER

uic

Specifies the group number and member number. Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

EXAMPLES

1
```
$ SET UIC [370,10]
```

This command establishes your UIC as [370,10]. You can now read or modify any files whose access is restricted to this UIC.

2
```
$ SET UIC [214,4]
$ SET DEFAULT [ANDERSON]
```

The SET UIC command sets your UIC to [214,4]; the SET DEFAULT command sets the default directory name to [ANDERSON].

3
```
$ SET UIC [GEORGE]
```

This example sets the UIC to be that of the user named GEORGE who is a member of the same group as the person issuing the SET UIC command. Note the similarity of this UIC format to the directory name format. Be sure not to use a UIC where a directory specification is needed.

4
```
$ SET UIC [VMS,GEORGE]
```

This example sets the UIC to be that of the user named GEORGE who is a member of the VAX/VMS group. The person issuing the SET UIC command need not be a member of the VAX/VMS group.

SET VERIFY

Controls whether command lines and data lines in command procedures are either displayed at the terminal or printed in a batch job log. The information displayed by the SET VERIFY command can help you in debugging command procedures.

FORMAT

SET [NO]VERIFY [=([NO]PROCEDURE, [NO]IMAGE)]

restrictions

None.

PARAMETERS

([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.

If you do not specify either of the verification keywords, then both types of verification are set (with SET VERIFY) or cleared (with SET NOVERIFY). If you specify only one keyword, then the type of verification you specify is set or cleared; the other one is not affected. Also, when you specify only one keyword, you can omit the parentheses.

DESCRIPTION

By default, when VAX/VMS processes interactive command procedures, it does not display the command lines at the terminal. System responses and error messages are always displayed.

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. Thus, 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.

The default setting for noninteractive jobs (such as batch jobs) is VERIFY; that is, all lines in the command procedure appear in the job listing.

EXAMPLES

1

```
$ 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 will display 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).

2

```
$ PROC_VER = F$ENVIRONMENT("VERIFY_PROCEDURE")
$ IMAGE_VER = F$ENVIRONMENT("VERIFY_IMAGE")
$ SET NOVERIFY
   .
   .
   .
$ TEMP = F$VERIFY(PROC_VER, IMAGE_VER)
```

This command procedure uses the lexical function F$ENVIRONMENT to save the current procedure and image verification setting. Then the SET NOVERIFY command turns off both procedure and image verification. Subsequently, the F$VERIFY function is used to restore the original verification settings.

3

```
$ 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.

4

```
$ SET VERIFY = PROCEDURE
```

In this example, procedure verification is turned on. If image verification was previously on, it remains on; if image verification was off, it remains off.

```
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

Modifies the characteristics of one or more mounted Files–11 volumes.

FORMAT

SET VOLUME *device-spec[:][,...]*

restrictions

Requires write (W) access to the index file on the volume. If you are not the owner of the volume, requires either a system UIC or system (SYSPRV) privilege.

PARAMETER

device-name[:][,...]

Specifies the name of one or more mounted Files–11 volumes.

QUALIFIERS

/ACCESSED[=n]

Requires operator (OPER) privilege.

Specifies the number of directories to be maintained in system space for ready access. If you specify a value greater than the current value, the new value is effective immediately. However, if you specify a value that is less than the current value, the new value is written to the home block and will not take effect until the next time the volume is mounted.

You can specify a number (n) in the range of 0 through 255. If you specify the qualifier /ACCESSED and omit the number of directories, a default value of 3 is used.

/DATA_CHECK[=(option[,...])]

Defines a default for data check operations following all reads and/or writes to the specified volume. You can specify the following options:

[NO]READ	Determines whether checking is performed following all read operations.
[NO]WRITE	Determines whether checking is performed following all write operations.

If you specify the /DATA_CHECK qualifier without specifying an option, the default qualifier /DATA_CHECK=WRITE is used. If you omit the /DATA_CHECK qualifier, no checking is performed.

/ERASE_ON_DELETE
/NOERASE_ON_DELETE (default)

Specifies that when a file on the volume is deleted, the space occupied by that file is erased. For more information on how the erasing is done, see the DELETE/ERASE command.

/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 number (n) in the range of 0 through 65535. If you specify the /EXTENSION qualifier without specifying a value, a default value of 0 (the VAX 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=(code)

Specifies the default protection to be applied to all files on the specified volume. Specify the code according to the rules given in Section 7.1.2 of the *VAX/VMS DCL Concepts Manual*.

Note that this attribute is not used while the volume is in use on a VAX/VMS system, but is provided to control the process use of the volume on RSX–11M systems. VAX/VMS always uses the default file protection; the protection can be changed with the DCL command SET PROTECTION/DEFAULT.

/HIGHWATER_MARKING
/NOHIGHWATER_MARKING

Specifies that all files on the disk volume are subject to file highwater marking. This qualifier applies to Structure Level 2 volumes only.

If you specify /HIGHWATER_MARKING, SET VOLUME sets the File Highwater Mark (FHM) volume attribute. FHM is a mechanism that guarantees that a user cannot read data that he has not written.

The /NOHIGHWATER_MARKING qualifier disables FHM for the volume.

/LABEL=volume-label

Specifies a 1- through 12-character alphanumeric name to be encoded on the volume. Lowercase letters are automatically changed to uppercase. The specified label remains in effect until it is explicitly changed (that is, dismounting the volume does not effect the label).

/LOG
/NOLOG (default)

Controls whether the SET VOLUME command displays the volume specification of each volume after the modification.

/MOUNT_VERIFICATION
/NOMOUNT_VERIFICATION

Specifies whether the volume is subject to mount verification.

/OWNER_UIC[=uic]

Sets the owner UIC of the volume to the specified UIC. Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

If you specify the /OWNER_UIC qualifier without specifying a UIC, the current process UIC is used.

/PROTECTION=(code)

Specifies the protection to be applied to the volume. The protection controls who can read, write, and delete files on the volume. If you do not specify a protection code, protection defaults to all types of access for all categories of user. Specify the code according to the rules given in Section 7.1.2 of the *VAX/VMS DCL Concepts Manual*.

When you specify a protection code, access type E (execute) indicates create access.

/REBUILD

Recovers caching limits for a volume that was improperly dismounted.

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.

/RETENTION=(min[,max])

Specifies the minimum (min) and maximum (max) retention times to be used by the file system to determine the expiration date for files on the volume.

When a file is created on the volume, the expiration date is initially set to the current time + max. Each time the file is accessed, the current time is added to the min time, and if this sum is greater than the expiration date, a new expiration date is computed.

If you omit the max 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-).

/UNLOAD (default)
/NOUNLOAD

Specifies whether the volume is unloaded (that is, spun down) when the DCL command DISMOUNT is issued.

/USER_NAME[=user-name]

Specifies a user name of up to 12 alphanumeric characters to be recorded on the volume. If you specify the /USER_NAME qualifier without specifying a user name, the current process user name is used.

/WINDOWS[=n]

Specifies the number of mapping pointers to be allocated for file windows. You can specify a number (n) in the range 7 through 80.

If you specify the /WINDOWS qualifier without specifying a number, a default value of 7 is used.

EXAMPLES

1
```
$ SET VOLUME/DATA_CHECK=(READ,WRITE) DBC5
```

This command requests that data checks be performed following all read and write operations to DBC5.

2
```
$ SET VOLUME/FILE_PROTECTION=(S:RWED,O:RWED,G:RE,W:RE) DBC5
```

This command sets the default protection to be applied to all files created on volume DBC5. System and owner are granted all types of access; group and world are permitted only to read and execute files on DBC5.

3
```
$ SET VOLUME/LABEL=LICENSES DBC5
```

This command encodes the label LICENSES on the volume DBC5. 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 DBA0:
```

This command specifies that 25 directories are to be maintained in system space for ready access for the volume DBA0. 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$DBA2:
%SET-I-MODIFIED, _NODE$DBA2: modified
```

The SET VOLUME/REBUILD command causes a rebuild operation to begin on the volume that is mounted on NODE$DBA2:. The /LOG qualifier directs SET VOLUME to display a notification message.

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.

FORMAT

SET WORKING_SET

restrictions

Working set limits cannot be set to exceed those defined in the user authorization file (UAF).

PARAMETERS

None.

DESCRIPTION

A process's working set is the collection of 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. The SET WORKING_SET command enables the user to change the working set size within the authorized limits.

QUALIFIERS

/ADJUST (default)
/NOADJUST

Enables or disables the system's changing of the process working set.

/EXTENT=n

Specifies the maximum number of 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 user authorization file.

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 or not confirmation of the SET WORKING_SET command is displayed.

/QUOTA=n

Specifies the maximum number of 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 Adjust Working Set Limit ($ADJWSL) system service, which is described in the *VAX/VMS System Routines Reference Volume*.

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 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 sets the working set size for any image in the process to 100.

SHOW

Displays information about the current status of the process, the system, or devices in the system.

FORMAT

SHOW *option*

restrictions

Restrictions are noted in the command descriptions for each SHOW option.

DESCRIPTION

The SHOW command options are summarized in Table DCL–16. For detailed information on each command option, see the descriptions of individual SHOW commands following Table DCL–16.

Table DCL–16 SHOW Command Options

Option	Displays
ACCOUNTING	Items for which accounting is enabled
ACL	The Access Control List associated with a system object
AUDIT	The security features that are enabled
BROADCAST	The current message broadcast status
CLUSTER	Cluster activity and performance
CPU	Current state of the attached processor (VAX–11/782 systems)
DEFAULT	The current default device and directory
DEVICES	The status of devices in the system
DEVICE/SERVED	The status of devices served by the MSCP server on a VAXcluster.
ERROR	The error count for the CPU, memory, and physical devices
INTRUSION	The contents of the breakin database
KEY	Key definitions created by the DEFINE/KEY command
LOGICAL	Current logical name assignments
MAGTAPE	The status and characteristics of a specific magnetic tape device
MEMORY	The availability and utilization of memory resources
NETWORK	The availability of network nodes, including the current node
PRINTER	Default characteristics of the specified printer
PROCESS	Attributes of the current process, including privileges, resource quotas, memory usage, priority, and accounting information

Table DCL–16 (Cont.) SHOW Command Options

Option	Displays
PROTECTION	The 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 the current 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
QUOTA	The current disk quota authorized for and used by a specific user on a specific disk
RMS_DEFAULT	The current default multiblock and multibuffer counts used by VAX RMS for file operations
STATUS	The status of the current job, including accumulated CPU time, open file count, and count of I/O operations
SYMBOL	Current symbol definitions
SYSTEM	A list of all processes in the system
TERMINAL	The device characteristics of a terminal
[DAY]TIME	The current date and time
TRANSLATION	The result of translating a logical name
USERS	Information about users currently on the system
WORKING_SET	The current working set size limit and quota

SHOW ACCOUNTING

The SHOW ACCOUNTING command displays items for which accounting is enabled. For a detailed description of these items, see the discussion of the ACCOUNTING command in the *VAX/VMS DCL Dictionary*.

FORMAT

SHOW ACCOUNTING

restrictions

None.

PARAMETERS

None.

DESCRIPTION

The SHOW ACCOUNTING command displays one or more of the following items for which accounting is currently enabled on your system:

Item	Meaning
PROCESS	Any process termination
INTERACTIVE	Interactive job termination
LOGIN_FAILURE	Login failures
SUBPROCESS	Subprocess termination
DETACHED	Detached job termination
BATCH	Batch job termination
NETWORK	Network job termination
PRINT	All print jobs
MESSAGE	User messages

SHOW ACL

Allows you to display the access control list (ACL) of an object.

FORMAT

SHOW ACL

restrictions

None.

PARAMETER

object-name

Specifies the name of the object whose ACL is to be displayed. No wildcard characters are allowed in the object-name specification.

DESCRIPTION

The SHOW ACL command enables you to display the access control list (ACL) of a system object. By default, the SHOW ACL command assumes an object type of file. If the object is any other type, you must include the /OBJECT_TYPE qualifier.

QUALIFIER

/OBJECT_TYPE=type

Defines the object type of the object whose ACL is to be displayed. The following keywords are used to specify the object type:

FILE (default)	The object is a Files–11 disk file.
DEVICE	The object is a device.
SYSTEM_GLOBAL_SECTION	The object is a system global section.
GROUP_GLOBAL_SECTION	The object is a group global section.
LOGICAL_NAME_TABLE	The object is a system logical name table.

EXAMPLE

```
$ SHOW ACL/OBJECT_TYPE=DEVICE TTA1
Object type: device,    Object name: VTA1
(IDENTIFIER=[SALES,FRANK],ACCESS=READ)
(IDENTIFIER=[123,321]+NETWORK,ACCESS=NONE)
    .
    .
    .
```

This SHOW ACL command displays the ACL of the device TTA1.

SHOW AUDIT

Provides a display that identifies which security auditing features are enabled and the events that they will report.

FORMAT

SHOW AUDIT

restrictions

Requires the SECURITY privilege.

PARAMETERS

None.

DESCRIPTION

The SHOW AUDIT command displays the set of features that have been enabled for auditing with the DCL command SET AUDIT. If no auditing has been enabled, the display reports briefly that security alarms are currently disabled. The display is directed to the current SYS$OUTPUT device.

It is useful to check which auditing features are enabled whenever you plan to add or delete features with a subsequent SET AUDIT command.

QUALIFIER

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

EXAMPLES

1
```
$ SHOW AUDIT
Security alarms currently disabled
```

The display produced by this SHOW AUDIT command reveals that security auditing is not enabled.

2

```
$ SET AUDIT/ALARM/ENABLE=ALL
$ SHOW AUDIT
Security alarms currently enabled for:
    ACL
    MOUNT
    AUTHORIZATION
    BREAKIN:      (DIALUP,LOCAL,REMOTE,NETWORK,DETACHED)
    LOGIN:        (BATCH,DIALUP,LOCAL,REMOTE,NETWORK,SUBPROCESS,DETACHED)
    LOGFAILURE:   (BATCH,DIALUP,LOCAL,REMOTE,NETWORK,SUBPROCESS,DETACHED)
    LOGOUT:       (BATCH,DIALUP,LOCAL,REMOTE,NETWORK,SUBPROCESS,DETACHED)
    FILE_ACCESS:
        FAILURE: (READ,WRITE,EXECUTE,DELETE,CONTROL)
        SUCCESS: (READ,WRITE,EXECUTE,DELETE,CONTROL)
        SYSPRV:  (READ,WRITE,EXECUTE,DELETE,CONTROL)
        BYPASS:  (READ,WRITE,EXECUTE,DELETE,CONTROL)
        GRPPRV:  (READ,WRITE,EXECUTE,DELETE,CONTROL)
        READALL: (READ,WRITE,EXECUTE,DELETE,CONTROL)
```

This SHOW AUDIT command reveals that security auditing has been enabled to provide alarms for all possible events.

3

```
$ SHOW AUDIT
Security alarms currently enabled for:
    ACL
    BREAKIN:      (DIALUP,LOCAL,REMOTE,NETWORK,DETACHED)
    FILE_ACCESS:
        FAILURE: (READ,WRITE,EXECUTE,DELETE,CONTROL)
        BYPASS:  (READ,WRITE,EXECUTE,DELETE,CONTROL)
```

This SHOW AUDIT command reveals that all terminals enabled as security operators will receive an alarm when:

- An access control list (ACL) access requests the alarm
- The system detects a possible breakin attempt
- A file access fails with read, write, execute, delete, or control access
- A file access with read, write, execute, delete, or control access is gained by means of the BYPASS privilege

4

```
$ SHOW AUDIT
Security alarms currently enabled for:
    BREAKIN:      (DIALUP,LOCAL,REMOTE,NETWORK,DETACHED)
    LOGIN:        (DIALUP)
    LOGOUT:       (DIALUP)
```

This SHOW AUDIT command reveals that the terminals enabled as security operators will receive an alarm whenever the system detects a possible breakin attempt, a dialup at login time, or whenever a dialup connection logs out.

SHOW BROADCAST

Displays the message classes that are currently affected by the SET BROADCAST command.

FORMAT

SHOW BROADCAST

restrictions *None.*

PARAMETERS

None.

DESCRIPTION

The SHOW BROADCAST command tells which classes of messages are being screened from your terminal by the SET BROADCAST command.

QUALIFIER

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

EXAMPLES

1
```
$ SHOW BROADCAST
Broadcasts are enabled for all classes
```

This example shows the display when all message classes are enabled for broadcast.

2
```
$ SHOW BROADCAST
Broadcasts are currently disabled for:
    MAIL
```

This SHOW BROADCAST display indicates that SET BROADCAST=NOMAIL is in effect.

```
3   $ SHOW BROADCAST
    Broadcasts are currently disabled for:
        GENERAL
        PHONE
        MAIL
        QUEUE
        SHUTDOWN
        URGENT
        DCL
        OPCOM
        USER1
        USER2
        USER3
        USER4
        USER5
        USER6
        USER7
        USER8
        USER9
        USER10
        USER11
        USER12
        USER13
        USER14
        USER15
        USER16
```

This example shows the display when SET BROADCAST=NONE is in effect.

SHOW CLUSTER

Invokes the VAX/VMS SHOW CLUSTER Utility (SHOW CLUSTER) to monitor and display cluster activity and performance. For a complete description of the Show Cluster Utility, including information about the SHOW CLUSTER command, see the *VAX/VMS Show Cluster Utility Reference Manual*.

FORMAT SHOW CLUSTER

SHOW CPU

Displays the current state of the attached processor in a VAX/VMS multiprocessing system.

FORMAT

SHOW CPU

restrictions

- Applies only to VAX/VMS multiprocessing systems.
- Requires change mode to kernel (CMKRNL) privilege.

PARAMETERS

None.

DESCRIPTION

Use the SHOW CPU command to display the state of the attached processor. The attached processor can be in one of the following states: INITIALIZE, IDLE, BUSY, EXECUTE, DROP, or STOP.

EXAMPLE

```
$ SHOW CPU
  Attached processor is in the EXECUTE state.
```

The display indicates that the attached processor is currently in the EXECUTE state.

SHOW DEFAULT

Displays the current default device and directory names, along with any equivalence strings. These defaults are applied whenever you omit a device and/or directory name from a file specification.

FORMAT SHOW DEFAULT

restrictions *None.*

PARAMETERS *None.*

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. You can change these defaults during a terminal session or in a batch job with the SET DEFAULT command, or by reassigning the logical name SYS$DISK.

EXAMPLES

1
```
$ SHOW DEFAULT
  DISK1:[ALPHA]
$ SET DEFAULT DISK5:[HIGGINS.SOURCES]
$ SHOW DEFAULT
  DISK5:[HIGGINS.SOURCES]
```

The SHOW DEFAULT command requests a display of 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
```
$ SHOW DEFAULT
  DISK0:[HIGGINS]
$ DEFINE  SYS$DISK  DISK3:
$ SHOW DEFAULT
  DISK3:[HIGGINS2]
```

The first SHOW DEFAULT command displays the current default device and directory names. The DEFINE command changes the equivalence name for the logical name SYS$DISK, thus changing the default device from DISK0 to DISK3. When you issue the SHOW DEFAULT command again, you see that the default directory name is now HIGGINS2.

3
```
$ SHOW DEFAULT
WORK:[SMITH]
$ DEFINE/TRANSLATION_ATTRIBUTES=CONCEALED XYZ WORK:[INVOICES.]
$ SET DEFAULT XYZ:[SALES]
$ SHOW DEFAULT
XYZ:[SALES]
```

In this example, the rooted-device logical name XYZ is specified as a concealed device. In this case, SHOW DEFAULT displays the rooted-device logical name without translation.

4
```
$ SET DEFAULT WORK:[BLUE]
$ SHOW DEFAULT
 WORK:[BLUE]
$ DEFINE FOO WORK:[BLUE.TEMP1],WORK:[BLUE.TEMP2]
$ SET DEFAULT FOO
$ SHOW DEFAULT
 FOO:[BLUE]
 =    WORK:[BLUE.TEMP1]
 =    WORK:[BLUE.TEMP2]
```

The logical name FOO is defined as a search list containing the directories [BLUE.TEMP1] and [BLUE.TEMP2] both on device WORK. 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.

Since the directory field has not been explicitly specified, the original [BLUE] directory remains in effect as the current default directory. Note however, that unless the current default directory syntax ([]) is explicity used, all file references will be to those directories contained in the search list.

5
```
$ SET DEFAULT NOSUCH:[NOWAY]
$ SHO DEFAULT
 NOSUCH:[NOWAY]
 %DCL-I-INVDEF, NOSUCH:[NOWAY] does not exist
```

The default has been set to a nonexistant device and directory. An informational message is displayed warning the user of this fact.

SHOW DEVICES

Displays the status of a device on the system.

FORMAT

SHOW DEVICES *[device-name[:]]*

restrictions

See qualifier descriptions.

PARAMETER

device-name[:]

Specifies the name of a device for which information is to be displayed. 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).
- 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.

If you issue the SHOW DEVICES command and specify neither a device name parameter nor any qualifier, the command provides a brief listing of characteristics of all devices on the system (with the exception of mailbox devices). To obtain information about a specific device or generic class of devices, specify a device name.

Use the /ALLOCATED qualifier for a list of devices that are currently allocated to processes; use the /MOUNTED qualifier for a list of the mounted devices.

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 issue 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, SHOW DEVICES displays information about that device. If you use certain qualifiers with SHOW DEVICES, information is displayed about those devices that currently have volumes mounted and/or that have been allocated to processes.

The device name displayed by the system uses the format ddcu, where dd is device code, c is the controller designation, and u is the unit number. If the system is part of a VAXcluster that is running with HSCs, the device name will include the node name using the format node$ddcu (where node refers to the node name).

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.

/FILES

Requires SYSPRV 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 that 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 which 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. In order 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 /NOSYSTEM qualifier is specified, the names of all files currently opened on the system are displayed.

If a file is read-protected from your UIC, the "No privilege" message is displayed instead of the file name. You must have SYSPRV (system privilege) or BYPASS privilege in order to display the file name.

A space in place of a file name represents a workfile (such as a temporary edit file) not entered in any directory. In order to have temporary file names displayed, 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 qualifiers. The functions of /FILES and these qualifiers are mutually exclusive.

/FULL

Displays a complete list of information about the devices.

/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 which currently have volumes mounted are displayed.

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

/SYSTEM
/NOSYSTEM

Controls whether the names of installed files and files opened by the system are displayed. Files opened by the system are those which have been opened without the use of an ancillary control process (ACP), such as INDEXF.SYS and QUOTA.SYS.

If you specify /NOSYSTEM with the /FILES qualifier, only files opened by processes are displayed. If you omit both /SYSTEM and /NOSYSTEM and specify the /FILES qualifier, the names of all files currently opened 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.

/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).

EXAMPLES

1

```
$ SHOW DEVICES
Device                  Device       Err.    Volume         Free  Trans Mount
 Name                   Status      Count     Label        Blocks Count Count
DBA0:                   Online mnt      0  VMS            47088    115     1
DBA1:                   Online mnt      0  USERPACK1      45216      2     1
DBA2:                   Online mnt      3  DOCUMENT        8068     20     1
DBA5:                   Online mnt      0  MASTERP        28668      1     1
DBA6:                   Online          0
DBA7:                   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
```

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 online; characteristics indicate whether the device is allocated, 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 DMA0
Disk NODE1$DMA0:, device type RK07, is online, allocated, mounted,
error logging enabled
 Error count                    0  Operations completed            1257
 Owner UIC                  [1,4]  Owner process name              VANNOY
 Owner process ID       202000C8   Dev Prot  S:RWED,O:RWED,G:RWED,W:RWED
 Reference count                2  Default buffer size              512
 Volume label       JAKE_X239      Relative volume no.                0
 Cluster size                   1  Transaction count                  2
 Free blocks                 3741  Maximum files allowed          13447
 Extend quantity                5  Mount count                        1
 Volume status            Process  ACP process name             DMA0BACP
 File ID cache size            64  Extent cache size                 64
 Quota cache size              64
Volume is subject to mount verification, file high-water marking
```

The SHOW DEVICES command requests a full listing of the status of the RK07 device DMA0. The device is located on NODE1 in a VAXcluster.

SHOW DEVICES

```
3   $ SHOW DEVICES/FULL NODE2$
Disk NODE2$DUA0:, device type RA81, is online, mounted,
error logging enabled
 Error count                    0  Operations completed            24195
 Owner UIC              [11,177]  Owner process name
 Owner process ID       20200000  Dev Prot  S:RWED,O:RWED,G:RWED,W:RWED
 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, you have requested a full display of information about each device on NODE2 in the VAXcluster. Information is shown here only for the first two devices: a mounted device and one that is not mounted.

SHOW DEVICES/SERVED

Displays information on devices served by the MSCP server on this node. The /SERVED qualifier is required.

FORMAT

SHOW DEVICES/SERVED

restrictions

None.

DESCRIPTION

This command displays information about the MSCP server and the devices it serves. This information is of most use to system managers.

QUALIFIER

/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.

/HOST

Displays the names of the processors that have MSCP-served devices on line. SYSGEN's MSCP/HOST command determines how many hosts in the 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.

/RESOURCE

Displays information on the resources available to the MSCP server for use in processing I/O requests for the devices it serves. You make these resources available to the MSCP server when you use SYSGEN's MSCP command to start the MSCP server and use the qualifiers listed below.

SHOW DEVICES/SERVED

Qualifier	Item Specified
/BUFFER	The amount of buffer space available to the MSCP server
/FRACTION	The maximum size, in pages, of the buffer granted to an I/O request; for transfers of more data than will fit a buffer of the size specified by this qualifier, several CI transfers are needed
/SMALL	The minimum size, in pages, of the buffer that the MSCP server can grant to an I/O request; if less than this amount of buffer space is available, the I/O request must wait until at least this much buffer space becomes available; when this much space becomes available, the MSCP server grants the request a buffer
/PACKETS	The number of I/O-request packets (CDRPs) available to the MSCP server for processing I/O requests

EXAMPLES

1
```
$ SHOW DEVICES/SERVED
MSCP Served Devices on BOSTON 12-JUN-1985 12:34:56.78

                                          Queue Requests
Device:          Status      Total Size   Current    Max     Hosts
    2$DBA0        AVAIL          340670         0      0         0
    2$DMA1       ONLINE           53790         0      0         2
    2$DMA0      OFFLINE           53790         0      0         0
```

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* column shows 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 BOSTON 12-JUN-1985 12:34:58.82

Request Count:
      0-7:        1       32-39:        8       88-103:        7
     8-15:        0       40-55:        0      104-127:        0
    16-23:        0       56-71:        8
    24-31:        0       72-87:       20

Operations Count:
    ABORT             0    ERASE             0    READ              0
    ACCESS            0    FLUSH             0    REPLACE           0
    AVAILABLE         0    GET CMD STS       0    SET CTL CHR      10
    CMP CTL DAT       0    GET UNT STS    2799    SET UNT CHR       0
    CMP HST DAT       0    ONLINE            0    WRITE             0
  Total            2809
```

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 10 set-controller-characteristics (SET CTL CHR) operations, but has performed no set-unit-characteristics (SET UNT CHR) operations.

3

```
$ SHOW DEVICES/SERVED/RESOURCE
           MSCP Served Devices on BOSTON 12-JUN-1985 12:34:58.82
                                                        Fragment Size
Resources:          Total      Free      In Use     Minimum     Maximum
    Buffer area:       64        64          0           4          32
    I/O Packets:      144       144
                  Current   Maximum
    Buffer wait:        0         0
```

This example shows the information displayed by the SHOW DEVICES /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 columns labeled *Fragment Size* show the minimum and the maximum size, in pages, of a buffer that an I/O request can obtain from the buffer area.

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 concurrently waited to obtain a buffer.

4

```
$ SHOW DEVICES/SERVED/HOST
           MSCP Served Devices on BOSTON 12-JUN-1985 12:34:58.82
                                              Queue Requests
Host:              Time of Connection         Current   Max   Devices
    HARVEY      10-JUN-1985 12:57:39.90            0     1         2
    DOC         11-JUN-1985 22:02:10.25            0     1         0
    GRUMPY      11-JUN-1985 22:02:10.25            0     0         1
    SLEEPY      11-JUN-1985 22:02:11.75            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 has currently 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-served devices that the listed hosts have on line.

SHOW ERROR

Displays the error count for all devices with error counts greater than 0.

FORMAT

SHOW ERROR

restrictions

None.

PARAMETERS

None.

QUALIFIERS

/FULL

Displays the error count for all devices, including those with no errors. (The error count is either 0 or a number greater than 0.)

/OUTPUT[=file-spec]
/OUTPUT=SYS$OUTPUT (default)

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

EXAMPLE

```
$ SHOW ERROR
```

Displays the error count for all devices with error counts greater than 0:

```
Device  Error Count
CPU         2
MEMORY      1
DBB1        9
```

SHOW INTRUSION

Displays the contents of the breakin database.

FORMAT SHOW INTRUSION

restrictions Requires the CMKRNL and SECURITY privileges.

DESCRIPTION

VAX/VMS stores information in the breakin 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 breakin attempts by using the SHOW INTRUSION command to display the contents of the breakin database.

The entries in the breakin database have the following format:

```
Intrusion      Type      Count      Expiration      Source
```

The information provided in the fields in each entry is as follows:

Intrusion	Class of intrusion. The type of evasive action that VAX/VMS 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 VAX/VMS. The SYSGEN parameter, LGI_BRK_TMO, controls how long VAX/VMS 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 breakin database, the operating system classifies login failures according to their source. The four classes of system intrusion are:

TERMINAL	Login failure originating from one terminal.
TERM_USER	Login failure originating from one terminal, using a valid username.
NETWORK	Login failure originating from a remote node using a valid username.
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:

TERMINAL	terminal:
TERM_USER	terminal:username
NETWORK	node::username
USERNAME	username

The type of evasive action that a security manager can take is based on the type of information provided. See the *Guide to VAX/VMS System Security* for details on how to use this information.

The breakin 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, VAX/VMS 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 will cease to be counted is displayed in the expiration field of the entry. When the number of login failures exceeds the number specified by the SYSGEN parameter, LGI_BRK_TERM, the breakin entry is classified as intruder.

When a breakin entry is promoted to intruder, VAX/VMS takes evasive action by blocking all login attempts from that particular source.

The duration of the evasive action is determined by the SYSGEN parameter LGI_HID_TIM. The absolute time at which the evasive action will end is displayed in the expiration field of the entry.

For information on breakin detection, prevention, and evasive actions, see the *Guide to VAX/VMS System Security*.

If you determine that an entry in the breakin database resulted from a user error and not a breakin attempt, you can remove an entry from the breakin database with the DELETE/INTRUSION command. See the DELETE /INTRUSION command for more details.

QUALIFIERS

/OUTPUT[=file-spec]

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 breakin database that is displayed. The valid keywords are:

ALL	All breakin entries. By default, all entries are displayed.
SUSPECT	Breakin entries for login failures that have occurred but have not yet passed the threshold necessary to be identified as intruder.
INTRUDER	Breakin entries for which the login failure rate was high enough to warrant evasive action.

EXAMPLES

1
```
$ SHOW INTRUSION/OUTPUT=INTRUDER.LIS
```

This command writes all the entries currently in the breakin database to the file INTRUDER.LIS.

2
```
$ SHOW INTRUSION/TYPE=INTRUDER
Intrusion   Type       Count    Expiration    Source
TERMINAL    INTRUDER     9      10:29:39.16   _LTA23:
NETWORK     INTRUDER     7      10:47:53.12   STAR::HAMM
```

As a result of the SHOW INTRUSION command, all intruder entries currently in the breakin database are displayed.

SHOW KEY

Displays the key definitions created by the DEFINE/KEY command.

FORMAT

SHOW KEY *[key-name]*

restrictions

See qualifier descriptions.

PARAMETER

key-name

Specifies the name of the key whose definition you want displayed. See the DEFINE/KEY command for a list of the valid key names.

DESCRIPTION

After you have defined keypad keys 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 find out the names of all the states in which you have created key definitions. Use the SET KEY command to change key states.

QUALIFIERS

/ALL

Requests that all key definitions in the current state be displayed. You can use the /STATE qualifier to request key definitions in other states. If you use the /ALL qualifier, do not specify a key name. If no state is specified, all key definitions in the current state are displayed. Use the /STATE qualifier to specify one or more states.

/BRIEF (default)
/NOBRIEF

Requests that only the key definition and state be displayed. The /BRIEF and /NOFULL qualifiers are equivalent.

/DIRECTORY

Requests the display of the names of all states for which keys have been defined. If you have no keys defined, the SHOW KEY/DIRECTORY command displays nothing. If you have not specified a state with a key definition, the SHOW KEY/DIRECTORY command displays DEFAULT for the state.

Note that the /DIRECTORY qualifier is incompatible with any of the other SHOW KEY qualifiers.

/FULL
/NOFULL (default)

Requests that all qualifiers that are associated with a definition be displayed. 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

Specifies the name of a state for which the specified key definitions are to be displayed. 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 /NOSTATE, key definitions in the current state are displayed.

EXAMPLE

```
$ DEFINE/KEY/TERMINATE PF1 "ATTACH GEORGE"
%DCL-I-DEFKEY, DEFAULT key PF1 has been defined
$ SHOW KEY PF1
DEFAULT keypad definitions:
  PF1 = "ATTACH GEORGE"
$ SHOW KEY/FULL PF1
DEFAULT keypad definitions:
  PF1 = "ATTACH GEORGE"  (noecho,terminate,noerase,nolock)
```

The SHOW KEY command displays both the definition and the state of the definition for the PF1 key. This is the default display.

The SHOW KEY/FULL command requests that additional information, concerning all qualifiers associated with the key definition, be included in the display.

SHOW LOGICAL

Displays all logical names in one or more logical name tables or displays the current equivalence string, or strings, assigned to a specified logical name or names. The SHOW LOGICAL command performs iterative translations.

FORMAT

SHOW LOGICAL *[logical-name[:],[...]]*

restrictions

Requires read (R) access to the table in which a logical name is cataloged to display information about the logical name.

PARAMETER

logical-name[:],[...]

Specifies one or more logical names for which the equivalence strings are to be displayed. The logical name can have from 1 to 255 characters (unless it is the name of a logical name table, in which case the maximum length is 31 characters). The logical name is translated iteratively up to a number of times determined by the system (from 9 to 11). That is, the equivalence strings resulting from a translation are examined to see if they are also logical names.

If you do not specify the logical name parameter, the SHOW LOGICAL command displays all logical names in the specified logical name tables. If no tables are specified and if no logical name is present, the SHOW LOGICAL command displays all logical names in the tables indicated by the logical name LNM$DCL_LOGICAL. Unless LNM$DCL_LOGICAL has been redefined for your process, LNM$DCL_LOGICAL translates to the process, job, group, and system tables, in that order. (The process, job, group, and system tables are referred to by the logical names LNM$PROCESS, LNM$JOB, LNM$GROUP, and LNM$SYSTEM.)

Wildcard characters are allowed in the logical name specification. However, if wildcard characters are used, iterative logical name translation is not performed.

DESCRIPTION

The SHOW LOGICAL command displays logical names in the specified table or tables. If you specify a logical name with the command, the system displays the equivalence name or names assigned to that logical name.

The SHOW LOGICAL command performs iterative translations. If a logical name has more than one equivalence string, then all translations at a level are displayed before going to the next level.

If you specify the /FULL qualifier, the SHOW LOGICAL command also displays the access mode of the logical name and any attributes that are associated with a logical name, equivalence string, and logical name table.

The SHOW TRANSLATION command is similar to the SHOW LOGICAL command. However, the SHOW TRANSLATION command searches the logical name tables for a specified logical name and returns the equivalence string for the first match found. Also, SHOW TRANSLATION is executed within DCL, whereas SHOW LOGICAL is not; thus, SHOW TRANSLATION does not cause the current image (if any) to exit.

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 for this qualifier is USER_MODE; thus, 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 system's default logical name table search list are displayed. (The logical name LNM$DCL_LOGICAL specifies the default logical name table search list.)

/DESCENDANTS
/NODESCENDANTS (default)

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.

/FULL

Provides more detailed information on the access mode and any attributes for each logical name, equivalence string, and logical name table displayed.

/GROUP

When you specify a logical name, indicates that only the group logical name table is to be searched. The /GROUP qualifier is synonymous with /TABLE=LNM$GROUP.

If you specify the /GROUP qualifier and you do not also specify a logical name, all entries in the group logical name table are displayed.

/JOB

When you specify a logical name, indicates that only the job logical name table is to be searched. The /JOB qualifier is synonymous with /TABLE=LNM$JOB.

If you specify the /JOB qualifier and you do not also specify a logical name, all entries in the job logical name table are displayed.

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

/PROCESS

When you specify a logical name, indicates that only the process logical name table is to be searched. The /PROCESS qualifier is synonymous with /TABLE=LNM$PROCESS.

If you specify the /PROCESS qualifier and you do not also specify a logical name, all entries in the process logical name table are displayed.

/STRUCTURE
/NOSTRUCTURE (default)

Controls whether the command displays the "family tree" of all logical name tables accessible to this process. 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 /STRUCTURE, you cannot specify any other qualifiers except /ACCESS_MODE, /FULL, and /OUTPUT.

/SYSTEM

When you specify a logical name, indicates that only the system logical name table is to be searched. The /SYSTEM qualifier is synonymous with /TABLE=LNM$SYSTEM.

If you specify the /SYSTEM qualifier and you do not also specify a logical name, all names in the system logical name table are displayed.

/TABLE=(name[,...])

Specifies the name of the tables to be searched. If you specify only one table, you can omit the parentheses.

You can use the /TABLE qualifier to specify a user-defined logical name table (created with the CREATE/NAME_TABLE command), to specify the process, group or system logical name tables, or to specify the process or system directory tables. You can also use wildcards to specify the table name.

If you specify the table name 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.

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" = "DBA2:"
  "SYS$SCRATCH" = "WORK6:[ODONNELL]"
```

The SHOW LOGICAL command requests a display of the current process logical names. (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 requests a display of the current equivalence name 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 requests a display of all current logical names in the group logical name table. (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 requests the equivalence name 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 equivalence strings.

5
```
$ SHOW LOGICAL/TABLE=LNM$GROUP/TABLE=LNM$SYSTEM SYS$DISK
  "SYS$DISK" = "ZZZ3:" (LNM$SYSTEM_TABLE)
```

The SHOW LOGICAL command is qualified by both the /TABLE=LNM$GROUP and /TABLE=LNM$SYSTEM qualifiers. The response indicates that the logical name SYS$DISK has an equivalence name in the system logical name table. When you issue two conflicting qualifiers, as in this example, only the last qualifier you specify is used.

6
```
$ SHOW LOGICAL/TABLE=LNM$PROCESS_DIRECTORY
```

This command 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 MAGTAPE

Displays the current characteristics and status of a specified magnetic tape device.

FORMAT

SHOW MAGTAPE *device-name[:]*

restrictions

None.

PARAMETER

device-name[:]

Specifies the name of the magnetic tape device for which you want to display the characteristics and status.

QUALIFIER

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, the output is sent to the current process default output stream or device, identified by the logical name SYS$OUTPUT.

If you enter /OUTPUT with no file specification, or if you omit the file name or the file type, SHOW is the default file name and LIS the default file type. If you enter a file specification, it may not include any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

EXAMPLE

```
$ SHOW MAGTAPE MTA0:
  MTA0: UNKNOWN, DENSITY=800, FORMAT=Normal-11
        Odd Parity
```

The SHOW MAGTAPE command displays the characteristics of the device MTA0:. The display shows the device type, density, and format (default or normal PDP–11).

It also displays the following characteristics:

Position lost	Write-locked
End-of-tape	Even parity
End-of-file	Odd parity
Beginning-of-tape	

SHOW MEMORY

Displays the availability and usage of those system resources that are related to memory.

FORMAT

SHOW MEMORY

restrictions *None.*

PARAMETERS

None.

DESCRIPTION

The information provided by this command can help you determine whether to change certain system memory resources to improve system performance. The system memory resources are:

- Physical memory
- Process entry slots and balance slots
- Nonpaged and paged dynamic memory
- Space in paging and swap files

When the SHOW MEMORY command is executed, a display is written to SYS$OUTPUT. Depending on which qualifiers you specify, the display will show the following memory resource statistics:

- Physical memory usage
- Bad page list
- Number of pages allocated to VMS
- Slot usage
- Fixed-size pool areas (packets)
- Dynamic memory usage (bytes)
- Paging file usage (pages)
- Lookaside list
- Dynamic memory

QUALIFIERS

/ALL (default)

Displays all available information, that is, information displayed by the /FILES, /PHYSICAL_PAGES, /POOL, and /SLOTS qualifiers.

/FILES

Displays information about the usage of each paging and swap file currently installed.

/FULL
When used with the /POOL or /FILES qualifier, displays additional information about the usage of each pool area or paging and swap file currently installed. This qualifier is ignored unless the /FILES or /POOL qualifier is explicitly specified.

/OUTPUT[=file-spec]
/NOOUTPUT
Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

/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.

/SLOTS
Displays information about the availability of PCB vector slots and balance slots.

EXAMPLES

```
1   $ SHOW MEMORY/PHYSICAL_PAGES

                System Memory Resources on  9-JUN-1985 16:11:30.76
Physical Memory Usage (pages):      Total❶       Free❷      In Use❸    Modified❹)
  Main Memory (4.00Mb)                 8192          97         7866           229

Of the physical pages in use, 1446 pages are permanently allocated to VMS.
```

Physical Memory Usage

Shows the utilization of physical memory.

❶ Total	Displays the 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	Displays the number of pages on the free page list.
❸ In Use	Displays the 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	Displays the number of pages on the modified page list.

Bad Page List

Shows the contents of the bad page list. This display is written only when there are pages on the bad page list.

Total	Displays the number of pages on the bad page list.
Dynamic	Displays the number of memory errors detected after the system was booted.
I/O Errors	Displays the number of errors detected during page fault handling.
Static	Displays the 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 VMS

Any SHOW MEMORY display that includes the physical memory display concludes with the number of pages permanently allocated to the VAX/VMS system. These pages include nonpaged executive code and data, the PFN data base, nonpaged dynamic memory, the interrupt stack, and the system page table.

2 $ SHOW MEMORY/SLOTS

```
              System Memory Resources on  9-JUN-1985 16:11:35.31
Slot Usage (slots):                 Total❶       Free❷    Resident❸      Swapped❹)
  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 slots.

❶ Total	Displays the number of process entry slots (the value of the SYSGEN parameter MAXPROCESSCNT) and balance slots (the value of the SYSGEN parameter BALSETCNT) permanently allocated when the system was bootstrapped.
❷ Free	Displays the number of slots that are not currently being used.
❸ Resident	Displays the number of slots that are currently being used by memory-resident processes. The number of balance 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 slots.
❹ Swapped	Displays the number of slots used by outswapped processes. For process entry slots, this number includes all processes that have been partially outswapped. For balance slots, this number includes those processes that have had their process bodies outswapped but have process headers that are still resident.

3
```
$ SHOW MEMORY/POOL

            System Memory Resources on  9-JUN-1985 16:11:39.97
Fixed-Size Pool Areas (packets):    Total❶       Free❷      In Use❸       Size❹)
  Small Packet (SRP) List               624          42          582           96
  I/O Request Packet (IRP) List         500         257          243          160
  Large Packet (LRP) List                51          14           37          640
Dynamic Memory Usage (bytes):       Total❺       Free❻      In Use❼      Largest❽
  Nonpaged Dynamic Memory           161792        3488       158304         1936
  Paged Dynamic Memory               65536       29312        36224        29296
```

Fixed-Size Pool Areas (packets)

Shows the use of the nonpaged pool areas that consist of fixed-size packets, the so-called lookaside lists.

❶ Total	Displays the total number of packets allocated for each list.
❷ Free	Displays the number of available packets on each list.
❸ In Use	Displays the number of packets in use on each list. This number is simply the total number of packets minus the number of free packets.
❹ Size	Displays the fixed block size for each list.

Dynamic Memory Usage (bytes)

Shows the use of the nonpaged and paged pool areas that allocate variably sized blocks.

❺ Total	Displays the total number of bytes set aside for each area.
❻ Free	Displays the total amount of free space in each dynamic memory area.
❼ In Use	Displays the amount of space currently allocated from each area. This number is simply the total size minus the number of free bytes.
❽ Largest	Displays the size of the largest contiguous block in each area. For paged pool, this number represents the largest block that can be successfully allocated. For nonpaged pool, an allocation request larger than this number will cause nonpaged pool to grow (if other constraints allow growth).

4
```
$ SHOW MEMORY/FILES

                System Memory Resources on  9-JUN-1985 16:11:45.83
Paging File Usage (pages):                      Free❷      In Use❸      Total❹)
  DISK$VMS08FEB:[SYS0.SYSEXE]SWAPFILE.SYS❶        12112         9888         22000
  DISK$VMS08FEB:[SYS0.SYSEXE]PAGEFILE.SYS         23838         6162         30000
```

Paging File Usage (pages)

Shows the usage of paging and swap files.

❶ Name	Displays the complete file specification of each swap or paging file. The names of the primary paging file and the primary swapping file (if this file exists), and the files installed by the bootstrap operation are always displayed. The names of any secondary paging or swap files installed by the SYSGEN Utility are displayed only if the process using the SHOW MEMORY command has read access to those files. If the process cannot read the file, the name is suppressed but the usage statistics are displayed.
❷ Free	Displays the number of available blocks in each paging and swap file currently installed.
❸ In Use	Displays the number of blocks allocated from each paging and swap file.
❹ Total	Displays the total size of each paging and swap file.

When the /FULL qualifier is included on the SHOW MEMORY command for displays of pool areas or paging file usage, additional information is included in the pool or files displays.

SHOW MEMORY

5 $ SHOW MEMORY/POOL/FULL

```
                System Memory Resources on  9-JUN-1985 16:11:49.74

Small Packet (SRP) Lookaside List                Packets      Bytes      Pages
    Current Total Size(1)                            624      59904        117
    Initial Size (SRPCOUNT)(2)                        50       4800         10
    Maximum Size (SRPCOUNTV)(3)                     1500     144000        282
    Free Space(4)                                     46       4416
    Space in Use(5)                                  578      55488
    Packet Size/Upper Bound (SRPSIZE)(6)                        96
    Lower Bound on Allocation(7)                                48

I/O Request Packet (IRP) Lookaside List          Packets      Bytes      Pages
    Current Total Size(1)                            500      80000        157
    Initial Size (IRPCOUNT)(2)                       500      80000        157
    Maximum Size (IRPCOUNTV)(3)                     1500     240000        469
    Free Space(4)                                    259      41440
    Space in Use(5)                                  241      38560
    Packet Size/Upper Bound (fixed)(6)                          160
    Lower Bound on Allocation(7)                                112

Large Packet (LRP) Lookaside List                Packets      Bytes      Pages
    Current Total Size(1)                             51      32640         64
    Initial Size (LRPCOUNT)(2)                        37      23680         47
    Maximum Size (LRPCOUNTV)(3)                       80      51200        100
    Free Space(4)                                     14       8960
    Space in Use(5)                                   37      23680
    Packet Size/Upper Bound (LRPSIZE + 64)(6)                   640
    Lower Bound on Allocation(7)                                480

Nonpaged Dynamic Memory
    Current Size (bytes)(8)          161792   Current Total Size (pages)(8)      316
    Initial Size (NPAGEDYN)(9)       149504   Initial Size (pages)(9)            292
    Maximum Size (NPAGEVIR)(10)      512000   Maximum Size (pages)(10)          1000
    Free Space (bytes)(11)             4256   Space in Use (bytes)(12)        157536
    Size of Largest Block(13)          1936   Size of Smallest Block(14)          16
    Number of Free Blocks(15)            19   Free Blocks LEQU 32 Bytes(16)        6

Paged Dynamic Memory
    Current Size (PAGEDYN)(8)         65536   Current Total Size (pages)(8)      128
    Free Space (bytes)(11)            29312   Space in Use (bytes)(12)         36224
    Size of Largest Block(13)         29296   Size of Smallest Block(14)          16
    Number of Free Blocks(15)             2   Free Blocks LEQU 32 Bytes(16)        1
```

Lookaside List

Displays information about each nonpaged pool area that contains fixed-size blocks.

❶ Current Size	Displays the current total size of each lookaside list. The "pages" column lists the number of physical pages permanently allocated to each list.
❷ Initial Size	Displays the initial size of each lookaside list and the name of the SYSGEN parameter that controls this size.
❸ Maximum Size	Displays the maximum size to which each lookaside list can grow and the name of the SYSGEN parameter that controls this size.
❹ Free Space	Displays the amount of free space on each list.
❺ Space in Use	Displays the amount of space currently allocated from each list.
❻ Packet Size	Displays the fixed block size for each list and the relation between this size and a SYSGEN parameter. (Note that the block size for the I/O request packet lookaside list cannot be varied.) Allocation requests must be smaller than this size in order to be allocated from a given list.
❼ Lower Bound	Displays the lower limit on allocation size permitted from each lookaside list.

Dynamic Memory

Shows the utilization of dynamic memory areas consisting of variably sized blocks.

❽ Current Size	Displays the current size (in bytes and pages) of each dynamic memory area.
❾ Initial Size	Displays the initial size of nonpaged dynamic memory and the name of the SYSGEN parameter (NPAGEDYN) that controls this size.
❿ Maximum Size	Displays the maximum size to which nonpaged dynamic memory can grow and the name of the SYSGEN parameter (NPAGEVIR) that controls this size.
⓫ Free Space	Displays the amount of free space in each dynamic memory area.
⓬ Space in Use	Displays the amount of space currently allocated from each area.
⓭ Largest Block	Displays the size of the largest contiguous area in each pool area.
⓮ Smallest Block	Displays the size of the smallest hole (free block) in each pool area.
⓯ Free Blocks	Displays the total number of free blocks in each pool area. The size of this number is a measure of pool fragmentation.
⓰ Blocks LEQU 32	Displays the number of free blocks that are 32 bytes long or shorter. This number is another measure of pool fragmentation because while these small blocks are unlikely to be allocated, they contribute to the allocation time whenever an allocation request is made.

6
```
$ SHOW MEMORY/FILES/FULL

              System Memory Resources on  9-JUN-1985 16:11:55.49
DISK$VMS08FEB:[SYS0.SYSEXE]SWAPFILE.SYS❶
    Free Blocks❷              12400    Blocks in Use❸                 9600
    Total Size (blocks)❹      22000    Paging File Number❺               0
    Swap Usage (processes)❻      45    Paging Usage (processes)❼         0
  This file is used exclusively for swapping.❽)
DISK$VMS08FEB:[SYS0.SYSEXE]PAGEFILE.SYS❶
    Free Blocks❷              23947    Blocks in Use❸                 6053
    Total Size (blocks)❹      30000    Paging File Number❺               3
    Swap Usage (processes)❻       0    Paging Usage (processes)❼        45
  This file can be used for either paging or swapping.❽
```

Paging File Usage

Shows the usage of paging and swap files.

❶ File Name	Displays the complete file specification of each paging or swap file, subject to the privilege restriction mentioned in the description of the normal display.
❷ Free Blocks	Displays the number of free blocks in each paging and swap file.
❸ Blocks in Use	Displays the number of blocks allocated from each paging and swap file.
❹ Total Size	Displays the size of each paging and swap file.
❺ File Number	Displays the internal paging file index assigned to each paging or swap file when it is installed.
❻ Swap Usage	Displays the number of processes that are currently assigned space in this file for the purpose of swapping. Swap file assignment can change over the life of a process.
❼ Paging Usage	Displays the number of processes that are currently paging to this file. Paging file assignment is made when a process is created and a process continues to page to that file. Note that the last two pieces of information are limited by the GROUP and WORLD privileges. That is, a process with neither privilege can only determine the paging and swap file assignments of itself and its subprocesses. A process with GROUP privilege can obtain this information about all processes that have the same group number. Only a process with WORLD privilege can obtain accurate paging and swap file information for the entire system.
❽ File Usage	Displays a line of text that describes whether the file is used exclusively for swapping or is used for both paging and swapping.

```
7   $ SHOW MEMORY
```

This command displays all memory resource information, that is, information displayed by the /FILES, /PHYSICAL_MEMORY, /POOL, and /SLOTS qualifiers.

SHOW NETWORK

Displays the availability of the local node as a member of the network[1] and the addresses and names of all nodes that are currently accessible to the local node. The SHOW NETWORK command also displays link and cost relationships between the local node and other nodes in the network.

FORMAT

SHOW NETWORK

restrictions

None.

PARAMETERS

None.

DESCRIPTION

The SHOW NETWORK command displays the current network in terms of the following five characteristics: node, links, cost, hops, and next hop to node. These characteristics are defined as follows:

Node	Identifies each available node in the network by its node address and node name.
Links	Shows the number of logical links between the local node and each available remote node.
Cost	Shows the total line cost of the path to a remote node. The cost for each line in the network is assigned by the system manager.
Hops	Shows the number of intermittent nodes plus the target node.
Next hop to node	Shows the outgoing physical line used to reach the remote node. The local node is identified by the term LOCAL. The node address and node name of the next hop to the target node are also displayed.

If you enter the SHOW NETWORK command and the network is unavailable at that time, you will receive the following display:

```
Network unavailable
```

QUALIFIERS

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT without a file specification, the output is sent to the current process default output stream or device, identified by the logical name SYS$OUTPUT.

[1] DECnet–VAX is available under separate license.

If you enter /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

EXAMPLE

```
$ SHOW NETWORK

VAX/VMS Network Status for local node 2.161 ARAKIS on 15-APR-1985
The next hop to the nearest area router is node 2.62 ZEUS.
Node            Links  Cost   Hops  Next Hop to Node
2.161 ARAKIS        0     0      0     Local   ->   2.161 ARAKIS
2.6 RAEL            0     3      3     UNA-0   ->   2.1   RAEL
2.2 PANGEA          0     9      5     UNA-0   ->   2.2   PANGEA
2.3 TWDEE           0     5      4     UNA-0   ->   2.63  AURORA
2.4 TWDUM           0     4      4     UNA-0   ->   2.4   TWDUM
2.11 NEONV          0     5      5     UNA-0   ->   2.11  NEONV
2.63 AURORA         0     4      4     UNA-0   ->   2.63  AURORA
                Total of 7 nodes.
```

If your local node is a nonrouting or end node and you enter the SHOW NETWORK command, you will receive the following message on your terminal.

```
This is a nonrouting node, and does not have any network
information. The designated router for node _nodename is
node_number_name.
```

SHOW PRINTER

Displays the device characteristics currently defined for a system printer.

FORMAT

SHOW PRINTER *device-name[:]*

restrictions

None.

PARAMETER

device-name[:]

Specifies the name of the printer for which characteristics are to be displayed.

DESCRIPTION

The SHOW PRINTER command displays the characteristics that are currently defined for the specified printer. Characteristics include printer type, page width, page length, line termination type, form feed setting, case setting, character translation settings, and wrapping.

Printer characteristics are established by the SET PRINTER command. You must have OPER (operator) privilege to use the SET PRINTER command.

QUALIFIER

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

EXAMPLE

```
$ 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
 Intermediate device: STAR$DBA1:
 Associated queue: LN01$PRINT
```

The SHOW PRINTER command requests a display of the characteristics of the printer LPA0.

SHOW PROCESS

Displays information about a process and any subprocess in the current process tree.

FORMAT

SHOW PROCESS *[process-name]*

restrictions

- Requires GROUP privilege to show other processes in the same group.
- Requires WORLD privilege to show processes outside your group.

PARAMETER

process-name

Specifies the name of the process about which information is to be displayed. Process names can have from 1 to 15 alphanumeric characters. Process names are linked to group numbers. The specified process must have the same group number in its user identification code (UIC) as the current process. You cannot specify the process name for a process outside of your group. To display information about a process outside of your group, you must use the qualifier /IDENTIFICATION=PID.

You cannot use the process name parameter when the /IDENTIFICATION qualifier is specified. If you include neither the process name nor the /IDENTIFICATION qualifier, the current process is assumed.

QUALIFIERS

/ACCOUNTING

Displays accumulated accounting statistics for the current terminal session.

/ALL

Displays all information available, that is, the default information as well as the information displayed by the /ACCOUNTING, /PRIVILEGES, and /QUOTAS qualifiers.

If the request is for the current process, it will also display information given by the /MEMORY and /SUBPROCESSES qualifiers. In addition, for the current process, the qualifier will display the process rights identifiers displayed by the /PRIVILEGES qualifier.

/CONTINUOUS

Displays continuously updated information about the process.

While the continuous display is running, you can press the V key to display 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 (PC) 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 (*).

To terminate the continuous display, press the E key.

The /CONTINUOUS qualifier may not be used with the /OUTPUT qualifier.

/IDENTIFICATION=pid

Specifies the process identification (PID) of the process about which information is to be displayed. 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 name parameter. If, in addition, you specify either the /MEMORY or /SUBPROCESSES qualifiers, the process identification (PID) value must be that of the current process.

/MEMORY

Displays the process's use of dynamic memory areas, which consist of variably sized blocks.

The /MEMORY qualifier is only allowed for the current process. If, in addition, the /IDENTIFICATION qualifier is used, the process identification (PID) value must be that of the current process.

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

The /OUTPUT qualifier may not be used with the /CONTINUOUS qualifier.

/PRIVILEGES

Displays user privileges and identifiers currently enabled for the process. See Table DCL–1-3 in the *VAX/VMS DCL Concepts Manual* for a list of the privilege keywords and a description of each privilege.

/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.

/SUBPROCESSES

Displays the process names of any subprocesses owned by the current process. If a hierarchy of subprocesses exists, the command displays the names in hierarchical order. This qualifier can be used only for the current process.

The /SUBPROCESSES qualifier is only allowed for the current process. If, in addition, the /IDENTIFICATION qualifier is used, the process identification (PID) value must be that of the current process.

EXAMPLES

1
```
$ SHOW PROCESS
 21-FEB-1985 15:35:19.39   VTA102:            User: MALIK
 Pid: 28200364   Proc. name: MALIK           UIC: [VMS,MALIK]
 Priority:   4   Default file spec: WORK5:[MALIK]
 Devices allocated: VTA102:
```

The default output of the SHOW PROCESS command displays:

- Date and time the SHOW PROCESS command is issued
- Device name of the current SYS$INPUT device
- User name
- Process identification number (PID)
- Process name
- User identification code (UIC)
- Base execution priority
- Default device
- Default directory
- Devices allocated to the process and volumes mounted, if any

2
```
$ SHOW PROCESS/ACCOUNTING
 21-FEB-1985 14:48:01.31   VTA102:            User: 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
```

3
```
$ SHOW PROCESS/PRIVILEGES
 21-FEB-1985 14:59:28.53   VTA102:            User: 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
```

4 $ SHOW PROCESS/QUOTAS

```
21-FEB-1985 15:00:28.79   VTA102:           User: 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
```

5 $ SHOW PROCESS/SUBPROCESSES

```
21-FEB-1985 15:44:59.39                     User: MALIK
Processes in this tree:
MALIK
  MALIK_1 (*)
    MALIK_2
```

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 FRED

```
                       Process FRED                      12:26:53

   State                CUR                Working set          269
   Cur/base priority    8/4                Virtual pages       1713
   Current PC           7FFEE07E           Buffered I/O         646
   Current PSL          03C00000           Page faults         3417
   Current user SP      7FF785A4           Event flags     C8000007
                                                           C0000000
   PID                  226006C0           Direct I/O           246
   UIC                  [VMS,FRED]         CPU time     00:00:13.82
   SYS$SYSROOT:[SYSEXE]SHOW.EXE
```

The /CONTINUOUS qualifier causes the display of information about process FRED to be updated continuously.

7 $ SHOW PROCESS/MEMORY

```
21-FEB-1985 14:59:04.48   VTA102:           User: 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 displays the use of dynamic memory areas for the current process, MALIK. These areas are described below:

Current size	Displays the current size (in bytes and pages) 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 hole (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	Displays the number of free blocks that are 32 bytes long or shorter. This number is another measure of pool fragmentation. Although these small blocks are unlikely to be allocated, they contribute to the allocation time whenever an allocation request is made.

SHOW PROTECTION

Displays the current 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 command.

FORMAT SHOW PROTECTION

restrictions *None.*

PARAMETERS *None.*

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 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

Displays information about queues and jobs that are currently in queues.

FORMAT

SHOW QUEUE *[queue-name]*

restrictions

None.

PARAMETER

queue-name

Specifies a queue name. If you omit the parameter, the system displays information about all queues that have been initialized for the system. In a VAXcluster, the SHOW QUEUE command by itself displays all initialized queues in the cluster.

You can use wildcard characters in the queue name parameter. The same rules that apply to wildcard characters for file specifications also apply for queue names. The default for the queue name parameter is asterisk (*), which means that information about all initialized queues is displayed.

DESCRIPTION

The SHOW QUEUE command can display information about queues as well as about jobs in a queue.

When you issue the SHOW QUEUE command with no qualifiers, the system lists the names and types of all the queues. If you have any jobs on any of the queues, information about those jobs is displayed. In a cluster, all queues available for the cluster are listed.

QUALIFIERS

/ALL

Displays the names of all job entries in the specified queues. By default, the SHOW QUEUE command displays only jobs that are owned by the current process.

If no queue name is specified, the /ALL qualifier displays all job entries on all queues.

/BATCH

Lists all batch queues and any jobs in those queues that are owned by the current process. If you use wildcard characters in the queue name parameter, information about all batch queues that match the queue name will be displayed.

/BRIEF (default)

Requests a brief listing of information about job entries in the queue. When you specify /BRIEF, the user name, job number, and queue name are displayed as well as the current form and stock mounted on each queue.

/DEVICE

Lists all printer, terminal, and server queues and displays any jobs in the queues that are owned by the current process. If you use wildcard characters in the queue name parameter, information about all printer, terminal, and server queues that match the queue name will be displayed.

/FILES

Requests a brief listing of information about job entries in the queue with the list of files associated with each job. The display includes a full file specification for each file in each job.

/FULL

Displays full information about all queues, and about any jobs in the queue that are owned by the current process. The information on jobs includes the full file specification, date and time of submission, and all settings that were specified for the job.

Information about a queue includes the queue name and type and all settings that have been set for the queue. Use this qualifier to find out which characteristics and forms have been set for all queues or for the queues specified.

If you use wildcard characters in the queue name parameter, information about all queues that match the queue name will be displayed.

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

EXAMPLES

1

```
$ SHOW QUEUE SYS_TEX
Batch queue SYS_TEX
  Jobname         Username     Entry  Status
  -------         --------     -----  ------
  CHAPTER1.JOB    SMITH         1388  Executing
```

The SHOW QUEUE command displays the current job entry on the batch queue SYS_TEX. If user SMITH has no job entry in the SYS_TEX batch queue, only information about the queue name and status is displayed.

2

```
$ SHOW QUEUE SYS_SERVE
%QUEMAN-E-NOSUCHQUEUE, no such queue
```

There is no queue named SYS_SERVE on the system.

```
3  $ SHOW QUEUE/DEVICE/FULL DELPHI_LPA0
Printer queue DELPHI_LPA0, on DELPHI::DELPHI_LPA0, mounted form COLOR
    /BASE_PRIORITY=4
    /DEFAULT=(FEED,FLAG,FORM=WHITE (stock=DEFAULT),TRAILER=ONE)
    /NOENABLE_GENERIC Lowercase /OWNER=[1,4] /PROTECTION=(S:E,O:D,G:R,W:W)

  Jobname         Username     Entry  Blocks  Status
  -------         --------     -----  ------  ------
  SYLOGIN         MARTIN         880      10  Printing
    Submitted  9-AUG-1985 12:49 /FORM=COLOR  /PRIORITY=4
  LOGIN           MARTIN         858       4  Pending
    Submitted  8-AUG-1985 17:27 /PRIORITY=4
```

The SHOW QUEUE command lists the current job entry for user MARTIN on the printer queue LPA0. The /FULL qualifier lists the submission information, the full file specification, and the current settings for both the job and the queue.

If the stock of the job's form matches the stock of the mounted form, then that job can print, as illustrated by job SYLOGIN. If the form of the job as submitted is undefined, then that job will take the default form of the queue. Since the form for job LOGIN was not defined when the job was submitted, job LOGIN will assume the default form for the queue. Note that LOGIN will remain in a pending state until the stock of the mounted form matches the stock of the default form.

```
4  $ SHOW QUEUE/DEVICE DELPHI*
Printer queue DELPHI_LPA0, on DELPHI::DELPHI_LPA0, mounted form DEFAULT
Printer queue DELPHI_LPB0, on DELPHI::DELPHI_LPB0, mounted form MEMO
(stock=DEFAULT)
```

The brief display of SHOW QUEUE includes the current form and stock mounted on each processing queue. The stock of the form for DELPHI_LPA0 is not displayed since the stock name is identical to the form name.

SHOW QUEUE/CHARACTERISTICS

Displays characteristic names and numbers that are available on queues. Characteristic names are installation-defined.

FORMAT

SHOW QUEUE/CHARACTERISTICS *[characteristic-name]*

restrictions

None.

PARAMETER

characteristic-name

Specifies the name of a characteristic. You can use wildcard characters in the characteristic name parameter. The same rules that apply to wildcard characters for file specifications also apply for characteristic names. The default for the characteristic name parameter is an asterisk (*), which means that the names of all characteristics on the system will be displayed.

DESCRIPTION

SHOW QUEUE/CHARACTERISTICS is used to find out which characteristics are available on the system. To find out which characteristics are available on a particular queue, use the SHOW QUEUE command with the /FULL qualifier.

Characteristic names are created by the DEFINE/CHARACTERISTICS command. You can specify a particular characteristic for a job using the /CHARACTERISTICS qualifier with the PRINT, SUBMIT, or SET QUEUE/ENTRY command. Characteristics are assigned to queues with the INITIALIZE/QUEUE, SET QUEUE, or START/QUEUE command.

QUALIFIERS

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

EXAMPLES

1

```
$ SHOW QUEUE/CHARACTERISTICS
Characteristic name                Number
-------------------                ------
REDINK                                  0
DAILY_WEATHER                           1
TEXMAC_JOB                              2
```

This example displays all the characteristics that have been defined for this system.

2

```
$ SHOW QUEUE/CHARACTERISTICS *INK
Characteristic name                Number
-------------------                ------
REDINK                                  0
BLUEINK                                 1
BROWNINK                                2
```

This SHOW QUEUE/CHARACTERISTICS command displays the names of all characteristics that end in the string INK.

SHOW QUEUE/FORM

Displays form names and numbers that are available on queues. Forms define the size and type paper and the layout of text that are used for print jobs.

FORMAT

SHOW QUEUE/FORM *[form-name]*

restrictions

None.

PARAMETER

form-name

Specifies the name of a form. You can use wildcard characters in the form name parameter. The same rules that apply to wildcard characters for file specifications also apply for form names. The default for the form name parameter is an asterisk (*), which means that the names of all forms on the system will be displayed.

DESCRIPTION

SHOW QUEUE/FORM is used to find out which forms are available on the system. To find out 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 using the /FORM qualifier with the PRINT or SET QUEUE/ENTRY command. Forms are assigned to queues with the INITIALIZE/QUEUE, SET QUEUE, or START/QUEUE command.

The default form type for all printer and terminal queues has the name DEFAULT and the number 0.

QUALIFIERS

/BRIEF (default)

Requests a brief listing of information about the forms. Only the form name, stock, number, and description of the form is displayed.

/FULL

Requests a full listing of all information about the forms on the system. Information includes form names, numbers, descriptions, and the DEFINE /FORM qualifier settings. The default is /BRIEF.

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

EXAMPLES

1
```
$ SHOW QUEUE/FORM DEFAULT
Form name                          Number   Description
---------                          ------   -----------
DEFAULT                                 0   System-defined default
```

This example displays only the default form.

2
```
$ SHOW QUEUE/FORM
Form name                          Number   Description
---------                          ------   -----------
132_51_STD (stock=DEFAULT)            102   132 by 51 (standard short)
40_66_STD (stock=DEFAULT)             103   40 by 66 (standard labels)
BLUE_PAPER_STOCK (stock=DIGITAL_8X11_STOCK1412TEA)
                                    22222   blue paper, DEC order# 22222
DEFAULT                                 0   System-defined default
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)
MEMO (stock=DEFAULT)                  110   LN03 indented memo format
```

This example displays all forms including the stock for each form.

3
```
$ 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)
```

This SHOW QUEUE/FORM command displays the names of all form types, including the stock, that begin with LN01. The names, stock for each form, numbers, and brief descriptions of those forms are listed.

SHOW QUEUE/FORM

```
$ 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
LNO1_LANDSCAPE (stock=DEFAULT)           105    132 by 66 (landscape)
    /LENGTH=66 /STOCK=DEFAULT /WIDTH=132
LNO1_LANDSCAPE_INDENTED (stock=DEFAULT)
                                         107    132 by 65 (landscape)
    /LENGTH=65 /SETUP=(LNO1_TOP_MARGIN_150) /STOCK=DEFAULT /WIDTH=132
LNO1_PORTRAIT (stock=DEFAULT)            106    80 by 60 (portrait)
    /LENGTH=60 /SETUP=(LNO1_PORTRAIT) /STOCK=DEFAULT /WIDTH=80
MEMO (stock=DEFAULT)                     110    LNO3 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 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.

FORMAT

SHOW QUOTA

restrictions

Requires read (R) access to the quota file in order to display the quotas of other users.

PARAMETERS

None.

DESCRIPTION

The SHOW QUOTA command indicates whether a quota exists for any specific user on a specific disk.

The display that results from the SHOW QUOTA command gives the quotas used, authorized, and available in blocks. The amount of overdraft permitted is also shown.

QUALIFIERS

/DISK[=device-name[:]]

Identifies the disk whose quotas are to be examined. SYS$DISK, the current default disk, is examined by default.

/USER=uic

Identifies which user's quotas are to be displayed. Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

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 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 displays the fact that the user with 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 illustrates a user with an overdrawn quota.

SHOW RMS_DEFAULT

Displays the current default multiblock count, multibuffer count, network transfer size, prologue level, and extend quantity that VAX RMS uses for file operations.

FORMAT

SHOW RMS_DEFAULT

restrictions *None.*

PARAMETERS

None.

QUALIFIER

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, 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

          Prologue    Extend Quantity
Process      0               0
System       0               0
```

The SHOW RMS_DEFAULT command displays the current process and system default multiblock and multibuffer counts for all types of files. It also displays the current process and system prologue level, extend quantity, and network transfer size.

SHOW STATUS

Displays the status of the current process.

FORMAT SHOW STATUS

restrictions *None.*

PARAMETERS *None.*

DESCRIPTION

You can use the SHOW STATUS command to verify that your current process is, in fact, running. The SHOW STATUS information can also indicate if the process is using an abnormal amount of CPU time, opening too many files, or accruing too many page faults.

The information displayed by SHOW STATUS is similar to that displayed by CTRL/T. (See the SET CONTROL command.)

EXAMPLE

```
$ SHOW STATUS
  Status on  15-APR-1985 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 displays the current status of the process, including:

- Current time and date
- Elapsed CPU time used by the current process
- Number of page faults
- Open file count
- Buffered I/O count
- Direct I/O count
- Current working set size
- Current amount of physical memory occupied

SHOW SYMBOL

Displays the current value of a local or global symbol. Symbols are defined with assignment statements (the = or := commands), by parameters passed to a command procedure file, or by the INQUIRE or READ commands.

FORMAT

SHOW SYMBOL *[symbol-name]*

restrictions

None.

PARAMETER

symbol-name

Specifies the name of the symbol whose value you want to display. The symbol name can contain from 1 to 255 characters, including alphanumeric characters from the DEC Multinational Character Set, underscores, or dollar signs. The symbol name must begin with a letter, an underscore, or a dollar sign. Wildcard characters are allowed in the symbol-name specification.

If you specify /ALL, you cannot specify a symbol name.

DESCRIPTION

The SHOW SYMBOL command searches the local symbol table of the current command level, then local symbol tables of preceding command levels, then the global symbol table of the specified symbol, and displays the first match it finds.

QUALIFIERS

/ALL

Requests that the current values of all symbols in the specified symbol table be displayed. If you specify /ALL and do not specify either /LOCAL or /GLOBAL, the SHOW SYMBOL command displays the contents of the local symbol table for the current command level.

/GLOBAL

Requests that only the global symbol table be searched for the specified symbol name.

If you specify both the /ALL and /GLOBAL qualifiers, all names in the global symbol table are displayed.

/LOCAL

Requests that only the local symbol table for the current command level be searched 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 an informational message is generated 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 requests that the current value of the symbol name PURGE be displayed. 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
      TIME == "SHOW TIME"
      LOG == "@LOG"
      $RESTART == "FALSE"
      $SEVERITY == "1"
      $STATUS == "%X00000001"
```

The SHOW SYMBOL command requests a display of all symbols defined in the global symbol table. Note that the symbols $RESTART, $STATUS, and $SEVERITY, which are maintained by the system, are also displayed.

```
3   $ SHOW SYMBOL/LOCAL TIME
    %DCL-W-UNDSYM, undefined symbol
```

The SHOW SYMBOL command requests that only the local symbol table be searched for the symbol TIME. The response indicates that TIME currently has no value.

SHOW SYSTEM

Displays a list of processes in the system and information about the status of each.

FORMAT

SHOW SYSTEM

restrictions *None.*

PARAMETERS

None.

QUALIFIERS

/BATCH

Requests the display of the batch jobs in the system. By default, all processes are displayed.

/FULL

Requests that the User Identification code (UIC) be displayed in addition to the default information. The UIC will be displayed on a second line, underneath the process name.

/NETWORK

Requests the display of the network processes in the system. By default, all processes are displayed.

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

/PROCESS (default)

Displays all processes in the system.

/SUBPROCESS

Requests the display of the subprocesses in the system. By default, all processes are displayed.

EXAMPLE

```
$ SHOW SYSTEM
VAX/VMS X2EN  on node DELPHI 24-FEB-1985 15:10:31.02   Uptime    0 12:06:30
  Pid     Process Name    State  Pri      I/O       CPU        Page flts Ph.Mem
22200080 NULL            COM      0        0   0 16:34:12.00          0    0
22200081 SWAPPER         HIB     16        0   0 00:01:07.92          0    0
22200202 Meg             LEF      4     6206   0 00:03:52.53      17509  174
22200085 ERRFMT          HIB      8     4123   0 00:00:45.35        145  191
22200086 CLUSTER_SERVER  HIB     10       23   0 00:00:01.63        130  167
22200087 OPCOM           RWAST    9     1114   0 00:00:39.60        189  274
22200088 JOB_CONTROL     HIB      8     6662   0 00:02:29.09        368  474
22200089 CONFIGURE       HIB     13       84   0 00:00:00.64        148  235
2220008A VAXsim_Monitor  HIB      8     1543   0 00:00:20.55        312  137
2220008C PARTITION_CHKR  LEF      4     3833   0 00:00:14.27        132  153
2220008D SYMBIONT_0001   HIB      4     2866   0 00:16:26.65       6751  329
2220008E Cerb Servant    LEF      4    12412   0 00:03:08.40        854  165
2220008F Monitor         LEF     15     1151   0 00:00:28.64       2396  350
22200090 NETACP          HIB      9    14820   0 00:02:42.85       8464  350
22200091 EVL             HIB      5      209   0 00:00:07.90       9996   49   N
22200092 REMACP          HIB      9      194   0 00:00:01.18        123   59
22200113 LIEBERWIRTH     PFW      4    16505   0 00:04:17.12      23318  912
22200196 Paul Beck       COM      4     7556   0 00:03:14.61      19376  280
22200119 THIEL           LEF      9     7680   0 00:01:36.09       8689  233
2220011D KATZ            LEF      9     6276   0 00:00:43.66       2006  200
2220019E SENN            LEF      4     2646   0 00:01:59.11       4847  280
222001A4 TAVANI          LEF      9    13514   0 00:01:49.52       3011  512
222001A6 C_EMACS         HIB      7    16165   0 00:05:10.86       6026  639   S
222001A7 PERKO           LEF      4     7572   0 00:02:13.84      16487  195
22200128 RAPPAPORT       LEF      4    15634   0 00:07:34.16      39591  450
2220022C ZALEWSKI        LEF      7     2297   0 00:00:52.86       6668  300
2220012E ANDERSON        LEF      9      671   0 00:00:12.02       1197  200
2220022F MORSE           LEF      4      741   0 00:00:13.13       2394  150
22200231 MSCPmount       LEF      6     1211   0 00:00:48.40       3793   92
222000B2 MCCOY           LEF      4    11710   0 00:01:54.90      12798  180
222000B5 cw hobbs ttg2   LEF      4     6076   0 00:01:26.10       8418  220
22200236 HOBBS           LEF      4      251   0 00:00:05.80        560  330
22200237 MAIL_6188       LEF      6      236   0 00:00:04.58        609  269   N
22200238 MAIL_6189       LEF      4      236   0 00:00:04.58        542  261   N
22200239 Mike Etzel TTG  COM      4      220   0 00:00:05.42        844  154
222000BC Greg A-z        LEF      9     9873   0 01:35:16.42      16557  949
222001C5 MARCHESANO      LEF      9      492   0 00:00:12.00       2171  150
222000C8 CARDOZA         LEF      8     9381   0 00:05:44.12      22090  227
2220014A RSPITZ          LEF      9     9512   0 00:02:03.49      10264  227
222000CB SCHAEFER        LEF      9     5676   0 00:01:56.31       9468  526
222000CC HUANG           LEF      9    43511   0 00:16:30.55      55396  163
222000D2 BLAIR           LEF      4    17315   0 00:05:25.43      62535  300
222000DE KENAH           LEF      7     4887   0 00:01:39.36       8800  544
2220015F MALIK           CUR      4    13671   0 00:03:10.51      19027  302
22200162 _TTH1:          LEF      4    19344   0 00:20:57.28      54894  269
222000E4 BATCH_970       LEF      6      669   0 00:00:09.79        639  386   B
22200167 ELFSTROM        HIB      5     3399   0 00:01:01.07      10761  192
222000E9 Greg A_z        LEF      6    12899   0 00:01:52.69       6624  447
```

The response displays:

- Process identification code (PID)—a 32-bit binary value that uniquely identifies a process
- Process name—a 1-15 character string used to identify a process
- Process state—the activity level of the process, such as COMputing, HIBernation, LEF (local event flag) wait
- Current priority—the priority level assigned to the process (the higher the number, the higher the priority)

- Total process I/O count ❶—the number of I/O operations involved in executing the process. This consists of both the direct I/O count and the buffered I/O count.
- Charged CPU time ❶—the amount of CPU time that a process has used thus far
- Number of page faults ❶—the number of exceptions generated by references to pages which are not in the process's working set
- Physical memory occupied ❶—the amount of space in physical memory that the process is currently occupying
- Process indicator—letter B indicates a batch job; letter S indicates a subprocess; letter N indicates a network process.
- User identification code (UIC)—an 8-digit octal number assigned to a process. This is only displayed if the /FULL qualifier is specified.

❶ 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 message:

– swapped out –

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[:]]*

restrictions

See qualifier descriptions.

PARAMETER

device-name[:]

Specifies the name of the terminal for which you want the characteristics displayed. If you do not specify a device name, the system displays characteristics of the current device assigned to the logical name 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.

QUALIFIERS

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

/PERMANENT

Requires LOG_IO or PHY_IO privileges.

Displays the current permanent characteristics of the specified terminal.

EXAMPLE

```
$ SHOW TERMINAL
Terminal: _TTE4:        Device_Type: VT102          Owner: FRANKLIN
   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           No Holdscreen
   No Eightbit         Broadcast           No Readsync         No Form
   Fulldup             No Modem            No Local_echo       No Autobaud
   No 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 SIXEL Graphics   No Soft Characters  Printer port        Numeric Keypad
   ANSI_CRT            No Regis            No Block_mode       Advanced_video
   Edit_mode           DEC_CRT
   Advanced_video     No Edit_mode       DEC_Crt
```

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.

SHOW TIME

Displays the current date and time. The DAY element is optional.

FORMAT

SHOW [DAY]TIME

restrictions *None.*

PARAMETERS

None.

EXAMPLE

```
$ SHOW TIME
  4-FEB-1985 00:03:45
```

The SHOW TIME command displays the current date and time at the terminal.

SHOW TRANSLATION

Searches one or more logical name tables for a specified logical name and returns the first equivalence name of the first match found.

FORMAT

SHOW TRANSLATION *logical-name*

restrictions

Requires read (R) access to a logical name table to display information about any logical name cataloged in that table.

PARAMETER

logical-name

Specifies the logical name for which you want to display the translation. The logical name can have from 1 to 255 alphanumeric characters.

DESCRIPTION

When you use the SHOW TRANSLATION command, you can specify the tables you want to search. If you do not specify a table, SHOW TRANSLATION searches the tables specified by the multivalued logical name LNM$DCL_LOGICAL.

Unless LNM$DCL_LOGICAL has been redefined for your process, the SHOW TRANSLATION command searches the process, job, group, and system logical name tables, in that order, for a specified logical name and returns the equivalence name of the first match found. (To see how LNM$DCL_LOGICAL is defined for your process, enter the command SHOW LOGICAL /TABLE=LNM$DIRECTORIES LNM$DCL_LOGICAL.)

If a table contains more than one logical name entry with the same name but each name has a different mode, then the SHOW TRANSLATION command returns the equivalence string for the name with the outermost (least privileged) mode.

SHOW TRANSLATION is similar to the SHOW LOGICAL command in that both commands return equivalence strings for logical names. However, the SHOW TRANSLATION command is executed within DCL, whereas the SHOW LOGICAL command executes an image. Therefore, SHOW TRANSLATION does not cause the current image to exit and does not deassign user mode logical names. Also, SHOW TRANSLATION does not display iterative translations of a name.

QUALIFIERS

/TABLE=name

Specifies the name of the table to be searched. If you specify the table name 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.

If you do not specify the /TABLE qualifier, the default is /TABLE=LNM$DCL_LOGICAL, as explained in the "Description" section.

EXAMPLES

1

```
$ SHOW TRANSLATION PAYROLL
  PAYROLL = DISK1:[ACCOUNTS.WORKING]FACTOR1.DAT;37 (LNM$PROCESS_TABLE)
```

The SHOW TRANSLATION command displays the current equivalence name of the logical name PAYROLL. The command also displays the name of the table where the logical name was found. In this example, PAYROLL was found in the process logical name table, LNM$PROCESS_TABLE.

2

```
$ DEFINE DISK  DBA1:
$ DEFINE/GROUP DISK  DBA2:
$ SHOW TRANSLATION DISK
  DISK = DBA1:(LNM$PROCESS_TABLE)
```

The DEFINE commands place entries for the logical name DISK in both the process and group logical name tables. Then, the SHOW TRANSLATION command shows the equivalence name associated with the logical name DISK. By default, the SHOW TRANSLATION command searches process, job, group, and system tables, in that order, and displays the first match that it finds. The logical name DISK from the process logical name table (LNM$PROCESS_TABLE) is displayed because it is found before the name DISK in the group table.

3

```
$ RUN ORION
^Y
$ SHOW TRANSLATION TERMINAL
  TERMINAL = _TTT3: (LNM$PROCESS_TABLE)
$ CONTINUE
```

The RUN command executes the image ORION.EXE. After CTRL/Y 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$SYSTEM USER
  USER = "DBA2:"  (LNM$SYSTEM_TABLE)
```

This command displays the equivalence string for the logical name USER, which is located in the table LNM$SYSTEM. LNM$SYSTEM is a logical name that translates to the name LNM$SYSTEM_TABLE, the system logical name table. Because a table name is specified, the SHOW TRANSLATION does not use the default search order. The command searches for the name only in the specified table.

5

```
$ DEFINE/TABLE=LNM$PROCESS_DIRECTORY MYPROC -
_$ TEST_TABLE, LNM$PROCESS
$ SHOW TRANSLATION/TABLE=MYPROC FILER
  FILER = "[SMITH.FILER]"   (TEST_TABLE)
```

This example defines a logical name (MYPROC) that is equated to two table names: TEST_TABLE and LNM$PROCESS. Because the logical name translates iteratively to a logical name table, MYPROC must be placed in a directory table. When you specify MYPROC with the SHOW TRANSLATION command, the tables TEST_TABLE and LNM$PROCESS are searched, in that order. The first match found is displayed.

SHOW USERS

Displays the terminal name, username, and process identification code (PID) of either specific interactive users or all interactive users on the system.

FORMAT

SHOW USERS *[username]*

restrictions

None.

PARAMETER

username

Specifies the user about whom you want information. If you specify a string, the system lists all interactive users whose usernames begin with the specified character string. For example, if the string MAR is specified as a username, a list of all interactive users whose usernames begin with MAR is displayed. If no user exists whose username matches the specified username string, an error message is displayed.

If you omit this parameter, a list of all interactive users is displayed.

QUALIFIER

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

EXAMPLES

1
```
$ SHOW USERS
          VAX/VMS Interactive Users
            4-FEB-1985 12:48:51.14
    Total number of interactive users = 14

  PID     Username      Process Name    Terminal
202000B3 <login>       _VTA9:          VTA9:          TTA7:
204000C4 AHO           AHO             VTA21:         LTA8:
2040013A ETZEL         M Etzel VTA43:  VTA43:         TTA1:
20400138 FRISSELLE     FRISSELLE       VTA42:         TTA5:
20400095 HOBBS         cw hobbs        VTA1:          TTD5:
204000DC HUANG         _VTA32:         VTA32:         LTA11:
204000B9 KUEHN         KUEHN           VTA14:         TTB0:
20400123 MALIK         MALIK           VTA38:         TTB6:
20400113 MCLAUGHLIN    MCLAUGHLIN      VTA35:         TTB3:
204000BC MURRAY        MURRAY          VTA16:         TTA2:
204000C2 OPERATOR      OPERATOR        VTA20:         OPA0:
2040012C PERRON        PERRON          VTA40:         TTA4:
2040013D POLLACK       POLLACK         VTA45:         TTA3:
20400097 STEEVES       STEEVES         VTA2:          LTA1:
```

The SHOW USERS command displays the process identification code (PID), the user name, process name, and terminal names (both virtual and physical) of all interactive users currently on the system. A user name of <login> indicates that someone is in the process of logging in.

2
```
$ SHOW USERS GARGARIAN
          VAX/VMS Interactive Users
           4-FEB-1985 16:45:14.14
    Total number of interactive users = 32

  PID     Username      Process Name    Terminal
20200115 GARGARIAN     GARGARIAN       VTA3:          TTA7:
```

The SHOW USERS command displays the process identification code (PID), the user name, process name, and terminal names of the interactive user GARGARIAN.

3
```
$ SHOW USERS J
          VAX/VMS Interactive Users
           4-FEB-1985 16:45:11.66
    Total number of interactive users = 32

  PID     Username      Process Name    Terminal
202000DB JACK          JACK            VTA19:         TTC1:
2020011D JEFFREYS      JEFFREYS        VTA20:         TTA6:
```

The SHOW USERS command displays the process identification code (PID), the user name, process name, and terminal names of all interactive users whose usernames begin with the letter J.

SHOW WORKING_SET

Displays the working set limit, quota, and extent assigned to the current process.

FORMAT

SHOW WORKING_SET

restrictions

None.

PARAMETERS

None.

QUALIFIER

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT 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 any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

EXAMPLE

```
$ SHOW WORKING_SET
  Working Set      /Limit= 180   /Quota= 350           /Extent= 1200
  Adjustment enabled    Authorized Quota= 350  Authorized Extent= 1200
```

The response to the SHOW WORKING_SET command indicates that the current process has a working set limit of 180 pages, a quota of 350 pages and that the current 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

Invokes the VAX Sort Utility (SORT) to reorder the records in a file into a defined sequence and to create either a new file of the reordered records or an address file by which the reordered records can be accessed. For a complete functional description of the Sort Utility, including more information about the SORT command, see the *VAX/VMS Sort/Merge Utility Reference Manual*.

FORMAT **SORT** *input-file-spec[,...] output-file-spec*

SPAWN

Creates a subprocess of the current process. Portions of the current process context are copied to the subprocess.

FORMAT

SPAWN *[command-string]*

restrictions

- The RESOURCE_WAIT state is required to spawn a process.
- Requires TMPMBX or PRMMBX user privilege.
- SPAWN does not manage terminal characteristics.
- The SPAWN and ATTACH commands cannot be used if your terminal has an associated mailbox.

PARAMETER

command-string

Specifies a command string to be executed in the context of the created subprocess. When the command completes, the subprocess terminates and control is returned to the parent process. If you use both the /INPUT qualifier and a command string, commands are obtained from the input file after the specified command string executes. The command string cannot exceed 132 characters.

DESCRIPTION

The SPAWN command enables you to create a subprocess of your current process with the following context copied from the parent process:

- All symbols except $RESTART, $SEVERITY, and $STATUS
- 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 context items, 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 the same base name as the parent process and a unique number. For example, if the parent process name is SMITH, the subprocess name can be SMITH_1, SMITH_2, and so on.

The LOGIN.COM file of the parent process is not executed for the subprocess since the context is copied separately, thus allowing for 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 will simultaneously attempt to use the same input or output stream when several processes share the same stream and you terminate a subprocess to which you are not currently attached (or 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 (see ATTACH) 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 automatically terminated.

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/line feed characters are prefixed to the subprocess's prompt string. By default, SPAWN copies the current setting of the parent process.

/CLI=cli
/NOCLI

Specifies the name of an alternate command language interpreter (CLI) to be used by the subprocess. If the /CLI qualifier is not specified, the CLI used will be the same as the parent process.

The CLI you specify must be located in SYS$SYSTEM and have the file type EXE.

/INPUT=file-spec

Specifies an input file containing one or more DCL commands to be executed by the spawned subprocess. If you specify a command string with the SPAWN command and an input file with the /INPUT qualifier, the command string is processed *before* the input file. Once processing of the input file is complete, the subprocess is terminated.

No wildcards are allowed in the file specification.

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 /INPUT.

Note that when NRO PPFs are used as implicit input (that is, /INPUT is not specified and SYS$INPUT is a NRO PPF), the SPAWN command can succeed. The following chart shows what happens.

Process Type	SYS$INPUT	Implicit Input
Interactive	NRO PPF	SYS$COMMAND
Non-interactive	NRO PPF	Null Device
Any	Any other	SYS$INPUT

If SYS$INPUT is a terminal, it cannot have an associated terminal mailbox.

/KEYPAD (default)
/NOKEYPAD

Determines whether keypad symbols and the current keypad state are copied from the parent process to the subprocess. By default, if you have established key definitions and/or states with the DEFINE/KEY and/or SET KEY commands, 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

Controls whether the assigned subprocess name is diplayed along with any messages indicating transfer of control between processes.

/LOGICAL_NAMES (default)
/NOLOGICAL_NAMES

Determines whether the system passes 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 the terminal at which you are logged in, notifying you that your subprocess has been completed or aborted. This qualifier should not be used unless you specify the /NOWAIT qualifier. /NOTIFY also cannot be specified when the SPAWN command is executed from within a non-interactive process.

Note that messages broadcast as a result of specifying the /NOTIFY qualifier are considered to be DCL messages. Therefore, if SET BROADCAST=NODCL is in effect, all such notification messages will be suppressed.

/OUTPUT=file-spec

Requests that the output from the SPAWN operation be written to the specified file.

Note that if both the /OUTPUT and /NOWAIT qualifiers are used interactively, and if SYS$COMMAND is specified as the /OUTPUT file specification, the output from both your parent process and your subprocesses will be displayed simultaneously at your terminal.

No wildcards are allowed in the file specification.

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 /OUTPUT.

Note that when NRO PPFs are used as implicit output, the SPAWN command can succeed. The following chart shows what happens.

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.

/PROCESS=subprocess-name

Specifies the name of the subprocess to be created. If you specify a process name that already exists, an error message is displayed. By default, 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 the /LOG qualifier has been specified, the assigned name of the subprocess is displayed.

/PROMPT[=string]

Specifies the prompt string for DCL to use in the subprocess. By default, SPAWN copies the current prompt from the parent process.

The string can consist of more than one character. All valid ASCII characters can be used in the string. In order to include spaces, lowercase letters or non-alphanumeric characters in your string, you must enclose the string in quotation marks. Otherwise, letters are automatically converted to uppercase, and leading and trailing spaces are removed.

If no string is specified, the DCL default prompt string "$ " is restored.

/SYMBOLS (default)
/NOSYMBOLS

Determines whether the system passes DCL global and local symbols to the subprocess. Note that the $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.

/WAIT (default)
/NOWAIT

Controls whether the system waits until the current subprocess is completed before allowing more commands to be issued in the parent process. The /NOWAIT qualifier allows you to issue new commands while the specified subprocess is running. When you use the /NOWAIT qualifier interactively, be sure to use the /OUTPUT qualifier as well so that output from the subprocess is directed to a file rather than to your terminal. Otherwise, your terminal will be used by more than one process at the same time.

Note that specifying the /NOWAIT qualifier will cause 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, will also affect all subprocesses sharing the input device. CTRL/Y, for example, will interrupt all such subprocesses.

This problem may be avoided by specifying /INPUT=NL:.

EXAMPLE

```
$ RUN MYPROG
      .
      .
      .
$ ^Y
$ SPAWN MAIL
%DCL-S-SPAWNED, process SMITH_1 spawned
%DCL-S-ATTACHED, terminal now attached to process SMITH_1
MAIL> READ
   .
   .
   .
MAIL> EXIT
%DCL-S-RETURNED, control returned to process SMITH
$ CONTINUE
```

The SPAWN command in this example allows you to enter the VAX/VMS Mail Utility without terminating the currently running program. After you exit from MAIL, control is returned to the parent process.

START/CPU

Starts the attached processor in a VAX/VMS multiprocessing system. The /CPU qualifier is required.

FORMAT

START/CPU

restrictions

- Applies only to VAX/VMS multiprocessing systems.
- Requires change mode to kernel (CMKRNL) privilege.

PARAMETERS

None.

DESCRIPTION

Use the START/CPU command to start the attached processor.

There are six possible states the attached processor can be in: STOP, INITIALIZE, IDLE, BUSY, EXECUTE, and DROP. If the attached processor is in the STOP state, the START/CPU command causes it to enter the INITIALIZE state. If the attached processor is in any state other than STOP, the START/CPU command has no effect, because the attached processor either is already running (the IDLE, BUSY, EXECUTE, or DROP state) or is initializing itself to prepare to run (the INITIALIZE state).

EXAMPLE

```
$ START/CPU
```

This command starts the attached processor in a VAX/VMS multiprocessing system.

START/QUEUE

Starts or restarts the specified queue. The /QUEUE qualifier is required.

FORMAT

START/QUEUE *queue-name[:]*

restrictions

Requires operator (OPER) privilege or execute (E) access to the specified queue.

PARAMETER

queue-name[:]

Specifies the name of the queue to be started or restarted.

DESCRIPTION

Use the START/QUEUE command to restart a queue that has been stopped for some reason. In order to use the START/QUEUE command, the queue must already have been initialized with the INITIALIZE/QUEUE command.

In general, when you want to start a newly created queue, you will use the INITIALIZE/QUEUE/START command to both initialize and start the queue with one command. If, for some reason, you want to start the queue at a later time, you can use the INITIALIZE/QUEUE command to initialize the queue and subsequently issue the START/QUEUE command to actually start the queue.

You can also use the START/QUEUE command to restart a queue after it has been stopped. When you issue the START/QUEUE command, you can reset the queue features.

Note that any qualifier that can be used with the INITIALIZE/QUEUE command can be specified with the START/QUEUE command, except for /START. The START/QUEUE command can override options that have been specified with the INITIALIZE/QUEUE command, such as /ENABLE_GENERIC, /DEFAULT, and /FORM_MOUNTED.

If the specified queue is running when you issue the START/QUEUE command, the system returns an error message. Use the SET QUEUE command to change the attributes of a running queue.

When you specify more than one of the following qualifiers, the system processes them in the order listed:

- /TOP_OF_FILE
- /BACKWORD or /FORWARD
- /SEARCH
- /ALIGN

QUALIFIERS

/ALIGN[=(option[,...])]

Determines whether alignment pages are printed when the queue is resumed. Alignment pages enable the operator to properly align the forms in the printer or terminal. Use this qualifier in restarting an output queue from a paused state. Possible options are:

MASK	Displays alphabetic characters as x's and numbers as 9's; nonalphanumeric characters are not masked. The default is not to mask.
n	Specifies the number of alignment pages to print. The value of *n* can be from 1 to 20; the default is 1.

/BACKWARD=n

Printing resumes after backing up one or more pages of the current file in the current job; the default is 1. Use this qualifier in restarting an output queue from a paused state.

/BASE_PRIORITY=n

Specifies the base process priority at which jobs are initiated from a batch queue or the base priority of the symbiont process for printer, terminal, or server queues. By default, if you omit the qualifier, jobs are initiated at the same priority as the base priority established by DEFPRI at system generation. The value n can be any decimal value from 0 through 15.

/BATCH
/NOBATCH (default)

Indicates that you are starting a batch queue. You cannot use the /BATCH qualifier unless the queue you are starting was initialized as a batch queue. If an existing queue is a batch queue, you can optionally use the /BATCH qualifier.

/BLOCK_LIMIT=([lowlim,]uplim)
/NOBLOCK_LIMIT

Limits the size of print jobs that can be executed on a printer or terminal 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 will be 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 in a pending state until the job limit for the queue is changed, enabling the job to execute.

The uplim parameter is a decimal number referring to the maximum number of blocks that will be accepted by the queue for a print job. If a print job is submitted that exceeds this value, the job remains in a pending state until the limit for the queue is changed, enabling the job to execute.

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 will execute in the queue. In order to specify only a lower job limit, you must use two sets of double quotation marks to indicate the upper specifier. For example, /BLOCK_LIMIT=(500,"") means any job with 500 or more

blocks will execute 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 will not run in the queue.

The /NOBLOCK_LIMIT qualifier cancels the /BLOCK_LIMIT setting previously established for that queue.

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

Specifies one or more characteristics for processing jobs on the queue. If only one characteristic is specified, you can omit the parentheses.

Each time you specify /CHARACTERISTICS, all previously set characteristics are erased. Only the characteristics specified with the qualifier are now established for the queue.

Queue characteristics are installation-specific. The characteristic parameter can be either a value from 0 through 127 or a characteristic name that has been defined by the DEFINE/CHARACTERISTIC command.

When users include the /CHARACTERISTICS qualifier with a PRINT or SUBMIT command, all the characteristics they specify must also be specified for the queue that will be executing the job. If not, the job will remain pending in the queue until the queue characteristics are changed or they delete the entry with the DELETE/ENTRY command. Users need not specify every characteristic of a queue with a PRINT or SUBMIT command as long as the ones they specify are a subset of the characteristics set for that queue. The job will also run if no characteristics are specified.

The /NOCHARACTERISTICS qualifier cancels any /CHARACTERISTICS settings previously established for that queue.

/CPUDEFAULT=time

Defines the default CPU time limit for batch jobs.

Specify the CPU default time as a delta time value, the numeric value 0, or the keyword NONE or INFINITE. The value 0 and the keyword INFINITE allow a job unlimited CPU time, subject to restrictions imposed by the /CPUMAXIMUM qualifier or the user authorization file. Specify NONE when a default CPU time limit is not needed.

The value for time cannot exceed the CPU time limit set by the /CPUMAXIMUM qualifier.

See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for information on specifying delta times.

/CPUMAXIMUM=time

Defines the maximum CPU time limit for batch jobs. Use this qualifier to override the CPU time limit specified in the user authorization file (UAF).

Specify the CPU maximum time as a delta time value, the numeric value 0, or the word NONE or INFINITE. Specify NONE when a maximum CPU time limit is not desired. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for information on specifying delta times.

A CPU time limit for processes is specified by each user record in the system UAF. You can also specify the following: a default CPU time limit for all jobs in a given queue and a maximum CPU time limit for all jobs in a given queue. See Table DCL–1 for information on what action is taken for each value specified and for the possible combinations of specifications.

/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. Once an option is set for the queue by the /DEFAULT qualifier, users do not have to specify that option in their PRINT commands.

The options are:

[NO]BURST[=keyword]	Specifies whether file burst pages will be printed. If the keyword is ALL (the default), burst pages are placed before each file in the print job. If the keyword is ONE, a burst page is placed before the first copy of the first file in the job.
[NO]FEED	Specifies whether a form-feed is automatically inserted at the end of a page.
FORM=type	Specifies the default form for a printer, terminal, or server queue. If a job is not submitted with an explicit form definition, then this form will be used to process the job. The systemwide default form, form=0, is the default value for this keyword. See also /FORM_MOUNTED.
[NO]FLAG[=keyword]	Specifies whether file flag pages will be printed. If the keyword is ALL (the default), flag pages are placed before each file in the print job. If the keyword is ONE, a flag page is placed before the first copy of the first file in the job.
[NO]TRAILER[=keyword]	Specifies whether file trailer pages will be printed. If the keyword is ALL (the default), trailer pages are placed at the end of each file in the print job. If the keyword is ONE, a trailer page is placed after the last copy of the last file in the job.

If you specify any of the keywords BURST, FLAG, TRAILER without specifying a value, the value ALL is used by default.

/DISABLE_SWAPPING
/NODISABLE_SWAPPING (default)

Controls whether batch jobs executed from a queue can be swapped in and out of memory.

/ENABLE_GENERIC (default)
/NOENABLE_GENERIC

Specifies whether files queued to a generic queue that does not specify explicit queue names in the /GENERIC qualifier can be placed in this execution queue for processing.

/FORM_MOUNTED=type

Specifies the form type for a printer, terminal, or server queue. If the stock of the mounted form is not identical to the stock of the default form, as indicated by the DCL command qualifier /DEFAULT=FORM=type, then all jobs submitted to this queue without an explicit form definition will enter a pending state. 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, then the job will enter a pending state. In both cases, the pending state will be maintained until the stock of the mounted form of the queue is identical to the stock of the form associated with the job.

Specify the form type using a numeric value or a form name that has been defined by the DEFINE/FORM command. Form types are installation-specific.

/FORWARD=n

Printing resumes after advancing one or more pages of the the current file in the current job; the default is 1. Use this qualifier in restarting an output queue from a paused state.

/GENERIC[=(queue-name[,...])]
/NOGENERIC (default)

Specifies that this is a generic queue and that jobs placed in it will be moved for processing to compatible execution queues.

By default, a generic queue is a generic printer queue. You establish a generic batch queue by including the /BATCH qualifier, a generic terminal queue by including the /TERMINAL qualifier, and a generic server queue by including the /PROCESSOR qualifier.

The /BATCH qualifier determines that an execution queue is a batch queue. The symbiont process determines whether queues are printer, terminal, or server queues; the standard symbiont sets this characteristic depending upon whether the output device is a printer or a terminal.

The /GENERIC qualifier optionally accepts a list of existing execution queues into which the generic queue can place work. If you list one or more queues, jobs can only be moved to the specified queues. If you specify the /BATCH qualifier, all queues specified with the /GENERIC qualifier must be batch queues.

If you do not specify any queue names with the /GENERIC qualifier, jobs can be moved to any execution queue that is initialized without the /NOENABLE_GENERIC qualifier, and that is the same type (batch, printer, terminal, or server) as the generic queue. If you are starting a generic server queue, the /PROCESSOR qualifier specified for the generic queue must match the /PROCESSOR qualifiers of the coordinate execution queues.

/JOB_LIMIT=n

Indicates the number of batch jobs that can be executed concurrently from the queue. The job limit default value for n is 1.

/LIBRARY=file-name
/NOLIBRARY

Specifies the file name for the device control library. When you are initializing a symbiont queue, you can use the /LIBRARY qualifier to specify an alternate device control library. The default library is

SYS$LIBRARY:SYSDEVCTL.TLB. Only a file name can be used as the parameter of the /LIBRARY qualifier. The system always assumes that the location of the file is in SYS$LIBRARY and that the file type is TLB.

/NEXT

Indicates that the current job will not resume printing when the queue is restarted. By default, the job that was executing when the queue stopped resumes printing if it has not been deleted. Use the /NEXT qualifier to abort the current job and start with the next job in the queue.

/ON=[node::]device[:] (printer, terminal, server queue)
/ON=node:: (batch queue)

Specifies the node and/or device on which this execution queue is located. For batch queues, only the node name can be specified. You can include both the node name and the device name for printer and terminal queues. By default, a queue executes on the same node from which you first start the queue. The default device parameter is the same as the queue name.

The node name is used only in VAXcluster systems; it must match the node name specified by the SYSGEN parameter SCSNODE for the processor on which the queue executes.

/OWNER_UIC=uic

Enables you to change the UIC of the queue. Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*. The default UIC is [1,4]. You must have OPER (operator) privilege in order to use this qualifier.

/PROCESSOR=file-name
/NOPROCESSOR

Allows users to specify their own print symbionts. The file name specifier can be any valid file name. The system supplies the device and directory name SYS$SYSTEM as well as the file type EXE.

If you use this qualifier for a output queue, it specifies that the symbiont image to be executed is SYS$SYSTEM:file-name.EXE. By default, SYS$SYSTEM:PRTSMB.EXE is executed.

If you use this qualifier for a generic queue, it specifies that the generic queue can place jobs only on queues established as server queues and that are executing the specified symbiont image.

The /NOPROCESSOR qualifier cancels the effect of a previous /PROCESSOR setting.

/PROTECTION=(codes)

Specifies the protection of the queue. By default, the queue protection is (SYSTEM:E, OWNER:D, GROUP:R, WORLD:W). If you include only one protection code, you can omit the parentheses. You must have OPER (operator) privilege in order to use this qualifier.

/RECORD_BLOCKING (default)
/NORECORD_BLOCKING

Determines whether the symbiont can concatenate (or block together) output records for transmission to the output device. If you specify /NORECORD_BLOCKING, the symbiont is directed to send each formatted record in a separate I/O request to the output device. For the standard VMS print symbiont, record blocking can have a significant performance advantage over single-record mode.

/RETAIN[=option]
/NORETAIN

Specifies that jobs be retained in the queue in a completed status after they have executed. By default, no jobs are retained in the queue after execution.

The option parameter can be the keyword ALL or ERROR. If ALL is specified, all jobs are retained in the queue after execution. If ERROR is specified, only jobs that have completed unsuccessfully are retained in the queue. If you omit the option parameter, ALL is assumed.

The /NORETAIN qualifier enables you to reset the queue to the default.

/SCHEDULE=[NO]SIZE

Specifies whether pending jobs in a printer or terminal queue are scheduled for printing based on the size of the job. When the default, /SCHEDULE=SIZE, is in effect, smaller jobs will be printed before longer ones.

If you issue this command while there are pending jobs in any queue, its effect on future jobs is unpredictable.

/SEARCH="search-string"

Enables you to resume printing the current file of the current job on the first page containing the specified string. The search string parameter can have from 1 to 63 characters. Enclose the search string in quotation marks.

Use this qualifier in restarting an output queue from a paused state.

/SEPARATE=(option[,...])
/NOSEPARATE

Specifies the job separation defaults for a printer or terminal queue. The job separation options are:

[NO]BURST	Specifies whether a burst page will be printed at the beginning of every job. Specifying BURST also results in a flag page being printed.
[NO]FLAG	Specifies whether a flag page will be printed at the beginning of every job.
[NO]TRAILER	Specifies whether a trailer page will be printed at the end of every job.
[NO]RESET=(module[,...])	Specifies a job reset sequence for the queue. The specified modules from the device control library are used to reset the device each time a job reset occurs.

/TERMINAL
/NOTERMINAL

Indicates that a generic queue will be associated with terminal queues instead of printer queues. The /TERMINAL qualifier allows all jobs entered in the generic queue to be moved to terminal queues with matching settings.

The /NOTERMINAL qualifier cancels the effect of a previous /TERMINAL setting.

/TOP_OF_FILE

Resumes printing at the beginning of the file. Resumes printing the file that is the current file when the queue is in a paused state. Use this qualifier only when restarting an output queue from a paused state.

/WSDEFAULT=n

Defines a working set default for a batch 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.

A positive integer in the range 1 through 65,535, 0, or the word NONE can be specified for n. If you specify 0 or NONE, the working set default value becomes the value specified either in the UAF or by the SUBMIT command (if specified). For more information see Table DCL–2.

You can also specify this qualifier for an output queue. Used in this context, it establishes the working set default of the symbiont process for a printer, terminal, or server queue when the symbiont process is created.

/WSEXTENT=n

Defines a working set extent for the batch 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.

A positive integer in the range 1 through 65,535, 0, or the word NONE can be specified for n. If you specify 0 or NONE, the working set value becomes the value specified either in the UAF or by the SUBMIT command (if specified). For more information see Table DCL–2.

You can also specify this qualifier for an output queue. Used in this context, it establishes the working set extent of the symbiont process for a printer, terminal, or server queue when the symbiont process is created.

/WSQUOTA=n

Defines the working set page size (working set quota) for a batch 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 a positive integer in the range 1 through 65,535, 0, or the word NONE as the value for n. If 0 or NONE is specified for n, the working set quota defaults to the value specified either in the UAF or by the SUBMIT command (if specified).

A working set default size and a working set quota (maximum size) are included in each user record in the system UAF and can be specified for individual jobs and/or for all jobs in a given queue. For more information see Table DCL–2.

You can also specify this qualifier for an output queue. Used in this context, it establishes the working set quota of the symbiont process for a printer, terminal, or server queue when the symbiont process is created.

EXAMPLES

1
```
$ START/QUEUE/BATCH SYS$BATCH
```

This command starts the batch queue named SYS$BATCH. The /BATCH qualifier indicates that this is a batch queue.

2
```
$ STOP/QUEUE  LPA0
$ START/QUEUE/TOP_OF_FILE  LPA0
```

The STOP/QUEUE command suspends operation of the printer queue LPA0. Then the START/QUEUE/TOP_OF_FILE command resumes operation. The file that was being printed when the queue was stopped is started again from the beginning.

3
```
$ INITIALIZE/QUEUE  LPA0
      .
      .
      .
$ START/QUEUE/DEFAULT=FLAG  LPA0
```

The INITIALIZE/QUEUE command 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.

4
```
$ START/QUEUE/DEFAULT=FORM=LN01_PORTRAIT LN01_PRINT
```

The START/QUEUE command restarts the the LN01_PRINT queue with the default form LN01_PORTRAIT.

START/QUEUE/MANAGER

Starts the queue manager for the batch/print facility and opens the job queue manager file. After the system is bootstrapped, you must execute this command before you can execute any other queue management or job submission command. The /QUEUE qualifier is optional, but you must specify the /MANAGER qualifier.

For more information, see the *VAX/VMS System Manager's Reference Manual*.

FORMAT

START/QUEUE/MANAGER *[file-spec]*

restrictions

Requires both operator (OPER) and system name (SYSNAM) privileges.

PARAMETER

file-spec

Specifies the name of the file to contain information about batch and print jobs, queues, and form definitions. The file specification parameter is used in VAXcluster systems or for specifying an alternate system job queue file. The default file specification is SYS$SYSTEM:JBCSYSQUE.DAT. Any elements that you omit from the file specification default to those of SYS$SYSTEM:JBCSYSQUE.DAT.

No wildcard characters are permitted in the file specification.

DESCRIPTION

The START/QUEUE/MANAGER command is generally included in the system startup procedure. You can omit the file specification if you want your system to use the default job queue manager file.

You must also ensure that the definition of each execution queue (by the INITIALIZE/QUEUE or START/QUEUE command) contains the /ON qualifier. The node name specified with the /ON qualifier must match the system's node name as defined by the SYSGEN parameter SCSNODE.

In a VAXcluster, you must include the file specification parameter to cause each system to access the same job queue manager file on a shared disk volume. The file specification must include at least the device and directory names.

QUALIFIER

/BUFFER_COUNT=n

Specifies the number of buffers in a local buffer cache to allocate for performing I/O operations to the system job queue file.

Specify a positive integer in the range of 1 through 127, or 0. If 0 is specified, the default value of 50 is used.

/EXTEND_QUANTITY=n

Specifies the number of blocks by which the system job queue file is extended, when necessary. This value is also used as the initial allocation size when the queue file is created.

Specify a positive integer in the range of 10 through 65,535, or 0. If 0 is specified, the default value of 100 is used.

/NEW_VERSION
/NONEW_VERSION (default)

Enables you to specify that a new version of the job queue manager file be created to supersede an existing version. The new file contains no information until you issue a subsequent INITIALIZE/QUEUE command.

/RESTART
/NORESTART (default)

The /RESTART qualifier specifies that the queue manager be restarted automatically on recovery from a job controller abort. In addition, batch and output queues are restored to the states that existed prior to the interruption of service. The job queue manager file that is opened is the same file that was open before the abort. Upon restarting, the job controller uses the default values for the /EXTEND_QUANTITY and /BUFFER_COUNT qualifiers. Previously set values are lost.

When the job controller incurs an internal fatal error, the process aborts and restarts itself. By default, the queue manager is not restarted. Intervention by a user with OPERATOR privilege is necessary to restart the queue manager and to restore the queueing environment using START/QUEUE/MANAGER and appropriate START/QUEUE commands.

Note: In order to prevent a looping condition, the job controller will not restart the queue manager if it detects an error within two minutes of starting the queue manager.

EXAMPLES

1
```
$ START/QUEUE/MANAGER
```

This command opens the default job queue manager file.

2
```
$ START/QUEUE/MANAGER DUA5:[SYSQUE]
```

This command opens the job queue manager file JBCSYSQUE.DAT on the cluster-accessible disk volume DUA5, in directory SYSQUE. You must mount the disk before you issue the START/QUEUE/MANAGER command.

STOP

Terminates execution of:

- A command, image, or command procedure that was interrupted by CTRL/Y
- A command procedure
- A subprocess or a detached process

FORMAT

STOP *[process-name]*

restrictions

- Requires GROUP privilege to stop other processes in the same group.
- Requires WORLD privilege to stop processes outside your group.

PARAMETER

process-name

Specifies the name of the process to be deleted. The process name can have from 1 to 15 alphanumeric characters. The specified process must have the same group number in its user identification code (UIC) as the current process. You cannot specify the process-name for a process outside of your group. To stop a process outside of your group, you must use the qualifier /IDENTIFICATION=pid.

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 any exit-handling routines, they are not given control. Use the EXIT command to terminate the image so that the exit-handling routines gain control.

Note that when an image has been interrupted by CTRL/Y and the RUN command is issued to execute another image, the interrupted image is terminated. However, in this case exit-handling routines will execute before the next image is run.

If the STOP command is executed from a noninteractive process (such as a batch job), the process terminates.

If you use CTRL/Y to interrupt a command procedure and then issue 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 code (PID), 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.

QUALIFIER

/IDENTIFICATION=pid

Specifies the process identification code (PID) that the system has assigned to the process. When you create a process with the RUN command, the RUN command displays the process identification code of the newly created process.

You can omit any leading zeros in specifying the PID.

EXAMPLES

1
```
$ RUN MYPROG
    .
    .
    .
CTRL/Y
Interrupt
$ STOP
```

The RUN command begins executing the image MYPROG. Subsequently, CTRL/Y interrupts the execution. The STOP command then terminates the image.

2
```
$ @TESTALL
    .
    .
    .
CTRL/Y
Interrupt
$ STOP
```

The @ (Execute Procedure) command executes the procedure TESTALL.COM. CTRL/Y interrupts the procedure. The STOP command 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 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.

STOP/CPU

Stops the attached processor in a VAX/VMS multiprocessing system. The /CPU qualifier is required.

FORMAT

STOP/CPU

restrictions

- Applies only to VAX/VMS multiprocessing systems.
- Requires change mode to kernel (CMKRNL) privilege.

PARAMETERS

None.

DESCRIPTION

Use the STOP/CPU command to stop the attached processor in a VAX/VMS multiprocessing system.

If the attached processor is not executing a process when the STOP/CPU command is issued, it enters the STOP state. If the attached processor is executing a process when the command is issued, it returns the process to the primary processor before entering the STOP state.

If the attached processor is already in the STOP state when the STOP/CPU command is issued, the command has no effect.

While the attached processor is in the STOP state, the primary processor will not schedule work for or make any requests of the attached processor.

EXAMPLE

```
$ STOP/CPU
```

This command stops the attached processor in a VAX/VMS multiprocessing system.

STOP/QUEUE

The STOP/QUEUE command causes the specified execution queue to pause. The /QUEUE qualifier is required.

FORMAT

STOP/QUEUE *queue-name[:]*

restrictions

Requires operator (OPER) privilege or execute (E) access to the queue.

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 can be 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, /FORWARD, /SEARCH, or /TOP_OF_FILE qualifier to have a print job restart at a different place.

For information on other STOP/QUEUE commands, see:

STOP/QUEUE/ABORT
STOP/QUEUE/ENTRY
STOP/QUEUE/MANAGER
STOP/QUEUE/NEXT
STOP/QUEUE/REQUEUE
STOP/QUEUE/RESET

EXAMPLES

1
```
$ STOP/QUEUE LPA0
```

This STOP/QUEUE command halts the current print job in the queue LPA0 and places that queue in the paused state.

2
```
$ STOP/QUEUE TEXTBATCH
.
.
.
$ START/QUEUE/BLOCK_LIMIT=500 TEXTBATCH
```

This STOP/QUEUE command halts all batch jobs that are currently executing on the queue TEXTBATCH and places that queue in the paused state. Later the START/QUEUE command releases the queue from the paused state. All the jobs that were halted will resume processing, but the START/QUEUE command now limits any further jobs to 500 blocks or smaller.

STOP/QUEUE/ABORT

Causes an executing job on the specified output queue to be stopped. The queue does not stop; more jobs are processed if they are available. The /QUEUE qualifier is optional, but you must specify the /ABORT qualifier.

FORMAT

STOP/QUEUE/ABORT *queue-name[:]*

restrictions

Requires operator (OPER) privilege, execute (E) access to the queue, or delete (D) access to the current job.

PARAMETER

queue-name

Specifies the name of the queue that contains the job you want to stop.

DESCRIPTION

The STOP/QUEUE/ABORT command is used to stop the job that is currently executing in a printer or terminal queue. The current job is deleted from the queue. (Since printer and terminal queues can have only one current job at a time, you do not specify a job entry number with /ABORT.) A user can issue this command to stop a job when that job is currently executing. Operators and system managers can use the command to abort the current job in a queue.

In general, the STOP/QUEUE/ABORT command is used to stop a print job that is no longer needed. Use the STOP/QUEUE/REQUEUE command to stop the current job and requeue it.

When you abort a print job with STOP/QUEUE/ABORT, the system attempts an orderly halt of the job. Assuming that the printing device is not malfunctioning, the print job completes the page that is currently printing. Then the job is removed from the queue. If the printer 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.

Note: If you accidentally issue the STOP/QUEUE/ABORT command for a queue that has problems, issue the STOP/QUEUE/RESET command to have the queue stopped in an orderly fashion.

EXAMPLE

```
$ STOP/QUEUE/ABORT LPA0
```

This example aborts the current print job on the queue LPA0. The next pending job in the queue will begin to execute. 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 output trailer pages, a trailer page is printed before the job is suspended.

STOP/QUEUE/ENTRY

Causes an executing job on the specified batch queue to be stopped. The queue does not stop; more jobs are processed if they are available. The /QUEUE qualifier is optional, but you must specify the /ENTRY qualifier.

FORMAT

STOP/QUEUE/ENTRY=*entry-number queue-name[:]*

restrictions

Requires operator (OPER) privilege, execute (E) access to the queue, or delete (D) access to the current job.

PARAMETER

queue-name

Specifies the name of the batch queue that contains the job you want to stop.

DESCRIPTION

The STOP/QUEUE/ENTRY command is used to stop a currently executing batch job. A user can issue the commands to stop a job when it is currently executing. Operators and system managers can use the commands to abort current batch jobs.

You can abort only a single current job with the STOP/QUEUE/ENTRY command, even if several jobs are currently executing in the batch queue. When you abort the batch job, the system attempts to stop the job in an orderly fashion, closing any open files and sending a message to the log file.

Use the STOP/QUEUE/REQUEUE command with the /ENTRY qualifier to stop the current batch job and requeue it.

Note: If you accidentally issue the STOP/QUEUE/ENTRY command for a queue that has problems, issue the STOP/QUEUE/RESET command to have the queue stopped in an orderly fashion.

EXAMPLE

```
$ STOP/QUEUE/ENTRY=365 SYS$BATCH
```

This example stops batch job number 365 currently executing on the SYS$BATCH queue. The next pending job in the queue will begin to execute.

STOP/QUEUE/MANAGER

Performs an orderly shutdown of the system job queue manager on the node from which the command is issued. The /QUEUE qualifier is optional, but you must specify the /MANAGER qualifier.

FORMAT

STOP/QUEUE/MANAGER

restrictions

Requires both operator (OPER) and system name (SYSNAM) privileges.

PARAMETERS

None.

DESCRIPTION

The STOP/QUEUE/MANAGER command carries out the following operations:

- Performs the STOP/QUEUE/NEXT operation for all output execution queues on the node from which the command is issued.
- Performs the STOP/QUEUE/ABORT operation for all current non-restartable jobs in all execution queues on the node from which the command is issued. Performs the STOP/QUEUE/REQUEUE operation for all current restartable jobs in all execution queues on that node.
- When all activity in all queues ceases, the STOP/QUEUE/MANAGER command closes the job queue manager file.

The STOP/QUEUE/MANAGER command is part of the SYS$SYSTEM:SHUTDOWN.COM procedure.

EXAMPLE

```
$ STOP/QUEUE/MANAGER
```

Performs a shutdown of all queues on the node from which the command is issued.

STOP/QUEUE/NEXT

Causes the specified queue to stop after all executing jobs have completed processing. The /QUEUE qualifier is optional, but you must specify the /NEXT qualifier.

FORMAT

STOP/QUEUE/NEXT *queue-name[:]*

restrictions

Requires operator (OPER) privilege or execute (E) access to the specified queue.

PARAMETER

queue-name[:]

Specifies the name of the queue that you want to stop.

DESCRIPTION

The STOP/QUEUE/NEXT command stops the queue completely after it allows any current jobs to complete execution. No new jobs can be initiated. Use the START/QUEUE command to restart the queue.

You should use the STOP/QUEUE/NEXT command before deassigning, deleting, merging, or requeuing a queue. Following this procedure allows all currently executing jobs to complete processing before changes are made to the queue.

If the printing device has failed, issue the STOP/QUEUE/RESET command to have the queue stopped in an orderly fashion.

EXAMPLES

1
```
$ STOP/QUEUE/NEXT BATCH1
```

This command prepares to stop the queue BATCH1. All currently executing jobs are allowed to complete, but no new jobs are allowed to initiate. Once the last current job has finished, the queue is stopped.

2
```
$ STOP/QUEUE/NEXT LPA0
$ SHOW QUEUE/ALL LPA0
Printer queue LPA0
$ DELETE/QUEUE LPA0
```

In this example, you want to delete the printer queue LPA0. First you issue the STOP/QUEUE/NEXT command to have the printer finish the current job. Then you issue the SHOW QUEUE/ALL command to be sure that there are no more jobs waiting in the queue. The system response indicates that there are no jobs in the queue. Now you are ready to issue the DELETE/QUEUE command.

STOP/QUEUE/REQUEUE

Causes an executing job on the specified queue to be stopped and requeued. The queue does not stop; more jobs are processed if they are available. The /QUEUE qualifier is optional, but you must specify the /REQUEUE qualifier. If you are requeueing a job on a batch queue, you must specify the /ENTRY qualifier.

FORMAT

STOP/QUEUE/REQUEUE*[=queue-name]*
queue-name[:]
STOP/QUEUE/ENTRY*=entry-number/REQUEUE*
[=queue-name] queue-name[:]

restrictions

Requires operator (OPER) privilege, execute (E) access to the queue, or delete (D) access to the current job.

PARAMETER

queue-name

Specifies the name of the queue that contains the job you want to stop. When you also specify a queue name as a parameter for the /REQUEUE qualifier, the job will be requeued to that queue.

DESCRIPTION

The STOP/QUEUE/REQUEUE command stops a currently executing job and requeues it for later processing. A user can issue the command to requeue a job when it is currently executing. Operators and system managers can use the command to abort or requeue current jobs.

If you include the queue name specifier with the /REQUEUE qualifier, STOP /QUEUE/REQUEUE transfers the current job to another queue. Otherwise, the job is requeued in the same queue.

You must use the /ENTRY qualifier with STOP/QUEUE/REQUEUE when issuing the command for a batch queue.

The STOP/QUEUE/REQUEUE command causes the system to requeue the job for later execution in the queue. A print job that has been checkpointed will resume printing at the checkpoint where it left off, unless you issue the SET/QUEUE/ENTRY/NOCHECKPOINT command before the job is reinitiated. Batch jobs generally restart at the beginning. You can use SET RESTART_VALUE commands in a batch job to avoid rerunning portions of the job that have successfully completed.

You can use the STOP/QUEUE/ABORT command to stop a current print job without requeueing it. The STOP/QUEUE/ENTRY command stops a current batch job without requeueing it.

Note: If you accidentally issue the STOP/QUEUE/REQUEUE command for a queue that has problems, issue the STOP/QUEUE/RESET command to have the queue stopped in an orderly fashion.

QUALIFIERS

/ENTRY=entry-number

Used with batch queues to stop a currently executing batch job. The entry-number parameter refers to the job entry number that was assigned to the job when it was submitted to the queue. The job entry number that you specify must match the job entry number of an executing job in order for the STOP/QUEUE/REQUEUE/ENTRY command to take effect.

You can only specify one entry number for each STOP/QUEUE/REQUEUE /ENTRY command.

/HOLD

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

/PRIORITY=n

Requires OPER (operator) or ALTPRI (alter priority) privilege to raise the priority value above the value of the SYSGEN parameter MAXQUEPRI.

You can use the /PRIORITY qualifier to change the priority of the aborted job. The n parameter can be any integer from 0 to 255. The default value for /PRIORITY is the same as the priority value that the job had when it was stopped.

Generally, the /PRIORITY qualifier is used to lower the priority of a job so that it will run at a time when the queue has no other jobs. No privilege is needed to set the priority lower than the MAXQUEPRI value.

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 job has been checkpointed, it will resume printing on LPB0 at the point where it stopped on LPA0.

2
```
$ STOP/QUEUE/REQUEUE/HOLD LPA0
   .
   .
   .
$ SET QUEUE/ENTRY=254/RELEASE
```

In this example, the job currently being printed on LPA0 is suspended and placed in the hold state. Later, when the SET QUEUE/ENTRY command is issued with the /RELEASE qualifier, the job state changes from holding to pending and waits until the job can begin processing. If the print job has been checkpointed, printing will resume where the job stopped.

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 successfully completed will not be rerun.

STOP/QUEUE/RESET

Abruptly stops the queue and returns control to the system. Any jobs that are currently executing are stopped immediately. The /QUEUE qualifier is optional, but you must specify the /RESET qualifier.

FORMAT

STOP/QUEUE/RESET *queue-name[:]*

restrictions

Requires operator (OPER) privilege or execute (E) access to the specified queue.

PARAMETER

queue-name[:]

Specifies the name of the queue that 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.

Use the START/QUEUE command to restart the queue. Current jobs that are restartable will be requeued for processing. Current jobs that are not restartable are aborted and must be resubmitted for processing. (Print jobs are restartable by default. Use the SUBMIT/RESTART command to make a batch job restartable.)

EXAMPLE

```
$ STOP/QUEUE/RESET LPA0
```

This command abruptly stops the printer queue LPA0. The current print job halts immediately.

```
$ STOP/QUEUE/RESET TEXBATCH
```

This command stops the TEXBATCH queue. Any current job that was submitted with the /RESTART qualifier will be requeued for processing when the queue is restarted. Current jobs that did not specify /RESTART must be resubmitted to the queue.

SUBMIT

Enters one or more command procedures in a batch job queue.

FORMAT

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

restrictions

- If you include a node name with the file specification parameter, you must use the /REMOTE qualifier.
- Requires operator (OPER) privilege, execute (E) access to the queue, or write (W) access to the queue.

PARAMETER

file-spec[,...]

Specifies the name of a file containing a command procedure to be submitted for batch job execution. You must specify a file name. If you do not specify a file type, the SUBMIT command uses the default file type COM. If you specify two or more files, separate the file specifications either with commas or plus signs. In either case, the files are concatenated and processed as a single input stream.

If the file specification contains a node name, the /REMOTE qualifier must be specified.

Full wildcard characters are allowed in the file specification.

DESCRIPTION

The SUBMIT command enters command procedures in the batch job queue. Each command procedure is contained in a file specified with the SUBMIT command. If you specify two or more files in the command line, the system considers the entries as a single job.

The system assigns a unique job entry number to each batch job in the queue. When you submit a batch job, by default the system displays both the job entry number it has assigned to the job, the name of the batch job queue in which your job has been entered, and the current job status, for example, executing, pending, or holding.

Once a batch job has been queued, the version of the file submitted is processed, even if a newer version of the file is created before the batch job runs.

When the system executes a command procedure submitted to a batch job queue, it creates a detached process to execute the commands. This process receives your disk and directory defaults and the same resource quotas and privileges that were given to your interactive process when you logged in.

Batch Job Output

When you submit a command procedure for processing by the SUBMIT command, all output from the command procedure is written to a log file. By default, the log file specification consists of the file name of the first command procedure file in the job and the file type LOG. VAX RMS uses this file specification when creating the output log file, thus implying that logical name translation will be performed. (Use the /NAME or /LOG_FILE

qualifier to assign a different name to the log file.) The log file is written to the directory defined by the logical name SYS$LOGIN in the UAF, unless you specify otherwise with the /LOG_FILE qualifier.

After the batch job finishes, the system queues the log file to SYS$PRINT and deletes the file after it has printed (unless you have specified /KEEP). If you do not want the log file printed, use the /NOPRINT qualifier. When you stop a batch job using the DELETE/ENTRY, STOP/IDENTIFICATION, or /STOP/QUEUE/ENTRY command, the log file is not queued for printing.

If multiple procedures are submitted, the job terminates as soon as any procedure exits with an error or severe (fatal) error status.

QUALIFIERS

/AFTER=time

Requests that the job be held until after a specific time. If the specified time has already passed, the job is queued for immediate processing.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values.

/BACKUP
/NOBACKUP

Selects files according to the dates of their most recent backup. This qualifier is only relevant when used with the /BEFORE or /SINCE qualifier. Use of the /BACKUP qualifier is incompatible with /CREATED, /EXPIRED, and /MODIFIED. The default is /CREATED.

/BEFORE[=time]
/NOBEFORE

Selects only those files that are dated before the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/BY_OWNER[=uic]
/NOBY_OWNER

Selects one or more files only if their owner user identification code (UIC) matches the specified owner UIC.

Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual*.

If the /BY_OWNER qualifier is specified without a UIC, the UIC of the current process is assumed.

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

Specifies one or more characteristics desired for the job. If you specify only one characteristic, you can omit the parentheses. Codes for characteristics can be either names or values from 0 to 127 and are installation-defined. Use the SHOW QUEUE/CHARACTERISTICS command to see which characteristics have been defined for your system. Use the SHOW QUEUE command with

the /FULL qualifier to see which characteristics are available on a particular queue.

When you include the /CHARACTERISTICS qualifier with a SUBMIT command, all the characteristics you specify must also be specified for the queue that will be executing the job. If not, the job will remain pending until the queue characteristics are changed, or until you delete the entry with the DELETE/ENTRY command. You need not specify every characteristic of a queue with a SUBMIT command as long as the ones you specify are a subset of the characteristics set for that queue. The job will also run if no characteristics are specified.

/CLI=filename

Enables you to specify a different command language interpreter (CLI) to use in processing the job. The file specification assumes the device name SYS$SYSTEM: and the file type EXE (SYS$SYSTEM:filename.EXE). If you do not specify the /CLI qualifier, the job is run by the CLI specified in the user's authorization record.

/CONFIRM
/NOCONFIRM (default)

Controls whether a request is issued before each individual SUBMIT operation to confirm that the operation should be performed on that file.

When the system issues the prompt, you can issue any of the following responses:

YES	NO	QUIT
TRUE	FALSE	CTRL/Z
1	0	ALL
	<RET>	

You can use any combination of upper- and lowercase letters for word responses. Word responses can be abbreviated to one or more letters (for example, T, TR, or TRU for TRUE). Affirmative answers are YES, TRUE, and 1. Negative answers are NO, FALSE, 0, and <RET> . QUIT or CTRL/Z indicates that you want to stop processing the command at that point. When you respond with ALL, the command continues to be processed, but no further prompts are given. If you type a response other than one of those in the list, the prompt will be reissued.

/CPUTIME=option

Defines a CPU time limit for the batch job. You can specify a delta time (see Section 2.5 of the *VAX/VMS DCL Concepts Manual*), the value 0, or the keyword NONE or INFINITE for the option.

When you need less CPU time than authorized, use the /CPUTIME qualifier to override the base queue value established by the system manager or the value authorized for you in the user authorization file (UAF). Specify 0 or INFINITE to request an infinite amount of time. Specify NONE when you want the CPU time to default to your user authorization file value or to the limit specified on the queue. Note that you cannot request more time than permitted by the base queue limits or your own user authorization file entry.

/CREATED (default)
/NOCREATED

Selects files based on their dates of creation. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /CREATED qualifier is incompatible with /BACKUP, /EXPIRED, and /MODIFIED.

/DELETE
/NODELETE (default)

Positional qualifier.

Controls whether files are deleted after processing. If you specify the /DELETE qualifier after the SUBMIT command word, all files in the job are deleted. If you specify the /DELETE qualifier following a file specification, only the associated file is deleted after it is processed.

The protection code on the input files must allow D (delete) access to the user identification code (UIC) of the user submitting the job in order for /DELETE to work.

/EXCLUDE=(file-spec[,...])
/NOEXCLUDE

Any files that match the listed file specifications are excluded from the SUBMIT operation. If you specify only one file, you can omit the parentheses.

Wildcard characters are supported for file specifications. However, you cannot use relative version numbers to exclude a specific version. The file specification can contain a directory specification, but you cannot include the device in the file specifications you supply with the /EXCLUDE qualifier.

/EXPIRED
/NOEXPIRED

Selects files according to the dates on which they will expire. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /EXPIRED qualifier is incompatible with /BACKUP, /CREATED, and /MODIFIED. The default is /CREATED.

/HOLD
/NOHOLD (default)

Controls whether or not the job is to be made available for immediate processing.

If you specify /HOLD, the job is not released for processing until you specifically release it with the /NOHOLD or /RELEASE qualifier of the SET QUEUE/ENTRY command.

/IDENTIFY (default)
/NOIDENTIFY

Controls whether the system displays the job entry number assigned to the job and the name of the queue in which the job was entered.

/KEEP
/NOKEEP

Controls whether the log file is deleted after it is printed. /NOKEEP is the default unless the /NOPRINTER qualifier is used.

/LOG_FILE[=file-spec]
/NOLOG_FILE

Controls whether a log file with the specified name is applied to the job or whether a log file is created. No wildcards are allowed in the file specification.

When you use the /LOG_FILE qualifier, the system writes the log file to the file you specify. If you use /NOLOG_FILE, no log file is created. If you use neither form of the qualifier, the log file is written to a file in your default directory that has the same file name as the first command file and a file type of LOG. By default, a log file is kept and is named after the first (or only) file in the job.

You can use the /LOG_FILE qualifier to specify that the log file be written to a different device. Logical names that occur 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 and the file type LOG. When you omit the /LOG_FILE qualifier, the job-name value used with /NAME must be a valid file name.

/MODIFIED
/NOMODIFIED

Selects files according to the dates on which they were last modified. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /MODIFIED qualifier is incompatible with /BACKUP, /CREATED, and /EXPIRED. The default is /CREATED.

/NAME=job-name

Specifies a string to be used as the job name and possibly as the file name for the batch job log file. The job name value can have from 1 to 39 characters. If the /LOG_FILE qualifier is omitted, the job-name value must be a valid file name. If characters other than alphanumerics, underscores, or dollar signs are used in the name, enclose the name in quotation marks.

If you do not specify /NAME, the name string defaults to the file name of the first, or only, file in the job. The job name is displayed by the SHOW QUEUE command.

/NOTIFY
/NONOTIFY (default)

Controls whether a message is broadcast to any terminal at which you are logged in, notifying you that your job has been completed or aborted.

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

Specifies from 1 to 8 optional parameters to be passed to the job. The parameters define values to be equated to the symbols named P1 through P8 in each command procedure in the job. The symbols are local to the specified command procedures.

If you specify only one parameter, you can omit the parentheses.

The commas delimit individual parameters. To specify a parameter that contains any special characters or delimiters, enclose the parameter in quotation marks. Each parameter can have up to 255 characters. However, the total length of all eight parameter strings of the /PARAMETER qualifier cannot exceed 480 characters.

/PRINTER[=queue-name]
/NOPRINTER

Controls whether the job log file is queued for printing when your job is completed. The /PRINTER qualifier allows you to specify a particular print queue. The default is /PRINTER. The default print queue for the log file is SYS$PRINT.

If you specify /NOPRINTER, /KEEP is assumed.

/PRIORITY=n

Requires OPER (operator) or ALTPRI (alter privilege) to raise the priority value of the SYSGEN parameter MAXQUEPRI value.

Specifies the job scheduling priority for the specified job. The priority value can be in the range of 0 through 255, where 0 is the lowest priority and 255 is the highest.

The default value for /PRIORITY is the value of the SYSGEN parameter DEFQUEPRI. No privilege is needed to set the priority lower than the MAXQUEPRI value.

The /PRIORITY qualifier has no effect on the process priority. The queue establishes the process priority.

/QUEUE=queue-name[:]

Specifies the name of the batch job queue in which the job is to be entered. If you do not specify /QUEUE, the job is placed in the default system batch job queue, SYS$BATCH.

/REMOTE

Indicates that the specified command procedure is to be executed at a remote node. The file specification must contain the name of the node on which the job resides. That node name must be the one on which the procedure is to be executed.

Note that, unlike the local case, multiple command procedures queued by a single SUBMIT/REMOTE command are considered separate jobs.

Not all of the SUBMIT qualifiers are compatible with /REMOTE. Only the following qualifiers may be specified with /REMOTE: /BACKUP, /BEFORE, /BY_OWNER, /CONFIRM, /CREATED, /EXCLUDE, /EXPIRED, /MODIFIED, and /SINCE.

/RESTART
/NORESTART (default)

Specifies whether the job will restart after a system failure or after a STOP /QUEUE/REQUEUE command.

/SINCE[=time]
/NOSINCE

Selects only those files that are dated after the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/USER=username

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

Allows you to submit a job on behalf of another user. The job will run exactly as if that user had submitted it. The job runs under that user's username and UIC. Accounting information is logged to that user's account. By default, the user identification comes from the requesting process.

The specifier can be any username that is validated on your system.

/WSDEFAULT=n

Defines a working set default for the batch job. You can specify a positive integer in the range 1 through 65,535, the value 0, or the keyword NONE for n.

Use this qualifier to override the base queue value established by the system manager or the value authorized in your user authorization file (UAF), provided you want to impose a lower value. Specify 0 or NONE if you want the working set value to default to either your UAF value or the working set default specified on the queue. You cannot request a higher value than your default.

/WSEXTENT=n

Defines a working set extent for the batch job. You can specify a positive integer in the range 1 through 65,535, the value 0, or the keyword NONE for n.

Use this qualifier to override the base queue value established by the system manager or the value authorized in your user authorization file (UAF), provided you want to impose a lower value. Specify 0 or NONE if you want the working set extent to default to either your UAF value or the working set extent specified on the queue. You cannot request a higher value than your default.

/WSQUOTA=n

Defines the maximum working set size for the batch job. This is the working set quota. You can specify a positive integer in the range 1 through 65,535, the value 0, or the keyword NONE for n.

Use this qualifier to override the base queue value established by the system manager or the value authorized in your user authorization file (UAF), provided you want to impose a lower value. Specify 0 or NONE if you want the working set quota to default to either your UAF value or the working set quota specified on the queue. You cannot request a higher value than your default.

EXAMPLES

1

```
$ SUBMIT AVERAGE
  Job AVERAGE (queue SYS$BATCH, entry 112) pending
```

The SUBMIT command enters the procedure AVERAGE.COM in the batch job queue. When the batch job completes, the log file AVERAGE.LOG is queued for printing and then deleted.

2

```
$ SUBMIT /PARAMETERS=(TXT,DOC,MEM) BACKUP -
$_AVERAGE, RUNMASTER
  Job AVERAGE (queue SYS$BATCH, entry 416) pending
```

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
```

The SUBMIT command enters the procedure TESTALL.COM as a batch job and specifies that the job be held for later processing. The job will not be released until the SET QUEUE/ENTRY/RELEASE command is issued. The /NAME qualifier requests that the batch job be identified as BATCH24.

SYNCHRONIZE

Places the process issuing this command in a wait state until a specified job completes execution.

FORMAT

SYNCHRONIZE *[job-name]*

restrictions

None.

PARAMETER

job-name

Specifies the name of the job 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.)

Do not use a job name that identifies more than one of your jobs because the synchronization will be unpredictable. To specify a job that does not have a unique name, use the /ENTRY qualifier to specify the job entry number. If you use the /ENTRY qualifier and if you also specify a job name, the job name is ignored.

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 currently in the system, the SYNCHRONIZE command issues 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.

QUALIFIERS

/ENTRY=entry-number

Specifies the system assigned entry number of the job. By default, the system displays the entry number when it successfully queues a job for execution; the entry number of a job is also displayed when you issue the SHOW QUEUE command.

If you specify both the /ENTRY qualifier and the job-name parameter, the job name is ignored.

/QUEUE=queue-name[:]

Specifies the name of the queue on which the job was entered or the queue on which the job is executing. If /QUEUE is not specified, the command assumes that the job is in the default batch job queue, SYS$BATCH.

EXAMPLES

1

```
$ SUBMIT/NAME=PREP FORMAT/PARAMETERS=(SORT,PURGE)
  Job PREP (queue SYS$BATCH, entry 219) started on queue SYS$BATCH
$ SUBMIT PHASER
```

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 line:

```
$ SYNCHRONIZE PREP
```

When this line is processed, the system verifies whether the job named PREP is currently executing in SYS$BATCH. (SYS$BATCH is the default queue for the SYNCHRONIZE command.) 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 issue subsequent commands from your terminal until the SYNCHRONIZE command completes and your process is released from the wait state.

TYPE

Displays the contents of a file or group of files on the current output device.

FORMAT

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

restrictions

None.

PARAMETER

file-spec[,...]

Specifies one or more files to be displayed. If you specify a file name and do not specify a file type, the TYPE command uses the default file type LIS.

If you specify two or more files, separate the file specifications with either commas or plus signs. The files are displayed in the order listed.

You can specify wildcard characters in place of the directory name, file name, file type, or file version number field. The TYPE command displays all files that satisfy the file description.

DESCRIPTION

When the TYPE command displays output, you can control the display in the following ways:

- Use 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 TYPE 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 TYPE to display the next file in the list.
- Use CTLR/S to temporarily suspend the output. Use CTRL/Q to resume the output display at the point of interruption.
- Use CTRL/O to suppress the display but not 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.
- Use CTRL/Y to interrupt the command execution. You can issue the CONTINUE command after pressing CTRL/Y, provided you have not issued an intervening command that calls up a new image. The CONTINUE command will cause TYPE to resume displaying the files where the interruption took place. If you press CTRL/Y to stop command execution entirely, you can use the EXIT command or any other DCL command (other than CONTINUE).

In addition, the /PAGE qualifier may be used to display text one screen at a time.

TYPE opens the specified file with shared read and write access. Therefore, any file that has its attributes set to shared write will be typed, even if it is currently opened by another user.

QUALIFIERS

/BACKUP

Selects files according to the dates of their most recent backup. This qualifier is only relevant when used with the /BEFORE or /SINCE qualifier. Use of the /BACKUP qualifier is incompatible with /CREATED, /EXPIRED, and /MODIFIED. The default is /CREATED.

/BEFORE[=time]

Selects only those files that are dated before the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

/BY_OWNER[=uic]

Selects one or more files only if their owner user identification code (UIC) matches the specified owner UIC.

Specify the UIC using standard UIC format as described in Section 7.1.1 of the *VAX/VMS DCL Concepts Manual.*

If the /BY_OWNER qualifier is specified without a UIC, the UIC of the current process is assumed.

/CONFIRM
/NOCONFIRM (default)

Controls whether a request is issued before each individual TYPE operation to confirm that the operation should be performed on that file.

When the system issues the prompt, you can issue any of the following responses:

YES	NO	QUIT
TRUE	FALSE	CTRL/Z
1	0	ALL
	<RET>	

You can use any combination of upper- and lowercase letters for word responses. Word responses can be abbreviated to one or more letters (for example, T, TR, or TRU for TRUE). Affirmative answers are YES, TRUE, and 1. Negative answers are NO, FALSE, 0, and <RET> . QUIT or CTRL/Z indicates that you want to stop processing the command at that point. When you respond with 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, the prompt will be reissued.

/CREATED (default)

Selects files based on their dates of creation. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /CREATED qualifier is incompatible with /BACKUP, /EXPIRED, and /MODIFIED.

/EXCLUDE=(file-spec[,...])

Any files that match the listed file specifications are excluded from the TYPE operation. If you specify only one file, you can omit the parentheses.

Wildcard characters are supported for file specifications. However, you cannot use relative version numbers to exclude a specific version. The file specification can contain a directory specification, but you cannot include the device in the file specifications you supply with the /EXCLUDE qualifier.

/EXPIRED

Selects files according to the dates on which they will expire. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /EXPIRED qualifier is incompatible with /BACKUP, /CREATED, and /MODIFIED. The default is /CREATED.

/MODIFIED

Selects files according to the dates on which they were last modified. This qualifier is relevant only when used with the /BEFORE or /SINCE qualifier. Use of the /MODIFIED qualifier is incompatible with /BACKUP, /CREATED, and /EXPIRED. The default is /CREATED.

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. If you do not enter the qualifier, or if you enter /OUTPUT 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 /OUTPUT with a partial file specification (for example, /OUTPUT=[JONES]), TYPE is the default file name and LIS the default file type. If you enter a file specification, it may not include any wildcard characters.

If you enter /NOOUTPUT, output is suppressed.

The /OUTPUT qualifier is incompatible with the /PAGE qualifier.

/PAGE
/NOPAGE (default)

Requests that output from the TYPE command be displayed one screen at a time. If more than one file has been requested, CTRL/Z may be used to cancel the display of the current file and to continue with the next file.

The /PAGE qualifier is incompatible with the /OUTPUT qualifier.

/SINCE[=time]

Selects only those files that are dated after the specified time.

You can specify either an absolute time or a combination of absolute and delta times. See Section 2.5 of the *VAX/VMS DCL Concepts Manual* for complete information on specifying time values. You can also use the keywords TODAY, TOMORROW, and YESTERDAY. If no time is specified, TODAY is assumed.

EXAMPLES

1

```
$ TYPE  COMMON.DAT
```

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.
    .
    .
    .
^O
This is the first line in the file BB.DAT.
    .
    .
    .
CTRL/Y
Interrupt
$ STOP
```

The TYPE command contains a 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. CTRL/Y interrupts the command,; the STOP command terminates the TYPE command.

3

```
$ TYPE LETTER*.MEM
April 1, 1985
    .
    .
    .
CTRL/Y
Interrupt
$ SHOW TIME
 15-APR-1985 15:48:07
$ CONTINUE
Sincerely yours,
    .
    .
    .
```

The TYPE command is issued to display all files whose names begin with the word LETTER and which have the file type MEM. While the files are being displayed, you press CTRL/Y to interrupt the processing to find out the time. After issuing the SHOW TIME command, you enter the CONTINUE command to resume the TYPE command operation.

4

```
$ TYPE MEXICO::NOTICE.TEXT/OUTPUT=TEMP.TEXT
```

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.

UNLOCK

Makes accessible a file that became inaccessible as a result of being improperly closed.

FORMAT

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

restrictions

None.

PARAMETER

file-spec[,...]

Specifies one or more files to be unlocked. If you include two or more file specifications, separate them with either commas or plus signs.

Wildcard characters are allowed in the file specifications.

QUALIFIERS

/CONFIRM
/NOCONFIRM (default)

Controls whether the UNLOCK command displays the file specification of each file before unlocking it, so that you can confirm whether the file should be unlocked. If you specify the /CONFIRM qualifier, you must respond to its prompt with a Y (YES) or a T (TRUE) and then press RETURN so that the UNLOCK command can unlock the file. If you enter anything else, such as N or NO, the file is not unlocked.

/LOG
/NOLOG (default)

Controls whether the UNLOCK command displays the file specification of each file that it has unlocked.

EXAMPLE

```
$ TYPE TST.OUT
%TYPE-E-OPENIN, error opening  DISK1:[STEVE]TST.OUT;3 as input
-SYSTEM-W-FILELOCKED, file is deaccess locked
$ UNLOCK TST.OUT
$ TYPE TST.OUT
```

The request to type the output file TST.OUT returns an error message indicating that the file is locked. The UNLOCK command unlocks it. Then the TYPE command is reissued to display the contents of the file.

```
$ UNLOCK NODE3::DISKO:[LISTS]MAILLIST3.LIS
$ COPY NODE3::DISKO:[LISTS]MAILLIST3.LIS *.*
```

You need a copy of the file MAILLIST3.list which is locked on remote NODE3. Issue the UNLOCK command first and then copy the file to your current node, disk, and directory.

WAIT

Places the current process in a wait state until a specified period of time has elapsed. 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*

restrictions

None.

PARAMETER

delta-time

Specifies the time interval to wait. The time must be specified according to the rules for specifying delta time values as described in Section 2.5 of the *VAX/VMS DCL Concepts Manual*. Note, however, that the delta time can contain only the hours, minutes, seconds, and hundredths of seconds fields; the days part must be omitted. Also, the delta time must begin with the number of hours and not a colon, even if the number of hours is zero.

Note that if you issue the WAIT command interactively, the WAIT command does not prompt you 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 ALPHA
$ WAIT 00:10
$ GOTO LOOP
```

The command procedure executes the program image ALPHA. 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 ALPHA again. The procedure loops until it is interrupted or terminated.

If the procedure is executed interactively, it can be terminated by pressing CTRL/C or CTRL/Y and issuing the STOP command or another DCL command that runs a new image in the process. If the procedure is executed in a batch job, it can be terminated with the DELETE/ENTRY command.

WRITE

Writes the specified data to the output file indicated by the logical name. The data is written as one record in the output file.

FORMAT

WRITE *logical-name expression[,...]*

restrictions

You can place qualifiers after the WRITE command or after the logical name; you cannot place qualifiers after the expression.

PARAMETERS

logical-name

Specifies the logical name assigned to the file to which a record is to be written. 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.

expression[,...]

Specifies data to be written as a single record to the output file. You can specify data items using character string expressions. See Section 5.7.1 of the *VAX/VMS DCL Concepts Manual* for more information on string expressions.

You can specify a list of expressions separated by commas; the command interpreter concatenates the items into a single record and writes the record to the output file.

The maximum size of any record that can be written is less than 1024 bytes, unless you specify the /SYMBOL qualifier, in which case the maximum is 2048 bytes. (See the "Description" section below.)

DESCRIPTION

The WRITE command can write records to sequential, relative, or indexed sequential access (ISAM) 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.

In order to write to a file, the file must be opened 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 explicitly opened 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. Thus, use the /SYMBOL qualifier where the record will contain approximately 1000 bytes or more.

QUALIFIERS

/ERROR=label

Specifies a label on a line in the command procedure to receive control if the write request results in an error. 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 routine specified for this qualifier takes precedence over any action statement indicated in an ON command.

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

/SYMBOL

Causes the expression to be interpreted and its expanded value placed in a 2048-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 if you need to write a very large record (see the preceding "Description" section).

Each expression specified must be a symbol. You cannot specify character string expressions (that is, strings in quotation marks) with the /SYMBOL qualifier.

If you do not use the /SYMBOL qualifier, the entire command, including the expression or expressions, is placed in a 1024-byte buffer, as explained in the "Description" section.

/UPDATE

Specifies that an existing record be modified (updated) using the specified record. You must be able to read and write to a file in order to use the /UPDATE qualifier. You can 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.

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/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
```

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 on the WRITE command; if an error occurs when a record is being written, control is transferred to the label WRITE_ERROR.

3

```
$ OPEN/APPEND OUTPUT_FILE TRNTO::DBA1:[PGM]PLAN.DAT
$ WRITE OUTPUT_FILE "BEGINNING PHASE 3"
```

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.

4

```
$ OPEN/APPEND MYFILE [JONES]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 [JONES]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.

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

VAX/VMS
DCL Dictionary
AA–Z200C–TE

READER'S COMMENTS

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

_____

_____

_____

_____

_____

_____

_____

_____

_____

Did you find errors in this manual? If so, specify the error and the page number.

_____

_____

_____

_____

_____

_____

_____

_____

_____

Please indicate the type of user/reader that you most nearly represent:

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code or Country _____

Do Not Tear - Fold Here and Tape

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SSG PUBLICATIONS ZK1-3/J35
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, NEW HAMPSHIRE 03062-2698

Do Not Tear - Fold Here

Cut Along Dotted Line

VAX/VMS
DCL Dictionary
AA–Z200C–TE

READER'S COMMENTS

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

_____

_____

_____

_____

_____

_____

_____

_____

_____

Did you find errors in this manual? If so, specify the error and the page number.

_____

_____

_____

_____

_____

_____

_____

_____

_____

Please indicate the type of user/reader that you most nearly represent:

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____
or Country

— — Do Not Tear - Fold Here and Tape —

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SSG PUBLICATIONS ZK1-3/J35
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, NEW HAMPSHIRE 03062-2698

— — Do Not Tear - Fold Here —

Cut Along Dotted Line