

VSI DECset for OpenVMS Code Management System Reference Manual

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

Software Version: DECset Version 12.7

VSI DECset for OpenVMS Code Management System Reference Manual



VMS Software

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

Legal Notice

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

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

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

Table of Contents

Preface	v
1. About VSI	v
2. Intended Audience	v
3. Document Structure	v
4. Related Documents	v
5. References to Other Products	v
6. OpenVMS Documentation	vi
7. VSI Encourages Your Comments	vi
8. Conventions	vi
Chapter 1. CMS Command Dictionary	1
ACCEPT GENERATION	1
ANNOTATE	1
CANCEL REVIEW	4
CONVERT LIBRARY	4
COPY CLASS	5
COPY ELEMENT	6
COPY GROUP	8
CREATE CLASS	9
CREATE ELEMENT	9
CREATE GROUP	10
CREATE LIBRARY	11
DELETE CLASS	12
DELETE ELEMENT	12
DELETE GENERATION	13
DELETE GROUP	15
DELETE HISTORY	16
DIFFERENCES	17
DIFFERENCES (2)	20
FETCH	21
HELP	21
INSERT ELEMENT	22
INSERT GENERATION	22
INSERT GROUP	23
MARK GENERATION	24
MODIFY CLASS	25
MODIFY ELEMENT	26
MODIFY GENERATION	27
MODIFY GROUP	28
MODIFY LIBRARY	29
MODIFY RESERVATION	30
REJECT GENERATION	30
REMARK	31
REMOVE ELEMENT	32
REMOVE GENERATION	32
REMOVE GROUP	33
REPLACE	34
RESERVE	35
RETRIEVE ARCHIVE	37
REVIEW GENERATION	38

SET ACL	38
SET LIBRARY	40
SET NOLIBRARY	41
SHOW ACL	41
SHOW ARCHIVE	41
SHOW CLASS	42
SHOW ELEMENT	43
SHOW GENERATION	43
SHOW GROUP	45
SHOW HISTORY	45
SHOW LIBRARY	47
SHOW RESERVATIONS	48
SHOW REVIEWS_PENDING	48
SHOW VERSION	49
UNRESERVE	49
VERIFY	50
Chapter 2. Parameter and Qualifier Glossary	53
2.1. CMS Command Parameters	53
2.2. CMS Command Qualifiers	60
Chapter 3. CMS Messages	91
3.1. Message Display	91
3.1.1. Severity Levels	91
3.1.2. Library Directory Specifications	91
3.1.3. Secondary Messages	92
3.2. CMS Messages	92
3.3. CMS Internal Diagnostic Messages	164

Preface

This reference manual describes all the commands available with the *Code Management System for OpenVMS (CMS)*. The commands are listed in alphabetical order.

1. About VSI

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

2. Intended Audience

This guide is intended for experienced programmers and technical users who enter CMS commands from the DCL command-line interface.

3. Document Structure

This reference contains the following information:

- A command dictionary that defines each command and provides an optional example
- A command parameter and qualifier glossary
- An error message listing providing suggested user action

4. Related Documents

The following documents might also be helpful when using CMS:

- The *VSI DECset for OpenVMS Installation Guide* contains instructions for installing CMS.
- The *Code Management System for OpenVMS Release Notes* contain added information on the use and maintenance of CMS.
- The *VSI DECset for OpenVMS Guide to the Code Management System* contains introductory and conceptual information about CMS.
- The *CMS Client User's Guide* describes the installation and use of the CMS Client software in a Microsoft Windows environment.
- The *VSI DECset for OpenVMS Code Management System Callable Routines Reference Manual* describes the set of CMS callable routines.
- The *Using VSI DECset for OpenVMS Systems* contains information on using the other components of DECset.

5. References to Other Products

Some older products that DECset components worked with previously may no longer be available or supported by VSI. References in this manual to such products serve as examples only and do not imply that VSI has conducted recent interoperability testing.

See the Software Product Description for a current list of supported products that are warranted to interact with DECset components.

6. OpenVMS Documentation

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

7. VSI Encourages Your Comments

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

8. Conventions

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

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

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

The following conventions are also used in this manual:

Convention	Meaning
Ctrl/ <i>x</i>	A sequence such as Ctrl/ <i>x</i> indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
PF1 <i>x</i>	A sequence such as PF1 <i>x</i> indicates that you must first press and release the key labeled PF1 and then press and release another key or a pointing device button.
Return	In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)
...	A horizontal ellipsis in examples indicates one of the following possibilities: • Additional optional arguments in a statement have been omitted. • The preceding item or items can be repeated one or more times. • Additional parameters, values, or other information can be entered.
.	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.

Convention	Meaning
()	In command format descriptions, parentheses indicate that you must enclose the options in parentheses if you choose more than one.
[]	In command format descriptions, brackets indicate optional choices. You can choose one or more items or no items. Do not type the brackets on the command line. However, you must include the brackets in the syntax for OpenVMS directory specifications and for a substring specification in an assignment statement.
[]	In command format descriptions, vertical bars separate choices within brackets or braces. Within brackets, the choices are options; within braces, at least one choice is required. Do not type the vertical bars on the command line.
{ }	In command format descriptions, braces indicate required choices; you must choose at least one of the items listed. Do not type the braces on the command line.
bold text	This typeface represents the introduction of a new term. It also represents the name of an argument, an attribute, or a reason.
<i>italic text</i>	Italic text indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error <i>number</i>), in command lines (/PRODUCER= <i>name</i>), and in command parameters in text (where <i>dd</i> represents the predefined code for the device type).
UPPERCASE TEXT	Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
Monospace type	Monospace type indicates code examples and interactive screen displays. In the C programming language, monospace type in text identifies the following elements: keywords, the names of independently compiled external functions and files, syntax summaries, and references to variables or identifiers introduced in an example.
-	A hyphen at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line.
numbers	All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radices—binary, octal, or hexadecimal—are explicitly indicated.

Chapter 1. CMS Command Dictionary

This section describes the CMS commands, listed in alphabetic order. Most of the commands include an example.

The parameter and qualifier glossary describes the parameters and qualifiers used with the CMS commands.

To use the command dictionary, look up a particular command for specific information about the command. Optionally, go to the glossary of command parameters or qualifiers for additional information.

ACCEPT GENERATION

ACCEPT GENERATION — Changes the review status of each specified element generation from pending to accepted, then removes it from the review pending list.

Format

ACCEPT GENERATION *element-expression* "remark"

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

Note

Use this command only on element generations with reviews pending (see the **REVIEW GENERATION** command for more information). If you access the generation after it has been accepted, CMS will not inform you of any review status.

Example

```
CMS> ACCEPT GENERATION EXAMPLE.SDML "this example cleared for publication"
%CMS-S-ACCEPTED, generation 3 of element DISKX:[PROJECT.CMSLIB]EXAMPLE.SDML
accepted
```

This example accepts the most recently created generation of the element EXAMPLE.SDML. The generation is removed from the review pending list.

ANNOTATE

ANNOTATE — Creates a line-by-line file listing of the changes made to each specified element generation, and places it in your current directory or a specified directory.

Format

ANNOTATE *element-expression*

Command Qualifiers	Defaults
/[NO]APPEND	/NOAPPEND
/[NO]CONFIRM	/NOCONFIRM
/FORMAT=(data-format,data-partition)	/FORMAT=(ASCII,RECORDS)
/FULL	See Qualifier Glossary
/GENERATION[=generation-expression]	/GENERATION=1+
/[NO]LOG	/LOG
/MERGE=generation-expression	/NOMERGE
/NOMERGE	
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/OUTPUT[=file-specification]	/OUTPUT=element-name.ANN

This command also creates an output file containing an annotated listing. By default, the file name is the same as the element name. The file type is .ANN. The annotated listing file contains two parts:

- A history
- A source file listing

The history includes the generation number, date, time, user, and remark associated with each generation of the element (and other file-related information when you use the /FULL qualifier). Generations are listed in reverse chronological order. The generation numbers of the specified generation and its ancestors are marked with an asterisk (*).

The source file listing contains all the lines inserted or modified from generation 1 to the specified generation. The listing does not show lines deleted from the file. However, when the same part of an element is updated multiple times, the lines which were added and removed can sometimes result in a “lines deleted” message in the listing.

CMS inserts consecutive line numbers in the listing unless editor-assigned line numbers exist. (Line numbers start with 1 for the first line and increase by 1 for each line.) The generation field starts at the first character position of each line. It contains the generation number of the most recent generation in which the line was inserted or modified. The generation field is blank if a line is unchanged since generation 1.

Example

```
CMS> ANNOTATE TIMECVT.BAS/GENERATION=3
%CMS-S-ANNOTATED, element DISKX:[WORK.CMSLIB]TIMECVT.BAS annotated
```

This example produces a file named TIMECVT.ANN, which contains the annotated listing of generation 3 for TIMECVT.BAS. The contents of TIMECVT.ANN are as follows:

```
Annotated listing for element TIMECVT.BAS in CMS Library DISKX:
[WORK.CMSLIB]
    25-APR-2005 15:50:29
```

```

4      15-APR-2005 10:01:55 JAMES "additional error checks"
*3     12-APR-2005 15:49:01 JAMES "add check for invalid delta time"
*2     27-MAR-2005 12:39:58 JAMES "jp - fixed length string required"
*1     25-MAR-2005 15:37:11 JAMES "time conversion program"

```

Annotated listing for element TIMECVT.BAS in CMS Library DISKX:
[WORK.CMSLIB]

```

      25-APR-2005 15:50:29
      1      10      rem Program to compute an absolute time given the
present time
      2      rem and a delta time. The result is written to a
file.
      3      4      20      OPTION TYPE = EXPLICIT
      5      DECLARE STRING DELTA_TIME
2      6      MAP (STRING_LEN) STRING ASC_TIME = 80
      7      DECLARE LONG RETCODE
      8      DIM LONG BINARY_DELTA(1)
      9      DIM LONG NOW(1)
     10      DIM LONG BINARY_CVT_TIME(1)
     11
     12      100      EXTERNAL LONG CONSTANT SS$_NORMAL
3      13      EXTERNAL LONG CONSTANT SS$_IVTIME
     14      EXTERNAL LONG FUNCTION LIB$ADDX
     15      EXTERNAL LONG FUNCTION LIB$SUBX
     16      EXTERNAL LONG FUNCTION LIB$INT_OVER
     17      EXTERNAL INTEGER FUNCTION SYS$BINTIM (STRING BY
DESC, LONG BY REF)
     18      EXTERNAL INTEGER FUNCTION SYS$GETTIM (LONG BY REF)
     19      EXTERNAL INTEGER FUNCTION SYS$ASCTIM (LONG BY
REF,STRING BY DESC, &
     20      LONG BY
REF,LONG BY REF)
     21      150      LET RETCODE = LIB$INT_OVER(0)
     22      PRINT "Input delta time"
     23      INPUT DELTA_TIME
     24      LET RETCODE = SYS$BINTIM ( DELTA_TIME,
BINARY_DELTA(0) )
3      25      175      IF (RETCODE = SS$_NORMAL) THEN GOTO 200
3      26      ELSE IF RETCODE = SS$_IVTIME THEN      &
3      27      PRINT , "INVALID TIME"
3      28      GOTO DONE
3      29      END IF
3      30      END IF
3      31      200      LET retcode = SYS$GETTIM(NOW(0))
3      32      IF (VAL( DELTA_TIME ) > 0 ) THEN      &
     33      retcode=LIB$ADDX(NOW(0),BINARY_DELTA
(0) ,BINARY_CVT_TIME(0))
     34      END IF
     35      LET retcode = SYS
$ASCTIM(,ASC_TIME,BINARY_CVT_TIME(0),)
     36      OPEN "TIME.TMP" FOR OUTPUT AS FILE #1
     37      PRINT #1,ASC_TIME
     38      CLOSE #1
     39      32767      Done:  END
     40

```

The element's history appears at the beginning of the TIMECVT.ANN file, which lists the records of the transactions that created each of the four generations. The third generation was annotated (ANNOTATE

TIMECVT.BAS/GENERATION=3), so changes made after generation 3 are not shown in the annotated listing. Generation 3 and its ancestors are marked with an asterisk in the history.

The source file listing shows each line of the file, including line numbers. The numbers on the far left are the generation numbers in which the line was most recently inserted or modified; the lines with no generation numbers have not changed since generation 1. CMS assigns the next column of numbers. The program itself includes the third column of numbers.

CANCEL REVIEW

CANCEL REVIEW — Changes the review status of each specified element generation from pending to none and removes it from the review pending list.

Format

CANCEL REVIEW *element-expression* *"remark"*

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

Use this command only on element generations with reviews pending (see the REVIEW GENERATION command for more information).

Example

```
$ CMS CANCEL REVIEW EXAMPLE.SDML "review the final version only"
%CMS-S-CANCELED, review of generation 2 of element DISKX:
[PROJECT.CMSLIB]EXAMPLE.SDML canceled
```

This example cancels the review of the element EXAMPLE.SDML, and removes it from the review pending list.

CONVERT LIBRARY

CONVERT LIBRARY — Creates a copy of an existing Version 2. *n* CMS library and converts the copy for use with this version of CMS. You do not need to convert libraries created with CMS Version 3. *n* or higher.

Format

CONVERT LIBRARY *V2-library-directory-specification* *V4-library-directory-specification*

To convert a library, create an empty directory to contain the new converted library. Conversion maintains all library setup in your existing library except for OpenVMS Access Control List (ACL) protection mechanisms assigned to the CMS library directory, or to any CMS library files in that directory. The conversion process neither propagates original OpenVMS directory or file ACLs, nor assigns them new default ACLs. You must assign the default protection on the converted library. For

example, to assign read, write, and delete access to the system, owner, and group categories, use the following command:

```
$ SET FILE/ACL=(DEFAULT_PROTECTION,S:RWD,O:RWD,G:RWD) CMSLIB_V4.DIR
```

The conversion process also maintains the reference copy directory. After converting the library, CMS automatically executes the VERIFY/REPAIR command to ensure that any existing reference copy directory is valid and current.

The CONVERT LIBRARY command will log an unusual occurrence in the history file.

Example

```
$ CREATE/DIRECTORY [PROJECT.CMSLIB_V4]
$ CMS
CMS> CONVERT LIBRARY
_V2 library: [PROJECT.CMSLIB_V2]
_Directory for current format library: [PROJECT.CMSLIB_V4]
%CMS-S-CREATED, CMS Library DISKX:[PROJECT.CMSLIB_V4] created
%CMS-I-LIBINSLIS, Library DISKX:[PROJECT.CMSLIB_V4] inserted into the
  library list
%CMS-I-CONELE, element DATAPROG.BAS converted
%CMS-I-CONGRP, group TESTGRP converted
%CMS-I-CONCLS, class ETMETAL converted
%CMS-I-CONRES, all reservations converted
%CMS-I-CONHIS, history file converted
%CMS-S-CONVERTED, Version 2 library converted to current format
%CMS-I-VERCLS, class list verified%CMS-I-VERCMD, command list verified
%CMS-I-VERELE, element list verified
%CMS-I-VERGRP, group list verified
%CMS-I-VERRES, reservation list verified
%CMS-I-VERFRE, internal free space list verified
%CMS-I-VERFRE, internal free space list verified
%CMS-I-VERFRE, internal free space list verified
%CMS-I-VERFRE, internal free space list verified
%CMS-I-VERFRE, internal free space list verified
%CMS-I-VERFRE, internal free space list verified
%CMS-I-VERFRE, internal free space list verified
%CMS-I-VERARC, archive control block verified
%CMS-I-VER2, internal contiguous space verified
%CMS-I-VERCON, control file verified%CMS-I-VEREDF, element DATAPROG.BAS
  verified
%CMS-I-VEREDFS, element data files verified
%CMS-I-REPAIRED, library [PROJECT.CMSLIB_V4] repaired
```

This example creates a new directory to contain the converted library, and then converts the old library.

CMS automatically issues VERIFY/REPAIR to ensure that both the library and any reference copies are valid. VERIFY/REPAIR repairs any invalid reference copies. If you receive reference copy errors, your library is still converted and available for use.

COPY CLASS

COPY CLASS — Uses an existing library class to copy and create a new class in the same library or in another library. The original class is left unchanged. The class attributes are copied in full, and an

identical set of element generations are inserted into the new class. The elements to be inserted into the new class must already exist in the target library.

Format

COPY CLASS *old-class-expression new-class-name "remark"*

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

CMS must be able to create one new class for each old class. When you use wildcards or a comma list in the input class specification, you must specify a wildcard as the output class specification.

Example

```
CMS> COPY CLASS INIT_CLASS INITX_CLASS "experimental version"
%CMS-S-COPIED, class DISKX:[PROJECT.CMSLIB]INIT_CLASS copied to INITX_CLASS
```

This command creates a new class named INITX_CLASS in the same library as the original class.

COPY ELEMENT

COPY ELEMENT

Format

COPY ELEMENT *old-element-expression new-element-name "remark"*

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

Uses an existing library element to copy and create a new element or elements in the same library or in another library. The original element or elements is left unchanged. The generation history, file characteristics, and element attributes are preserved and copied in full.

If the existing element has the reference copy attribute enabled (that is, if it was created or modified with /REFERENCE_COPY), the reference copy attribute is also enabled for the new element (assuming the reference copy attribute is established for the library).

If the existing element is reserved when you enter COPY ELEMENT, CMS informs you of the condition and proceeds with the transaction. The new element is not reserved, regardless of whether the original element is reserved at the time of the copy transaction.

If a generation of the element is marked pending review, CMS informs you of the condition and asks whether you want to proceed. If you type YES, CMS records the transaction as an unusual occurrence

and proceeds with the command. The new element is not marked as pending review, regardless of whether the original element is marked at the time of the copy. If you type NO or press Return or Ctrl/Z, no further action is taken.

CMS must be able to create one new element for each old element. When you use wildcards, a group name, or a comma list in the input element specification, CMS builds a list of elements to be copied. CMS uses this list as the point of reference during the copy transactions. If the output element specification does not allow CMS to create a new element for each element in the input list, the results might not be what you intend. For example, the following combination of wildcard expressions produces only one new element:

```
input element specification - *.FOR
output element specification - NDATA.*
```

The first element matching the input specification (*.FOR) produces one new element named NDATA.FOR. Each successive element matching the input specification generates an error message because CMS can create only one unique element name from the given combination of wildcard expressions.

Examples

1. CMS> **COPY ELEMENT INIT.FOR INITX.FOR "experimental version"**
%CMS-S-COPIED, element DISKX:[PROJECT.CMSLIB]INIT.FOR copied to
INITX.FOR

This example creates the new element INITX.FOR in the same library as the old element.

2. \$ **CREATE/DIRECTORY [RELEASE.CMSLIB]**
\$ **CMS**
CMS> **CREATE LIBRARY [RELEASE.CMSLIB] "follows development library"**
%CMS-S-CREATED, CMS Library DISKX:[RELEASE.CMSLIB] created
CMS> **COPY ELEMENT *.* /LIBRARY=[PROJECT.CMSLIB] *.* "loading elements"**
%CMS-I-COPIED, element DISKX:[PROJECT.CMSLIB]INIT.FOR copied to
DISKX:[RELEASE.CMSLIB]INIT.FOR
%CMS-I-COPIED, element DISKX:[PROJECT.CMSLIB]INITX.FOR copied to
DISKX:[RELEASE.CMSLIB]INITX.FOR
%CMS-I-COPIED, element DISKX:[PROJECT.CMSLIB]MSGDOC.FOR copied to
DISKX:[RELEASE.CMSLIB]MSGDOC.FOR
%CMS-I-COPIED, element DISKX:[PROJECT.CMSLIB]OUTPUT.FOR copied to
DISKX:[RELEASE.CMSLIB]OUTPUT.FOR
%CMS-I-COPIED, element DISKX:[PROJECT.CMSLIB]SEARCH.FOR copied to
DISKX:[RELEASE.CMSLIB]SEARCH.FOR
%CMS-I-COPIED, element DISKX:[PROJECT.CMSLIB]ARGCHK.FOR copied to
DISKX:[RELEASE.CMSLIB]ARGCHK.FOR
%CMS-S-COPIES, 6 copies completed

CMS> **SHOW HISTORY**
History of CMS Library DISKX:[RELEASE.CMSLIB]
9-MAY-2005 11:23:43 SMITH CREATE LIBRARY DISKX:[RELEASE.CMSLIB]
"follows development library"
9-MAY-2005 11:26:00 SMITH COPY ELEMENT /LIBRARY=DISKX:[PROJECT.CMSLIB]
INIT.FOR INIT.FOR "loading elements"
9-MAY-2005 11:26:04 SMITH COPY ELEMENT /LIBRARY=DISKX:[PROJECT.CMSLIB]
INITX.FOR INITX.FOR "loading elements"
9-MAY-2005 11:26:07 SMITH COPY ELEMENT /LIBRARY=DISKX:[PROJECT.CMSLIB]
MSGDOC.FOR MSGDOC.FOR "loading elements"
9-MAY-2005 11:26:15 SMITH COPY ELEMENT /LIBRARY=DISKX:[PROJECT.CMSLIB]

```

      OUTPUT.FOR OUTPUT.FOR "loading elements"
9-MAY-2005 11:26:17 SMITH COPY ELEMENT/LIBRARY=DISKX:[PROJECT.CMSLIB]
      SEARCH.FOR SEARCH.FOR "loading elements"
9-MAY-2005 11:26:19 SMITH COPY ELEMENT/LIBRARY=DISKX:[PROJECT.CMSLIB]
      ARGCHK.FOR A RGCHK.FOR "loading elements"
CMS> SHOW GENERATION/DESCENDANTS INIT.FOR
Element generations in CMS Library DISKX:[RELEASE.CMSLIB]
INIT.FOR
  2      6-MAR-2005 17:34:04 SMITH "header offset and additional
      free space added"
  1      6-MAR-2005 17:26:10 SMITH "initialization routines"

```

This example creates a new directory for a new library, and copies all the elements from the library [PROJECT.CMSLIB] into the new library [RELEASE.CMSLIB]. The new elements are being created in a separate library, so CMS can create new elements with the same names as the old elements; you can enter a null string for the second parameter (for the new element name). In this case, CMS supplies the value *.*.

The SHOW HISTORY command executed after the copy transaction indicates that the library history contains only records of transactions performed on the new library (CREATE LIBRARY and COPY transactions). The SHOW GENERATION/DESCENDANTS command shows the generation history for one of the elements. The COPY ELEMENT transaction preserves the generation history for each element; thus, the record of replacement transactions (also the CREATE ELEMENT transaction that produced generation 1 of the element) is maintained from the old element to the new.

COPY GROUP

COPY GROUP — Uses an existing library group to copy and create a new group in the same library or in another library. The original group is left unchanged. The group attributes are copied in full, and an identical set of elements and groups are inserted into the new group. The elements to be inserted into the new group must already exist in the target library.

Format

COPY GROUP *old-group-expression new-group-name "remark"*

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

CMS must be able to create one new group for each old group. When you use wildcards or a comma list in the input group specification, you must specify a wildcard as the output group specification.

Example

```

CMS> COPY GROUP INIT_GROUP INITX_GROUP "experimental version"
%CMS-S-COPIED, group DISKX:[PROJECT.CMSLIB]INIT_GROUP copied to INITX_GROUP

```

This command creates a new group named INITX_GROUP in the same library as the original group.

CREATE CLASS

CREATE CLASS — Creates one or more classes. After creating a class, you can place any related set of element generations in that class by using the **INSERT GENERATION** command. The **CREATE CLASS** command does not automatically place any generations in the created class. For more information on classes, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

Format

CREATE CLASS *class-name* [, ...] **"remark"**

Command Qualifiers	Defaults
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

Example

```
CMS> CREATE CLASS INTERNAL_RELEASE "for internal use only"
%CMS-S-CREATED, class DISKX:[PROJECT.CMSLIB]INTERNAL_RELEASE created
```

This example creates a class named **INTERNAL_RELEASE**. After establishing the class name, you can place element generations in the class with the **INSERT GENERATION** command.

CREATE ELEMENT

CREATE ELEMENT — Creates the first generation of a new element or elements by moving the input file into a CMS library.

Format

CREATE ELEMENT *element-expression* **"remark"**

Command Qualifiers	Defaults
/[NO]CONCURRENT	/CONCURRENT
/[NO]CONFIRM	/NOCONFIRM
/HISTORY= "string"	/NOHISTORY
/NOHISTORY	
/INPUT[=file-specification]	See Qualifier Glossary
/[NO]KEEP	/NOKEEP
/[NO]LOG	/LOG
/NOTES= "string"	/NONOTES
/NONOTES	
/POSITION=n	See Qualifier Glossary

Command Qualifiers	Defaults
/[NO]REFERENCE_COPY	See Qualifier Glossary
/[NO]RESERVE	/NORESERVE
/[NO]REVIEW	/NOREVIEW

By default, CMS searches for the file in your current default directory. You can direct CMS to use a file in a different directory by specifying the /INPUT qualifier. After creating the element, CMS deletes all versions of the file used to create the new element. If you specify either the /KEEP or /RESERVE qualifiers, CMS does not delete the file (even if the library is set with /NOKEEP or /NORESERVE).

When you create an element, you can also define the history, concurrent, notes, position, reference copy, and review attributes for the element or establish a reservation.

CMS stores the creation date and time, revision date and time, file attributes, and file revision number of the file used to create generation 1 of the new element. When you fetch or reserve a generation of an element, CMS restores the times, attributes, and file revision number associated with the file used to create the element generation. You can also use the SHOW GENERATION/FULL command to display this information.

To change the creation remark associated with the element or generation 1 of the element, use the MODIFY ELEMENT or MODIFY GENERATION command.

If you specify the /NOTES qualifier, you must specify the /POSITION qualifier on the same command line.

Example

```
CMS> CREATE ELEMENT INIT.FOR "initialization routines"
%CMS-S-CREATED, element DISKX:[PROJECT.CMSLIB]INIT.FOR created
```

This example creates an element named INIT.FOR from a file with the same name in the current, default directory, then deletes all versions of that file in the current, default directory.

CREATE GROUP

CREATE GROUP — Creates one or more empty groups.

Format

```
CREATE GROUP group-name[, ...] "remark"
```

Command Qualifiers	Defaults
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

After creating a group, you can place any related set of elements or groups in that group by using the INSERT ELEMENT or INSERT GROUP command. The CREATE GROUP command does not automatically place any elements or groups in the created group. For more information about groups, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

Example

```
CMS> CREATE GROUP TIME_TST "files for time tests"
%CMS-S-CREATED, group DISKX:[PROJECT.CMSLIB]TIME_TST created
```

Creates a group named TIME_TST. After establishing the group name, you can place elements in the group with the INSERT ELEMENT command.

CREATE LIBRARY

CREATE LIBRARY — Builds CMS control files in a directory so it can be used as a CMS library.

Format

CREATE LIBRARY *directory-specification*[,...] **"remark"**

Command Qualifiers	Defaults
/AFTER[=directory-specification]	See Qualifier Glossary
/BEFORE[=directory-specification]	See Qualifier Glossary
/[NO]CONFIRM	/NOCONFIRM
/[NO]CREATE_DIRECTORY	/NOCREATE_DIRECTORY
/[NO]CONCURRENT	/NOCONCURRENT
/[NO]EXTENDED_FILENAMES	/NOEXTENDED_FILENAMES
/[NO]KEEP	/NOKEEP
/[NO]LOG	/LOG
/[NO]LONG_VARIANT_NAMES	/NOLONG_VARIANT_NAMES
/[NO]REFERENCE_COPY=directory-specification	/NOREFERENCE_COPY
/REVISION_TIME[=option]	See Qualifier Glossary

After creating a library with this command, you can enter CMS commands to manipulate the library. Entering the CREATE LIBRARY command automatically sets your current CMS library to the library (or libraries) specified. You can use CREATE LIBRARY only once on a library.

To create more than one library at a time, enter the CREATE LIBRARY command with more than one directory specification. The directory specifications must be separated by commas. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

When you execute this command, CMS defines logical names beginning with CMS\$. These names are used by subsequent CMS commands. Do not define logical names beginning with CMS\$, because this prefix is reserved for use by CMS.

You cannot specify both the /AFTER and /BEFORE qualifiers on the same command line.

Example

```
CMS> CREATE LIBRARY [RELEASE.CMSLIB] "follows development library"
%CMS-S-CREATED, CMS Library DISKX:[RELEASE.CMSLIB] created
```

This example creates a CMS library in the subdirectory [RELEASE.CMSLIB]. The library does not contain any elements yet. Subsequent CMS commands refer to the library contained in [RELEASE.CMSLIB] until the user logs out or enters a SET LIBRARY or another CREATE LIBRARY command.

DELETE CLASS

DELETE CLASS — Deletes one or more classes from a CMS library.

Format

DELETE CLASS class-expression "remark"

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/REMOVE_CONTENTS	See Qualifier Glossary

The class must exist and must not contain any element generations prior to deletion. If the /REMOVE_CONTENTS qualifier is not specified, and the class contains generations, CMS issues an error message and does not delete the class.

Even though a class is deleted, records of transactions that created and used the class are kept in the library history. You can reuse the deleted class name to create a new class. However, there is no distinction between the two classes in the project history, except their transactions are separated by entries for DELETE CLASS and CREATE CLASS commands.

To determine which generations belong to a class, use the SHOW CLASS command with the /CONTENTS qualifier.

You cannot perform the following operations:

- Delete a class containing any element generations. (Use the /REMOVE_CONTENTS qualifier to remove all generations during deletion; use the REMOVE GENERATIONS command to remove all generations prior to deletion.)
- Delete a class that has read-only access. (Use the MODIFY CLASS/NOREAD_ONLY command to change the access to the class.)

Example

```
CMS> DELETE CLASS PRE_RELEASE "no longer necessary"
%CMS-S-DELETED, class DISKX:[PROJECT.CMSLIB]PRE_RELEASE deleted
```

This example deletes the class named PRE_RELEASE.

DELETE ELEMENT

DELETE ELEMENT — Deletes an element from a CMS library.

Format

DELETE ELEMENT *element-expression* "remark"

Command Qualifiers	Defaults
/[NO]CONFIRM	/CONFIRM
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

If the element is set to /REFERENCE_COPY and there is a current reference copy directory for the CMS library, CMS deletes the corresponding file from the reference copy directory.

There cannot be any existing reservations for the element. The element cannot be a member of a group, nor can one of its generations belong to a class or be under review. If one of the element's generations is under review, use the CANCEL REVIEW command to remove it from the review list. If an element is reserved, you must unreserve or replace it before you can delete the element. If the element belongs to any groups or has generations in any classes, use the REMOVE ELEMENT or REMOVE GENERATION command to remove it.

Even though an element is deleted, records of transactions that created and used the element are kept in the library history. You can reuse the deleted element name to create a new element. However, there is no distinction between the two elements in the library history, except their transactions are separated by entries for DELETE ELEMENT and CREATE ELEMENT commands.

You cannot perform the following operations:

- Delete an element that belongs to a group or has a generation in a class.
- Delete an element that has a generation reserved.
- Restore a deleted element.
- Delete an element that has a generation under review.

Example

```
CMS> DELETE ELEMENT INITX.FOR "x-version no longer required"
Delete element INITX.FOR? [Y/N] (N): Y
%CMS-I-DELETED, element DISKX:[PROJECT.CMSLIB]INITX.FOR deleted%CMS-S-
DELETIONS, 1 deletion completed
```

This example uses INITX.FOR as an experimental module; when it is no longer needed, you can delete it from the library.

DELETE GENERATION

DELETE GENERATION — Deletes information about one or more generations of an element.

Format

DELETE GENERATION *element-expression* "remark"

Command Qualifiers	Defaults
/AFTER=generation-expression	See Qualifier Glossary
/ARCHIVE[=file-specification]	/NOARCHIVE
/BEFORE=generation-expression	See Qualifier Glossary
/[NO]CONFIRM	/CONFIRM
/FROM[=generation-expression]	See Qualifier Glossary
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/TO=generation-expression	See Qualifier Glossary

After you delete a generation, it cannot be restored to its former place in the element in the CMS library. You can archive the contents of the generation using the /ARCHIVE qualifier and later restore the contents of the generation (see the *VSI DECset for OpenVMS Guide to the Code Management System*).

If the generation or range of generations to be deleted has a direct descendant generation (that is, a descendant generation on the same line of descent), the changes associated with those generations are combined, and those changes are combined with the changes in the descendant generation. If there is no descendant generation (the generation or range of generations to be deleted is at the end of the line of descent), the changes associated with those generations are discarded. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

Use the /GENERATION qualifier to specify a single generation. /GENERATION=1+ is the default. You can also specify a range of generations with either the /AFTER or /FROM qualifier to delimit the beginning of a range, and either the /BEFORE or /TO qualifier to delimit the end of a range. You can pair these sets of qualifiers to specify ranges with inclusive or exclusive endpoints (see the following restrictions).

If you delete the latest generation on the main line of descent of an element that has the **reference copy** attribute, CMS deletes the generation's reference copy and creates a new reference copy corresponding to the generation that is now the latest generation on the main line of descent.

The following restrictions apply to this command:

- You cannot delete generation 1 of an element.
- You cannot delete a generation that has variants.
- You cannot delete a generation that has a review pending.
- You cannot delete a reserved generation.
- You cannot delete a generation that is in a class.
- All generations in the specified range of generations to be deleted must be on the same line of descent.
- You cannot use /GENERATION in combination with /AFTER, /BEFORE, /FROM, or /TO.
- You cannot specify /AFTER and /FROM on the same command line.
- You cannot specify /BEFORE and /TO on the same command line.

Example

```
CMS> DELETE GENERATION SAMPLE.PAS/AFTER=1/BEFORE=V1
_Remark: delete a range
%CMS-E-NOGENDELETED, no generations of DISKX:[PROJECT.CMSLIB]SAMPLE.PAS
deleted
-CMS-E-VARINRANGE, range has variants

CMS> DELETE GENERATION/ARCHIVE/FROM=2A1/TO=2A1+/NOCONFIRM SAMPLE.PAS
_Remark: delete the variant range and archive the deleted generations
%CMS-S-GENDELETED, 3 generations of element DISKX:
[PROJECT.CMSLIB]SAMPLE.PAS deleted
```

The first command specifies that all generations be deleted between generation 1 and the generation in class V1. CMS could not delete the generations because it found variants for the indicated generations.

The second command specifies a range of generations to be deleted from and including the variant generation 2A1 to and including the latest variant generation of the element SAMPLE.PAS. In this case, CMS deleted three generations of the element. The /ARCHIVE qualifier directs CMS to save the deleted generations in an archive file in your default directory.

To display the descendants of a generation and the classes containing the generations, use the SHOW GENERATION/DESCENDANTS/MEMBER command.

DELETE GROUP

DELETE GROUP — Deletes one or more groups from a CMS library.

Format

DELETE GROUP group-expression "remark"

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/REMOVE_CONTENTS	See Qualifier Glossary

If a group is not empty and the /REMOVE_CONTENTS qualifier has not been specified, or it belongs to another group, CMS returns an error and does not delete the group.

Even though a group is deleted, records of transactions that created and used the group are kept in the library history. You can reuse the deleted group name to create a new group. However, there is no distinction between the two groups in the library history, except their transactions are separated by entries for DELETE GROUP and CREATE GROUP commands.

To determine which elements and groups belong to a group, use the SHOW GROUP command with the /CONTENTS qualifier.

You cannot perform the following operations:

- Delete a group containing any elements or groups. Use the /REMOVE_CONTENTS qualifier to remove elements from the group during deletion; use the REMOVE ELEMENT command to remove elements from a group prior to deletion.)
- Delete a group belonging to another group. Use the /REMOVE_CONTENTS qualifier to remove groups from the group during deletion; use the REMOVE GROUP command to remove subgroups prior to deletion.
- Delete a group that has read-only access. Use the MODIFY GROUP/NOREAD_ONLY command to change the access to the group.

Example

```
CMS> DELETE GROUP TIME_TST "superseded by comparison tests"
%CMS-S-DELETED, group DISKX:[PROJECT.CMSLIB]TIME_TST deleted
```

This example deletes the group TIME_TST.

DELETE HISTORY

DELETE HISTORY — Deletes all or part of the library history.

Format

DELETE HISTORY “remark”

Command Qualifiers	Defaults
/BEFORE=date-time	/BEFORE=current-time
/[NO]CONFIRM	/CONFIRM
/[NO]LOG	/LOG
/OBJECT=object-name-expression	See Qualifiers Glossary
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/[NO]TRANSACTIONS=(keyword,...)	/TRANSACTIONS=ALL
/USER=username	See Qualifiers Glossary

CMS writes the deleted history records to a file named HISTORY.DMP in your current, default directory. CMS cannot access this file as a history file.

The DELETE HISTORY command does not delete the library creation history record.

Whenever you delete some of the library history, CMS records two transactions. As with other commands that modify the contents of the library, CMS records the deletion transaction. In addition, CMS logs a REMARK transaction at the point that corresponds to the /BEFORE value. If you do not specify the /BEFORE qualifier, the default is /BEFORE=current-time. The REMARK transaction record includes the following remark: “PREVIOUS HISTORY DELETED”. Both the REMARK and the DELETE HISTORY transactions are unusual transactions. When you use the SHOW HISTORY command, CMS identifies unusual transactions by displaying an asterisk (*) in the first column of the transaction record.

Example

```
CMS> DELETE HISTORY/BEFORE=TODAY "old history in HISTORY.DMP"
Confirm DELETE HISTORY/BEFORE=10-MAY-2005 [Y/N] (N): y
%CMS-S-HISTDEL, 89 history records deleted
```

This example deletes all the library history prior to the current day.

The following code shows the first few records contained in the HISTORY.DMP file:

```
Deleted dump for CMS Library DISKX:[PROJECT.CMSLIB]
 6-MAR-2005 17:07:50 SNAKE  CREATE ELEMENT OUTPUT.FOR "ASCII format"
 6-MAR-2005 17:26:10 MARTIN CREATE ELEMENT INIT.FOR "init routines"
 8-MAR-2005 12:33:09 LISA   RESERVE INIT.FOR(1) "change header offset"
 9-MAR-2005 17:34:04 RONALD REPLACE INIT.FOR(2) "header offset and
    additional free space added"
.
.
.
```

The CREATE LIBRARY transaction is not deleted from the library history.

DIFFERENCES

DIFFERENCES — Compares the contents of two files.

Format

DIFFERENCES file1 [file2]

Command Qualifiers	Defaults
/[NO]APPEND	/NOAPPEND
/FORMAT= (data-format,data-partition, [no]generation-differences)	See Qualifiers Glossary
/FULL	See Qualifiers Glossary
/IGNORE=(keyword[,...])	See Qualifiers Glossary
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/OUTPUT[=file-specification]	/OUTPUT=file1.DIF
/NOOUTPUT	
/[NO]PAGE_BREAK	/NOPAGE_BREAK
/[NO]PARALLEL	/NOPARALLEL
/SKIP=number-of-lines	See Qualifiers Glossary
/WIDTH=n	See Qualifiers Glossary

Parameter Qualifier	Defaults
/GENERATION[=generation-expression]	

If CMS finds differences, it creates a file named file1.DIF in your current default directory (unless you specified the /OUTPUT qualifier). If the files are the same, CMS issues a message to that effect and does not create a differences file. By default, CMS compares two files that are not located in a CMS library. However, you can direct CMS to use element generations from the current library by specifying the /GENERATION qualifier on one or both of the file-name parameters.

If both file1 and file2 are element generations, both generations must reside in the same library of the library search list or an error will occur.

A difference is defined as one of the following:

- A line (or lines) that are in one file but not in the other.
- A replacement of n lines by m lines (n might not be equal to m).

Only the lines that differ are displayed in the differences file unless you specify the /FULL qualifier. This qualifier directs CMS to include a complete listing in the output file, including identical text and differences between file1 and file2.

A heading at the beginning of the differences file includes the name of the user who entered the command, the date and time the command was entered, and the file specifications of the two files being compared. If you direct CMS to use element generations and you specify the /FORMAT option generation-differences, the differences listing contains a section labeled “Generation Differences” that contains the replacement history for the element. Each generation used in the comparison is identified by an asterisk (*) in the first column of the transaction record.

The differences between the files are found in a section labeled “Text Differences.” Each difference is formatted with the line (or lines) from the first file, followed by the differing lines from the second file. If a difference consists of lines in one file but not the other, only the lines from the file containing the additional text are displayed.

If you specify the /SKIP, /SENTINEL, and /IGNORE qualifiers on the same command line, they are processed in the following order:

1. /IGNORE=HISTORY
2. /IGNORE=NOTES
3. /SKIP
4. /SENTINEL
5. /IGNORE options other than HISTORY or NOTES

For example, if you specify /SKIP=5 and /SENTINEL=("sushi","bar"), DIFFERENCES disregards the first five lines in each of the compared files, and then searches the remainder of each file for the sentinel character strings “sushi” and “bar.”

The following restrictions apply when you use the DIFFERENCES command with a CMS library search list:

1. If both file1 and file2 are element generation specifications, both elements must reside in the same library as the library search list.
2. During a library search list, CMS displays error messages if it cannot locate one or both of the specified elements in the first library. CMS displays error messages for each library of the library search list in which the element is not found.

Parameter Qualifier

/GENERATION[=generation-expression]

Directs CMS to search for an element generation in the current CMS library. The /GENERATION qualifier must immediately follow the element name to which it applies. If you specify /GENERATION but do not provide a generation expression, CMS uses the latest generation on the main line of descent.

You can specify a generation indirectly by using a class name, the plus operator, the semicolon, or relative generation offsets. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

Example

In the following example, the files FIB1.PAS and FIB2.PAS are used as input to the DIFFERENCES command. The contents of these files are as follows:

FIB1.PAS;1:

```
PROGRAM FIBONACCI (OUTPUT);
VAR FOLD1, FOLD2, FNEW, I: INTEGER;
BEGIN
    FOLD1 := 0; FOLD2 := 0; FNEW := 1; I := 1;
    REPEAT      WRITELN(I, FNEW);
        FOLD1 := FOLD2; FOLD2 := FNEW; FNEW := FOLD1 + FOLD2;
        I := I + 1;    UNTIL FNEW > (MAXINT DIV 2);
END .
```

FIB2.PAS;1:

```
PROGRAM FIBONACCI (OUTPUT);
VAR FOLD1, FOLD2, FNEW, I: INTEGER;
BEGIN
    WRITELN('  I              FNEW');
    FOLD1 := 0; FOLD2 := 0; FNEW := 1; I := 1;
    REPEAT      WRITELN(I:3, FNEW:20);
        FOLD1 := FOLD2; FOLD2 := FNEW; FNEW := FOLD1 + FOLD2;
        I := I + 1;    UNTIL FNEW > (MAXINT DIV 2);
END .
```

```
CMS> DIFFERENCES FIB1.PAS FIB2.PAS
%CMS-I-DIFFERENT, files are different
```

This example writes the differences between FIB1.PAS and FIB2.PAS to the file FIB1.DIF. The contents of FIB1.DIF are as follows:

```
CMS File Comparison Utility
Files Compared By SMITH On 30-APR-2005 15:30:37
(1) DISKX:[WORK.TESTS]FIB1.PAS;1
(2) DISKX:[WORK.TESTS]FIB2.PAS;1
+++++
File DISKX:[WORK.TESTS]FIB2.PAS;1 Line 4
      2)      WRITELN('  I              FNEW');
-----
+++++
```

```
File DISKX:[WORK.TESTS]FIB1.PAS;1 Line 6
1)      WRITELN(I, FNEW);File DISKX:[WORK.TESTS]FIB2.PAS;1 Line 6
2)      WRITELN(I:3, FNEW:20);
-----
**** End of Differences ****
```

The first difference consists of an additional line in the file FIB1.PAS. That line is displayed with a heading to indicate that it is line 4 of the file FIB1.PAS. The second difference shows that a line from FIB2.PAS differs from a line in FIB1.PAS. The differing lines from both files are included in the report.

DIFFERENCES (2)

DIFFERENCES (2) — Compares the contents of two classes.

Format

DIFFERENCES /CLASS class-name1 class-name2

Command Qualifiers	Defaults
/[NO]APPEND	/NOAPPEND
/CLASS	See Qualifiers Glossary
/FULL	See Qualifiers Glossary
/IGNORE=FIRST_VARIANT	See Qualifiers Glossary
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/OUTPUT[=file-specification]	/OUTPUT=class-name1.DIF
/NOOUTPUT	
/[NO]PARALLEL	/NOPARALLEL
/SHOW=(options[,...])	See Qualifiers Glossary
/WIDTH=n	See Qualifiers Glossary

If CMS finds differences, it creates a file named class-name1.DIF in your current default directory (unless you specified the /OUTPUT qualifier). If the classes are the same, CMS issues a message to that effect and does not create a differences file.

Only those generations that differ between classes are displayed in the differences file unless you specify the /FULL qualifier. This qualifier directs CMS to include a complete listing in the output file, including identical generations and differences between classes.

The /CLASS qualifier is required and must be entered before all other qualifiers.

If you specify the /IGNORE=first-variant qualifier on the same command line, CMS ignores differences where the generation in one class is the first variant of the generation in the other class. For example, 2C1 is the first variant of generation 2. DIFFERENCES/CLASS does not note when generation 2C1 is a member of the first class and generation 2 is a member of the second class.

To direct CMS to ignore either variant or generation differences between classes, use the /SHOW qualifier with the appropriate keyword. If the /SHOW qualifier is not specified, all differences are

shown. If an element generation is a member of one class, but other generations of that element do not exist in the other class, the difference is shown regardless of whether the /SHOW qualifier was specified.

FETCH

FETCH — Fetches a copy of one or more specified element generations from a CMS library to your current, default directory. The element is not reserved, and CMS does not allow you to replace a fetched element. CMS allows you to fetch an element that is reserved, and notifies you of any current reservations for the element.

Format

FETCH *element-expression* "remark"

The presence or absence of a remark determines whether the **FETCH** transaction is recorded in the library history. If you enter a remark, CMS records the transaction in the history file. If you enter a null remark, CMS does not record the transaction in the history file.

If a version of a file with the same name as the element exists in your current, default directory when you execute the fetch transaction, CMS notifies you. A new version is then created with the next higher version number.

When you fetch a generation of an element from a CMS library, CMS restores the file creation time, revision time, revision number, and record format and attributes. The file placed in your directory has the same creation and revision times as the file used to create the generation you are fetching.

Example

```
CMS> FETCH INIT.FOR "check for correct spelling"
Element INIT.FOR currently reserved by:
      (1)      SMITH      2      30-APR-2005 15:48:35.65  "to add new routine"
%CMS-S-FETCHED, generation 2 of element DISKX:[PROJECT.CMSLIB]INIT.FOR
      fetched
```

This example fetches the latest generation on the main line of descent of element INIT.FOR. CMS indicates that the element is reserved and continues with the fetch transaction.

HELP

HELP — Provides online help for CMS at both the DCL and the CMS subsystem command level.

Format

```
RESET { topic
      command
      command /qualifier
      command option
      command option /qualifier }
```

At DCL level, type **HELP CMS** or **CMS HELP** for information. At CMS subsystem level, type **HELP** for information. If you omit a parameter after **HELP CMS**, you get an overview of the CMS HELP facility.

This overview displays the general syntax of a CMS command, and all the keywords for which you can obtain more information.

The command `HELP CMS HELP` explains how the HELP topics are organized.

INSERT ELEMENT

INSERT ELEMENT

Format

INSERT ELEMENT *element-expression group-expression "remark"*

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/IF_ABSENT	See Qualifier Glossary
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

Inserts one or more elements or groups into one or more specified groups. The groups must exist. When you use the INSERT ELEMENT command with a group name, you insert the contents of the group. For example, if you insert group A into group B with the INSERT ELEMENT command, group B will contain the contents of group A at the time of the insertion transaction. If the contents of group A change at a later time, the contents of group B are not affected. To insert a group into another group, use the INSERT GROUP command.

Note that you cannot insert the following:

- An element into a group that has read-only access
- An element into a group if the element already belongs to the group
- An element into a group in another library (the element and group must be in the same library)

Example

```
CMS> INSERT ELEMENT DBAS EXAMPLES "more examples for book"
%CMS-I-INSERTED, element DISKX:[PROJECT.CMSLIB]ARTFIG.CXS inserted into
DISKX:[PROJECT.CMSLIB]group EXAMPLES
%CMS-I-INSERTED, element DISKX:[PROJECT.CMSLIB]SNAKE.TXT inserted into
DISKX:[PROJECT.CMSLIB]group EXAMPLES
```

Inserts the contents of group DBAS into group EXAMPLES.

INSERT GENERATION

INSERT GENERATION — Places one or more specified element generations into one or more classes. The class or classes must exist. See the CREATE CLASS command for more information.

Format

INSERT GENERATION *element-expression class-expression "remark"*

Command Qualifiers	Defaults
/ALWAYS	See Qualifier Glossary
/BEFORE=date-time	See Qualifier Glossary
/[NO]CONFIRM	/NOCONFIRM
/IF_ABSENT	See Qualifier Glossary
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/[NO]SUPERSEDE	/NOSUPERSEDE

A class can contain only one generation of an element. You cannot insert any generations into a class that has the read-only attribute. (See the **MODIFY CLASS** command for more information.)

The following restrictions apply to this command:

- You cannot insert an element generation into a class that has read-only access.
- You cannot insert a generation into a class in another library (the generation and class must be in the same library).
- A class can contain only one generation of any particular element.

Example

```
CMS> INSERT GENERATION INIT.FOR, SPEC.TXT/GENERATION=3/IF_ABSENT/CONFIRM
_Class name: BASELEVEL_1 "inserting generation 3 for final baselevel
Insert generation 3 of element INIT.FOR into class BASELEVEL_1? [Y/N]
(N): Y
%CMS-I-GENINSERTED, generation 3 of element DISKX:[PROJECT.CMSLIB]INIT.FOR
inserted into class DISKX:[PROJECT.CMSLIB]BASELEVEL_1
Insert generation 3 of element SPEC.TXT into class BASELEVEL_1? [Y/N]
(N): Y
%CMS-I-GENINSERTED, generation 3 of element DISKX:[PROJECT.CMSLIB]SPEC.TXT
inserted into class DISKX:[PROJECT.CMSLIB]BASELEVEL_1
CMS-I-INSERTIONS, 2 insertions completed
```

This example inserts generation 3 of both elements **INIT.FOR** and **SPEC.TXT** into the class **BASELEVEL_1**. The **/IF_ABSENT** qualifier indicates that the generations should be inserted only if they are not found in the class. The **/CONFIRM** qualifier directs CMS to prompt you for confirmation before each insertion.

INSERT GROUP

INSERT GROUP — Inserts one or more groups into one or more specified groups. Both groups must exist.

Format

INSERT GROUP subgroup-expression group-expression "remark"

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/IF_ABSENT	See Qualifier Glossary
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

If you use the INSERT GROUP command to insert group A into group B, the elements accessible through group B change as the contents of group A change. A group cannot be a member of itself; that is, it cannot be a subgroup of itself. For example, you cannot insert group A into group B if group A already contains group B.

The following restrictions apply to this command:

- You cannot insert a group into another group that has read-only access.
- You cannot create recursive groups (a group cannot directly or indirectly be a member of itself).
- You cannot insert a group from one library into a group in another library (both groups must be in the same library).
- You can insert a group into another group only once.

Example

```
CMS> INSERT GROUP USER_MANUAL CODE_AND_DOCS "user documentation"
%CMS-S-INSERTED, group DISKX:[PROJECT.CMSLIB]USER_MANUAL inserted into
group DISKX:[PROJECT.CMSLIB]CODE_AND_DOCS
```

This example inserts the group USER_MANUAL into the group CODE_AND_DOCS. As long as group USER_MANUAL belongs to group CODE_AND_DOCS, any changes to the contents of USER_MANUAL are reflected in the contents of CODE_AND_DOCS. Any element accessible through USER_MANUAL is also accessible through CODE_AND_DOCS.

MARK GENERATION

MARK GENERATION — Marks the review status of the specified element generation from none to pending, and inserts it into the review pending list.

Format

MARK GENERATION element-expression "remark"

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/[NO]LOG	/LOG

Command Qualifiers	Defaults
/OCCLUDE[=option,...]	/OCCLUDE=ALL

Use the REVIEW GENERATION command to review the element generation. Use one of the following commands to change the review status of the element generation: ACCEPT GENERATION, REJECT GENERATION, or CANCEL REVIEW. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

You can only use this command on element generations that do not have reviews pending.

Example

```
$ CMS MARK GENERATION/GENERATION=1X1 SPEC.COM
_Remark: check this gen out before reinserting into class
%CMS-S-MARKED, generation 1X1 of element DISKX:[PROJECT.CMSLIB]EXAMPLE.SDML
marked for review
```

This example marks a specific generation for review. CMS adds the generation to the review pending list.

MODIFY CLASS

MODIFY CLASS — Changes the characteristics of one or more specified classes.

Format

MODIFY CLASS class-expression /qualifier “remark”

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/[NO]LOG	/LOG
/NAME=class-name	See Qualifier Glossary
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/[NO]READ_ONLY	See Qualifier Glossary
/REMARK= “string”	See Qualifier Glossary

You can alter the following characteristics:

- The name of the class.
- The access to the class (READ_ONLY or NOREAD_ONLY). You cannot change the contents, the name, or the remark of a class set to READ_ONLY.
- The creation remark associated with the class.

Use the SHOW CLASS command to display class characteristics.

The following restrictions apply to this command:

- You cannot modify a class that has read-only access. If a class has read-only access, you must change it to NOREAD_ONLY access to change the contents of the class or any other characteristics.
- You must specify one or more of the following qualifiers: /NAME, /[NO]READ_ONLY, or /REMARK.

Example

```
CMS> MODIFY CLASS PRE_RELEASE/NOREAD_ONLY/NAME=PRE_RELEASE_V3
_Remark: include additional functions
%CMS-S-MODIFIED, class DISKX:[PROJECT.CMSLIB]PRE_RELEASE modified
```

This example renames class PRE_RELEASE to PRE_RELEASE_V3. Because PRE_RELEASE was set to READ_ONLY, you must use the /NOREAD_ONLY qualifier to modify the class.

MODIFY ELEMENT

MODIFY ELEMENT — Changes the characteristics of one or more specified elements.

Format

MODIFY ELEMENT *element-expression* /*qualifier* "remark"

Command Qualifiers	Defaults
/[NO]CONCURRENT	See Qualifier Glossary
/[NO]CONFIRM	/NOCONFIRM
/HISTORY= "string"	See Qualifier Glossary
/NOHISTORY	
/[NO]LOG	/LOG
/NAME=element-name	See Qualifier Glossary
/NOTES= "string"	See Qualifier Glossary
/NONOTES	
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/POSITION=n	See Qualifier Glossary
/[NO]REFERENCE_COPY	See Qualifier Glossary
/REMARK= "string"	See Qualifier Glossary
/[NO]REVIEW	See Qualifier Glossary

You can alter the following characteristics:

- The **concurrent** attribute of the element
- The history string inserted in the element history when the element is reserved or fetched
- The element name
- The notes string and related **position** attribute

- The **reference copy** attribute of the element
- The creation remark associated with the element
- The **review** attribute

Use the SHOW ELEMENT command to display element characteristics.

If the **history**, **notes**, or **position** attribute is modified on an element that has the **reference copy** attribute, CMS creates an updated reference copy for the element.

The following restrictions apply to this command:

- You cannot modify an element if it is set to read-only access.
- You can modify only the **reference copy**, **remark**, and **review** attributes of an element that has a generation reserved.
- If you specify /NOTES, you must specify /POSITION on the same command line.
- You must specify only one or more of the following qualifiers:

```
/[NO]CONCURRENT
/[NO]HISTORY
/NAME
/[NO]NOTES
/POSITION
/[NO]REFERENCE_COPY
/REMARK
/[NO]REVIEW
```

Example

```
CMS> MODIFY ELEMENT INIT.FOR/NOCONCURRENT
_Remark: no more changes, other than those already discussed
%CMS-S-MODIFIED, element DISKX:[PROJECT.CMSLIB]INIT.FOR modified
```

This example sets the element INIT.FOR to NOCONCURRENT access. Only one person can reserve the element at a time.

MODIFY GENERATION

MODIFY GENERATION — Changes the characteristics associated with a specified generation of one or more elements.

Format

MODIFY GENERATION *element-expression* /*qualifier* "remark"

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

Command Qualifiers	Defaults
/REMARK= "string"	See Qualifier Glossary

Use the SHOW GENERATION command to display generation characteristics.

This command does not change a generation's review status. See the ACCEPT GENERATION, CANCEL GENERATION, MARK GENERATION, REJECT GENERATION, and REVIEW GENERATION commands for more information.

You must specify either the /GENERATION or /REMARK qualifiers.

Example

```
CMS> MODIFY GENERATION/GENERATION=5/REMARK="Obsolete"
_Element expression: SPEC.TXT
_Remark: Marked obsolete
%CMS-S-MODIFIED, generation 5 modified
```

This example specifies that a new remark be substituted for the creation remark of generation 5 of element SPEC.TXT. You must also specify the element expression and its associated remark, which is logged in the history file.

MODIFY GROUP

MODIFY GROUP — Changes the characteristics of one or more specified groups.

Format

MODIFY GROUP group-expression /qualifier "remark"

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/[NO]LOG	/LOG
/NAME=group-name	See Qualifier Glossary
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/[NO]READ_ONLY	See Qualifier Glossary
/REMARK= "string"	See Qualifier Glossary

You can alter the following characteristics:

- The name of the group.
- The access to the group (READ_ONLY or NOREAD_ONLY). You cannot change the contents of a group set to READ_ONLY.
- The creation remark associated with the group.

Use the SHOW GROUP command to display group characteristics.

The following restrictions apply to this command:

- You cannot change the attributes of a group set to READ_ONLY. If a group has read-only access, you must change it to NOREAD_ONLY to change any other characteristics.
- You must specify one or more of the following qualifiers: /NAME, /[NO]READ_ONLY, or /REMARK.

Example

```
CMS> MODIFY GROUP TESTS/READ_ONLY "coordinate before changing contents"
%CMS-S-MODIFIED, group DISKX:[PROJECT.CMSLIB]TESTS modified
```

This example sets the group TESTS to READ_ONLY. After the group is set to READ_ONLY, the contents cannot be changed.

MODIFY LIBRARY

MODIFY LIBRARY — Establishes or removes the connection between the current CMS library and a reference copy directory.

Format

MODIFY LIBRARY /qualifier "remark"

Command Qualifiers	Defaults
/[NO]CONCURRENT	/NOCONCURRENT
/[NO]CONFIRM	/NOCONFIRM
/[NO]EXTENDED_FILENAMES	/NOEXTENDED_FILENAMES
/[NO]KEEP	/NOKEEP
/[NO]LOG	/LOG
/[NO]LONG_VARIANT_NAMES	/NOLONG_VARIANT_NAMES
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/NOREFERENCE_COPY	
/REVISION_TIME[=option]	See Qualifier Glossary

The reference copy directory cannot be a CMS library. This command does not add files to or delete any files from a reference copy directory. After establishing a reference copy directory for a library, subsequent transactions that create new element generations on the main line of descent also update the reference copy directory (if the element is set to /REFERENCE_COPY).

You must use the MODIFY ELEMENT/NOREFERENCE_COPY command on the elements in the library before using the MODIFY LIBRARY/NOREFERENCE_COPY command.

If you specify MODIFY LIBRARY/REFERENCE_COPY and the reference copy directory is already set, CMS first verifies all the files found in that directory. The contents of the directory must match the elements that have /REFERENCE_COPY set in the CMS library. If CMS finds discrepancies or if elements are set with /NOREFERENCE_COPY and there are existing reference copies for those elements, CMS advises you to use VERIFY/REPAIR.

Use the SHOW LIBRARY command to display library characteristics.

Example

```
CMS> MODIFY LIBRARY/REFERENCE_COPY=[WORK.REFCOPY] "current test area"
%CMS-S-MODIFIED, library [WORK.CODELIB] modified
```

This example establishes the reference copy directory [WORK.REFCOPY] for the current CMS library. To update the reference copy directory, an element must be set to /REFERENCE_COPY. If these two conditions are met, each transaction that creates a new mainline element generation also updates the reference copy directory.

MODIFY RESERVATION

MODIFY RESERVATION — Changes the remark associated with a particular reservation of an element elements.

Format

MODIFY RESERVATION *element-expression* /*qualifier* "remark"

Command Qualifiers	Defaults
/[NO]CONFIRM	/CONFIRM
/GENERATION[=generation-expression]	See Qualifiers Glossary
/IDENTIFICATION NUMBER=n	See Qualifiers Glossary
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/REMARK= "new-remark"	See Qualifiers Glossary

Use the SHOW RESERVATIONS command to display reservation characteristics.

The /REMARK qualifier is required with this command. The qualifier value is a quoted string that CMS uses to replace the original reservation remark. If multiple reservations exist, use the /GENERATION and /IDENTIFICATION_NUMBER qualifiers to further identify the reservation whose remark you want to change.

Example

```
CMS> MODIFY RESERVATION/GENERATION=5/REMARK="Add new feature"
_Element expression: SPEC.TXT
_Remark: "Modified the reservation remark"
%CMS-S-MODIFIED, reservation SPEC.TXT modified
```

This changes the original reservation remark associated with generation 5 of the element SPEC.TXT to "Add new feature".

REJECT GENERATION

REJECT GENERATION — Changes the review status of each specified element generation from pending to rejected, and removes it from the review pending list.

Format

REJECT GENERATION *element-expression* **"remark"**

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

You can use this command only on element generations that have reviews pending (see the REVIEW GENERATION command for more information).

If you try to reserve a generation that has been rejected, CMS issues a message stating that the generation was rejected and then prompts you for confirmation. Additionally, further generations created from a rejected generation are marked for review, regardless of the element's review attribute. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

Example

```
$ CMS REJECT GENERATION EXAMPLE.SDML "don't change this until it is fixed"
%CMS-S-REJECTED, generation 1X1 of element DISKX:
[PROJECT.CMSLIB]EXAMPLE.SDML rejected
```

This example rejects the latest generation of EXAMPLE.SDML, which was on the review pending list. This generation of EXAMPLE.SDML remains rejected until you specify a MARK GENERATION command.

REMARK

REMARK

Format

REMARK **"remark"**

Command Qualifiers	Defaults
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/UNUSUAL	See Qualifier Glossary

Places a remark in the library history. When you let CMS prompt you for the remark, the length of the remark cannot exceed 254 characters. When you enter the remark on the command line, the length of the remark cannot exceed 256 characters. The remark is recorded in the library history in the following format:

```
date time username REMARK "remark"
```

For more information on remarks, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

Example

```
CMS> REMARK "all transactions from this point use modules for new system"
%CMS-S-REMARK, remark added to history file
```

This example adds the remark enclosed in quotation marks to the library history.

REMOVE ELEMENT

REMOVE ELEMENT — Removes one or more elements from one or more groups.

Format

REMOVE ELEMENT *element-expression group-expression "remark"*

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/IF_PRESENT	See Qualifiers Glossary
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

This command does not delete the elements from the library, but there is no longer any association between the elements and the groups.

You cannot remove elements from a group that has read-only access (see the description of the MODIFY GROUP command).

Example

```
CMS> REMOVE ELEMENT *.* A2 "remove all elements from group"
%CMS-S-REMOVED, element DISKX:[PROJECT.CMSLIB]SEARCH.FOR removed from group
DISKX:[PROJECT.CMSLIB]A2
%CMS-S-REMOVED, element DISKX:[PROJECT.CMSLIB]ARGCHK.FOR removed from group
DISKX:[PROJECT.CMSLIB]A2      .      .      .
```

This example removes all the elements from the group A2.

REMOVE GENERATION

REMOVE GENERATION — Removes one or more element generations from one or more classes.

Format

REMOVE GENERATION *element-expression class-expression "remark"*

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/IF_PRESENT	See Qualifiers Glossary

Command Qualifiers	Defaults
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

The command does not delete the element generation from the library, but the element generation is no longer associated with the class.

To remove one element generation from a class and replace it with another generation of the same element, use the INSERT GENERATION command with the /SUPERSEDE qualifier.

You cannot remove a generation from a class with read-only access (see the MODIFY CLASS command).

Example

```
CMS> REMOVE GENERATION USER.DOC PRE_RELEASE_V3
_Remark: internal documentation is online
%CMS-S-GENREMOVED, generation 2 of element DISKX:[PROJECT.CMSLIB]USER.DOC
removed from class DISKX:[PROJECT.CMSLIB]PRE_RELEASE_V3
```

This example removes generation 2 of USER.DOC from class PRE_RELEASE_V3.

REMOVE GROUP

REMOVE GROUP — Removes one or more groups from another group or groups.

Format

REMOVE GROUP group-expression1 group-expression2 "remark"

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/IF_PRESENT	See Qualifiers Glossary
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

The command does not delete the group from the library, but there is no longer any association between the two groups. Removing group A from group B means that the contents of group A are no longer accessible through group B.

You cannot remove a group from a group with read-only access (see the description of the MODIFY GROUP command).

Example

```
CMS> REMOVE GROUP A1 A2 "remove group from group"
%CMS-S-REMOVED, group DISKX:[PROJECT.CMSLIB]A1 removed from group DISKX:
[PROJECT.CMSLIB]A2
```

This example removes group A1 from group A2.

REPLACE

REPLACE — Returns each specified element reservation from your default directory to the current CMS library, which creates a new generation of the element.

Format

REPLACE *element-expression* "remark"

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/IDENTIFICATION_NUMBER=n	See Qualifiers Glossary
/IF_CHANGED	See Qualifiers Glossary
/INPUT[=file-specification]	See Qualifiers Glossary
/INSERT_INTO_CLASS=	See Qualifiers Glossary
(class-expression)	
/[NO]KEEP	/NOKEEP
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/[NO]RESERVE	/NORESERVE
/VARIANT=variant-letter	/NOVARIANT
/NOVARIANT	

You can direct CMS to use a file other than the one located in your default directory by specifying the /INPUT qualifier. After the reserved generation is replaced, CMS deletes the file used to create the new generation (and any earlier versions of the file in the same directory). If you specify either the /KEEP or /RESERVE qualifier, CMS does not delete the file. You cannot replace a reserved generation held by another user unless you hold BYPASS process privilege, or you are granted BYPASS access to the element by an access control entry (see the *VSI DECset for OpenVMS Guide to the Code Management System*). After the replace transaction is completed, the reservation is ended.

To insert newly created generations into one or more classes automatically, use the /INSERT_INTO_CLASS qualifier. This qualifier combines the functions of the REPLACE and INSERT GENERATION commands and prevents unwanted updates from occurring between individual REPLACE and INSERT GENERATION commands. To use this qualifier, you require EXECUTE access to the INSERT GENERATION command.

The /NOCONFIRM qualifier does not override the confirmation prompt issued when you unreserve a concurrent reservation, or when you cancel another user's reservation.

The number of the new generation is the number of its predecessor with the rightmost level number increased by 1. For example, if you reserve generation 1A1, CMS creates generation 1A2 when you replace it. CMS also stores the creation date and time, revision date and time, and file revision number of the file used to create the new generation. When you fetch or reserve a generation of an element, CMS restores the times and file revision number associated with the file used to create the element generation. You can also the SHOW GENERATION/FULL command to display this information.

CMS reports an error if you try to create a generation that is already in the library (see the description of the /VARIANT qualifier in the Command Qualifier Glossary).

The REPLACE command checks for other current reservations and concurrent replacements of the element, and whether you are replacing another user's reservation. If any of these situations occur, CMS prompts whether you want to proceed with the command. If you type NO, or press Return or Ctrl/Z, the command is not executed. If you type YES, CMS executes the command and records the transaction as an unusual occurrence.

If you have more than one reservation of an element, or if you are replacing a concurrent reservation made by another user (that is, if there is any ambiguity), you must specify the exact reservation to be replaced (see the *VSI DECset for OpenVMS Guide to the Code Management System*). To do this, use either the /GENERATION or /IDENTIFICATION_NUMBER qualifier.

You can use /GENERATION as long as the concurrent reservations are not on the same generation. If you have more than one concurrent reservation for the same generation, you must identify the specific reservation to be replaced. Each reservation is assigned an identification number. Use the SHOW RESERVATIONS command to determine the identification number of each reservation.

The identification number appears in parentheses at the beginning of each line. If you use the /IDENTIFICATION_NUMBER qualifier, you do not need to use the /GENERATION qualifier; when both are used, CMS ignores the /GENERATION qualifier.

If the **reference copy** attribute is enabled for an element and REPLACE creates the new generation on the main line of descent, CMS creates a new reference copy in the reference copy directory for the element and deletes the old copy from the reference copy directory.

Replacing an Element with Defined Attributes

If you reserve a generation of an element with an embedded history and then replace it, the REPLACE command ignores the history; CMS will not copy the history into your CMS library. If you add text to the file in or above the history (relative to #B), or in or below the history (relative to #H), the REPLACE command issues an error message and the command is not executed.

If you reserve a file with embedded notes and then replace it, the REPLACE command does not copy the notes to the CMS library. If, while editing the file, you insert text that looks like an embedded note, it is deleted when the file is replaced.

For more information about concurrent reservations and replacements, and on embedded histories and notes, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

Example

```
CMS> REPLACE FILEIO.BLI
_Remark: descriptor bug fixed
%CMS-S-GENCREATED, generation 14 of element DISKX:
[PROJECT.CMSLIB]FILEIO.BLI created
```

This example creates a new generation on the main line of descent of element FILEIO.BLI.

RESERVE

RESERVE — Sends a copy of the specified element generation of each specified element to your current, default directory (or to another location if you specify the /OUTPUT qualifier).

Format

RESERVE *element-expression* **"remark"**

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/HISTORY= "string"	See Qualifiers Glossary
/NOHISTORY	
/[NO]LOG	/LOG
/MERGE=generation-expression	/NOMERGE
/NOMERGE	
/NOCONCURRENT	See Qualifiers Glossary
/NOTES= "string"	See Qualifiers Glossary
/NONOTES	
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/OUTPUT[=file-specification]	See Qualifiers Glossary
/NOOUTPUT	
/POSITION=column-number	See Qualifiers Glossary

Each element is marked as reserved. After you modify the file, return your changes to the library with the REPLACE command. You can also cancel the reservation using the UNRESERVE command.

You can reserve more than one generation of the same element if concurrent reservations are allowed (see the *VSI DECset for OpenVMS Guide to the Code Management System*). If a generation of an element is reserved by you or another user, or a generation of the element is under review or has been rejected, CMS displays the current reservations and review comments, and prompts whether you want to proceed with the command. If you type YES, you are added to the list of current reservers. The transaction is recorded as an unusual occurrence (see the *VSI DECset for OpenVMS Guide to the Code Management System*). If you type NO or press Return or Ctrl/Z, no action is taken.

If a version of the output file exists in your default directory when you enter the RESERVE command, CMS notifies you. A new version is then created with the next higher version number.

When you retrieve a generation of an element from a CMS library, CMS restores the file creation and revision times. The file placed in your directory has the same creation and revision times as the file used to create the generation that you are reserving. If you specify /MERGE, the file placed in your default directory has the current creation and revision times.

You cannot use /NOCONFIRM to override the prompt generated when you try to reserve a generation of an element that is already reserved by you or by another user.

Examples

1. CMS> **RESERVE SYNTAX.PAS**
 Remark: **add syntax for RECORD declaration**
 Element SYNTAX.PAS currently reserved by:

```
(1) JERRYH      1      24-JUL-2005 16:17:45 "implement FOR loop syntax"
Proceed?  [Y/N] (N): YES
%CMS-S-RESERVED, generation 1 of element DISKX:
[PROJECT.CMSLIB]SYNTAX.PAS reserved
```

This example creates a concurrent reservation for the element SYNTAX.PAS. Because you typed YES in response to the Proceed? prompt, generation 1 of the element is reserved.

2. CMS> **RESERVE COPY.BLI/GENERATION=12/MERGE=11A1**
 _Remark:merging new I/O routines with library self checking
 %CMS-W-MERGECONFLICT, 31 changes successfully merged with 3 conflicts
 %CMS-S-RESERVED, generation 12 of element DISKX:[PROJECT.CMSLIB]COPY.BLI
 reserved and merged with generation 11A1

Merges generation 11A1 into generation 12 of the element COPY.BLI and reserves generation 12. The version delivered to the user directory contains the changes from both generations. Thirty-one changes were successfully merged. There were three conflicts between the two generations. You must resolve these conflicts by editing the file. For more information on merging, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

RETRIEVE ARCHIVE

RETRIEVE ARCHIVE — Retrieves one or more generations of an element from one or more archive files.

Format

RETRIEVE ARCHIVE file-expression

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/[NO]LOG	/LOG
/OUTPUT[=file-specification]	See Qualifiers Glossary
/NOOUTPUT	

CMS retrieves the highest numbered generation from the archive file and places a copy of the generation in your default directory with the same file name and file type of the element whose generation was originally deleted (unless you specify another name or location with the /OUTPUT qualifier).

You can specify that a particular generation be retrieved from the archive file with the /GENERATION qualifier. If you omit /GENERATION, CMS retrieves the highest numbered generation in the archive file. CMS creates one file for each retrieved generation. You do not need to have a library currently set to use this command, because the RETRIEVE ARCHIVE command does not interact with the CMS library.

Example

```
CMS> RETRIEVE ARCHIVE/GENERATION=2A3 SAMPLE.CMS_ARCHIVE
%CMS-S-RETRIEVED, generation 2A3 of element DISKX:
[PROJECT.CMSLIB]SAMPLE.PAS
retrieved from DISKX:[WORK]SAMPLE.CMS_ARCHIVE;1
```

This example retrieves generation 2A3 from the file SAMPLE.CMS_ARCHIVE in your default directory. CMS names generation 2A3 to its original element name SAMPLE.PAS and places it in your default directory.

REVIEW GENERATION

REVIEW GENERATION — Associates a review remark with one or more specified element generations.

Format

REVIEW GENERATION element-expression "remark"

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

The review status of the generation must be pending. To display the remarks associated with the generation, enter the SHOW REVIEWS_PENDING command.

You can use this command only on element generations that have reviews pending (for more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*).

Example

```
$ CMS REVIEW GENERATION EXAMPLE.SDML "looks ok to me - JEFF"
%CMS-S-REVIEWED, generation 3 of element DISKX:[PROJECT.CMSLIB]EXAMPLE.SDML
reviewed
```

The latest generation of the element EXAMPLE.SDML is pending review; the REVIEW GENERATION command allows users to associate review comments with that generation. The generation can then be accepted, canceled, or rejected. Use the SHOW REVIEWS_PENDING command to display all generations under review in the library.

SET ACL

SET ACL — Manipulates the access control lists (ACLs) on various objects in the CMS library.

Format

SET ACL /OBJECT_TYPE=type object-expression "remark"

Command Qualifiers	Defaults
/ACL[=(ace[,...])]	See Qualifiers Glossary
/AFTER=ace	See Qualifiers Glossary
/[NO]CONFIRM	/NOCONFIRM
/DEFAULT	See Qualifiers Glossary

Command Qualifiers	Defaults
/DELETE	See Qualifiers Glossary
/LIKE=object-specification	See Qualifiers Glossary
/[NO]LOG	/LOG
/NEW	See Qualifiers Glossary
/OBJECT_TYPE=type	See Qualifiers Glossary
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/REPLACE=(ace[,...])	See Qualifiers Glossary

ACLs are used to control access to individual CMS commands. ACLs are also used to control access to elements, groups, and classes, as well as on the lists containing these entities. You can also put an ACL on the entire library and the library history. For more information on using ACLs, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

When you use the SET ACL/DELETE command to delete a list of access control entries (ACEs) from the ACL, CMS terminates the operation if it encounters an identifier that does not exist on the system. If this occurs, continue the operation with the remaining ACEs specified.

Examples

1. CMS> **SET ACL/OBJECT_TYPE=ELEMENT SAMPLE.PAS/ACL=(IDENTIFIER=WALLEN, -**
_CMS> ACCESS=RESERVE+CONTROL) "setting up acl on element"
%CMS-S-MODACL, modified access control list for element DISKX:
[PROJECT.CMSLIB] SAMPLE.PAS

This example assigns an ACL on the element SAMPLE.PAS, specifying that the user holding the identifier WALLEN has RESERVE and CONTROL access to the element.

2. CMS> **SET ACL/OBJECT_TYPE=LIBRARY ELEMENT_LIST -**
_CMS> /ACL=((IDENTIFIER=WALLEN, OPTIONS=DEFAULT, ACCESS=FETCH) , -
_CMS> (IDENTIFIER=WALLEN, ACCESS=CREATE+CONTROL)) ""
%CMS-S-MODACL, modified access control list for subtype DISKX:
[PROJECT.CMSLIB] ELEMENT_LIST

This example shows how to assign two separate ACLs on the element list. The first ACL specifies a default ACL to be inherited by newly created elements in the library. The second ACL allows the user holding the identifier WALLEN to create elements in the library.

3. CMS> **SET ACL/OBJECT_TYPE=ELEMENT/DEFAULT SAMPLE.PAS ""**
%CMS-S-MODACL, modified access control list for element DISKX:
[PROJECT.CMSLIB] SAMPLE.PAS
SHOW ACL/OBJECT_TYPE=ELEMENT SAMPLE.PAS
ACLs in CMS Library DISKX: [PROJECT.CMSLIB] SAMPLE.PAS
(IDENTIFIER=[PROJECT, WALLEN], ACCESS=FETCH)

The SET ACL/OBJECT_TYPE command causes the default ACL from the element list (see Example 2) to be placed on the existing element SAMPLE.PAS (new elements will inherit this default ACL automatically). The SHOW ACL command displays this ACL.

4. CMS> **SET ACL/OBJECT_TYPE=CLASS BL1/ACL=**
(IDENTIFIER=[DEV, *]+LIBRARIAN, -
_CMS> ACCESS=INSERT+REMOVE) ""

```
%CMS-S-MODACL, modified access control list for class DISKX:
[PROJECT.CMSLIB]BL1
```

This example assigns an ACL allowing INSERT and REMOVE access to class BL1 for users in group DEV holding the LIBRARIAN identifier.

5. CMS> **SET ACL/OBJECT_TYPE=CLASS/LIKE=BL1 BL2 ""**
 %CMS-S-MODACL, modified access control list for class DISKX:
 [PROJECT.CMSLIB]BL2CMS>
SHOW ACL/OBJECT_TYPE=CLASS
 ACLs in CMS Library DISKX:[PROJECT.CMSLIB]BL1
 (IDENTIFIER=[DEV,*]+LIBRARIAN,ACCESS=INSERT+REMOVE) BL2
 (IDENTIFIER=[DEV,*]+LIBRARIAN,ACCESS=INSERT+REMOVE)

The /LIKE qualifier causes the ACL from the class BL1 (see Example 4) to be placed on the class BL2. The SHOW ACL command displays the ACL on both classes BL1 and BL2.

SET LIBRARY

SET LIBRARY — Enables access to an existing CMS library or list of libraries.

Format

SET LIBRARY directory-specification[,...]

Command Qualifiers	Defaults
/[NO]LOG	/LOG
/[NO]VERIFY	/VERIFY

Future CMS commands automatically refer to the library (or libraries) identified in the SET LIBRARY command. The SET LIBRARY command defines logical names beginning with CMS\$ that allow CMS commands to refer implicitly to the library. Do not define logical names beginning with CMS\$ because this prefix is reserved for CMS.

The SET LIBRARY command performs some consistency checks on the directory to verify that it is a valid CMS library (unless you specify the /NOVERIFY qualifier). If the library is not valid, you receive an error message.

You must create the library using the CREATE LIBRARY command. During each session in which you want to use your CMS library, you must use the SET LIBRARY command before accessing a library. This command is not required if you have just created a library (see the CREATE LIBRARY command) because the CREATE LIBRARY command performs an implicit SET LIBRARY.

To create a search list, specify multiple libraries on the SET LIBRARY command. This enables you to manipulate several libraries with one command. You must include commas between the directory specifications.

Examples

1. CMS> **SET LIBRARY [WORK.CMSLIB]**
 %CMS-S-LIBIS, CMS library is DISKX:[WORK.CMSLIB]

This example sets the CMS library to the existing library [WORK.CMSLIB].

```

2. CMS> SET LIBRARY [WORK.CMLSIB], [TEST.CMSLIB]
%CMS-I-LIBIS, library is DISK$:[WORK.CMSLIB]
%CMS-I-LIBINSLIS, library DISK$:[TEST.CMSLIB] inserted at end of library
list
%CMS-S-LIBSET, library set
-CMS-I-SUPERSEDE, library list superseded

```

This example sets (or resets, if there was an existing library or libraries) the current library to contain two libraries.

SET NOLIBRARY

SET NOLIBRARY — Removes one or more libraries from the current library search list.

Format

SET NOLIBRARY [directory-specification[,...]]

Command Qualifiers	Defaults
/[NO]LOG	/LOG

For more information on search lists, see the *VSI DECset for OpenVMS Guide to the Code Management System*. If all libraries are removed from the list, the logical name CMS\$LIB is deassigned.

SHOW ACL

SHOW ACL — Displays the access control list (ACL) associated with the specified object or objects.

Format

SHOW ACL object-expression /OBJECT_TYPE=type

Command Qualifiers	Defaults
/[NO]APPEND	/NOAPPEND
/OBJECT_TYPE = type	See Qualifiers Glossary
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/OUTPUT[=file-specification]	/OUTPUT=SYS\$OUTPUT

Example

```

CMS> SHOW ACL/OBJECT_TYPE=ELEMENT SAMPLE.PAS
ACLs in CMS Library DISKX:[PROJECT.CMSLIB]SAMPLE.PAS
      (IDENTIFIER=[PROJECT,WALLEN], ACCESS=FETCH)

```

This example displays the ACE on element SAMPLE.PAS.

SHOW ARCHIVE

SHOW ARCHIVE — Displays information about the contents of one or more specified archive files.

Format

SHOW ARCHIVE file-expression

Command Qualifiers	Defaults
/[NO]APPEND	/NOAPPEND
/BRIEF	See Qualifiers Glossary
/FULL	See Qualifiers Glossary
/INTERMEDIATE	See Qualifiers Glossary
/OUTPUT[=file-specification]	/OUTPUT=SYSS\$OUTPUT

Example

```
CMS> SHOW ARCHIVE/BRIEF SAMPLE.CMS_ARCHIVE
25-JAN-2005 17:08:47 JONES DISKX:[WORK.CMSLIB] SAMPLE.PAS(2A1 through 2A3)
"delete
the variant range and archive the deleted generations"
```

This example displays information about the archive file, consisting of the date and time entered with the DELETE GENERATION command, the name of the person who archived the file, the library in which the original element resided, the generations archived into the file, and the remark entered on the DELETE GENERATION command.

SHOW CLASS

SHOW CLASS — Lists the names of all established classes in alphabetical order, along with the associated creation remarks, in a CMS library.

Format

SHOW CLASS [class-expression]

Command Qualifiers	Defaults
/[NO]APPEND	/NOAPPEND
/BRIEF	See Qualifiers Glossary
/[NO]CONTENTS	/NOCONTENTS
/FULL	See Qualifiers Glossary
/INTERMEDIATE	See Qualifiers Glossary
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/OUTPUT[=file-specification]	/OUTPUT=SYSS\$OUTPUT

Example

```
CMS> SHOW CLASS
```

```
Classes in CMS Library DISKX:[RELEASE.CMSLIB]
INTERNAL_RELEASE    "for internal use only"
MESSAGESTEST        "filter evolution checks"
PASCAL_CLASS        "PASCAL tests"
```

This example displays the class name and the creation remark for each class in the library.

SHOW ELEMENT

SHOW ELEMENT — Lists the name of each specified element in alphabetical order, along with the remark logged at the time the element was created or modified, in a CMS library. You can also specify qualifiers that provide information about element attributes, concurrent access, and the groups to which the element belongs.

Format

SHOW ELEMENT [**element-expression**]

Command Qualifiers	Defaults
/[NO]APPEND	/NOAPPEND
/BRIEF	See Qualifiers Glossary
/FORMAT= "string"	/NOFORMAT
/NOFORMAT	
/FULL	See Qualifiers Glossary
/INTERMEDIATE	See Qualifiers Glossary
/[NO]MEMBER	/NOMEMBER
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/OUTPUT[=file-specification]	/OUTPUT=SYS\$OUTPUT

Example

```
CMS> SHOW ELEMENT/FORMAT="$ CMS FETCH #E"/OUTPUT=FETCH.COM
```

This example produces a file named FETCH.COM that contains a FETCH command for each element in the library. The contents of a file produced by this command might look like the following example:

```
$ CMS FETCH INIT.FOR
$ CMS FETCH INITX.FOR
$ CMS FETCH MSGDOC.FOR
$ CMS FETCH OUTPUT.FOR
$ CMS FETCH SEARCH.FOR
```

SHOW GENERATION

SHOW GENERATION — Lists information about a specific generation of an element or a group of elements.

Format

SHOW GENERATION [**element-expression**]

Command Qualifiers	Defaults
/[NO]ANCESTORS	/NOANCESTORS
/[NO]APPEND	/NOAPPEND
/BEFORE=date-time	See Qualifiers Glossary
/BRIEF	See Qualifiers Glossary
/[NO]DESCENDANTS	/NODESCENDANTS
/FORMAT= "string"	/NOFORMAT
/FROM=generation-expression	/FROM=1
/FULL	See Qualifiers Glossary
/INTERMEDIATE	/See Qualifiers Glossary
/[NO]MEMBER	/NOMEMBER
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/OUTPUT[=file-specification]	/OUTPUT=SYS\$OUTPUT
/SINCE=date-time	See Qualifiers Glossary

This command displays the transaction record of the latest generation on the main line of descent. You can also use **SHOW GENERATION** to display the element name and generation number of all element generations belonging to a specified group or class.

The following restrictions apply to this command:

- You cannot specify the **/ANCESTORS** and **/DESCENDANTS** qualifiers on the same command line.
- You cannot specify the **/DESCENDANTS** and **/FROM** qualifiers on the same command line.
- If you specify the **/FROM** qualifier, you must also specify the **/ANCESTORS** qualifier on the same command line.

Example

```
1. CMS> SHOW GENERATION CCM.FOR
   Element generations in CMS Library DISKX:[TAYLOR.CMSLIB]
   CCM.FOR          2          7-DEC-2005 14:15:51 SMITH "header changed"
```

This example displays the element name, generation number, date, time, and remark associated with the latest main-line generation for element **CCM.FOR**.

```
2. CMS> SHOW GENERATION/GENERATION=RELEASE5 -
   _CMS> /FORMAT="$ CMS FETCH #E/GENERATION=#G"/OUTPUT=FETCH_CLASS.COM
```

Produces a file named **FETCH_CLASS.COM** that contains a **FETCH** command for each element belonging to the class named **RELEASE5**. The **FETCH** command retrieves the element of the correct generation from the **RELEASE5** class. The contents of a file produced by this command might look like the following:

```

$ CMS FETCH INIT.FOR/GENERATION=6
$ CMS FETCH OUTPUT.FOR/GENERATION=7
$ CMS FETCH SEARCH.FOR/GENERATION=3

CMS>SHOW GENERATION/MEMBER/DESCENDANTS TEST.FOR
Element generations in CMS Library
DISKX:[PROJECT.CMSLIB]
  TEST.FOR
    2      18-OCT-2005 15:04:54 SMITH "header changed"
    1      26-SEP-2005 13:38:04 SMITH ""
      Member list:          RELEASE1

```

This command displays the generation number, date, time, remark, and the name of any classes to which each generation belongs for all the generations of the element TEST.FOR.

SHOW GROUP

SHOW GROUP — Lists the names of all established groups in alphabetical order, along with the remark logged at the time each group was created or modified, in a CMS library.

Format

SHOW GROUP [group-expression]

Command Qualifiers	Defaults
/[NO]APPEND	/NOAPPEND
/BRIEF	See Qualifiers Glossary
/CONTENTS[=n]	/NOCONTENTS
/NOCONTENTS	
/FULL	See Qualifiers Glossary
/INTERMEDIATE	See Qualifiers Glossary
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/OUTPUT[=file-specification]	/OUTPUT=SYSS\$OUTPUT

Example

```

CMS> SHOW GROUP
Groups in CMS Library DISKX:[PROJECT.CMSLIB]
ARTWORK      "graph display"
OUT_DEFS     "output definition files"
TEST_ART     "tests for graphs"

```

This example displays the group name and creation remark associated with the group, by default. Because no group name is specified on the command line, CMS lists all groups in the library.

SHOW HISTORY

SHOW HISTORY — Displays a chronological list of transactions performed on a CMS library.

Format

SHOW HISTORY [**object-name-expression**]

Command Qualifiers	Defaults
/[NO]APPEND	/NOAPPEND
/BEFORE=date-time	See Qualifiers Glossary
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/OUTPUT[=file-specification]	/OUTPUT=SYS\$OUTPUT
/SINCE=date-time	See Qualifiers Glossary
/[NO]UNUSUAL	/NOUNUSUAL
/USER=username	See Qualifiers Glossary

You can limit the number of transactions displayed by specifying different parameters and qualifiers on the command. CMS records all transactions that update the library. The following transactions are not logged in the library history:

ANNOTATE	SET LIBRARY
DIFFERENCES	SHOW commands
FETCH (no remark)	VERIFY (no qualifiers)
RETRIEVE ARCHIVE	

The SHOW HISTORY command displays a chronological list of transactions. The transaction records are displayed in the following format:

```
date time username command remark
```

date

Displays the date the command was entered, in the following format:

```
[d]d-mmm-yyyy
```

For example:

```
9-JUN-2005
```

time

Displays the time the command was entered, in the following format:

```
hh:mm:ss
```

For example:

```
17:12:12 09:02:33
```

username

Displays the user name of the person who entered the command.

command

Identifies the command that was entered. The string “CMS” does not appear as part of the command. The command name, option name (if any), and any parameters are displayed. If the command operates on a particular generation of an element, the element name is followed by the generation number enclosed in parentheses; for example, SEMANTICS.PAS(3). Qualifiers that indicate some modification of the library are logged; for example, the /NAME qualifier with a MODIFY command.

“remark”

Specifies a character string that was logged in the history file with this command, and is usually used to explain why the command was entered. The remark is enclosed in quotation marks. If no remark was entered, a null remark (") is displayed.

A space separates each item in the transaction record.

The following example shows a transaction record:

```
11-AUG-2005 11:54:15 TAYLOR REPLACE SYNTAX.PAS(3)
    "RECORD and FOR loop syntax compatible"
```

This record shows that on August 11, 2005, at 11:54, user TAYLOR entered the REPLACE command on SYNTAX.PAS, which created generation 3.

Any command that causes CMS to record an unusual occurrence displays an asterisk (*) in the first column.

Because transaction records for elements, groups, or classes deleted from a library are kept (see DELETE ELEMENT, DELETE GROUP, and DELETE CLASS), SHOW HISTORY can display records for elements and classes that do not currently exist. If a deleted name is reused, SHOW HISTORY does not distinguish between the old and new histories.

SHOW LIBRARY

SHOW LIBRARY — Displays the directory specification of the current CMS library.

Format

SHOW LIBRARY

Command Qualifiers	Defaults
/[NO]APPEND	/NOAPPEND
/BRIEF	See Qualifiers Glossary
/FULL	See Qualifiers Glossary
/INTERMEDIATE	See Qualifiers Glossary

SHOW LIBRARY also checks whether the directory is a valid CMS library (unless you specify the /BRIEF qualifier). If you do not designate a CMS library for the current session (with SET LIBRARY or CREATE LIBRARY), this command issues an error message.

Example

```
CMS> SHOW LIBRARY
```

Your CMS library list consists of: `DISKX:[PROJECT.CMSLIB]`

This example displays the device and directory where the current CMS library is located.

SHOW RESERVATIONS

SHOW RESERVATIONS — Provides information about current reservations and concurrent replacements in effect at the time the command is issued.

Format

SHOW RESERVATIONS [**element-expression**]

Command Qualifiers	Defaults
/[NO]APPEND	/NOAPPEND
/BEFORE=date-time	See Qualifiers Glossary
/CONCURRENT	See Qualifiers Glossary
/NOREPLACEMENTS	See Qualifiers Glossary
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/OUTPUT[=file-specification]	/OUTPUT=SYS\$OUTPUT
/SINCE=date-time	See Qualifiers Glossary
/USER=username	See Qualifiers Glossary

For each element that has one or more generations reserved, the output consists of the element name and the following:

- For each reservation – identification number, user, reserved generation number, date, time, and remark
- For each concurrent replacement – user, replaced generation number, date, time, and remark

The elements are listed in alphabetical order.

SHOW REVIEWS_PENDING

SHOW REVIEWS_PENDING — Displays information about element generations that currently have review pending status.

Format

SHOW REVIEWS_PENDING [**element-expression**]

Command Qualifiers	Defaults
/[NO]APPEND	/NOAPPEND
/BEFORE=date-time	See Qualifiers Glossary

Command Qualifiers	Defaults
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/OUTPUT[=file-specification]	/OUTPUT=SYS\$OUTPUT
/SINCE=date-time	See Qualifiers Glossary
/USER=username	See Qualifiers Glossary

For each element with generations under review, the output consists of the element name, user, generation number, date, time, replacement remark, and any associated review comments that were established with the REVIEW GENERATION command. The elements are listed in alphabetical order.

SHOW VERSION

SHOW VERSION — Displays the version number of the current CMS system.

Format

SHOW VERSION

Command Qualifiers	Defaults
/[NO]APPEND	/NOAPPEND
/OUTPUT[=file-specification]	/OUTPUT=SYS\$OUTPUT

UNRESERVE

UNRESERVE — Cancels one or more existing reservations of a generation of an element.

Format

UNRESERVE element-expression "remark"

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/IDENTIFICATION_NUMBER=n	See Qualifiers Glossary
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL

You cannot unreserve a generation held by another user unless you hold BYPASS privilege, or are granted BYPASS access to the element by an access control entry (see the *VSI DECset for OpenVMS Guide to the Code Management System*).

If you have more than one reservation of an element, or if you are replacing a concurrent reservation made by another user (that is, if there is any ambiguity), you must specify the exact reservation to be canceled (see the *VSI DECset for OpenVMS Guide to the Code Management System*). Use either the /GENERATION qualifier or the /IDENTIFICATION_NUMBER qualifier.

You can use /GENERATION as long as the concurrent reservations are not on the same generation. If you have more than one concurrent reservation for the same generation, you must identify the specific reservation to be canceled. Each reservation is assigned an identification number. Use the SHOW RESERVATIONS command to determine the identification number of each reservation.

The identification number appears in parentheses at the beginning of each line. If you use the /IDENTIFICATION_NUMBER qualifier, you do not need to use the /GENERATION qualifier. CMS ignores the /GENERATION qualifier when both are used.

The /NOCONFIRM qualifier, which is the default, does not override the confirmation prompt issued when you make a concurrent replacement or replace another user's reservation.

Example

```
$ SET PROCESS/PRIVILEGE=BYPASS
$ CMS UNRESERVE SAMPLE.TXT "unreserving on behalf of SMITH"
Element A.TXT currently reserved by:
(1) SMITH 1 17-FEB-2005 13:43:52 "sample module for program"
Unreserve (1) SAMPLE.TXT generation 1, held by SMITH? [Y/N] (N): Y
%CMS-S-UNRESERVED, element DISKX:[SMITH.WORK.CMSLIB]SAMPLE.TXT unreserved
```

This example shows the cancellation of another user's existing reservation (by using the BYPASS privilege) of the latest generation of the element SAMPLE.TXT. If concurrent reservations of element SAMPLE.TXT exist, you need to specify the exact reservation to be canceled by using either the /GENERATION or /IDENTIFICATION qualifier.

VERIFY

VERIFY — Performs a series of consistency checks on your library.

Format

VERIFY [**element-expression**]

Command Qualifiers	Defaults
/[NO]CONFIRM	/CONFIRM
/[NO]CONTROL	/NOCONTROL
/[NO]LOG	/LOG
/OCCLUDE[=option,...]	/OCCLUDE=ALL
/[NO]RECOVER	/NORECOVER
/[NO]REPAIR	/NOREPAIR

If you enter VERIFY under normal conditions, the command executes successfully, indicating that the information in your library is correct. If the data in the library is invalid, the command returns an error message indicating that the verification of the library contains an error.

Use the /CONTROL qualifier to verify that the CMS library control file is valid. Use the VERIFY or VERIFY/NOCONTROL command to confirm that CMS libraries are fully valid. Use the /RECOVER and /REPAIR qualifiers of the VERIFY command to correct some of the errors discovered by VERIFY.

Recovery and repair transactions are marked as unusual occurrences in the library history. For more information about the VERIFY transaction, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

You cannot do the following:

- Specify the /RECOVER and /REPAIR qualifiers on the same command line.
- Specify an element expression parameter when you use the /RECOVER qualifier.

Chapter 2. Parameter and Qualifier Glossary

This glossary contains two parts: a list of command parameters and a list of command qualifiers. Many of the commands described in the CMS Command Dictionary use the same parameters and qualifiers as other commands. Their descriptions in this glossary help to complete the information already presented in the CMS Command Dictionary section.

2.1. CMS Command Parameters

This glossary lists the CMS command parameters in alphabetical order.

Glossary

class-expression	Specifies: • The class (or classes) to be deleted from the CMS library. See the DELETE CLASS command. • An established class into which the element generation is being placed. The class must not have the read-only attribute. See the INSERT GENERATION and REPLACE commands. • The class being modified. See the MODIFY CLASS command. • The class from which the element generation is to be removed. This class must not have read-only access. See the REMOVE GENERATION command. • One or more classes to be listed. If you do not specify a class name, CMS lists all the classes in the library. See the SHOW CLASS command. A class expression can be a class name, a wildcard, expression, or a list of these separated by commas.
class-name	Specifies a name for the class. CMS reports an error if you specify a name that is already used for an existing class or group. Class and group names must be unique. If a previously used class or group name has been removed with the DELETE CLASS or DELETE GROUP command, you can reuse that name with CREATE CLASS. Wildcards are not allowed.
command /qualifier	Specifies a CMS command with an appropriate qualifier. For example: <pre>\$ HELP CMS DIFFERENCES/PARALLEL</pre> This command gives you help at DCL level on the PARALLEL qualifier on the DIFFERENCES command.
command option	Specifies a CMS command with an appropriate option. For example:

\$ **HELP CMS SHOW ELEMENT**
\$ **HELP CMS CREATE CLASS**

These commands give you help at DCL level on the SHOW ELEMENT and CREATE CLASS commands.

command option/qualifier

Specifies a CMS command with an option and a qualifier. For example:

\$ **HELP CMS CREATE ELEMENT/RESERVE**

This command gives you help at DCL level on the /RESERVE qualifier on the CREATE ELEMENT command.

command

Gives information about CMS either at DCL level or at CMS subsystem level. At DCL level, the DCL command HELP CMS provides online help on CMS commands, qualifiers, and other topics. For example:

\$ **HELP CMS**

To get help on a specific CMS command, such as the CREATE ELEMENT command, type the command after HELP CMS. For example:

\$ **HELP CMS CREATE ELEMENT**

You can get help at the CMS subsystem level by typing either HELP or HELP and the specific command. For example:

CMS> **HELP CREATE ELEMENT**

directory-specification

Specifies:

- One or more valid OpenVMS directories. Each directory must not contain any files. A directory to be used as a CMS library cannot be your current default directory. See the CREATE LIBRARY command.
- One or more existing CMS libraries. The directory used as the CMS library cannot be your current, default directory. The directory specification must conform to OpenVMS conventions; it can also be a logical name. See the SET LIBRARY command.
- One or more existing CMS libraries in the current library search list. The directory specification must conform to OpenVMS conventions; it can also be a logical name that translates to a search list. If you do not supply a directory specification, CMS removes all libraries from the current library search list. See the SET NOLIBRARY command.

If you specify more than one OpenVMS directory, you must separate the directory specifications with commas. Wildcards are not allowed.

element-expression

An element expression can be an element name, group name, wildcard expression, or a list of these separated by commas. This parameter specifies:

- One or more elements. See the ACCEPT GENERATION, REVIEW GENERATION, or CANCEL REVIEW command.
- One or more elements. For each element, CMS lists the transaction record of the most recent generation of the main line of descent. If you do not supply an element expression, CMS lists a transaction record for each element in the library. See the SHOW GENERATION command.
- One or more elements to be created. If you do not specify the /INPUT qualifier (or if you specify /INPUT without a value), the element name must correspond to an existing file in your current directory. The name cannot be the same as any existing element name in the library. Do not use the file name 00CMS, which is reserved for library control files. Generation 1 of the new element is created. An element expression can also be a list of element names separated by commas or a wildcard expression. See the CREATE ELEMENT command.
- One or more elements to be deleted from the library. See the DELETE ELEMENT command.
- One or more generations of an element. See the DELETE GENERATION command.
- One or more elements. If you specify a group name, CMS annotates each element in the group. If you use wildcards, CMS produces one annotated listing file for each matching element. By default, the most recent generation of an element on the main line of descent is annotated. See the ANNOTATE command.
- One or more generations of an element to be retrieved from the library. By default, CMS fetches the most recent generation on the main line of descent. See the FETCH command.
- One or more elements to be inserted into the group. See the INSERT ELEMENT command.
- One or more elements whose generations are to be inserted into the class. By default, the most recent generation on the main line of descent is inserted. See the INSERT GENERATION command.
- One or more elements or groups of elements whose generations are to be marked with pending review status. See the MARK GENERATION command.
- One or more elements or elements whose generations are to be modified. See the MODIFY ELEMENT or MODIFY GENERATION command.

- One or more elements or elements whose reservation remarks are to be modified. See the **MODIFY RESERVATION** command.
- One or more elements whose generations are to be rejected. See the **REJECT GENERATION** command.
- One or more elements to be removed from one or more groups. When you use the wildcard characters in the element expression, **/IF_PRESENT** is the default. (CMS does not return an error message if this group does not contain the element being removed.) See the **REMOVE ELEMENT** command.
- One or more generations of elements to be removed from one or more classes. When you use the wildcard characters in the element expression, **/IF_PRESENT** is the default. (CMS does not return an error message if the class does not contain a generation of the element.) See the **REMOVE GENERATION** command.
- One or more reserved generations of an element to be replaced. If you specify more than one element (with either a group name or wildcard expression), each file indicated by the element expression must exist in the same directory. When you use wildcards, CMS creates an input element list based on the list of element generations you have reserved. See the **REPLACE** command.
- The element (or elements) from which a generation is to be reserved. By default, CMS reserves the most recent generation on the main line of descent of each element designated by the element-expression. See the **RESERVE** command.
- One or more elements to be listed. If you do not supply an element expression, CMS lists all the elements in the library. See the **SHOW ELEMENT** command.
- One or more elements for which CMS displays current reservation and concurrent replacement information. If you do not specify an element expression, information is displayed for all elements. See the **SHOW RESERVATIONS** command.
- One or more elements to be searched for generations with pending reviews. By default, all elements are searched. See the **SHOW REVIEWS_PENDING** command.
- One or more elements whose reservations are to be canceled. See the **UNRESERVE** command.
- One or more elements to be verified. If you do not supply an element expression, CMS verifies every file in the library. You cannot specify an element expression parameter if you use the **/RECOVER** qualifier. See the **VERIFY** command.

file-expression

Specifies the name of one or more archive files. A file expression can be a filename.type specification, wildcard expression, or a list of these separated by commas.

file1	Specifies the first file to be compared. If you specify the / GENERATION qualifier, this parameter must be an element name; otherwise, an OpenVMS file specification is assumed.
file2	Specifies the second file to be compared. If you specify the / GENERATION qualifier, this parameter must be an element name; otherwise, an OpenVMS file specification is assumed. CMS follows these rules when you do not provide a second file specification: ● If you direct CMS to take <i>file1</i> from a location that is not a CMS library, CMS uses the next lower file version in the same directory as <i>file1</i>. ● If you direct CMS to take <i>file1</i> from a CMS library (by specifying / GENERATION), CMS uses the latest default directory version of <i>file1</i> as the second input file.
generation-expression	Used with the /GENERATION qualifier, this value specifies a particular generation of an element. The first generation of an element is generation 1, and subsequent generations are numbered sequentially by adding 1 to the prior generation number. If you do not specify a generation-expression value, or you do not use the /GENERATION qualifier at all, the default generation value of 1+ is used (the latest generation).
group-expression	Specifies: ● The group (or groups) to be deleted. See the DELETE GROUP command. ● One or more groups into which the element (or elements) is to be inserted. See the INSERT ELEMENT command. ● The group into which the subgroup expression is to be inserted. See the INSERT GROUP command. ● The group to be modified. See the MODIFY GROUP command. ● The group from which one or more elements are to be removed. See the REMOVE ELEMENT command. ● The group to be listed. If you do not supply a group expression, CMS lists all the groups in the library. See the SHOW GROUP command. A group expression can be one or more group names, a wildcard expression, or a list of these separated by commas.
group-expression1	Specifies one or more groups to be removed. Wildcards and a comma list are allowed. When you use wildcard characters or a comma list in the group name, /IF_PRESENT is the default. (CMS does not return an error message if <i>group-expression2</i> does not contain <i>group-expression1</i> .)

group-expression2	Specifies one or more groups from which the groups in <i>group-expression1</i> are to be removed. Wildcards and a comma list are allowed.
group-name	Specifies a name for the group. CMS reports an error if you specify an existing group or class name. (Group and class names must be unique.) However, if a previously used group or class name has been removed with the DELETE GROUP or DELETE CLASS command, you can reuse that name with CREATE GROUP. A group name can also be a list of group names separated by commas. Wildcards are not allowed.
new-class-name	Specifies the name for the new class. The name cannot be the same as any existing class name in the target library. Wildcards are allowed. If you specify more than one class with COPY CLASS, you must use a wildcard character for the new class name.
new-element-name	Specifies the name for the new element. The name cannot be the same as any existing element name in the target library. The file-name component cannot be 00CMS because this name is reserved for CMS. Wildcards are allowed. If you specify more than one element with COPY ELEMENT, you must use a wildcard character for the new element name.
new-group-name	Specifies the name for the new group. The name cannot be the same as any existing group name in the target library. Wildcards are allowed. If you specify more than one group with COPY GROUP, you must use a wildcard character for the new group name.
object-expression	Specifies: ● One or more objects whose ACLs are to be modified. See the SET ACL command. ● The CMS library object whose ACL is to be displayed. See the SHOW ACL command. Wildcards and a comma list are allowed. The object expression depends on the object type (see the / OBJECT_TYPE qualifier). For example, if the object type is CLASS, the object expression must be the name of a class in the CMS library. The same principle applies to elements and groups. However, if the object type is LIBRARY, the object expression must be one or more of the following keywords: ELEMENT_LIST CLASS_LIST GROUP_LIST HISTORY LIBRARY_ATTRIBUTES These keywords are referred to as object subtypes. You can abbreviate object subtypes. Wildcards are not allowed.

The object name can also be the name of a CMS command. If /OBJECT_TYPE is specified as COMMAND, one of the following occurs depending on the associated command:

- SET ACL modifies the ACL on the given command. Commands that contain two words must be specified with an underscore, for example, INSERT_ELEMENT.
- SHOW ACL displays the ACL for the given command.

object-name-expression	Specifies an object whose history information is to be either displayed or delete. With the SHOW HISTORY and DELETE HISTORY commands, the object name expression can be one or more class names, element names, group names, a wildcard expression, or a list of any of these separated by commas. (If you do not specify an object name expression, CMS lists or deletes the transaction records for all classes, elements, and groups in the library.)
old-class-expression	Specifies one or more existing classes to be copied. If you specify more than one class to be copied, you must use a wildcard character for the new class name. An old class expression can be a class name, a wildcard expression, or a list of these separated by commas.
old-element-expression	Specifies one or more existing elements to be copied. If you specify more than one element to be copied, you must use a wildcard character for the new element name. An old element expression can be an element name, a group name, a wildcard expression, or a list of these separated by commas.
old-group-expression	Specifies one or more existing groups to be copied. If you specify more than one group to be copied, you must use a wildcard character for the new group name. An old group expression can be a group name, a wildcard expression, or a list of these separated by commas.
“remark”	Specifies a character string for the creation remark of a new element, a class, an element, a group, a new library, a specified element generation, newly created generations, or the reservation to be logged in the history file with the CMS commands. With the CREATE ELEMENT command, the remark is stored with both the element and its first generation. If no remark was entered with the COPY ELEMENT command, the remark from the old element is used for the creation remark of the new element, but a null remark (") is logged in the history file. This parameter also specifies a character string or string to be logged in the history file with a CMS command, which explains why the command was entered. This parameter and the string on the /REMARK qualifier are unrelated for the following CMS commands: ● MODIFY GROUP

- MODIFY ELEMENT
- MODIFY CLASS
- MODIFY GENERATION
- MODIFY RESERVATION

These remarks are enclosed in quotation remarks. If no remark is entered, a null remark ("") is logged.

subgroup-expression	Specifies one or more groups to be inserted into a second group (indicated by group-expression). A subgroup expression can be a group name, a wildcard expression, or a list of these separated by commas.
topic	Specifies a subject related to CMS. For example, help on the topic OVERVIEW consists of general information on CMS and pointers to other topics that would be of interest to new users. Help on CLASSES defines the concept of a class and points to help on commands that manipulate classes.
V2-library-directory-specification	Specifies the directory specification of the existing CMS library you want to convert.
V4-library-directory-specification	Specifies the directory specification of the new CMS library you want to create. This directory must be empty.

2.2. CMS Command Qualifiers

This glossary lists the CMS command qualifiers in alphabetical order.

Glossary

/ACL[=(ace[,...])]	Specifies one or more ACEs to be modified. When no ACE is specified, the entire ACL is affected. Separate multiple ACEs with commas and enclose the list in parentheses. The specified ACEs are inserted at the beginning of the ACL unless the /AFTER qualifier is used.
/AFTER=ace	With the SET ACL command, this qualifier indicates that all ACEs specified with the /ACL qualifier are added after the ACE specified with the /AFTER qualifier. By default, any ACEs added to the ACL are placed at the top of the list.
/AFTER=[directory-specification]	With the SET LIBRARY command, this qualifier instructs CMS to insert new libraries into the existing library search list (that you previously specified by using a comma list with the CREATE LIBRARY or SET LIBRARY command) immediately following the existing specified directory. If you omit the directory specification, CMS adds the libraries to the end of the list. You cannot specify both /AFTER and /BEFORE on the same command line. By default, the SET LIBRARY command's library list supersedes any existing search list.

	With the CREATE LIBRARY command, this qualifier instructs CMS to insert new libraries into the existing library search list immediately following the existing specified directory. The specified directory must be in the existing library search list. If you omit the directory specification, CMS automatically adds the libraries (in the order you specify) to the end of the list. You cannot specify both /AFTER and /BEFORE on the same command line. If neither /AFTER nor /BEFORE is specified, the CREATE LIBRARY command's library list supersedes any existing search list.
/AFTER=generation-expression	With the DELETE GENERATION command, this qualifier specifies the start of a range of generations that are to be deleted, <i>excluding</i> the specified generation. You cannot specify both /AFTER and /FROM, or both /AFTER and /GENERATION. You must specify the end of the range with either the /BEFORE or /TO qualifier.
/ALWAYS	Directs CMS to insert the element generation into the class in all cases. If the class contains a generation from the specified element, that generation is removed before the new one is inserted.
/ANCESTORS /NOANCESTORS (D)	Displays the transaction records of the specified element generation and all its ancestors. The transaction records are listed in reverse chronological order. If you do not specify a particular generation, the list begins with the latest generation on the main line of descent. The ancestors of a main-line generation are all the preceding generations back to the first generation of an element. The ancestors of a variant-line generation are all preceding generations on the variant line of descent, and any generations back to the first generation on the main line.
/APPEND /NOAPPEND (D)	Controls whether CMS appends the command output to an existing file, or creates a new file. If you specify /APPEND and the output file does not exist, CMS creates a new file. If you do not provide a file specification (see the description for /OUTPUT), the output is appended to SYS\$OUTPUT. With the ANNOTATE command, this qualifier controls whether CMS appends the history and source file listing to an existing file, or creates a new file. If you specify /APPEND and the output file does not exist, CMS creates a new file. If you do not provide an output file specification (see the description for /OUTPUT), CMS searches your default directory for a file with the element file name and the file type .ANN. With the DIFFERENCES command, this qualifier controls whether CMS appends the command output to an existing file, or creates a new file. If you specify /APPEND and the output file does not exist, CMS creates a new file. If you do not provide an output file specification (see the description for /OUTPUT), CMS searches your current, default directory for a file with the file name specified in the <i>file1</i> or <i>class-name1</i> parameter and the file type .DIF.

<code>/ARCHIVE[=file-specification]</code> <code>/NOARCHIVE (D)</code>	Specifies a file to which CMS writes archived generation information. If the file specification is omitted, CMS creates a file with the same name as each element, assigns a file type of <code>.CMS_ARCHIVE</code> , and places it in your default directory.
<code>/BEFORE=date-time</code>	With the <code>INSERT GENERATION</code> command, this qualifier inserts the element generation based on the <code>/GENERATION</code> qualifier or the latest generation on the main line of descent if there is no qualifier. The generation inserted is the latest ancestor of the generation determined above that pre-dates the specified time. If no generation pre-dates that time (i.e. generation 1 is later) then no generation is inserted. The time value can be an absolute, delta, or combination time value, or one of the following keywords: <code>TODAY</code>, <code>TOMORROW</code>, or <code>YESTERDAY</code>. If the time value is a future value, CMS uses the current time. If no time value is supplied, CMS used <code>TODAY</code>. With the <code>SHOW HISTORY</code> command, this qualifier lists all history information before a specified date. By default, the time is the current date and time. The time value can be an absolute, delta, or combination time value, or one of the following keywords: <code>TODAY</code>, <code>TOMORROW</code>, or <code>YESTERDAY</code>. With the <code>SHOW REVIEWS_PENDING</code> command, this qualifier specifies that only generations placed under review before the indicated date and time are to be displayed. By default, all generations with reviews pending are displayed. The date value can be an absolute, delta, or combination time value, or one of the following keywords: <code>TODAY</code>, <code>TOMORROW</code>, or <code>YESTERDAY</code>. With the <code>SHOW RESERVATIONS</code> or <code>SHOW GENERATION</code> command, this qualifier lists the current reservations or generations that were executed prior to the specified time. The date value can be an absolute, delta, or combination time value, or one of the following keywords: <code>TODAY</code>, <code>TOMORROW</code>, or <code>YESTERDAY</code>. By default, CMS displays all existing reservations or generations.
<code>/BEFORE=date-time</code> <code>/BEFORE=current-time (D)</code>	With the <code>DELETE HISTORY</code> command, deletes all the history information before a specified time. A single entry is made in the history file specifying that a section of the history data has been removed. This entry is made at the location in the history file where the lines were deleted. The time value can be an absolute, delta, or combination time value, or one of the following keywords: <code>TODAY</code> , <code>TOMORROW</code> , or <code>YESTERDAY</code> . If the time value is a future value, CMS uses the current time.
<code>/BEFORE[=directory-specification]</code>	With the <code>SET LIBRARY</code> command, this qualifier instructs CMS to insert new libraries into the existing library search list (that you previously specified by using a comma list with the <code>CREATE LIBRARY</code> or <code>SET LIBRARY</code> command) immediately in front of the existing specified directory. If you omit the directory specification, CMS adds the libraries to the front of the list. You cannot specify both <code>/AFTER</code> and <code>/BEFORE</code> on the same command line. By default, the <code>SET LIBRARY</code> command's library list supersedes any existing search list.

With the CREATE LIBRARY command, this qualifier instructs CMS to insert new libraries into the existing library search list immediately in front of the existing specified directory. The specified directory must be in the existing library search list. If you omit the directory specification, CMS automatically adds the libraries (in the order you specify) to the front of the list. You cannot specify both /AFTER and /BEFORE on the same command line. If neither /AFTER nor /BEFORE is specified, the CREATE LIBRARY command's library list supersedes any existing search list.

/BEFORE=generation-expression

With the DELETE GENERATION command, specifies the end of a range of generations that are to be deleted, *excluding* the specified generation. You cannot specify both /BEFORE and /TO, or both /BEFORE and /GENERATION. You must specify the start of the range with either the /AFTER or /FROM qualifier. A generation-expression value must be entered.

/BRIEF
/FULL
/INTERMEDIATE (D)

The /BRIEF qualifier displays:

- The name of the element, the generations archived into this file, the name of the person who archived the file, the date and time, the remark entered into the DELETE GENERATION command, and the name of the library in which the original element resided. See the SHOW ARCHIVE command.
- Class names. See the SHOW CLASS command.
- Element names. See the SHOW ELEMENT command.
- Only the element names and generation numbers for each specified generation. See the SHOW GENERATION command.
- Group names. See the SHOW GROUP command.
- The directory specification. It does not lock the library, even if the default library is locked by another user. However, if the library needs recovery, it is not detected until another operation is performed. See the SHOW LIBRARY command.

The /FULL qualifier displays:

- Complete generation file information for each archived generation (see the SHOW ARCHIVE command).
- The name, creation remark, and **read-only** attribute (if established) for each class or group (see the SHOW CLASS or SHOW GROUP command).
- The name, creation remark, and the attributes in effect for the specified elements (see the SHOW ELEMENT command).
- Standard CMS transaction information, the element, generation number, date, time, user, and remark, and also produces information about the file creation date and time, file revision

number, and record format and attributes (see the SHOW GENERATION command).

- The current library directory specification, number of elements, groups, classes, reservations, concurrent replacements, reviews pending, and the current reference copy directory specification, if any (see the SHOW LIBRARY command).

The /INTERMEDIATE qualifier displays:

- Generation history for the archived generations (see the SHOW ARCHIVE command).
- Name and creation remark for each class or group (see the SHOW CLASS or SHOW GROUP command).
- Name and creation remark associated with the element (see the SHOW ELEMENT command).
- Element name, generation number, date, time, user, and remark associated with the transaction that created the generations (see the SHOW GENERATION command).
- Directory specification of the current CMS library (see the SHOW LIBRARY command).

/CLASS

With the DIFFERENCES command, compares the contents of two classes. If the particular generation of any element differs between the classes, CMS creates a file that contains a listing of the differences in members. If the classes are identical, CMS issues a message to that effect and does not create a differences file. The /CLASS qualifier must be specified prior to any other qualifiers.

/CONCURRENT (D) /NOCONCURRENT

With the CREATE ELEMENT and CREATE LIBRARY commands, specifies whether this element or library can have multiple reservations. After you create the element or library, you can change concurrent access by using the MODIFY ELEMENT and MODIFY LIBRARY commands. With the SHOW GENERATIONS command, lists only concurrent reservations and concurrent replacements. The use of this qualifier overrides any library-wide setting.

/CONCURRENT

With the SHOW RESERVATIONS command, lists only concurrent reservations and concurrent replacements (concurrent replacements are not displayed if you also specify the /NOREPLACEMENTS qualifier). The /CONCURRENT qualifier is the default for the SHOW RESERVATIONS command, and is not negatable.

/[NO]CONCURRENT

With the RESERVE command, specifies whether this element can have multiple reservations. If you do not specify this qualifier, the existing concurrent access is not changed. The use of this qualifier overrides any library-wide setting.

/CONFIRM (D) /NOCONFIRM

Controls whether CMS prompts you for confirmation before each transaction.

With the VERIFY command, this qualifier controls whether CMS prompts you for confirmation before deleting any invalid reference copies during a VERIFY/REPAIR operation. In some cases, CMS might find invalid reference copies (for example, if there is one valid reference copy and the remaining reference copies are invalid), so it automatically deletes the invalid copies without prompting for confirmation.

When you run CMS in interactive mode, with or without the /CONFIRM qualifier, type YES, ALL, TRUE, or 1, to execute the transaction. Type NO, QUIT, FALSE, 0, or press Return or Ctrl/Z to halt the transaction. If you type any other character, CMS continues to prompt until you type an acceptable response. See the DELETE and COPY commands for more information. CMS does not prompt for confirmation in batch mode.

With the UNRESERVE command, the /NOCONFIRM qualifier does not override the confirmation prompt issued when you make a concurrent replacement, or you replace another user's reservation.

With the RESERVE command, you cannot use /NOCONFIRM to override the prompt generated when you try to reserve a generation of an element that is already reserved by you or by another user.

With the REPLACE command, the /NOCONFIRM qualifier does not override the confirmation prompt issued when you make a concurrent replacement, or when you cancel another user's reservation.

/CONTENTS[=*n*]
/NOCONTENTS (D)

With the SHOW GROUP command, this qualifier lists the group name and creation remark, along with the names of any elements or groups contained within the specified group.

You can specify an integer value (*n*) that directs CMS to display nested groups up or down to and including the level indicated by *n*. When you specify /CONTENTS without a value, CMS displays one level of contents. You can also specify /CONTENTS=ALL to display all levels of contained groups.

With the SHOW CLASS command, this qualifier controls whether CMS identifies the element generations belonging to each class. If you specify /CONTENTS, CMS displays the class name and creation remark, along with the element name and generation number for each generation in the class.

/CONTROL
/NOCONTROL (D)

With the VERIFY command, this qualifier verifies that the CMS library control file is valid. Use the VERIFY or VERIFY/NOCONTROL command to confirm that CMS libraries are fully valid.

/CREATE_DIRECTORY
/NOCREATE_DIRECTORY
(D)

With the CREATE LIBRARY command, this qualifier creates a directory to contain the library. A CMS library must reside in a directory that has been initialized for use solely by CMS.

/DEFAULT

Creates an ACL for one or more specified objects as if the object were newly created. The /DEFAULT qualifier propagates the DEFAULT

option ACEs in the ACL of the entity list to the ACL of the specified object. You can only use this qualifier with an object that is a library entity (an element, class, or group).

/DELETE

Indicates that the ACEs specified with the /ACL qualifier are to be deleted. If no ACEs are specified with the /ACL qualifier, the entire ACL is deleted. If the /ACL qualifier specifies an ACE that does not exist in the ACL of the specified object, you are notified that the ACE does not exist, and the delete operation continues on to the next ACE on the ACL, if any exists.

/DELETE[=file-specification] /NODELETE (D)

Controls whether all versions of the unreserved file are deleted. If you omit /DELETE, the files are not deleted. If you specify /DELETE without a file specification, all versions of the unreserved files are deleted from your current directory. The file specification allows you to specify a different location and the name of the file to be deleted.

/DESCENDANTS /NODESCENDANTS (D)

Displays the transaction records of the specified element generation and of all its descendants. The transaction records are listed in reverse chronological order. If you do not specify a particular generation, the list begins with generation 1.

The descendants of a generation consist of all the successor generations, including those on variant lines of descent. You can use this command to determine whether any variant lines of descent exist for a particular element.

/EXTENDED_FILENAMES /NOEXTENDED_FILENAMES (D)

With the CREATE LIBRARY and MODIFY LIBRARY commands, this qualifier specifies whether extended file names will be allowed. This qualifier only applies to CMS libraries on OpenVMS systems that support extended file names.)

/FORMAT=(data-format,data-partition) /FORMAT=(ASCII,RECORDS) (D)

Controls whether the history and source file listing is formatted, and specifies the type of formatting. You can specify one or both formatting parameters in any order.

data-format

Specifies the type of format. The following table lists the possible values for data formats.

Data Format Option	Action
ASCII (D)	Specifies that each byte of the data be displayed as an ASCII character. This option is most useful when files contain textual data. If no data partition is specified, data is partitioned into records. This option is the default.
DECIMAL	Specifies that each value be displayed as a decimal numeral. If no data partition is specified, data

Data Format Option	Action
	is partitioned into longwords. You cannot specify both DECIMAL and RECORDS.
HEXADECIMAL	Specifies that each value be displayed as a hexadecimal numeral. If no data partition is specified, data is partitioned into longwords. You cannot specify both HEXADECIMAL and RECORDS.
OCTAL	Specifies that each value be displayed as an octal numeral. If no data partition is specified, data is partitioned into longwords. You cannot specify both OCTAL and RECORDS.

data-partition

Specifies the type of data partition. A data partition is the size that data in each record is broken into before it is formatted. The following table lists the possible values for data partitions.

Data Partition Option	Action
BYTE	Specifies that the data displayed be partitioned into bytes.
LONGWORD	Specifies that the data displayed be partitioned into longword values. This is the default partitioning for DECIMAL, HEXADECIMAL, or OCTAL.
RECORDS (D	Specifies that no further partitioning of data is to occur beyond the record partitioning already in the file. This partitioning is most useful when the files contain textual data. You can only specify RECORDS by itself or in conjunction with ASCII. It is mutually exclusive with all other options. This value is the default.
WORD	Specifies that the data displayed be partitioned into word values.

/FORMAT=(data-format,data-partition,[no]generation-differences)

Controls whether the output file is formatted, specifies the type of formatting, and controls whether a list of generation differences

is included in the DIFFERENCES output. You can specify the parameters in any order.

data-format

Specifies the type of format. The following table lists the possible values for data formats.

Data Format Option	Action
ASCII (D)	Specifies that data be presented as if each byte represents a value in the ASCII character set. This option is most useful when files contain textual data. If no data partition is specified, data is partitioned into records. This option is the default.
DECIMAL	Specifies that each value be displayed as a decimal numeral. If no data partition is specified, data is partitioned into longwords. You cannot specify both DECIMAL and RECORDS.
HEXADECIMAL	Specifies that each value be displayed as a hexadecimal numeral. If no data partition is specified, data is partitioned into longwords. You cannot specify both HEXADECIMAL and RECORDS.
OCTAL	Specifies that each value be displayed as an octal numeral. If no data partition is specified, data is partitioned into longwords. You cannot specify both OCTAL and RECORDS.

data-partition

Specifies the type of data partition. A data partition is the size that data in each record is broken into before it is formatted. The following table lists the possible values for data partitions.

Data Partition Option	Action
BYTE	Specifies that the data displayed be partitioned into bytes. Records are not partitioned further unless the data-format option indicates otherwise.

Data Partition Option	Action
LONGWORD	Specifies that the data displayed be partitioned into longword values. This is the default partitioning for DECIMAL, HEXADECIMAL, or OCTAL.
RECORDS (D)	Specifies that no further partitioning of data is to occur beyond the record partitioning already in the file. This partitioning is most useful when the files contain textual data. You can specify RECORDS only by itself or in conjunction with ASCII. It is mutually exclusive with all other options. This value is the default.
WORD	Specifies that the data displayed be partitioned into word values. Data records are not partitioned further unless the data format indicates otherwise.

generation-differences

Specifies whether a list of generation differences is to be included in the DIFFERENCES output. This option is applicable only if two element generations are compared by the DIFFERENCES command. In any other case, this option is ignored. The following table lists the available keywords for the /GENERATION parameter qualifier.

Generation Differences Option	Action
GENERATION_DIFFERENCES	Specifies that a list of differences is to be included in the output
NOGENERATION_DIFFERENCES (D)	Specifies that no list of differences is included in the output

/FORMAT= "string"
/NOFORMAT (D)

Controls whether the output of the SHOW ELEMENT or SHOW GENERATION command is formatted. You can use the /FORMAT qualifier in combination with the /OUTPUT qualifier to set up a command file. With this command file, you can execute a CMS command or a DCL command on a specified set of elements (such as all the elements in a group or class).

The format string can contain printing characters; within the format string, CMS recognizes #E (and #e) as the element format parameter, and #G (and #g) as the generation number format parameter. For each line of output (one line per generation or element), CMS displays the format string and replaces each occurrence of #E (or #e) or #G (or #g)

with the element name or generation number, respectively. To include a number sign in the output line, type it twice (##). When you specify the /FORMAT qualifier, CMS does not generate the heading normally produced by the SHOW ELEMENT or SHOW GENERATION command.

To set up a command file, you specify a format string consisting of a command, including the dollar sign (\$) prompt and one of the following parameters:

- An element format parameter (for example, /FORMAT="\$CMSFETCH #E")
- A format parameter, #E or #G (for example, /FORMAT="\$CMSFETCH #E/GENERATION=#G")

When you execute the SHOW ELEMENT or SHOW GENERATION command with these qualifiers, CMS creates a command file containing a list of FETCH commands that use each element in the denoted set as parameters.

/FROM=generation-expression
/FROM=1 (D)

With the SHOW GENERATION command, this qualifier specifies the generation that begins the list of ancestors. There is no default; a value is required. You must specify the /ANCESTORS and /FROM qualifiers on the same command.

With the DELETE GENERATION command, this qualifier specifies the start of a range of generations that are to be deleted, *including* the specified generation. You cannot specify both /FROM and /AFTER, or both /FROM and /GENERATION. You must specify the end of the range with either the /BEFORE or /TO qualifier.

/FULL

If used with the DIFFERENCES command, directs CMS to include the following information about the file used to create each generation:

- Creation time
- Revision time
- Revision number
- Record format
- Record attributes

CMS also indicates deleted lines in the source listing. Each set of one or more deleted lines is identified by a count of the deleted lines.

If used with the DIFFERENCES/CLASS command, CMS displays a complete listing of members for each class being compared (including identical and differing generations).

/GENERATION=generation-expression

Specifies which reserved generation of the element is to be replaced. If you have more than one reservation of the same element generation,

you must use the /IDENTIFICATION_NUMBER qualifier to replace the reservation.

You can specify a generation indirectly by using a class name, the plus operator, the semicolon, or relative generation offsets. See the *VSI DECset for OpenVMS Guide to the Code Management System*.

/
GENERATION[=generation-
expression]

Enables you to specify a generation indirectly by using a classname, the plus operator, the semicolon, or relative generation offsets. See the *VSI DECset for OpenVMS Guide to the Code Management System*.

This qualifier is used with different commands to supply the following information:

- A particular generation of the element to be rejected. If you omit /GENERATION, CMS rejects the most recently created generation with a review pending. You specify this qualifier only if more than one generation of an element is under review. See the REJECT GENERATION command.
- Which generation of the element is to have its review pending status canceled. If you omit /GENERATION, CMS cancels the review of the most recently created generation with a review pending. Specify this qualifier only if more than one generation of an element is under review. See the CANCEL REVIEW command.
- Which reserved generation of an element is to be modified. If more than one reservation exists, you must use the /IDENTIFICATION_NUMBER qualifier to identify the reservation to modify. See the MODIFY RESERVATION command.
- Directs CMS to remove a particular generation of an element from one or more classes in the library. The generation must currently exist in the class. If you use a wildcard or a list of class names for the class expression, CMS deletes the particular generation from each specified class. See the REMOVE command.
- A particular generation to be retrieved from the archive file. If you omit /GENERATION, CMS retrieves the highest numbered generation in the archive file. See the RETRIEVE ARCHIVE command.
- The generation of the element with which to associate the review remark. If you omit /GENERATION, CMS uses the most recently created generation with a review pending. You specify this qualifier only if more than one generation of an element is under review. See the REVIEW GENERATION command.
- A particular generation of the element to be accepted. If you omit /GENERATION, CMS accepts the most recently created generation with a review pending. You specify this qualifier only if more than one generation of an element is under review. See the ACCEPT GENERATION command.

- That only reviews pending for generations matching the generation expression are to be displayed. By default, all generations with reviews pending are displayed. See the `SHOW REVIEWS_PENDING` command.
- The generation about which you want information. When you use the `/GENERATION` qualifier with the element-name parameter, the transaction record of the indicated generation is displayed. See the `SHOW GENERATION` command.
- Lists reservations for a specific generation or set of generations in a class. See the `SHOW RESERVATIONS` command.

`/`
`GENERATION[=generation-`
`expression]`
`/GENERATION=1+ (D)`

Specifies which generation of the element is to be marked as having review pending status. If you omit `/GENERATION`, CMS marks the most recent generation on the main line of descent.

You can specify a generation indirectly by using a class name, the plus operator, the semicolon, or relative generation offsets. See the *VSI DECset for OpenVMS Guide to the Code Management System*.

`/`
`GENERATION[=generation-`
`expression]`
`/GENERATION=1+ (D)`

Enables you to specify a generation indirectly by using a class name, the plus operator, the semicolon, or relative generation offsets.

In addition, this qualifier specifies the following information with different commands:

- Which generation of the element is to be marked as having review pending status. If you omit `/GENERATION`, CMS marks the most recent generation on the main line of descent. See the `MARK GENERATION` command.
- Which generation of the element is to be modified. If you omit `/GENERATION`, CMS modifies the most recent generation on the main line of descent. See the `MODIFY GENERATION` command.
- A particular generation of the element to be reserved. If you omit `/GENERATION`, CMS reserves the most recent generation on the main line of descent. See the `RESERVE` command.
- A particular generation of the element to be deleted. By default, the most recent generation on the main line of descent is deleted. You cannot combine `/GENERATION` with any of the following qualifiers: `/FROM`, `/TO`, `/AFTER`, and `/BEFORE`. See the `DELETE GENERATION` command.
- A particular generation of the element to be retrieved. If you omit `/GENERATION`, CMS fetches the most recent generation on the main line of descent. See the `FETCH` command.
- A particular generation of the element to be annotated. If you omit `/GENERATION`, CMS annotates the most recent generation on the main line of descent. The history contains a description of every generation of the element, including those created after

the specified generation. (Generations created after the specified generation are not marked with an asterisk.) See the ANNOTATE command.

- A particular generation of the element to be inserted into the class. If you omit /GENERATION, the INSERT GENERATION command uses the latest generation on the main line of descent.

/HISTORY= "string"

Specifies that the history is to be included in the retrieved file. The quoted string specifies the format of the history. The quoted string must contain the characters #H or #B (lowercase is allowed) and can contain other printing characters. To include a quotation mark in the output history string, type it twice ("). To include a number sign (#) in the output history string, type it twice (##). For a detailed explanation of the history attribute, see the *VSI DECset for OpenVMS Guide to the Code Management System*.)

/HISTORY= "string"

Establishes the **history** attribute for the element. If an element has a **history** attribute, its history (which is similar to that produced by the ANNOTATE command) is included in the file when you retrieve it with the FETCH or RESERVE command.

The quoted string specifies the format of the history. The quoted string must contain the characters #H or #B (lowercase is allowed) and can contain other printing characters. To include a quotation mark in the output history string, type it twice ("). To include a number sign (#) in the output history string, type it twice (##). For a detailed explanation of the history attribute, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

/HISTORY= "string"
/NOHISTORY (D)

Establishes the **history** attribute for the element. If an element has the **history** attribute, its history is included in the file when you retrieve it with the FETCH or RESERVE command.

The quoted string specifies the format of the history. The quoted string must contain the characters #H or #B (lowercase is allowed) and can contain other printing characters. To include a quotation mark in the output history string, type it twice ("). To include a number sign (#) in the output history string, type it twice (##). For a detailed explanation of the history attribute, see the *VSI DECset for OpenVMS Guide to the Code Management System*.)

/

IDENTIFICATION_NUMBER= n This qualifier is required when:

- You have multiple reservations of the same generation of an element.
- Multiple users have reserved the same generation of an element, and you have BYPASS privilege to the element.

You can use /IDENTIFICATION_NUMBER instead of /GENERATION when you have multiple reservations. Use the SHOW RESERVATIONS command to determine the identification number of

each reservation. The identification number appears in parentheses at the beginning of each line or before the user name.

`/IF_ABSENT`

Directs CMS to:

- Insert subgroup-expression into group-expression only if group-expression does not already contain it (see the INSERT GROUP command).
- Insert the element only if the group does not already contain that element (see the INSERT ELEMENT command).
- Insert the element generation into the class only if a generation of that element is not already in the class (see the INSERT GENERATION command).

CMS takes no action and does not return an error if:

- Subgroup-expression already belongs to group-expression (see the INSERT GROUP command).
- The element already belongs to the group (see the INSERT ELEMENT command).
- A generation of the element is already in the class (see the INSERT GENERATION command).

`/IF_CHANGED`

Specifies that a new generation is to be created only if the input file is different from the generation that was reserved. CMS automatically creates a new generation, regardless of the existence of any differences.

CMS deletes the input file from the specified location after the new generation is created (unless you specify the /KEEP or /RESERVE qualifier).

`/IF_PRESENT`

Directs CMS to remove:

- The element from the group if it belongs to the group (see the REMOVE ELEMENT command).
- Any generation of the element that exists in the class (see the REMOVE GENERATION command).
- Group-expression1 only if it belongs to group-expression2 (see the REMOVE GROUP command).

CMS takes no action and does not return an error if:

- The element does not belong to a group (see the REMOVE ELEMENT command).
- The class does not contain a generation from the element (see the REMOVE GENERATION command).
- Group-expression 1 does not belong to group-expression 2 (see the REMOVE GROUP command).

When you use wildcard characters or a comma list in the group name, /IF_PRESENT is the default.

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

Specifies one or more of the following keywords. Each keyword indicates a type of special character to be ignored during the comparison.

Keyword	Ignored Characters
CASE	Directs CMS to ignore any differences between the case of alphabetic characters (A through Z, a through z).
FORM_FEEDS	Directs CMS to remove form feed characters as it compares records from the two files.
HISTORY	Directs CMS to ignore element generation history as it compares a file with a generation. At least one of the files must be an element generation with the history attribute enabled.
LEADING_BLANKS	Directs CMS to remove leading blanks and tabs as it compares records from the two files.
NOTES	Directs CMS to ignore notes as it compares a file with a generation. At least one of the files must be an element generation with the notes attribute enabled.
SPACING	Directs CMS to compress multiple blanks and tabs into a single space as it compares records from the two files.
TRAILING_BLANKS	Directs CMS to remove trailing blanks and tabs as it compares records from the two files.

If the HISTORY or NOTES keyword is specified, the history or notes text is not used for the comparison, and is also removed from the output generated by DIFFERENCES. For all other options, the output generated by DIFFERENCES contains the original records used for the comparison, instead of the modified form of the records designated by the /IGNORE qualifier.)

/IGNORE=FIRST_VARIANT

Directs CMS to ignore differences where the generation in one class is the first variant of a generation in the other class. For example, generation 2C1 is a first variant of generation 2; and generation 4B6D1 is a first variant of generation 4B6.

/INPUT[=file-specification] Specifies a file to be used as input for the replacement transaction. If you use the **/INPUT** qualifier but do not supply a file specification, CMS searches your current, default directory for a file with the same name as the element specified on the command line. When you specify **/INPUT**, CMS deletes the input file from the specified location after the new generation is created (unless you specify the **/KEEP** or **/RESERVE** qualifier).

CMS must be able to:

- Match the input element list with the list of elements indicated by the element expression parameter. If you use wildcards in the **/INPUT** file specification to generate more than one input file, you must also use wildcards in the element expression parameter. See the **REPLACE** command.
- Create a unique element for each file in the input file list. If you use wildcards in the **/INPUT** file specification to specify more than one input file, you must also use wildcards in the element-name parameter. See the **CREATE ELEMENT** command.

/INSERT_INTO_CLASS=(class-expression) Specifies one or more classes into which newly created generations of elements are to be inserted during a **REPLACE** operation. The new generation of each element replaced is inserted into the specified class or classes. If no new generation is created, no insertion takes place. Use of this qualifier requires **EXECUTE** access to the **INSERT GENERATION** command.

The class expression can be a class name, wildcard expression, or a list of class names separated by commas. If only one class name is specified, the parentheses can be omitted.

For each class specified by this qualifier, new generations are always inserted, regardless of whether the class already contains a generation of the element. CMS removes previous generations from the class before inserting the new generation. This is similar to the function of the **INSERT GENERATION/ALWAYS** command. All insertions occur in the library in which the new generations are created. Occlusion does not apply to classes specified by this qualifier.

CMS performs the following access control entry (ACE) actions after performing those associated with the **REPLACE** command or with the elements being replaced:

- actions associated with the **INSERT GENERATION** command
- actions associated with the classes into which generations are inserted

If other insertion options, such as **/SUPERSEDE**, are required, create a temporary class and use that class during the **REPLACE** operation. Then specify the temporary class as the value for an **INSERT GENERATION/GENERATION** operation using the necessary additional options.

<code>/KEEP</code> <code>/NOKEEP (D)</code>	Controls whether CMS deletes the disk files used to replace or create an element. The library-wide default can be set in the <code>CREATE LIBRARY</code> and <code>MODIFY LIBRARY</code> commands, but can be overridden at the element level, as follows: ● In the <code>REPLACE</code> command, this qualifier controls whether the file used to create the new element generation is deleted from your directory. If you omit both <code>/KEEP</code> and <code>/RESERVE</code>, the files are deleted. ● In the <code>CREATE ELEMENT</code> command, this qualifier controls whether CMS deletes all versions of the file used to create the new element. If you specify <code>/KEEP</code>, CMS does not delete the file.
<code>/LIBRARY[=directory-specification]</code>	Identifies a valid CMS library that is the location of the element, group, or class specified by the old- <code>n-expression</code> parameter. When you specify an alternate library, the new- <code>n-name</code> parameter is optional. If you do not specify a value for <code>/LIBRARY</code>, the current CMS library is used.
<code>/LIKE=object-specification</code>	Indicates that the ACL of the specified object is to replace the ACL of the object (or objects) specified with <code>SET ACL</code>. Any existing ACEs are deleted before the ACL specified by <code>/LIKE</code> is copied. The type of the source and destination objects must be the same. Also, the specified object must be in the same library as the object (or objects) specified with <code>SET ACL</code>. Wildcard characters are not allowed.
<code>/LOG (D)</code> <code>/NOLOG</code>	Controls whether CMS displays success and informational messages on the default output device. If the command executes successfully, CMS displays a success message. If you specify <code>/NOLOG</code>, success and informational messages are suppressed. Any warning, error, or fatal error messages are displayed regardless of whether <code>/LOG</code> or <code>/NOLOG</code> is specified.
<code>/</code> <code>LONG_VARIANT_NAMES</code> <code>/</code> <code>NOLONG_VARIANT_NAMES (D)</code>	With the <code>CREATE LIBRARY</code> and <code>MODIFY LIBRARY</code> commands, this qualifier specifies whether long variant names will be allowed. Long variant names can consist of an alphabetic string (including underscores) up to 255 characters in length.
<code>/MEMBER</code> <code>/NOMEMBER (D)</code>	Lists the element name, creation remark or generation number, and the names of any groups or classes to which the element or element generation belongs.
<code>/MERGE=generation-expression</code> <code>/NOMERGE (D)</code>	Controls whether another generation of the element (called the merge generation) is to be merged with the generation that is being reserved or fetched (called the retrieved generation). If you specify the <code>/MERGE</code> qualifier, CMS merges the lines of the two generations and delivers a single copy of the file to your default directory. The file placed in your directory has the current creation and revision times. The merge generation cannot be on the same line of descent as the retrieved generation. When there is a conflict between

blocks of one or more lines, CMS includes the conflicting lines and flags the conflict.

This qualifier also combines two generations of an element and creates a single file containing the annotated listing. The parameter on the /MERGE qualifier specifies the generation merged into the retrieved generation. This qualifier creates a file containing the text common to both generations and the changes made to both generations. When changes that are not identical are made in the same position of the common ancestor, the changes from both generations are included in the resulting file and are marked as a conflict. By default, generations are not merged.

For an explanation of how two generations are merged and how CMS treats conflicts between the generations, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

/NAME=class-name

Specifies the new name for the class, element, or group. The following rules apply to the commands that use this qualifier:

- You cannot use the file name 00CMS. This name is reserved for CMS. If you specify the /NAME qualifier, you cannot use wildcards or a comma list in the element-name parameter, nor can you use a wildcard for the /NAME qualifier. If an element is set to /REFERENCE_COPY, CMS creates a new reference copy of the element in the reference copy directory. See the MODIFY ELEMENT command.
- If a previously used class or group name was removed with the DELETE CLASS or DELETE GROUP command, you can reuse that name. Wildcards and comma lists are not allowed. If you specify the /NAME qualifier, you cannot use wildcards or a comma list in the class-expression parameter, nor can you use a wildcard for the /NAME qualifier. You cannot change the name of a class that has read-only access. See the MODIFY CLASS command.
- Wildcards and comma lists are not allowed in group names. If you specify the /NAME qualifier, you cannot use wildcards or a comma list in the group name parameter, nor can you use a wildcard for the /NAME qualifier. See the MODIFY GROUP command.

/NEW

Indicates that any existing ACEs in the ACL of the object specified with SET ACL are to be deleted. To use the /NEW qualifier, you must specify a new ACL or ACE with the /ACL qualifier.

/NOHISTORY

With the RESERVE and FETCH commands, this qualifier prevents CMS from including or appending the element history in or to the file. If you omit /NOHISTORY and the retrieved element has the **history** attribute, CMS includes the element history in the output file, or the element history is included in the file when it is delivered to your current, default directory. An element has the **history** attribute if you specify the /HISTORY qualifier on the CREATE ELEMENT or MODIFY ELEMENT command.

With the MODIFY ELEMENT command, this qualifier might also delete any existing **history** attribute. If both /HISTORY and /NOHISTORY are omitted, any existing **history** attribute remains unchanged.

/NONOTES

With the FETCH or RESERVE command, this qualifier specifies that notes are not to be embedded in the output file. If you omit /NONOTES, and the retrieved element has the **notes** attribute, CMS embeds notes in the output file. An element has the **notes** attribute if you specify the /NOTES qualifier on the CREATE ELEMENT or MODIFY ELEMENT command.

With the MODIFY ELEMENT command, this qualifier cancels any current **notes** attribute and the corresponding **position** attribute. If both /NOTES and /NONOTES are omitted, any existing **notes** attribute remains unchanged.

/NOOUTPUT

Specifies that the default system-generated output file is not to be created, as follows:

- With the RESERVE command, this qualifier specifies that the generation is to be reserved, but that no output file is to be created.
- With the DIFFERENCES command, this qualifier directs CMS to execute a comparison without creating an output file. This form of the comparison might be significantly faster because CMS stops the transaction when it encounters the first difference.
- With the FETCH command, this qualifier specifies that the fetch operation is to be performed along with any history processing and error checking, but that no output file is to be created. By default, an output file with the same name as the element is created.

For all other commands that generate an output file, the automatic file generation cannot be negated with the /NOOUTPUT qualifier.

/NOREPLACEMENTS

When combined with the /CONCURRENT qualifier in the SHOW RESERVATIONS command, this qualifier indicates that concurrent replacements are not to be displayed. With this combination, only the concurrent reservations would be displayed.

/NOTES= "string"

With the MODIFY ELEMENT or CREATE ELEMENT command, this qualifier establishes the **notes** attribute for the element. If an element has a **notes** attribute, notes are appended to the lines of the file when it is retrieved by the FETCH or RESERVE command. The default with the CREATE ELEMENT command is /NONOTES.

With the RESERVE command, this qualifier specifies that notes are to be appended to the lines of the file as it is retrieved by the RESERVE operation. The qualifier overrides the element's **NONOTES** attribute, if one is established.

With the FETCH command, this qualifier might temporarily establish the **notes** attribute for the element, regardless of whether the element

previously had the **notes** attribute enabled. If neither the /NOTES nor /NONOTES qualifier is specified for an element, but the element has the **notes** attribute enabled, notes are appended to the lines of the file when it is retrieved by the FETCH or RESERVE command.

With the FETCH and RESERVE commands, if you specify /NOTES, you must also specify /POSITION. For a detailed explanation, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

The quoted string specifies the format of the note. The quoted string can contain text or the characters #G, #g, or both. For a detailed explanation of the **notes** attribute, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

/OBJECT_TYPE=type

With the SET ACL or SHOW ACL command, this qualifier specifies the type of the object whose ACL is being modified or displayed. There is no default object type; therefore, this qualifier is required. The type or object type must be one of the following keywords:

- CLASS
- ELEMENT
- GROUP
- LIBRARY
- COMMAND (used with SET ACL only)

/OCCLUDE[=option,...]
/OCCLUDE=ALL (D)

Controls whether CMS selects the first instance of the specified object, or all instances of the specified object in the library search list. The options field contains one or more keywords associated with the name of the object. The options field can consist of the following keywords:

- ALL (equivalent to ELEMENT, GROUP, and CLASS or OTHER)
- ELEMENT
- NOELEMENT
- GROUP
- NOGROUP
- CLASS
- NOCLASS
- OTHER
- NOOTHER
- NONE (equivalent to NOELEMENT, NOGROUP, and NOCLASS or to NOOTHER)

You can specify either ALL or NONE, or any combination of the [NO]ELEMENT, [NO]GROUP, and [NO]CLASS keywords.

CMS performs occlusion for all objects automatically; that is, CMS selects only the first occurrence of a specified object.

If you do not specify an element expression on the VERIFY command, the default is /OCCLUDE=NONE. If you specify an element expression, the default is /OCCLUDE=ALL.

The /OCCLUDE qualifier is used with many commands, including but not limited to most of the CREATE, DELETE, INSERT, MODIFY, REMOVE, and SHOW commands. Other commands include ANNOTATE, DIFFERENCES, REMARK, VERIFY, etc.)

/OUTPUT[=file-specification]

Directs CMS to write output to the specified file. If you omit the /OUTPUT qualifier (or if you specify /OUTPUT but do not provide a file specification), CMS creates one of the following, depending on which CMS command was used:

- A file with the same name as the element in your default directory.
- A file named HISTORY.DMP. The default is /OUTPUT=HISTORY.DMP. (See the DELETE HISTORY command.) Output generation cannot be negated with the /NOOUTPUT qualifier.
- A new file if you do not specify /APPEND (see the ANNOTATE command).
- A file with the same name as the element (see the OUTPUT command).

If you specify /NOOUTPUT with certain commands, CMS is prevented from creating an output file. However, the /OUTPUT generation in other commands cannot be negated, but can redirect output with a file-specification entry.

With the RESERVE command, if you reserve more than one element (by specifying wildcards or a group name for the element expression parameter), CMS creates successive versions of the file indicated by /OUTPUT.

With the RETRIEVE ARCHIVE command, if you retrieve generations from more than one archive file and do not specify wildcards in the output file specification, CMS creates successive versions of the file indicated by /OUTPUT. Output generation cannot be negated with the /NOOUTPUT qualifier.

With the SHOW REVIEWS_PENDING command, this qualifier directs CMS to write output to the specified file, except for any warning and error messages, which are written to SYS\$OUTPUT and SYS\$ERROR. CMS creates a new file if you do not specify /APPEND. If you omit the /OUTPUT qualifier (or if you specify /OUTPUT but

do not provide a file specification), CMS directs output to the default output device (SYS\$OUTPUT). If you omit either the file name or file-type component, CMS supplies the missing component from the default specification.

With the ANNOTATE command, this qualifier directs CMS to write output to the specified file. CMS creates a new file if you do not specify /APPEND. If you omit the /OUTPUT qualifier (or if you specify /OUTPUT but do not provide a file specification), CMS creates a file with the element file name and the file type .ANN. Wildcards are allowed. The default is /OUTPUT=element-name .ANN. Output generation cannot be negated with the /NOOUTPUT qualifier.

If you ANNOTATE or FETCH more than one element (by specifying wildcards or a group name for the element expression parameter), and you do not specify wildcards in the output file specification, CMS creates successive versions of the file indicated by /OUTPUT. Output generation cannot be negated in the ANNOTATE command with the /NOOUTPUT qualifier.

With the DIFFERENCES command, this qualifier directs CMS to write output to the specified file. CMS creates a new file if you do not specify /APPEND. If you omit the /OUTPUT qualifier (or if you specify /OUTPUT but do not provide a file specification), CMS creates a file with the file name specified by the **file1** or **class-name1** parameter and the file type .DIF. If you specify a file name but omit the file-type component, CMS writes the output to a file with the specified file name and a file type of .DIF. The default is /OUTPUT=file1.DIF.

Obviously, the intent of SHOW commands is to generate an output file. Therefore, their automatic file generation cannot be negated with the /NOOUTPUT qualifier.

/PAGE_BREAK
/NOPAGE_BREAK (D)

Controls whether CMS allows page breaks in the output file. Page breaks are converted to the string "<PAGE>" in the output file. Use /PAGE_BREAK to request the inclusion of page breaks in the output file.

/PARALLEL
/NOPARALLEL (D)

Controls whether the differing lines from the two files are formatted side by side. Note that this qualifier can only be used with record partitioning. If you specify /PARALLEL, the differences from the first file are displayed on the left and the differences from the second file are displayed on the right. The heading of the differences report displays the file specification of the first file on the left and the file specification of the second file on the right.

The default width of the listing is 132. Use the /WIDTH qualifier to control the width. Vertical lines separate the text on the left side of the report from the text on the right side. The text from each of the files is allotted equal space (half the width of the full report).

If a line from one of the files being compared is longer than half the width of the full report, the line is truncated on the right. A plus sign (+) printed at the end of the line indicates that the line is truncated.

/POSITION=column-number

Specifies the column in which the note is to be placed. The column number is required and must be an integer in the range 1 to 511. The **notes** attribute or the /NOTES qualifier is required with the /POSITION qualifier.

If the length of the line is less than the specified column number, the note appears at the column number. If the length of the line is greater than or equal to the column number, the note is placed at the next tab stop after the end of the line. (Tab stops are at positions 9 and every 8 characters thereafter.)

/POSITION=n

Establishes the **position** attribute (the character position where the note generated by the /NOTES qualifier begins or is to begin on the line). The value *n* is required and must be an integer in the range 1 to 511. The /NOTES qualifier is required with the /POSITION qualifier. If the length of the line is less than *n*, the note appears at position *n*. If the length of the line is greater than or equal to *n*, the note is placed at the next tab stop after the end of the line. (Tab stops are at position 9 and every 8 characters thereafter.) For more information about the position attribute, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

With the CREATE ELEMENT command, the note is placed to the right of the text of the line.

With the MODIFY ELEMENT command, a file that has a **position** attribute must have a **notes** attribute. The /POSITION qualifier can be used only if the element has an established **notes** attribute, or the /NOTES qualifier is used on the same MODIFY ELEMENT command. If you omit /POSITION, any current **position** attribute remains unchanged.

/READ_ONLY
/NOREAD_ONLY (D)

Establishes or alters the **read-only** attribute of a class or group. To change the characteristics of a read-only class or a READ_ONLY group, you must set the class to NOREAD_ONLY. NOREAD_ONLY is the default attribute of a class or group when it is created with the CREATE CLASS or CREATE GROUP command.

/RECOVER
/NORECOVER (D)

Controls whether the VERIFY command cancels an incomplete transaction. You use the /RECOVER qualifier when a transaction with the library is incomplete and the rollback mechanism does not automatically cancel the command (see the *VSI DECset for OpenVMS Guide to the Code Management System*). For example, you must use the /RECOVER qualifier if the OpenVMS system fails while a CMS command is updating the library.

If a CMS command is terminated and the library is left in an inconsistent state, CMS recognizes that the last transaction was incomplete and automatically initiates command rollback (see the *VSI*

DECset for OpenVMS Guide to the Code Management System) to return the library to a valid format.

If you have set up a restrictive protection scheme and a system failure occurs during a CMS transaction and leaves your library in an inconsistent state, the VERIFY/RECOVER command should be executed by the same person who was using CMS at the time of the system failure, or by a person with sufficient privileges.

/ Specifies a valid OpenVMS directory to be used for reference copies
REFERENCE_COPY=directory- of library elements. The directory cannot be a CMS library, nor should
specification it be a subdirectory of a CMS library directory. Wildcards are not
/ allowed with the CREATE LIBRARY command.
NOREFERENCE_COPY(D)

- If you use the CREATE LIBRARY command to create a search list of more than one library, you should specify a different reference copy directory for each library in the search list.
- If you use the MODIFY LIBRARY command on a search list of more than one library, you should specify a reference copy directory for each library in the search list.

Although CMS allows different libraries to be assigned the same reference copy directory, it is strongly recommended that you assign each CMS library its own unique reference copy directory. If you specify only one reference copy directory for more than one library, CMS creates or uses one reference copy directory for the entire search list, *not* one reference copy directory for each library in the search list.

Use the /NOREFERENCE_COPY qualifier to remove the connection between the current CMS library and the current reference copy directory. Wildcards are not allowed. This qualifier is the default with the CREATE LIBRARY command.

With the MODIFY ELEMENT command, this qualifier controls whether CMS maintains a reference copy of the element. You must establish a reference copy directory.

The presence of the **reference copy** attribute for an element is inherited from the library; that is, if a reference copy directory is established for the library, the attribute is automatically enabled for the element. Specify /NOREFERENCE_COPY to override the **reference copy** attribute.

If you establish a reference copy directory for the CMS library, CMS creates a reference copy of the new element and updates the reference copy directory each time you create a new main-line generation of that element. When CMS places a file in the reference copy directory, it also deletes any earlier versions of that file in the reference copy directory.

/REMARK= "string"

This qualifier specifies:

- A new remark to be substituted for the creation remark associated with the generation. This qualifier is required. (See the MODIFY GENERATION command.)
- A new remark to be substituted for the creation remark associated with the class. You cannot change the remark of a class set to READ_ONLY. (See the MODIFY CLASS command.)
- A new remark to be substituted for the creation remark associated with the element. (See the MODIFY ELEMENT command.)
- A new remark to be substituted for the creation remark associated with the group. (See the MODIFY GROUP command.)
- A new remark to be substituted for the current remark associated with the reservation. This qualifier is required. (See the MODIFY RESERVATION command.)

/REMOVE_CONTENTS

Controls whether CMS removes items from the specified group or class prior to deletion. When /REMOVE_CONTENTS is specified with either the DELETE GROUP or DELETE CLASS command, CMS displays an informational message for each item that is removed from the specified group or class.

**/REPAIR
/NOREPAIR (D)**

Controls whether the VERIFY command repairs a file in the CMS library. You should use the /REPAIR qualifier if VERIFY issues a message concerning one of the following conditions:

- Element data files in the library were not closed by CMS.
- The checksum of elements in the library is invalid.
- Generations in the library have an invalid maximum record size.
- A data block was not found on the first pass.
- The reference copy for an element is missing.
- A reference copy is found for an element with the /NOREFERENCE_COPY qualifier.
- There are duplicate reference copies for an element.
- The reference copy of an element is invalid.
- There is an inconsistency in a backpointer list for a class or generation.
- There are lost free blocks or inconsistencies in block count and block attributes.

If reference copies need repairing, VERIFY/REPAIR creates or deletes files as necessary to correct the information in the reference copy directory.

If a file was not closed by CMS, VERIFY/REPAIR repairs the OpenVMS Record Management Services (RMS) file header so the file can be successfully verified. If the checksum of a file does not correspond to the contents of the file, VERIFY/REPAIR recalculates the checksum so the library can be verified.

If any of these conditions exist, data might have been changed in the library by methods other than the normal updating of the library with CMS commands. For example, a file might have been opened and modified with a text editor. You might want to find out why the files could not be verified. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

/REPLACE=(ace[,...])

Deletes the 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 current ACL.

/RESERVE
/NORESERVE (D)

Specifies the following:

- With the REPLACE command, this qualifier controls whether the new generation of the element created by the replacement is reserved. If you specify the /RESERVE qualifier, the generation is reserved and the element files are not deleted from your current, default directory. The list of concurrent replacements is updated as if /RESERVE had been omitted. For information on concurrent reservations and replacements, see the *VSI DECset for OpenVMS Guide to the Code Management System*.
- With the CREATE ELEMENT command, this qualifier controls whether the new element is to be reserved after it is created. When you specify /RESERVE, CMS does not delete the file used to create the element. Generation 1 of the newly created element is automatically reserved. If you omit both the /RESERVE and the /KEEP qualifiers, CMS deletes all versions of the file used to create the element, unless the /KEEP qualifier was set as the library-wide default with the CREATE LIBRARY or MODIFY LIBRARY command.

/REVIEW
/NOREVIEW (D)

Specifies that new generations of the element are marked for review. By default, new generations of the element are marked for review only if the reserved generation either was rejected or has a review pending. If you specify CREATE ELEMENT/REVIEW, generation 1 of the element is also marked for review.

You can change the **review** attribute with the MODIFY ELEMENT command.

/[NO]REVIEW

Controls whether new generations of the element are marked for review. If you specify /REVIEW, new generations of the element are marked for review. If you specify /NOREVIEW, new generations are marked only if the reserved generation either is rejected or has a review pending. If you do not specify this qualifier, the existing **review** attribute is not changed.

To determine whether an element has the **review** attribute enabled, use the SHOW ELEMENT/FULL command.

/REVISION_TIME[=option]

Controls whether CMS uses the original file revision time or the file storage time when a file is retrieved from the CMS library. The options field can contain one of the following keywords:

ORIGINAL (D)
STORAGE_TIME

Use the ORIGINAL keyword to indicate that the original revision time of files placed in a CMS library should be restored unchanged upon their retrieval. This is the default behavior.

Use the STORAGE_TIME keyword to indicate that the time when a file was stored in a CMS library (through a CREATE ELEMENT or REPLACE transaction) should be substituted for its original revision time upon retrieval.

/SENTINEL=(“begin-delimiter”, “end-delimiter”)

Specifies a pair of strings used to delimit a section of text to be ignored during the comparison of both files. The delimiters can be up to 256 characters per line and must be unique. Any text between and including the delimiters is treated as if it did not exist. If you do not enclose the sentinel strings in quotation marks, they are converted to uppercase before the comparison of the files. Sentinel strings can contain any characters, but if you include spaces or tabs, they must be enclosed in quotation marks.

Sentinel strings can appear anywhere in a file. If text delimited by a sentinel pair crosses record boundaries, the text after the delimited region appears in its own record in the output file; it is not appended to the contents of the record in which the begin delimiter was found.

/SHOW=(options[,...])

Specifies the type of information that is generated from a DIFFERENCES/CLASS command. The options field can contain one of the following keywords:

Keyword	Difference Type
GENERATION_DIFFERENCES	Directs CMS to show when complete generations differ between classes; variants are ignored.
VARIANT_DIFFERENCES	Directs CMS to show when generations and variants differ between classes. One generation will be a variant of the other if at some point in the path between one generation and the other a generation is reserved and replaced /VARIANT.

If an element generation is a member of one class but not the other, the difference is always shown regardless of whether the /SHOW qualifier is specified.

/SINCE=date-time

Specifies the following:

- With the SHOW REVIEWS_PENDING command, this qualifier specifies that only generations placed under review after the indicated date and time are to be displayed. By default, all generations with reviews pending are displayed. The date value can be an absolute, delta, or combination time value, or one of the following keywords: TODAY, TOMORROW, or YESTERDAY.
- With the SHOW HISTORY command, this qualifier specifies that only those history entries dated on or after the given time are to be displayed. The time value can be an absolute, delta, or combination time value, or one of the following keywords: TODAY, TOMORROW, or YESTERDAY. If you omit the /SINCE qualifier, all transactions that occurred since the library was created are displayed. If you specify the /SINCE qualifier but do not specify a value, CMS defaults to /SINCE=TODAY.
- With the SHOW RESERVATIONS command, this qualifier lists current reservations that were executed after the specified time. The time value can be an absolute, delta, or combination time value, or one of the following keywords: TODAY, TOMORROW, or YESTERDAY.
- With the SHOW GENERATIONS command, this qualifier lists current generations that were transacted after the specified time. The time value can be an absolute, delta, or combination time value, or one of the following keywords: TODAY, TOMORROW, or YESTERDAY.

/SKIP=number-of-lines

Indicates the number of lines at the beginning of each file (or generation) that are to be ignored during the comparison of both files. You must specify a non-negative integer value indicating the number of lines to be ignored.

/SUPERSEDE
/NOSUPERSEDE (D)

Controls whether CMS removes a generation of the element that exists in the class and replaces it with the specified generation. (Using /SUPERSEDE is the equivalent of using the REMOVE GENERATION command before the INSERT GENERATION command.

If you specify /SUPERSEDE and there is no generation of the specified element already in the class, an error message is issued and the generation is not inserted into the class. You cannot use the /IF_ABSENT qualifier on the same command line as the /SUPERSEDE qualifier to override this action.

If you omit the /SUPERSEDE qualifier and a generation of the element already exists in the class, an error message is issued and no change is made to the library.

/TO=generation-expression

With the DELETE GENERATION command, this qualifier specifies the end of a range of generations to be deleted, *including* the specified generation. You cannot specify both /TO and /BEFORE, or both /TO and /GENERATION. You must specify the start of the range with either the /AFTER or /FROM qualifier. You must specify a generation-expression value.

/[NO]TRANSACTIONS=
(keyword,...)
/TRANSACTIONS=ALL
(D)

Displays or deletes all transaction records generated by a specific command. You can specify the following keywords with this qualifier:

ACCEPT	FETCH	REMOVE
ALL	INSERT	REPLACE
CANCEL	MARK	RESERVE
COPY	MODIFY	REVIEW
CREATE	REJECT	SET
DELETE	REMARK	UNRESERVE
		VERIFY

If you specify more than one keyword, you must enclose the keyword list in parentheses. The /TRANSACTIONS qualifier directs CMS to list or delete transaction records for only the listed keywords. The /NOTTRANSACTIONS qualifier directs CMS to list or delete transaction records for all keywords except the listed keywords.

/UNUSUAL

Specifies that the remark string placed in the history file be marked as an unusual occurrence, so it appears marked with an asterisk in the output from SHOW HISTORY and is included in the output from a SHOW HISTORY/UNUSUAL command.

/UNUSUAL
/NOUNUSUAL (D)

Controls whether transactions recorded as unusual occurrences are displayed. If you specify /UNUSUAL, only unusual transactions are displayed. /NOUNUSUAL displays all transactions, including unusual occurrences. The following CMS transactions are defined as unusual occurrences:

- A RESERVE command specifying an element that is already reserved
- A concurrent replacement
- A VERIFY/RECOVER command
- A VERIFY/REPAIR command
- A CONVERT LIBRARY command
- A REMARK/UNUSUAL command

/USER=username

Specifies the following:

- With the DELETE HISTORY command, this qualifier deletes all transaction records created by the specified user.

- With the SHOW REVIEWS_PENDING command, this qualifier specifies that only reviews pending for generations created by the indicated user are to be displayed. By default, all generations with reviews pending are displayed.
- With the SHOW HISTORY command, this qualifier lists the transactions executed by the specified user.
- With the SHOW RESERVATIONS command, this qualifier lists current reservations that were executed by the specified user. If you do not specify this qualifier, CMS lists reservations held by all users.

/VARIANT=variant-letter
/NOVARIANT (D)

Controls whether a variant generation is created. If you specify the /VARIANT=variant-letter qualifier, the number of the created generation is the predecessor's number, followed by the variant letter, followed by the number 1.

If two or more users have concurrently reserved the same element generation, the replaced generations cannot be on the same line of descent. Thus, one can be replaced as a main-line generation and the rest must be replaced as variants. For more information on creating variant generations, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

/VERIFY (D)
/NOVERIFY

Enables you to set a library without performing the locking and verification process CMS normally performs. This speeds up the SET LIBRARY operation and enables a CMS library to be set even if the library is locked by another user. However, if the library needs recovery, the condition is not detected until another transaction is attempted.

/WIDTH=n

Specifies the limit for the width of the differences report. The value *n* is required and must be an integer in the range 48 to 511. If *n* is less than 48, 48 is used. If *n* is more than 511, 511 is used. The default width is the same as the width of the output device.

The width of the report is rounded down to the nearest multiple of 8 minus 1. CMS rounds down so if you have specified the /PARALLEL qualifier, CMS correctly interprets the horizontal tabs in the file on the right. For example, if you specify a value of 100 on the /WIDTH qualifier, the actual width is 95.

Chapter 3. CMS Messages

This section lists the CMS system messages alphabetically. The messages are accompanied by explanations and, where applicable, suggested actions to recover from errors.

3.1. Message Display

Messages are displayed on the current device identified by the logical name SYS\$OUTPUT. For an interactive user, this device is a terminal; for batch job users, it is the batch job log file. If the logical device SYS\$ERROR is different from SYS\$OUTPUT, the system writes warning, error, and fatal error messages to that device as well.

3.1.1. Severity Levels

The severity level of a message is included in the status message. Success and informational messages inform you that CMS has performed your request.

Warning messages indicate that the command might have performed some, but not all, of your request, and that you might need to verify command or program output.

Error messages indicate that CMS is unable to perform the requested function, and you must correct the problem and enter the command again.

Fatal messages indicate that CMS has discovered inconsistencies in itself or the CMS library, and normal processing cannot continue.

3.1.2. Library Directory Specifications

CMS displays the library directory specification for each status message containing a CMS object. These include the following:

- Elements
- Groups
- Classes
- The library (and its object subtypes)

This enables you to determine the exact library to which an object belongs when multiple libraries are involved in the operation. For example:

```
$ CMS SET LIBRARY [WORK.CMSLIB], [PROJ.CMSLIB]
.
.
.
$ CMS RESERVE EXAMPLE.TXT/OCCLUDE=NOELEMENT ""
%CMS-S-RESERVED, generation 1 of element DISKX:[WORK.CMSLIB]EXAMPLE.TXT
reserved
%CMS-S-RESERVED, generation 1 of element DISKX:[PROJ.CMSLIB]EXAMPLE.TXT
reserved
```

3.1.3. Secondary Messages

Often, CMS displays more than one message as a result of the execution of a command. When this occurs, CMS displays the primary message that indicates the status return, followed by one or more secondary messages that provide specific information. For example:

```
$  
CMS FETCH SRC.C"  
%CMS-E-NOFETCH, error fetching element SRC.C  
-CMS-E-NOTFOUND, element SRC.C not found
```

Primary messages contain the percent sign (%) prefix; secondary messages contain the hyphen (-) prefix.

3.2. CMS Messages

ABSTIM, 'qualifier ' time value must be absolute

Type: Error

Description: You must use an absolute time value for the /BEFORE and /SINCE qualifiers.

User Action: Correct the time value and enter the command again.

ACCEPTANCES, 'count ' generation(s) accepted

Type: Success

Description: CMS has successfully marked the indicated number of generations as accepted and removed them from the review pending list.

User Action: None.

ACCEPTED, generation 'gen-number ' of element 'name ' accepted

Type: Success

Description: CMS has successfully marked the indicated generation of the specified element as accepted and removed it from the review pending list.

User Action: None.

ACCVIORD, access violation reading routine argument at virtual address 'address'

ACCVIOWT, access violation referencing routine argument at virtual address 'address'

Type: Fatal

Description: You have passed an unreadable or unwritable argument to a CMS routine.

User Action: Check the number of arguments in the call to CMS. In addition, check to make sure you are using the correct passing mechanism, and you are passing the arguments in the correct order.

ALPHACHAR, in a class name the first character must be alphabetic

Type: Error

Description: You have specified a class name that does not begin with a letter.

User Action: Correct the class name and enter the command again.

ALRDYEXISTS, 'name ' is already an element name

ALRDYEXISTS 'name ' is already a class name

ALRDYEXISTS, 'name ' is already a group name

ALRDYEXISTS, 'name ' is already a reference copy name

Type: Error

Description: You have specified a group, element, or class name that is already being used in the library, or a reference copy name that already exists in the reference copy directory. All these names must be unique.

User Action: Use a different name and enter the command again.

ALRDYINCLS, class 'name ' already contains a generation from element 'name '

Type: Error

Description: You have tried to insert an element generation into a class that already contains a generation from that element. Classes can contain only one generation from a particular element.

User Action: If you want to insert the new generation into the class, you must first remove the old generation from the class. (See the description of the REMOVE GENERATION command, or the description of the /ALWAYS qualifier for the INSERT GENERATION command.)

ALRDYINGRP, element is already in group

ALRDYINGRP, group is already in group

Type: Error

Description: The specified element or group already belongs to the specified containing group.

User Action: None.

ALRDYMARKED, generation 'gen-number ' of 'name ' already marked for review

Type: Error

Description: You have tried to mark a generation for review that is already marked.

User Action: None.

ANNOTATED, element 'name' annotated

Type: Success

Description: CMS has successfully created an annotated listing file of the indicated element.

User Action: None.

ANNOTATIONS, 'count' annotation(s) completed

Type: Success

Description: CMS has successfully created annotated listings for the indicated number of elements.

User Action: None.

ARGCONFLICT, argument conflict (routine and file both specified)

Type: Error

Description: There are several CMS routines that do not accept both a routine and a file for input or output.

User Action: Check the arguments that you have specified in your calls to CMS.

ARGCOUNTERR, incorrect number of arguments

Type: Error

Description: You have passed an incorrect number of arguments to the CMS\$GET_STRING routine. The CMS\$GET_STRING routine expects two arguments, one for the string identifier and a corresponding string to be filled in by CMS.

User Action: Check the arguments you have specified in the routine call.

AUTOREC, attempting automatic VERIFY/RECOVER

Type: Informational

Description: CMS has detected that recovery is necessary. An automatic VERIFY/RECOVER transaction will be attempted. If you do not have the required access privileges, automatic recovery will fail.

User Action: None.

AUTORECSUC, automatic VERIFY/RECOVER was successful

Type: Informational

Description: CMS has successfully recovered your library with an automatic VERIFY/RECOVER operation.

User Action: None.

BADBUG, there is an unrecoverable bug in CMS or something it calls

Type: Fatal

Description: A rollback attempt to recover has failed.

User Action: This is not a situation that a user can resolve. Contact your VSI support representative.

BADCALL, invalid call to routine 'name '

Type: Fatal

Description: CMS has detected something wrong with one or more of the arguments passed to a CMS routine. This message can be followed by CMS\$_INVFETDB, CMS\$_INVLENGTH, CMS\$_INVLIBDB, CMS\$_INVRDARG, CMS\$_INVSTRDES, CMS\$_INVWRTARG, CMS\$_MAXARG, CMS\$_MINARG, or other error messages.

User Action: See the descriptions of the secondary messages for more information. Do not use SS \$_CONTINUE to continue processing when CMS returns CMS\$_BADCALL.

BADCRC, incorrect checksum in element 'name '

Type: Warning

Description: An element in your library contains an incorrect checksum.

User Action: Use the VERIFY/REPAIR command to recalculate the checksum.

BADFORMAT, invalid format combination

Type: Error

Description: CMS has detected an invalid format argument passed to a CMS routine. Only one data-format type can be specified, and only one data-partition type can be specified. The combination of data format and data partition must be valid.

User Action: Correct the mistake in the format information passed.

BADLIB, there is something wrong with your library

Type: Fatal

Description: CMS has discovered an inconsistency or error in your CMS library.

User Action: A BADLIB message indicates an inconsistency in your library that CMS cannot fix. You must use a backup copy of your library to be sure that you are using a consistent database.

BADCRETIME, file 'name' was created after the last successful transaction

BADLENSTR, 'structure' block length is 'count'; it should be 'count'

BADLSTSTR, 'object' list does not hold this 'object'

BADORDSTR, 'object' block 'name' is out of order

Type: Error

Description: CMS discovered an error while verifying your library.

User Action: See the descriptions of the primary messages for more information.

BADLST, 'object' list for generation; 'element' ('number') is corrupted

Type: Error

Description: CMS encountered an error in an internal list related to a generation in the library being verified.

User Action: Reissue the VERIFY/REPAIR command, or restore a backup copy of your library.

BADPTR, cannot finish reading 'object' list; bad pointer 'address'

BADTYPSTR, 'object' block type is 'number' ; it should be 'number'

BADVERSTR, 'object' block version is 'number' ; it should be 'number'

BCKPTRSTR, 'object' back pointer is 'address' ; previous block is 'address'

Type: Error

Description: CMS discovered an error while verifying your library.

User Action: See the descriptions of the primary messages for more information.

BADREF, reference copy for element 'name' is bad

Type: Error

Description: The reference copy for the indicated element is incorrect.

User Action: Use VERIFY/REPAIR to re-create the reference copy for the element.

BADREFHDR, reference copy file 'name' not closed by CMS

Type: Error

Description: The header file for a reference copy of an element in the library was edited by an application other than CMS.

User Action: Use VERIFY/REPAIR to re-create the reference copy of the element.

BADVERSION, file 'name ' has a bad version

Type: Error

Description: There is a duplicate element data file in the library, and CMS is unable to determine which of the two files is the valid element.

User Action: Examine the contents of the element data files to see if one of them is incorrect (perhaps it has no control records, or it is an incomplete file). Delete the incorrect file and enter the command again.

BUG, there is something wrong with CMS or something it calls

Type: Fatal

Description: CMS has discovered a bug or inconsistency in itself.

User Action: This is not a situation that a user can resolve. Contact your VSI support representative.

CANCELLATIONS, 'count ' reviews canceled

Type: Success

Description: The indicated number of generations marked for review has been successfully removed from the review pending list.

User Action: None.

CANCELED, review of generation 'gen-number ' of element 'name ' canceled

Type: Success

Description: The indicated generation of the specified element marked for review has been successfully removed from the review pending list.

User Action: None.

CANTRESMRG, cannot resolve merge, marked as conflict

Type: Warning

Description: CMS encountered an inconsistency during a merge operation. The lines reported as conflicting may not be from the corresponding portions of the two generations being merged.

User Action: If necessary, perform the merge operation manually.

CNTSTR, 'object ' count is 'count ' ; it should be 'count '

Type: Error

Description: This is a secondary message to BADLIB.

User Action: This message indicates a situation that CMS cannot fix. You must use a backup copy of your library. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

COMPARED, 'file-spec ' and 'file-spec ' compared

Type: Success

Description: CMS has successfully compared the indicated files.

User Action: None.

CONFLICTS, 'count ' merge conflict(s)

Type: Warning

Description: CMS has detected one or more conflicts during the merge transaction. The merge conflicts are flagged with asterisks in the output file.

User Action: Edit the output file to resolve the conflicts and delete the conflict asterisks. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

CONTROL C, operation aborted by Ctrl/C

Type: Warning

Description: You pressed Ctrl/C to abort an operation in progress.

User Action: Reenter the command to start the operation from the beginning.

CONVERTED, Version 2 library converted to Version 3 format

Type: Success

Description: CMS has successfully converted a Version 2 library to Version 3.0 format. Once you have converted the library, you can use CMS Version 3.0 to manipulate it.

User Action: None.

CONVERTLIB, 'directory ' is a Version 'number ' library and cannot be used without conversion

Type: Error

Description: You have tried to set your library to a directory that may contain library control structures created by a previous version of CMS. Before you can use the library, you must convert it.

User Action: Use the CONVERT LIBRARY command to convert a Version 2 library to Version 3.0 or higher. If the library is a Version 1 library, you should try to use the CMSCONVERTLIB program that was installed on your system with Version 2; however, conversion of Version 1 libraries is no longer supported.

CONVNOTNEC, library 'name ' is already in the current format

Type: Error

Description: You have attempted to convert a library that is already in the current format. No conversion is necessary.

User Action: None.

COPIED, 'operation-type ' 'library ' 'object-name ' copied to 'library ' 'object-name '

Type: Success

Description: CMS has successfully copied the indicated element.

User Action: None.

COPIES, 'number ' copy operation(s) completed

Type: Success

Description: CMS has successfully copied the indicated number of elements.

User Action: None.

CREATED, library 'directory-spec' created

CREATED, class 'name' created

CREATED, element 'name' created

CREATED, group 'name' created

Type: Success

Description: CMS has successfully created the indicated library, class, element, or group.

User Action: None.

CREATED, element 'name ' created, generation 1 reserved

Type: Success

Description: CMS has successfully created the indicated element and reserved the first generation.

User Action: None.

CREATES, 'count ' creations completed

Type: Success

Description: CMS has successfully created one or more classes, elements, or groups.

User Action: None.

DEFAULTDIR, default directory cannot be a CMS library

Type: Error

Description: You cannot set your CMS library to your default directory. In addition, you cannot specify your default directory as a parameter to the CREATE LIBRARY command, or set default to a CMS library when attempting to enter a CMS command.

User Action: Change your default directory so it is different from your CMS library and enter the command again.

DELETED, class 'name' deleted

DELETED, element 'name' deleted

DELETED, group 'name' deleted

Type: Success

Description: CMS has successfully deleted the indicated class, element, or group.

User Action: None.

DELETIONS, 'count ' deletion(s) completed

Type: Success

Description: CMS has successfully deleted the indicated number of classes, elements, or groups.

User Action: None.

DIFFCLASS, classes are different

Type: Informational

Description: CMS has detected differences between the compared classes.

User Action: None.

DIFFERENT, files are different

Type: Informational

Description: CMS has detected differences between the compared files.

User Action: None.

DUPEDF, 'name ' is a duplicate element data file

Type: Error

Description: VERIFY has detected a duplicate element data file.

User Action: Use VERIFY/REPAIR to remove the extra file.

DUPREF, 'name ' is a duplicate reference copy

Type: Error

Description: VERIFY has detected a duplicate reference copy.

User Action: Use VERIFY/REPAIR to remove the extra file.

EDFMIS, element data file 'name ' is missing

Type: Error

Description: An element name listed in the control file does not have a corresponding file in the library.

User Action: Use a backup tape to obtain the most recent copy of the element file.

ELEEXISTS, 'name ' is already an element name

Type: Error

Description: You have specified an element name that is already being used in the library. Element names must be unique.

User Action: Use a different name and enter the command again.

ELEMULTRES, element 'name ' is reserved more than once by you

Type: Error

Description: CMS cannot determine which reservation you want to replace or unreserve.

User Action: Remove the ambiguity by using either the /IDENTIFICATION_NUMBER or the /GENERATION qualifier. If you have multiple reservations on the same generation, you must use /IDENTIFICATION_NUMBER. To determine the correct identification number, use SHOW RESERVATIONS.

ELEXPIGN, element expression ignored

Type: Warning

Description: You have supplied an element expression for a VERIFY/RECOVER transaction.

User Action: Reenter the command without the element expression.

ENDPTRSTR, 'object ' end pointer is 'address '; last block is 'address '

Type: Error

Description: This is secondary to BADLIB; it indicates an inconsistency in the database that CMS cannot fix.

User Action: Use a backup copy of the library. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

EOF, end of file

Type: Warning

Description: The end-of-file was encountered. This message is used for communication between CMS and user-supplied callback routines.

User Action: None.

ERRACCEPTANCES, 'count ' generation(s) accepted and 'count ' error(s) occurred

Type: Error

Description: CMS has successfully marked the indicated number of generations for acceptance and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRANNOTATIONS, 'count ' element(s) annotated and 'count ' error(s) occurred

Type: Error

Description: CMS has successfully annotated the indicated number of elements and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRCANCELTIONS, 'count ' review(s) canceled and 'count ' error(s) occurred

Type: Error

Description: CMS has successfully removed the indicated number of the generations marked for review from the review pending list and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRCLOSE, error closing 'file name '

Type: Error

Description: CMS was unable to close the indicated file.

User Action: See the descriptions of the secondary messages for more information. If the problem involves any library files, use VERIFY.

ERRCOPIES, 'number ' copy operation(s) completed and 'number ' error(s) occurred

Type: Error

Description: CMS has successfully copied the indicated number of elements and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRCREATES, 'count ' creations completed and 'count ' error(s) occurred

Type: Error

Description: CMS has successfully completed the indicated number of create transactions and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRDELETIONS, 'count ' deletion(s) completed and 'count ' error(s) occurred

Type: Error

Description: CMS has successfully deleted the indicated number of elements, groups, or classes and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRELEHIS, error in element history data

Type: Error

Description: CMS encountered erroneous data in the element history. The element history is included in the output file when you fetch or reserve an element that has the **history** attribute defined.

User Action: Examine the file in your default directory (do not edit the library element file) to see if there is any non-history data in the element history.

ERREMOVALS, 'count ' removal(s) completed and 'count ' error(s) occurred

Type: Error

Description: CMS has successfully removed the indicated number of elements or groups and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERREPLACEMENTS, 'count ' element(s) replaced and 'count ' error(s) occurred

Type: Error

Description: CMS has successfully replaced the indicated number of elements and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRESERVATIONS, 'count ' element(s) reserved and 'count ' error(s) occurred

Type: Error

Description: CMS has successfully reserved the indicated number of elements and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRETRIEVALS, 'count ' generation(s) retrieved and 'count ' error(s) occurred

Type: Error

Description: CMS has successfully retrieved the indicated number of generations and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRFETCHES, 'count ' element(s) fetched and 'count ' error(s) occurred

Type: Error

Description: CMS has successfully fetched the indicated number of elements and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRGENDELETIONS, 'count ' element generations deleted and 'count ' error(s) occurred

Type: Error

Description: CMS has successfully deleted the indicated number of element generation ranges, and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRHISLINE, missing beginning (ending) history control line in file 'file name '

Type: Error

Description: CMS cannot find the top or bottom control line of the files history data.

User Action: Edit the file and restore the missing history control line. The beginning and ending control lines should be both identical and the same as those surrounding the original replacement history text.

ERRINSERTIONS, 'count ' insertion(s) completed and 'count ' error(s) occurred

Type: Error

Description: CMS has successfully inserted the indicated number of elements, groups, or generations, and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRMARKS, 'count ' generation(s) marked and 'count ' error(s) occurred

Type: Error

Description: CMS has successfully marked the indicated number of generations for review and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRMODACLS, 'count ' access control list(s) (ACLs) modified and 'count ' error(s) occurred

Type: Error

Description: CMS has successfully modified the indicated number of access control lists and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRMODIFIES, 'count ' modification(s) completed and 'count ' error(s) occurred

Type: Error

Description: CMS has successfully completed the indicated number of modifications and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRPAREXP, error parsing 'type ' expression

Type: Error

Description: You specified a class, element, or group with illegal syntax.

User Action: Correct the expression and enter the command again. See the descriptions of the secondary messages for more information.

ERRREJECTIONS, 'count ' generation(s) rejected and 'count ' error(s) occurred

Type: Error

Description: CMS has successfully rejected the indicated number of element generations and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRREVIEWS, 'count ' generation(s) reviewed with 'count ' error(s)

Type: Error

Description: CMS has successfully associated a review comment with the indicated number of generations and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRUNRESERVES, 'count ' element(s) unreserved with 'count ' error(s)

Type: Error

Description: CMS has successfully canceled the reservations for the indicated number of elements and encountered one or more errors during the transaction.

User Action: See the descriptions of the secondary messages for more information.

ERRVER2, internal contiguous space verified with errors

ERRVERARC, archive control block verified with errors

ERRVERCLS, class list verified with errors

ERRVERCMD, command list verified with errors

Type: Error

Description: CMS encountered one or more errors while verifying your library.

User Action: Use VERIFY/REPAIR. If it is unable to correct the problem, use a backup copy of your library. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

ERRVERCON, control file verified with errors

ERRVEREDFS, element data files verified with errors

ERRVERELE, element list verified with errors

ERRVERFRE, internal free space list verified with errors

Type: Error

Description: CMS encountered one or more errors while verifying your library.

User Action: Use VERIFY/REPAIR. If it is unable to correct the problem, restore a backup copy of your library. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

ERRVERGEN, generation 'name' verified with errors

Type: Error

Description: CMS encountered one or more errors while verifying an element generation your library.

User Action: Reissue the VERIFY/REPAIR command, or restore a backup copy of your library.

ERRVERGRP, group list verified with errors

ERRVERREFS, reference copies verified with errors

ERRVERRES, reservation list verified with errors

ERRVERSTR, internal string list structure verified with errors

Type: Error

Description: CMS encountered one or more errors while verifying your library.

User Action: Use VERIFY/REPAIR. If it is unable to correct the problem, restore a backup copy of your library. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

EXCLUDE, object excluded from consideration

Type: Success

Description: Return this value from a callback routine to indicate that the current piece of data is to be excluded from the action of the CMS routine. (See *CMS Callable Routines* for more information about the CMS\$_EXCLUDE return code.)

User Action: None.

EXIT, EXIT can be used only to leave subsystem level**Type:** Error**Description:** You entered CMS EXIT from DCL command level.**User Action:** None.**EXTENDEDLIB, library 'name' contains extended file names and cannot be used with this version of CMS****Type:** Error**Description:** The library contains element names that use the extended file names character set.**User Action:** Use the CMS version installed with extended file name support.**EXTFOUND, element 'name' is an extended file name****Type:** Error**Description:** CMS encountered an element with an extended filename.**User Action:** To rename the element, use the following command:

```
$ MODIFY ELEMENT/NAME=new_element_name old_element_name
```

FETCHED, generation 'gen-number' of element 'name' fetched**FETCHED, generation 'gen-number' of element 'name' fetched and merged with 'gen-number'****Type:** Success**Description:** CMS has successfully fetched the specified element generation from the library. CMS also returns this message when you execute a FETCH/MERGE transaction.**User Action:** None.**FETCHES, 'count ' element(s) fetched****Type:** Success**Description:** CMS has successfully fetched *count* elements.**User Action:** None.

FILEXISTS, file already exists, 'file-spec ' created

Type: Informational

Description: There is already a version of the indicated file in the directory that CMS is using as output for a command.

User Action: None.

FILINUSE, file 'file name ' in use

Type: Informational

Description: CMS is attempting to open a file that is currently locked by another user. Under these circumstances, CMS will wait and then try again.

User Action: None.

FIXCRC, checksum of element 'name ' repaired

Type: Informational

Description: CMS has repaired the checksum of the indicated element.

User Action: None.

FIXHDR, file header of element data file 'name ' repaired

Type: Informational

Description: CMS has repaired the file header of the indicated element.

User Action: None.

FREEBLKCON, 'count ' free blocks consolidated into other free blocks

Type: Informational

Description: During a VERIFY/REPAIR operation, CMS has consolidated free block space in the library to improve efficiency and performance.

User Action: None.

GENCREATED, generation 'gen-number' of element 'name' created

GENCREATED, generation 'gen-number' of element 'name' created and reserved

Type: Success

Description: CMS has successfully created the indicated element generation in the library.

User Action: None.

GENDELETED, 'count ' generation(s) of element 'name ' deleted

Type: Success

Description: CMS has successfully deleted the indicated number of generations of the specified element.

User Action: None.

GENDELETIONS, 'count ' element generations deleted

Type: Success

Description: CMS has successfully deleted generations of the indicated number of elements.

User Action: None.

GENEXISTS, generation 'gen-number ' already exists

Type: Error

Description: You have tried to execute a replacement transaction that would create a generation that already exists in the library.

User Action: To save the changes you have made to the file, you must create a variant generation.

GENINSERTED, generation 'gen-number ' of element 'name ' inserted into class 'name '

Type: Success

Description: CMS has successfully inserted the indicated element generation into the specified class.

User Action: None.

GENMULTRES, generation 'gen-exp ' of element 'name ' is reserved more than once by you

Type: Error

Description: CMS cannot determine which reservation you want to replace or unreserve.

User Action: Remove the ambiguity by using the /IDENTIFICATION_NUMBER qualifier. To determine the correct identification number, use SHOW RESERVATIONS.

GENNOINSERT, error inserting 'name ' into class 'name '

Type: Error

Description: CMS cannot insert the element generation into the class.

User Action: See the descriptions of the secondary messages for more information.

GENNOREMOVE, error removing 'name ' from class 'name '

Type: Error

Description: CMS cannot remove the element generation from the class.

User Action: See the descriptions of the secondary messages for more information.

GENNOTANC, generation 'gen-number ' is not an ancestor of 'gen-number '

Type: Error

Description: You have specified a generation that is either a descendant of, or not on the same line of descent as, another generation.

User Action: Correct the generation number and enter the command again.

GENNOTFOUND, generation 'gen-number ' of 'name ' not found

Type: Error

Description: CMS was unable to find the indicated element generation in the library.

User Action: Check to see that you are requesting a valid element generation.

GENNOTRES, generation 'gen-exp ' is not reserved by you

Type: Error

Description: You have specified a generation number that does not correspond to a reservation belonging to you.

User Action: Check your reservation generation and enter the command again.

GENRECSIZE, generation 'name' has 'size' -byte records; 'name' has 'size' -byte records

Type: Error

Description: CMS cannot merge generations with fixed-length records if the records are not of the same length.

User Action: Reserve the generation with the smaller records and convert those records to the larger size. Replace the generation and attempt the merge again, this time using the newly created generation.

GENREMOVED, generation 'gen-number' of element 'name' removed from class 'name'

Type: Success

Description: CMS has successfully removed the indicated element generation from the class you specified.

User Action: None.

GENRESREV, generation reserved or marked for review

Type: Error

Description: A deletion range cannot contain generations that are reserved or marked for review.

User Action: If the generation is reserved, either replace or unreserve it. If the generation is marked for review, accept or reject it or cancel the review. Then enter the command again.

GENTOODEEP, generation edits are nested too deeply

Type: Error

Description: The generation edits contained in the element data file are nested more deeply than the maximum allowed (400). This is a fairly rare occurrence caused by the method CMS uses to store generation edits, and is outside of the user's control.

User Action: Use the DELETE GENERATION command to delete old or unneeded generations, then enter the command again. Alternatively, you can use the reserved generation to create a new element. The generation contained in the old element can then be unreserved.

HASFILES, directory 'directory-spec' contains files

Type: Informational

Description: You have tried to create a CMS library in a directory that already contains files.

User Action: You must specify an empty directory or subdirectory as the parameter to the CREATE LIBRARY command.

HASMEMBERS, class 'name' contains one or more generations

HASMEMBERS, group 'name' contains one or more elements

Type: Error

Description: The class or group you specified contains one or more elements. A class or a group must be empty before you can delete it.

User Action: Remove any elements that belong to the group or class (see the description for REMOVE ELEMENT, REMOVE GENERATION, or REMOVE GROUP), then enter the command again.

HISNOTSTM, history file record format is not stream_lf

Type: Error

Description: The 00CMS.HIS file must be in stream_lf (line feed) format. Note that if you edit the history file, the format will no longer be stream_lf.

User Action: Use VERIFY/REPAIR.

HISTDEL, 'number ' history records deleted

Type: Success

Description: CMS has successfully deleted part or all of your library history. By default, the deleted history is placed in the file HISTORY.DMP in your default directory.

User Action: None.

IDENTICAL, files are identical

Type: Success

Description: The two files compared by CMS are identical.

User Action: None.

IDENTCLASS, classes are identical

Type: Informational

Description: The two classes compared by CMS are identical.

User Action: None.

IDENTNOTRES, reservation 'integer ' is not reserved by you

Type: Error

Description: You have specified a reservation identification number that does not correspond to a reservation belonging to you.

User Action: Enter a valid reservation identification number.

ILLACT, illegal review action

Type: Error

Description: This is a return status from the callable routine. You have specified a value for the review action that does not correspond to any known actions.

User Action: See the description of CMS\$REVIEW_GENERATION in the *VSI DECset for OpenVMS Code Management System Callable Routines Reference Manual* and correct the action value.

ILLARCREC, illegal record in archive file

Type: Error

Description: The archive file contains a record that has an invalid format, indicating corruption of the archive file.

User Action: Restore a new copy of the archive file from backup and enter the command again.

ILLCHAR, illegal character in 'string '

Type: Error

Description: You have specified a character that is not allowed in this context.

User Action: Correct the syntax and enter the command again.

ILLCLSNAM, illegal class name

Type: Error

Description: You have specified a class name incorrectly. Class names can consist of alphanumeric characters (A–Z, a–z, 0–9), hyphens (-), underscores (_), dollar signs (\$), and a period (.).

User Action: Correct the class name syntax and enter the command again.

ILLCONREC, illegal control record in library element

ILLDATREC, illegal data record format in element file

Type: Fatal

Description: CMS has encountered an illegal control record in a library element.

User Action: These messages indicate that, through some means other than CMS, the element file has become corrupted. Under these circumstances, you should substitute a recent backup copy of the element file to insure the integrity of the data. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

ILLEGALDEV, illegal device name

Type: Error

Description: You have included an illegal device specification in a directory or file specification.

User Action: Correct the device specification and enter the command again.

ILLELENAM, illegal element name

Type: Error

Description: You have specified an illegal element name.

User Action: Correct the syntax and enter the command again.

ILLELEXP, 'string ' is an illegal element expression

Type: Error

Description: You have specified an illegal element expression.

User Action: Correct the syntax and enter the command again.

ILLFORMAT, illegal /FORMAT qualifier string

Type: Error

Description: You have specified an illegal format string. In a format string, the number sign (#) indicator must be followed by a second number sign or the letter E.

User Action: Correct the format string and enter the command again.

ILLGEN, illegal generation expression

Type: Error

Description: You have specified an illegal generation expression.

User Action: Correct the generation expression and enter the command again.

ILLGRPNAM, illegal group name

Type: Error

Description: You have specified the group name incorrectly. Group names can consist of alphanumeric characters (A–Z, a–z, 0–9), hyphens (-), underscores (_), and dollar signs (\$).

User Action: Correct the syntax of the group name and enter the command again.

ILLHIST, illegal history string

Type: Fatal

Description: The history string for an element in your library contains more than one history format parameter (#H or #B).

User Action: Use the MODIFY ELEMENT/HISTORY command to establish the correct history string.

ILLNAME, 'name ' is for CMS use only

Type: Error

Description: You have used a name that is reserved for CMS use.

User Action: See *Chapter 1, "CMS Command Dictionary"* for information on name limitations.

ILLNOTE, illegal notes string

Type: Fatal

Description: The notes string for an element in your library contains more than one notes format parameter (#G).

User Action: Use the MODIFY ELEMENT/NOTES command to establish the correct notes string for the element.

ILLOBJTYP, illegal object type

Type: Error

Description: You have specified an illegal value for the object type.

User Action: See the *VSI DECset for OpenVMS Code Management System Callable Routines Reference Manual* for legal object type values.

ILLPAR, illegal format parameter in /HISTORY string

ILLPAR, illegal format parameter in /NOTES string

Type: Error

Description: You have given an illegal parameter in the value specified for the /HISTORY or /NOTES qualifier.

User Action: Correct the syntax and enter the command again.

ILLPOSVAL, /POSITION value must be from 1 to 511

Type: Error

Description: You have specified a position value that is not within the allowed range.

User Action: Correct the position value and enter the command again.

ILLREFDIR, reference copy directory cannot be a CMS library

Type: Error

Description: You cannot direct CMS to use a CMS library as a reference copy directory.

User Action: Make sure you are specifying a nonlibrary directory.

ILLRMK, illegal remark

Type: Error

Description: You have specified an illegal remark. CMS does not accept control characters in the remark. You can use only printable characters in a remark text string.

User Action: Correct the remark and enter the command again.

ILLSEQ, illegal sequence value

Type: Fatal

Description: The element file is identified as sequenced, but the format of the file is incorrect.

User Action: Substitute a backup copy of the element file. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

ILLSUBTYP, illegal object subtype: 'subtype' in SET ACL or SHOW ACL command

Type: Error

Description: You have specified an illegal object subtype.

User Action: Change the object subtype to one of the allowable keywords.

User Action: See the *VSI DECset for OpenVMS Guide to the Code Management System* for allowable keywords.

ILLVAR, illegal variant specification

Type: Error

Description: You have specified an illegal value for the /VARIANT qualifier (the value must be a single alphabetic character).

User Action: Correct the syntax and enter the command again.

INCLIBVER, library version is 'num', maximum supported version is 'num'

Type: Error

Description: An attempt was made to access a CMS library that has a database version number greater than that supported by the current version of CMS.

User Action: Upgrade to the appropriate version of CMS.

INCRANGSPEC, incomplete range specification

Type: Error

Description: A deletion range must have both beginning and ending generations specified with /FROM (or /AFTER) and /TO (or /BEFORE).

User Action: Correct the syntax and enter the command again.

INSERTED, element 'name' inserted into group 'name'

INSERTED, group 'name' inserted into group 'name'

Type: Success

Description: CMS has successfully inserted the indicated element or group into the group you specified.

User Action: None.

INSERTIONS, 'count' insertion(s) completed

Type: Success

Description: CMS has successfully inserted the indicated number of generations or elements.

User Action: None.

INUSE, library 'directory-spec' is in use, please wait

Type: Informational

Description: The library is currently being used by someone else.

User Action: None. You will need to wait until the other transaction is finished. CMS will automatically continue execution of your command as soon as the library is free. It is not necessary to reenter the command or press Ctrl/C.

INVFETDB, invalid fetch data block passed to CMS

Type: Error

Description: You have specified an invalid fetch data block (FDB). This message is always preceded by CMS\$_BADCALL.

User Action: Check to make sure you are using the correct passing mechanism for the FDB argument in the call to CMS.

INVFIXMRS, generation has fixed-length records with maximum record size of zero

Type: Error

Description: CMS has found a generation created from a file with fixed-length records that has a stored maximum record size of zero bytes. Because RMS does not allow the creation of a file with fixed-length records of length zero, CMS will be unable to fetch or reserve this generation.

User Action: Use VERIFY/REPAIR.

INVLENGTH, invalid library data block length

Type: Fatal

Description: This message indicates an incorrect library data block length. This message is always preceded by CMS\$_BADCALL.

User Action: Check to see that you have included a call to CMS\$SET_LIBRARY before calls to other CMS routines.

INVLIBDB, invalid library data block passed to CMS

Type: Error

Description: You have specified an invalid library data block. This message is always preceded by CMS\$_BADCALL.

User Action: Check to see that you have included a call to CMS\$SET_LIBRARY before calls to other CMS routines.

INVSTRDES, invalid string descriptor at virtual address 'address '

Type: Fatal

Description: You have passed an incorrect argument to a CMS routine. This message is always preceded by CMS\$_BADCALL.

User Action: Check the number of arguments in the call to CMS. In addition, check to make sure you are using the correct passing mechanism, and you are passing the arguments in the correct order.

ISMEMBER, element belongs to a group**ISMEMBER, group belongs to a group****ISMEMBER, generation belongs to a class**

Type: Error

Description: You cannot delete an element that belongs to a group or a class.

User Action: You must remove the element from any groups or classes (or remove the group from any groups) that it belongs to before you can delete it.

ISRESERVED, a generation of this element is reserved

Type: Error

Description: You can only modify the review attribute, reference copy attribute, and remark field of an element that is reserved. You cannot delete an element that is reserved.

User Action: Replace the element or cancel the reservation before continuing.

LIBALRINLIS, library 'directory-spec ' is already in the library list

Type: Error

Description: A library can appear only once in the library list; you cannot enter it twice.

User Action: Use the SET NOLIBRARY command to remove the library from the list before reentering it in its new position.

LIBINSLIS, library 'directory-spec ' inserted...

Type: Informational

Description: The indicated directory was inserted into the library list.

User Action: None.

LIBIS, library is 'directory-spec '

Type: Informational

Description: Your CMS library is set to the indicated directory.

User Action: None.

LIBLISMOD, library list modified

Type: Informational

Description: The CMS library list was modified.

User Action: None.

LIBLISNOTMOD, library list not modified

Type: Informational

Description: The CMS library list was not modified.

User Action: None.

LIBNOTINLIS, library 'directory-spec ' not found in the library list

Type: Error

Description: The indicated library was not found in the library list.

User Action: None.

LIBREMLIS, library 'directory-spec ' removed from the library list

Type: Informational

Description: The indicated directory was removed from the library list.

User Action: None.

LIBSET, library set

Type: Success

Description: CMS has successfully set your library. This message is returned from CMS \$SET_LIBRARY; however, CMS does not pass this message to the callback message routine.

User Action: None.

LONGVARFOUND, generation 'name ' contains a long variant name

Type: Error

Description: CMS has encountered a generation with a long variant name in your library during an attempt to change the LONG_VARIANT_NAMES attribute.

User Action: Use DELETE GENERATION to remove this and any other generation that contains along variant name.

MANCONLIB, 'directory' is prior to CMS V3.5-04 library and cannot be used with the current version.

Type: Error

Description: You have tried to set your library to a directory that may contain library control structures created by a previous version of CMS. Before you can use the library, you must convert it.

User Action: Please see release notes to convert a version 3 library to current format.

MARKED, generation 'gen-number ' of element 'name ' marked for review

Type: Success

Description: CMS has successfully marked the indicated generation of the specified element for review and placed it on the review pending list.

User Action: None.

MARKS, 'count ' generation(s) marked for review

Type: Success

Description: CMS has successfully marked the indicated number of generations for review and placed them on the review pending list.

User Action: None.

MAXARG, routine called with 'count ' arguments; maximum allowed is 'count '

Type: Fatal

Description: You have tried to pass too many arguments to a CMS routine. This message is always preceded by CMS\$_BADCALL.

User Action: Remove any extra arguments from the routine call.

MERGECONFLICT, 'count ' changes successfully merged with 'count ' conflicts

Type: Warning

Description: CMS has encountered conflicts during the merge transaction. The merge conflicts are flagged with asterisks in the output file.

User Action: Edit the output file to resolve the conflicts and delete the conflict asterisks. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

MERGECOUNT, 'count ' changes successfully merged with no conflicts

Type: Informational

Description: CMS did not encounter any conflicts during the merge transaction.

User Action: None.

MERGED, generations 'gen-exp ' and 'gen-exp ' merged

Type: Informational

Description: CMS merged the indicated elements during an annotate transaction.

User Action: None.

MINARG, routine called with 'count ' arguments; minimum required is 'count '

Type: Fatal

Description: You have not provided enough arguments for a CMS routine. This message is always preceded by CMS\$_BADCALL.

User Action: Check the number of arguments in the calls to CMS.

MISBLKSTR, a 'number ' block was not hit during pass 1

Type: Error

Description: CMS has detected an error in the database. This condition prevents the proper functioning of the library and the cause of this error should be investigated.

User Action: Use a backup copy of your library.

MISMATCON, control records in library element are mismatched

Type: Fatal

Description: CMS has encountered an error in the library element format.

User Action: Substitute a recent backup copy of the element. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

MODACL, modified access control list for class 'name'

MODACL, modified access control list for element 'name'

MODACL, modified access control list for group 'name'

MODACL, modified access control list for class list

Type: Success

Description: CMS has successfully modified the ACL for the indicated library object.

User Action: None.

MODACL, modified access control list for element list

MODACL, modified access control list for group list

MODACL, modified access control list for history access

MODACL, modified access control list for library access

Type: Success

Description: CMS has successfully modified the ACL for the indicated library object.

User Action: None.

MODACL, modified access control list for 'name' command

Type: Success

Description: CMS has successfully modified the ACL for the indicated library object.

User Action: None.

MODACLS, 'count' access control list(s) modified

Type: Success

Description: CMS has successfully modified the ACLs of the indicated number of library objects.

User Action: None.

MODIFIED, class 'name' modified

MODIFIED, element 'name' modified

MODIFIED, group 'name' modified

MODIFIED, library 'directory-spec' modified

Type: Success

Description: CMS has successfully modified the attributes of the indicated class, element, group, or library.

User Action: None.

MODIFICATIONS, 'count ' modification(s) completed

Type: Success

Description: CMS has successfully modified the attributes of the indicated number of classes, elements, or groups.

User Action: None.

MSSBLKSTR, 'count ' 'type-number ' type blocks found on pass 1, and 'count ' blocks found on pass 2

Type: Error

Description: CMS has encountered an internal error in the library control structure.

User Action: Use VERIFY/REPAIR. If it is unable to correct the problem, use a backup copy of your library. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

MULTCALL, multiple calls made to CMS\$PUT_STRING

Type: Warning

Description: A callback routine has invoked the CMS\$PUT_STRING routine more than once during a single execution of the callback routine.

User Action: Check to make sure that the callback routine is doing what you intend. If CMS \$PUT_STRING is called more than once during a single invocation of a callback routine, the string buffer is overwritten with each call to CMS\$PUT_STRING.

MULTPAR, multiple format parameters in /HISTORY string

MULTPAR, multiple format parameters in /NOTES string

Type: Error

Description: CMS has encountered more than one format parameter in a history or notes string.

User Action: Remove the extra format parameter and enter the command again.

MUSTBEDIR, 'string ' must be a directory specification

Type: Error

Description: CMS expects a directory specification where you have used the indicated string.

User Action: Correct the parameter and enter the command again.

MUSTBEPOS, position value must be a positive integer

MUSTBEPOS, width value must be a positive integer

Type: Error

Description: You cannot use negative width or position values.

User Action: Correct the parameter and enter the command again.

MUTEXC, 'qualifier ' and 'qualifier ' are mutually exclusive qualifiers

Type: Error

Description: You cannot specify the indicated qualifiers on the same command line.

User Action: Enter the command again using only one of the qualifiers.

NEEDNUMBER, generation number must end with an integer

Type: Error

Description: You have specified a generation number in an illegal format.

User Action: Correct the generation number and enter the command again.

NEEDPERIOD, element name must include or end with a period

Type: Error

Description: The element name you have specified does not contain a period.

User Action: Correct the element name parameter and enter the command again.

NETNOTALL, network access not allowed

Type: Error

Description: You cannot use CMS to access libraries on remote nodes.

User Action: None.

NOACCEPT, error accepting 'name '

Type: Error

Description: CMS was unable to remove the generation of the specified element from the review pending list and mark it as accepted.

User Action: See the descriptions of the secondary messages for more information.

NOACCESS, no 'type' access to class 'name'

NOACCESS, no 'type' access to element 'name'

NOACCESS, no 'type' access to group 'name'

NOACCESS, no 'type' access to class list

Type: Error

Description: The ACL for the specified object does not give you the required access to perform the operation.

User Action: Use the SET ACL command to modify the ACL of the appropriate object.

NOACCESS, no 'type' access to element list

NOACCESS, no 'type' access to group list

NOACCESS, no 'type' access to history

NOACCESS, no 'type' access to library

Type: Error

Description: The ACL for the specified object does not give you the required access to perform the operation.

User Action: Use the SET ACL command to modify the ACL of the appropriate object.

NOACCESS, no 'type' access to 'name' command

Type: Error

Description: The ACL for the specified object does not give you the required access to perform the operation.

User Action: Use the SET ACL command to modify the ACL of the appropriate object.

NOACE, specified access control entry does not exist

Type: Warning

Description: The ACE specified on the following line does not exist in the access control list.

User Action: Specify an existing ACE and enter the command again.

NOANNOTATE, error annotating 'name '

Type: Error

Description: CMS was unable to annotate the indicated element.

User Action: See the descriptions of the secondary messages for more information.

NOBACKUP, no backup for data file 'name '

Type: Fatal

Description: CMS is unable to recover the library because the required file is missing. If a transaction is stopped so the library contains a new element file that is incomplete, the earlier version of the file is required for recovery. If someone has purged the CMS library, or if some other action has been taken that deletes the previous, complete version of the library file, CMS cannot recover the library.

User Action: Use a backup copy of the element file. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

NOBCKPTR, 'object ' 'name ' has no back pointer to 'object ' 'name '

Type: Error

Description: CMS encountered an internal error in the library control structure.

User Action: Reissue the VERIFY/REPAIR command, or restore a backup copy of your library.

NOCANCEL, error canceling review of 'name '

Type: Error

Description: CMS was unable to remove the generation of the specified element from the review pending list and cancel the review.

User Action: See the descriptions of the secondary messages for more information.

NOCHANGES, no changes

Type: Informational

Description: You have replaced an element that has not been modified since you reserved it. CMS creates a new generation that is identical to the generation that was reserved.

User Action: None.

NOCLOSE, error closing file 'name '

Type: Fatal

Description: CMS was unable to close the indicated file.

User Action: You must close the file using a non-CMS procedure, then use VERIFY.

NOCLS, no classes found

Type: Warning

Description: CMS did not find any classes that matched the input specification.

User Action: None.

NOCMD, no commands found

Type: Warning

Description: There are no CMS commands that match the input specification.

User Action: None.

NOCOMMALIST, comma list not allowed in this context: 'string '

Type: Error

Description: You cannot use a comma list when modifying the name of a class, element, or group.

User Action: You must issue the command for each object one at a time.

NOCOMPARE, error comparing files

Type: Error

Description: CMS was unable to execute the DIFFERENCES command.

User Action: See the descriptions of the secondary messages for more information.

NOCONCUR, element 'name ' is set to NOCONCURRENT access

Type: Error

Description: If an element is set to NOCONCURRENT, only one person can reserve it at a time.

User Action: To reserve the element, you must set it to /CONCURRENT, or wait until it is no longer reserved.

NOCONRES, element 'name' is reserved with NOCONCURRENT access

Type: Error

Description: You have attempted to reserve an element that is already reserved; in this case, the reservation transaction specified NOCONCURRENT access for the duration of the reservation.

User Action: You must wait until the element is replaced, or until the reservation is canceled.

NOCONVERT, error converting Version 2 library to Version 3 format

Type: Error

Description: CMS was unable to convert your Version 2 library.

User Action: Check any additional messages. Before you use CONVERT LIBRARY again, delete any new files that might have been created in the new directory; you must supply an empty directory, which CONVERT LIBRARY can use for the Version 3.0 library.

NOCOPY, error copying 'name'

Type: Error

Description: CMS was unable to copy the indicated element.

User Action: See the descriptions of the secondary messages for more information.

NOCREATE, error creating library 'directory-spec'

NOCREATE, error creating class 'name'

NOCREATE, error creating element 'name'

NOCREATE, error creating group 'name'

Type: Error

Description: CMS was unable to create the indicated library, class, element, or group.

User Action: See the descriptions of the secondary messages for more information.

NODEFACL, no default access control list associated with object type

Type: Error

Description: The /DEFAULT qualifier can be used only with object types ELEMENT, CLASS, and GROUP.

User Action: Specify a supported object type and enter the command again.

NODELACCESS, you do not have DELETE access to the element

Type: Error

Description: CMS cannot delete the specified element due to insufficient file privileges or to an ACL restriction.

User Action: Add DELETE access to the element file; or, if an ACL is in place, change the list to support DELETE access.

NODELETE, error deleting class 'name'

NODELETE, error deleting element 'name'

NODELETE, error deleting group 'name'

NODELETE, error deleting history records

Type: Error

Description: CMS was unable to delete the indicated class, element, group, or history records.

User Action: See the descriptions of the secondary messages for more information.

NODELETIONS, no 'name ' deletions performed

Type: Error

Description: CMS was unable to delete the indicated items.

User Action: See the descriptions of the secondary messages for more information.

NODELGEN1, you cannot delete generation 1

Type: Error

Description: You cannot delete generation 1 of an element.

User Action: Remove generation 1 from the deletion range specification and retry the operation.

NOELE, no elements found

Type: Warning

Description: The current library does not contain any elements that match the element name specified on the command line.

User Action: None.

NOELEENT, file 'file name ' does not have an element entry

Type: Error

Description: CMS has detected a file in your library that does not have a corresponding entry in the control file. This file has been put there by means other than CMS. If you enter the VERIFY/REPAIR command on the library in this state, CMS deletes the extraneous file.

User Action: Remove the file from the library directory.

NOEXTENDED, extended file names are not available

Type: Error

Description: An attempt has been made to set the extended filename flag on a library that does not allow extended file names. Extended file names are available only on certain OpenVMS Alpha systems.

User Action: Ask your system manager if extended file names are available on your system.

NOEXTENDEDREF, extended file names are not available for reference copy directory

Type: Error

Description: An attempt has been made to use a reference copy directory that does not allow extended file names.

User Action: Specify a reference copy directory on a disk that allows extended file names.

NOFETCH, error fetching element 'name '

Type: Error

Description: CMS was unable to fetch the indicated element. This message can be displayed during a fetch transaction, and also during a differences transaction. CMS executes a fetch transaction to use a library file in a differences transaction.

User Action: See the descriptions of the secondary messages for more information.

NOFILE, no input file found

Type: Error

Description: This message is displayed if CMS is unable to find an input file for the primary input stream for a differences transaction.

User Action: Check the arguments in the call to CMS.

NOGENBEFORE, no generation exists before 'date '

Type: Error

Description: All generations of the specified element were created after the date specified by the /BEFORE qualifier.

User Action: Reissue the command, and specify a later date with the /BEFORE qualifier.

NOGENDELETED, no generations of element 'name ' deleted

Type: Error

Description: CMS encountered an error while deleting generations of the indicated element.

User Action: See the descriptions of the secondary messages for more information.

NOGENS, no generations found in the deletion range

Type: Error

Description: The specified deletion range does not contain any generations.

User Action: None.

NOGRP, no groups found

Type: Warning

Description: The current library does not contain any groups that match the group name specified in the command line.

User Action: None.

NOHIS, no history records found

Type: Warning

Description: The current library does not contain any records of transactions associated with the specified object.

User Action: None.

NOHISNOTES, history and notes will not be included in output file

Type: Informational

Description: CMS will not provide history or notes information in the output file because the generation being fetched or reserved has fixed-length records.

User Action: To see the history and notes information for the generations you are retrieving, use the ANNOTATE command.

NOHISPAR, no history parameter specified in history string

Type: Error

Description: You did not include #H or #B in the history string.

User Action: Correct the history string and enter the command again.

NOINSERT, error inserting 'name ' into group 'name '

Type: Error

Description: CMS was unable to insert the indicated element or group into the indicated group.

User Action: See the descriptions of the secondary messages for more information.

NOINPUT, no input data supplied by callback routine

Type: Error

Description: Your callback routine has not provided CMS with input.

User Action: Check to see that the callback routine contains a call to the CMS\$PUT_STRING routine.

NOMARK, error marking 'name '

Type: Error

Description: CMS was unable to mark a generation of the specified element for review.

User Action: See the descriptions of the secondary messages for more information.

NOMODACL, error modifying access control list for class 'name'

NOMODACL, error modifying access control list for element 'name'

NOMODACL, error modifying access control list for group 'name'

NOMODACL, error modifying access control list for class list

Type: Error

Description: CMS was unable to modify the access control list for the indicated library object.

User Action: See the descriptions of the secondary messages for more information.

NOMODACL, error modifying access control list for element list

NOMODACL, error modifying access control list for group list

NOMODACL, error modifying access control list for history access

NOMODACL, error modifying access control list for library access

Type: Error

Description: CMS was unable to modify the access control list for the indicated library object.

User Action: See the descriptions of the secondary messages for more information.

NOMODACL, error modifying access control list for 'name' command

Type: Error

Description: CMS was unable to modify the access control list for the indicated library object.

User Action: See the descriptions of the secondary messages for more information.

NOMODARG, arguments do not specify any modifications to 'object '

Type: Error

Description: You have not provided the necessary arguments to change the characteristics of the specified element, group, class, or library.

User Action: Add a qualifier to your command line specifying a modification and enter the command again. If you are using a CMS\$MODIFY_xxx routine, check the arguments in the call to CMS.

NOMODIFY, error modifying class 'name'

NOMODIFY, error modifying element 'name'

NOMODIFY, error modifying group 'name'

NOMODIFY, error modifying library 'directory-spec'

Type: Error

Description: CMS was unable to modify the attributes for the indicated class, element, group, or library.

User Action: See the descriptions of the secondary messages for more information.

NOOBJ, no objects found

Type: Warning

Description: The current library does not contain any objects that match the objects you specified.

User Action: None.

NOOBJTYP, no object type specified

Type: Error

Description: You have not specified an object type.

User Action: Use the /OBJECT_TYPE qualifier to specify the object type.

NORECOVER, error recovering library

Type: Error

Description: CMS is unable to recover your library.

User Action: Use a backup copy of the library. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

NOREF, error referencing 'directory-spec '

Type: Error

Description: CMS is unable to use the indicated directory.

User Action: See the descriptions of the secondary messages for more information.

NOREFDIR, no reference copy directory set for current library

Type: Error

Description: You have tried to give an element the reference copy attribute in a library that does not have an associated reference copy directory.

User Action: Either specify a reference copy directory for the library using the MODIFY LIBRARY command, or do not specify the reference copy attribute for the element.

NOREFELE, reference copy found for element with the /NOREFERENCE_COPY qualifier.

Type: Error

Description: CMS has discovered a reference copy for an element with the /NOREFERENCE_COPY qualifier.

User Action: Use VERIFY/REPAIR to delete the unwanted reference copy.

NOREJECT, error rejecting 'name '

Type: Error

Description: CMS was unable to remove the generation of the specified element from the review pending list and mark it as canceled.

User Action: See the descriptions of the secondary messages for more information.

NOREMARK, error adding remark to library

Type: Error

Description: CMS was unable to enter your remark in the library history.

User Action: See the descriptions of the secondary messages for more information.

NOREMOVAL, error removing 'name ' from group 'name '

Type: Error

Description: CMS was unable to remove the indicated element or group from the indicated group.

User Action: See the descriptions of the secondary messages for more information.

NOREPAIR, error repairing library

Type: Error

Description: CMS was unable to repair your library.

User Action: Use a backup copy of your library. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

NOREPBCKPTR, 'object ' 'name ', backpointer to 'object ' 'name ' not repaired

Type: Error

Description: CMS was unable to repair the specified backpointer list.

User Action: Reissue the VERIFY/REPAIR command.

NOREPCMD, command 'command-name ' not repaired

Type: Error

Description: CMS was unable to add the indicated command to the library's list of commands.

User Action: See the descriptions of the secondary messages for more information.

NOREPEDF, cannot repair 'name '

Type: Error

Description: CMS is unable to repair the indicated element data file.

User Action: Use a backup copy of the element file. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

NOREPGENMRS, maximum record size of generation 'number ' of element 'name ' not repaired

Type: Error

Description: CMS was unable to repair the maximum record size stored for the indicated generation.

User Action: Either correct the conditions indicated by the secondary messages and use VERIFY/REPAIR again, or delete the element and create it again.

NOREPLACE, error replacing 'name '

Type: Error

Description: CMS was unable to replace the indicated element in your library.

User Action: See the descriptions of the secondary messages for more information.

NOREPREF, cannot repair reference copy for 'name '

Type: Error

Description: CMS is unable to repair the indicated reference copy.

User Action: Use a backup copy of the reference copy. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

NOREPRO, cannot replace a read-only element

Type: Error

Description: The protection associated with the element file has been changed since you reserved it.

User Action: Change the file protection to allow both read and delete access.

NOREPSEQDATA, cannot repair sequence data for element

Type: Error

Description: During a VERIFY/REPAIR operation, CMS encountered an inconsistency in the internal attributes of a sequenced element. CMS cannot repair this type of corruption.

User Action: Replace the element with a previous, backup copy.

NORES, no reservations found

Type: Warning

Description: None of the elements in your library are reserved.

User Action: None.

NORESERVATION, error reserving 'name '

Type: Error

Description: CMS was unable to reserve the indicated element.

User Action: See the descriptions of the secondary messages for more information.

NORESNOCON, cannot reserve element 'name ' as NOCONCURRENT (other reservers)

Type: Error

Description: You cannot specify the /NOCONCURRENT qualifier when you reserve an element that is already reserved by someone else.

User Action: Reserve the element concurrently, or wait until the element is replaced or the reservation is canceled.

NORESRO, cannot reserve a read-only element

Type: Error

Description: The element file does not allow the required access for you to reserve the element.

User Action: Change the access so you have both read and delete access.

NORETRIEVE, error retrieving generation 'number ' from archive 'file-spec '

Type: Error

Description: CMS was unable to retrieve the indicated generation from the indicated archive file.

User Action: See the descriptions of the secondary messages for more information.

NOREV, no reviews pending

Type: Warning

Description: None of the generations specified have reviews pending.

User Action: None.

NOREVIEW, error reviewing 'name '

Type: Error

Description: CMS was unable to associate a review remark with the associated generation of the specified element.

User Action: See the descriptions of the secondary messages for more information.

NOREVPEND, generation 'gen-number ' of 'name ' does not have a review pending

Type: Error

Description: The indicated generation of the specified element has no reviews pending.

User Action: None.

NOREVSPEND, no reviews pending for element 'name '**Type:** Error**Description:** The indicated element has no generations with reviews pending.**User Action:** None.**NORMAL, normal successful completion****Type:** Success**Description:** This code indicates the successful completion of a user-supplied routine that is invoked from a callable CMS routine.**User Action:** None.**NOSINCE, error executing /SINCE operation****Type:** Error**Description:** CMS is unable to display the library history in the method indicated by /SINCE.**User Action:** See the descriptions of the secondary messages for more information.**NOSRCHLST, search lists are not allowed in this context****Type:** Error**Description:** You cannot use an OpenVMS search list as your default library and set your CMS library search list to a single library. Also, you must not define CMS\$LIB for your own purposes; the CMS\$LIB logical name is reserved for CMS use only.**User Action:** None.**NOSUPERSEDE, element 'name ' is not in class 'name ', cannot supersede****Type:** Error**Description:** You cannot use /SUPERSEDE if the class does not already contain a generation from the specified element.**User Action:** See the description of the /ALWAYS, /IF_ABSENT, and /SUPERSEDE qualifiers for the INSERT GENERATION command in *Chapter 1, "CMS Command Dictionary"*.

NOTBYCMS, data file 'name ' not closed by CMS

Type: Error

Description: Someone has used something other than CMS to access your CMS library.

User Action: Use VERIFY/REPAIR.

NOTCOMPLETED, last transaction was not completed

Type: Error

Description: The last library transaction was interrupted before completion. Because an incomplete transaction leaves your library in a potentially inconsistent state, you cannot manipulate the library until you have recovered the library.

User Action: Use VERIFY/RECOVER.

NOTCMSLIB, 'directory-spec ' is not a CMS library

Type: Error

Description: You have specified a directory that is not a valid CMS library.

User Action: Correct the directory specification and enter the command again.

NOTCRELIB, first history record is not a CREATE LIBRARY transaction

Type: Warning

Description: The first history record of every history file should be a CREATE LIBRARY transaction. It is likely that the history file has been edited. This condition does not prevent CMS from using the file.

User Action: None.

NOTDIRDES, 'generation ' is not a direct descendant of 'generation '

Type: Error

Description: The first generation indicated must be a descendant on the same line of descent as the second generation indicated.

User Action: None.

NOTESVALREQ, /NOTES value required for the position attribute

Type: Warning

Description: If you establish the **position** attribute for an element, you must also supply a notes string.

User Action: Establish a notes string for the element.

NOTFOUND, class 'name' not found

NOTFOUND, element 'name' not found

NOTFOUND, group 'name' not found

NOTFOUND, reservations for 'name' not found

Type: Error

Description: CMS was unable to find the indicated class, element, group, or reservation.

User Action: None.

NOTLOGGED, 'command-name ' commands are not logged in the library history

Type: Informational

Description: You have attempted to show the history of ANNOTATE, DIFFERENCES, or SHOW commands. CMS does not log these commands in the library history; however, CMS does not prevent you from including them in the syntax of the SHOW HISTORY command.

User Action: None.

NOTNOREF, not all elements are set to /NOREFERENCE_COPY

Type: Error

Description: You cannot issue the MODIFY LIBRARY/NOREFERENCE_COPY command unless all elements are set with /NOREFERENCE_COPY.

User Action: Use the MODIFY ELEMENT/NOREFERENCE_COPY command on all elements with the reference copy attribute and issue the MODIFY LIBRARY command again. Use the SHOW ELEMENT/FULL command to list the reference copy attribute of your elements.

NOTRESBYOU, element 'name' is not reserved by you

Type: Error

Description: Unless you have BYPASS access, you cannot replace an element that you have not reserved.

User Action: None. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

NOTSET, library not set

Type: Error

Description: CMS was unable to set your library to the specified directory.

User Action: Make sure the directory is a valid library (or an empty directory for CREATE LIBRARY) and enter the command again.

NOTTHERE, element is not in class

NOTTHERE, element is not in group

NOTTHERE, group is not in group

Type: Error

Description: CMS is unable to execute a command such as REMOVE GENERATION, REMOVE ELEMENT, or REMOVE GROUP because the element or group you specified does not exist in the specified location.

User Action: Use the /CONTENTS qualifier with the SHOW CLASS or SHOW GROUP command.

NOUNRESERVE, error unreserving 'name'

Type: Error

Description: CMS was unable to unreserve the indicated element.

User Action: See the descriptions of the secondary messages for more information.

NOVERIFY, error verifying library

Type: Error

Description: CMS was unable to verify your library.

User Action: For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

NOWLDCARD, wildcards are not allowed in this context: 'string '

Type: Error

Description: You cannot use wildcard characters in the indicated context.

User Action: Correct the syntax and enter the command again.

NULLARG, required argument is null

Type: Fatal

Description: The CMS routine call requires an argument that you have not supplied.

User Action: Check to see that you have specified all the necessary arguments, and that you have supplied placeholders where required.

NULLSTR, null string is not allowed in this context: 'string '

Type: Error

Description: You have specified a null string.

User Action: Make sure you have supplied the correct arguments, and have used placeholders where necessary.

ONEPERIOD, only one period is allowed in an element name

Type: Error

Description: The element name that you specified contains more than one period.

User Action: Make sure the element name contains only one period and enter the command again.

OPENARC, error opening archive file 'file-spec '

Type: Error

Description: CMS was unable to open the indicated archive file for input or output, depending on the operation.

User Action: See the descriptions of the secondary messages for more information.

OPENIN, error opening 'file-spec' as input

OPENIN1, error opening 'file-spec' as input

OPENIN2, error opening 'file-spec' as input

Type: Error

Description: CMS was unable to open the indicated file. OPENIN1 and OPENIN2 are used to identify the first and second file specifications given as arguments to DIFFERENCES.

User Action: See the descriptions of the secondary messages for more information.

OPENOUT, error opening 'file-spec ' for output

Type: Error

Description: CMS was unable to open the indicated file for output.

User Action: See the descriptions of the secondary messages for more information.

OVERDRAFT, disk quota exceeded; using overdraft for 'file-spec '

Type: Informational

Description: You have exceeded your disk quota during a CMS transaction. In this case, CMS uses overdraft quota to finish the transaction.

User Action: Reset your quota or remove some files so you can continue.

POSVALREQ, /POSITION value required for the notes attribute

Type: Error

Description: If you establish the **notes** attribute for an element, you must also supply a position value.

User Action: Establish a position value for the element.

PROCEEDING, proceeding...

Type: Informational

Description: The library is now available for your transaction.

User Action: None.

QUALCONFLICT, conflicting qualifiers

Type: Error

Description: You have specified conflicting qualifiers on the same command line.

User Action: Correct the syntax and enter the command again.

READERR, error reading from 'filename'

READIN, error reading 'filename'

Type: Error

Description: CMS was unable to read the indicated file.

User Action: See the descriptions of the secondary messages for more information.

READONLY, class 'name' is set to READ_ONLY

READONLY, group 'name' is set to READ_ONLY

Type: Error

Description: You cannot modify the characteristics of a class or group that is set to READ_ONLY.

User Action: Change the class or group to NOREAD_ONLY, then modify the other characteristics.

RECGRP, inserting 'name ' into 'name ' would create a recursive group

Type: Error

Description: You cannot insert one group into another if there is already a connection between the two groups.

User Action: Make sure you are establishing the group hierarchy that you intend.

RECNOTNEC, recovery is not necessary; the library is in a safe state

Type: Error

Description: You have entered VERIFY/RECOVER for a functioning library.

User Action: None.

RECOVERED, library 'directory-spec ' recovered

Type: Success

Description: CMS has successfully recovered your library.

User Action: None.

REFMISMAT, element has /REFERENCE_COPY attribute, library does not

Type: Error

Description: The element you specified has the /REFERENCE_COPY attribute enabled, but there is no reference copy directory set for the library. This situation can occur when using a library that was created with CMS Version 2, because that version did not enforce any correlation between the attributes.

User Action: Use VERIFY/REPAIR or MODIFY ELEMENT/NOREFERENCE_COPY to correct the mismatched attributes.

REFMISS, reference copy for element 'name ' is missing

Type: Error

Description: An element with the /REFERENCE_COPY attribute does not have a corresponding reference copy.

User Action: Use VERIFY/REPAIR to create a new reference copy for the element.

REFREPAIR, reference copies must be repaired – use VERIFY/REPAIR

Type: Warning

Description: This message appears after MODIFY LIBRARY has been used to change the reference copy directory, if the contents of the directory do not agree with the contents of the CMS library. CMS must repair the files in the reference copy directory before further operations can be done.

User Action: Use either VERIFY/REPAIR or MODIFY LIBRARY to change the reference copy directory. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System* and the description of the VERIFY command in *Chapter 1, "CMS Command Dictionary"*.

REJECTED, generation 'gen-number ' of element 'name ' rejected

Type: Success

Description: CMS has successfully marked a generation as rejected and removed it from the review pending list.

User Action: None.

REJECTIONS, 'count ' generation(s) rejected

Type: Success

Description: CMS has successfully marked the indicated number of generations as rejected and removed them from the review pending list.

User Action: None.

REMARK, remark added to history file

Type: Success

Description: CMS has successfully added your remark to the history file.

User Action: None.

REMOVALS, 'count ' removals completed

Type: Success

Description: CMS has successfully completed the indicated number of removals from one or more classes or groups.

User Action: None.

REMOVED, element 'name' removed from group 'name'

REMOVED, group 'name' removed from group 'name'

Type: Success

Description: CMS has successfully removed the indicated element or group from the indicated containing group.

User Action: None.

REPAIRED, library 'directory-spec ' repaired

Type: Success

Description: CMS has successfully repaired your library.

User Action: None.

REPBADLST, 'object ' list for generation 'name ' repaired

Type: Informational

Description: CMS has successfully repaired the specified condition in your library.

User Action: None.

REPBADTYP, 'object ' block type repaired

Type: Informational

Description: CMS has successfully repaired the specified condition in your library.

User Action: None.

REPBACKPTR, 'object ' 'name ', backpointer to 'object ' 'name ' repaired

Type: Informational

Description: CMS has successfully repaired the specified condition in your library.

User Action: None.

REPCMD, command 'command-name ' repaired

Type: Success

Description: CMS has successfully added *command-name* to the library's list of commands.

User Action: None.

REPCNTSTR, 'object ' count repaired

Type: Informational

Description: CMS has successfully repaired the specified condition in your library.

User Action: None.

REPDEL, extraneous file 'name ' deleted

Type: Informational

Description: VERIFY/REPAIR has deleted a file from the library. The file did not have a corresponding entry in the CMS control file. See the description of the error NOELEENT for more information.

User Action: None.

REPEDF, element 'name' repaired

Type: Informational

Description: CMS has either:

- Changed the file CRC as stored in the database to match the existing file
- Repaired the file header to make CMS the owner of the file again

User Action: Investigate the cause of the error. Run VERIFY to see if you receive a BADCRC message, which can indicate that the element data file has been altered in an uncontrolled way.

REPENDPTR, 'object' list end pointer repaired

Type: Informational

Description: CMS has successfully repaired the specified condition in your library.

User Action: None.

REPGENMRS, maximum record size of generation 'number' of element 'name' repaired to 'size' bytes

Type: Informational

Description: CMS has repaired the stored maximum record size for the indicated generation.

User Action: None.

REPLACEMENTS, 'count' element(s) replaced

Type: Success

Description: CMS has successfully replaced the indicated number of elements in the library.

User Action: None.

REPMISBLK, a type 'object' missed block repaired

Type: Informational

Description: CMS has successfully repaired the specified condition in your library.

User Action: None.

REPREF, reference copy for element 'name' repaired

Type: Informational

Description: CMS has repaired the indicated reference copy.

User Action: None.

RESERVATIONS, 'count' element(s) reserved

Type: Success

Description: CMS has successfully reserved the indicated number of elements.

User Action: None.

RESERVED, generation 'gen-number' of element 'name' reserved

RESERVED, generation 'gen-number' of element 'name' reserved and merged with 'gen-number'

Type: Success

Description: CMS has successfully reserved the indicated element generation.

User Action: None.

RETRIEVALS, 'count' generation(s) retrieved

Type: Success

Description: CMS has successfully retrieved the indicated number of generations.

User Action: None.

RETRIEVED, generation 'gen-number' of element 'name' retrieved from 'archive-file'

Type: Success

Description: CMS has successfully retrieved the indicated element generation.

User Action: None.

REVIEWED, generation 'gen-number' of element 'name' reviewed

Type: Success

Description: CMS has successfully associated a review remark with the indicated generation of the specified element.

User Action: None.

REVIEWS, 'count ' generation(s) reviewed

Type: Success

Description: CMS has successfully associated a review remark with the indicated number of generations.

User Action: None.

REVPENDING, a generation of this element has a review pending

Type: Error

Description: You cannot delete an element if one of its generations has a review pending.

User Action: Accept or reject the generation, or cancel the review; then enter the command again.

SAMELINE, generations 'gen-exp ' and 'gen-exp ' are on the same line of descent

Type: Error

Description: You cannot merge two element generations that are on the same line of descent.

User Action: Make sure you are specifying the correct generations.

SEQFAIL, illegal sequence value in library element

Type: Error

Description: A sequence value field in a record of an element file has been corrupted.

User Action: Use a backup copy of the element. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

SEQISMAT, one or more generations are sequenced, element is not

Type: Error

Description: During a VERIFY operation, CMS encountered an inconsistency in the internal attributes of a sequenced element. CMS cannot repair this type of corruption.

User Action: Replace the element with a previous, backup copy.

SEQUENCED, normal successful completion – sequenced open

Type: Success

Description: A CMS\$FETCH_OPEN transaction was completed successfully.

User Action: None.

SIZEMISMAT, cannot merge generations with different-sized records

Type: Error

Description: CMS cannot merge two generations with fixed-length records if the records are not of the same length.

User Action: Reserve the generation with the smaller records and convert those records to the larger size. Then replace the generation and attempt the merge again, this time using the newly created generation.

STARTHIS, library history starts at 'date '

Type: Error

Description: You have specified a /BEFORE date prior to the date of the CREATE LIBRARY transaction for this library.

User Action: Correct the date and enter the command again.

STOPPED, wildcard action terminated

Type: Success

Description: You have terminated a wildcard action from a user-supplied routine that is invoked by a callable CMS routine.

User Action: None.

SUPERSEDED, library list superseded

Type: Informational

Description: The previously existing library list has been superseded by the library or libraries you specified.

User Action: None.

SYSTIMDIF, last transaction executed at 'time '

Type: Error

Description: This message is secondary to SYSTIMERR.

User Action: See the explanation of the error SYSTIMERR.

SYSTIMERR, system time has been set incorrectly

Type: Error

Description: CMS has detected an error in system time. CMS displays this message if the current system time is less than the time of the last transaction. One of three situations might have occurred: the system time was inadvertently set back; at some point the system was set ahead and then later corrected (and any transactions that were executed during this time period might appear to have occurred in the future); or there is a difference between the times of two nodes on a cluster.

User Action: Check to see that the system time is correct. If your system is part of a cluster, check to see that the system times for all the nodes on the cluster are correct. Correct the system times and use VERIFY/REPAIR if necessary. (For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.)

TIMEORDER, /BEFORE and /SINCE time values cannot be resolved

Type: Error

Description: You have specified an incorrect sequence of time values for this combination of qualifiers. (The /BEFORE value cannot indicate a time that occurs prior to the /SINCE value.)

User Action: Check to make sure you are entering the correct values and enter the command again.

TOOLONG, 'string ' is too long; maximum is 'count ' character(s)

Type: Error

Description: You have provided a character string that is longer than allowed in this situation.

User Action: Correct the syntax and enter the command again.

TOOMANYLIBS, too many libraries in library list

Type: Error

Description: You have attempted to specify more than 128 libraries in a library list.

User Action: Delete some libraries from the library list and enter the command again.

TRUNCLST, 'object ' list for generation 'element ' ('number ') truncated

Type: Error

Description: CMS truncated an internal list during a VERIFY/REPAIR operation.

User Action: Reissue the VERIFY/REPAIR command, or restore a backup copy of your library.

TRYAGNLAT, library locked; try again later

Type: Error

Description: The library has been locked long enough to cause CMS to exit.

User Action: Wait until the library is unlocked and enter the command again.

UNDEFLIB, library is undefined

Type: Warning

Description: Your CMS library is undefined.

User Action: Make sure you are specifying a valid CMS library.

UNFOUT, unformatted output cannot be specified for an output file

Type: Error

Description: If you provide an output file for the CMS\$DIFFERENCES routine, you cannot also direct CMS to provide unformatted output.

User Action: Make sure you are passing the correct arguments to CMS.

UNRESERVED, element 'name' unreserved

Type: Success

Description: CMS has successfully canceled the reservation for the indicated element.

User Action: None.

UNRESERVES, 'count' element(s) unreserved

Type: Success

Description: CMS has successfully canceled the indicated number of reservations.

User Action: None.

UNSUPFRMT, 'filename' is in an unsupported format

Type: Error

Description: CMS does not support the file organization, record format, or record attributes of the indicated file.

User Action: See the *VSI DECset for OpenVMS Guide to the Code Management System* for information about supported file formats.

USERECOVER, use VERIFY/RECOVER

Type: Error

Description: Your library is in an inconsistent state.

User Action: Use VERIFY/RECOVER. See the description of the VERIFY command in *Chapter 1, "CMS Command Dictionary"* for more information.

USEREPAIR, use VERIFY/REPAIR

Type: Error

Description: This message is displayed when your library contains one or more files that have an invalid checksum or do not have a checksum, or when someone has accessed your library using means other than CMS.

User Action: Use VERIFY/REPAIR. See the description of the VERIFY command in *Chapter 1, "CMS Command Dictionary"* for more information.

USERERR, error returned from a user-supplied routine

Type: Error

Description: A callback routine that you have supplied has returned an error to CMS.

User Action: Varies according to situation.

USESETLIB, use SET LIBRARY

Type: Error

Description: Your CMS library is undefined.

User Action: Use the SET LIBRARY command to access a library.

VARINRANGE, range has variants

Type: Error

Description: A deletion range cannot contain the root of a variant line of descent.

User Action: Delete the variant lines of descent explicitly, then delete the original range.

VARLETTER, variant letter is misplaced in generation number

Type: Error

Description: You have incorrectly specified a generation number.

User Action: Correct the generation number syntax and enter the command again.

VER2, internal contiguous space verified

VERARC, archive control block verified

VERCLS, class list verified

VERCMD, command list verified

Type: Informational

Description: CMS has successfully verified the indicated portion of the library.

User Action: None.

VERCON, control file verified

Type: Informational

Description: CMS has successfully verified the indicated portion of the library.

User Action: None.

VEREDF, element 'name' verified

Type: Informational

Description: CMS has verified the indicated element data file.

User Action: Investigate the cause of the error. Run VERIFY to see if you receive a BADCRC message, which indicates that the element data file has been altered in an uncontrolled way.

VEREDFERR, element 'name' verified with errors

Type: Error

Description: CMS has encountered errors during the verification transaction.

User Action: See the descriptions of the secondary messages for more information.

VEREDFS, element data files verified

Type: Informational

Description: CMS has successfully verified the elements in your library.

User Action: None.

VERELE, element list verified

VERFRE, internal free space list verified

VERGRP, group list verified

Type: Informational

Description: These messages are displayed during the verification of the library data structures.

User Action: None.

VERIFIED, library 'directory-spec ' verified

Type: Success

Description: CMS has successfully verified your library.

User Action: None.

VERLMTERR, 'directory-spec ' requires a higher version limit

Type: Error

Description: Any directory that you intend to use as a CMS library must have a version limit of at least 2.

User Action: Change the version limit of the directory and enter the command again.

VERREF, reference copy for element 'name ' verified

Type: Informational

Description: CMS has verified the indicated reference copy.

User Action: None.

VERREFERR, reference copy for 'name' verified with errors

Type: Error

Description: CMS has encountered errors during the verification transaction. This message might appear during a VERIFY operation.

User Action: See the descriptions of the secondary messages for more information.

VERREFERRW, reference copy for 'name' verified with errors

Type: Warning

Description: CMS has encountered errors during the verification transaction. This message might appear during a MODIFY LIBRARY operation.

User Action: See the descriptions of the secondary messages for more information.

VERREFS, reference copies verified

Type: Informational

Description: CMS has successfully verified the reference copies for the elements in your library.

User Action: None.

VERRES, reservation list verified

VERSTR, internal string list structure verified

Type: Informational

Description: These messages are displayed during the verification of the library data structures.

User Action: None.

WAITING, library 'directory-spec' is still in use

WAITING, file 'name' is still in use

Type: Informational

Description: The INUSE message displayed earlier is still in effect. Another user is still accessing the library.

User Action: None. CMS automatically continues execution of your command as soon as the library is free. It is not necessary to reenter the command or press Ctrl/C.

WILDCONFLICT, wildcard conflict between element name and /INPUT

Type: Error

Description: You have provided a wildcard input file specification without the corresponding element name wildcard.

User Action: Correct the syntax and enter the command again.

WILDNEEDED, Wildcard needed: 'string '

Type: Error

Description: You have used a comma list in a context where a wildcard is necessary to resolve an expression, for example, in MODIFY ELEMENT/NAME or COPY ELEMENT.

User Action: Correct the syntax and enter the command again.

WILDVER, version number wildcard not allowed in element expressions

Type: Error

Description: CMS does not allow version number wildcards in element creation or replacement transactions.

User Action: Correct the syntax and enter the command again.

WRITEERR, error writing to 'file name '

Type: Error

Description: CMS was unable to write to the indicated file.

User Action: See the descriptions of the secondary messages for more information. If the problem involves any library files, use the VERIFY command.

ZEROADD, address of zero required in call frame entry

Type: Error

Description: Your call to the CMS\$UNRESERVE routine does not contain the required address of zero in the call frame.

User Action: Correct the arguments in the call.

ZLENBLK, zero-length block found during pass 2**Type:** Error**Description:** CMS has discovered a error in the control file.**User Action:** Use a backup copy of the library. For more information, see the *VSI DECset for OpenVMS Guide to the Code Management System*.

3.3. CMS Internal Diagnostic Messages

The following table lists the CMS internal diagnostic messages. If you should receive any of these messages, please contact your VSI product support representative.

Name	Message	Severity
ARCORDER	Batch archive request out of order	Fatal
ARCMAXREQ	Batch archive request too large	Fatal
ARCMAX	Too many batch archive calls	Fatal
BADDATASTR	Structure field of the passed LDB is invalid	Fatal
BADARGCOUNT	'routine' called with wrong number of arguments	Fatal
BADARGRANGE	'routine' called with too few or too many arguments	Fatal
BADBEGINTRA	Begin transaction called out of order or illegally	Fatal
BADCONFIRM	Command confirmation procedure failed	Fatal
BADINTER	Interlocking called out of order or illegally	Fatal
BADIOB	Bad IOB	Fatal
BADIOBLENGTH	The passed IOB has an invalid length	Fatal
BADLENGTH	Bad IOB length	Fatal
BOTHNULL	Both buffer pointers are null	Fatal
BUFOUTRANGE	Buffer Pointer out of range	Fatal
CALLFINDFIRST	Call CMS\$DIR_FIND first	Fatal
CANTBEWILD	Inconsistent wildcard markers	Fatal
CANTDEFINE	Cannot define logical name	Fatal
CANTDELETE	Cannot delete logical name	Fatal
CANTTRAN	Cannot translate logical name	Fatal

Name	Message	Severity
CANTREACH	Cannot get here from there	Fatal
CANTRESMRG	Cannot resolve merge, marked as conflict	Warning
CONVFAIL	Conversion failure	Fatal
CNCMISMAT	Mismatch of disable/enable CTRL-C operations	Fatal
DELINTRA	Delete not allowed in a transaction	Fatal
DELTATIME	Operand number of routine is a delta time	Fatal
EXIERR	Unknown error in exit handler	Fatal
EXINODEC	Cannot declare exit handler	Fatal
EXINOCAN	Cannot cancel exit handler	Fatal
FAOCNTMISSING	FAO count is missing from signal vector	Fatal
FAOCNTWRONG	FAO count exceeds size of message vector	Fatal
FAOMISMATCH	FAO count does not match expected value	Informational
FLDTOOBIG	Field size exceeds maximum numeric field size	Fatal
FUNNYTIME	Illegal time value	Fatal
GENSTGOVF	Generation string storage overflow	Fatal
GENSTKOVF	Generation stack overflow	Fatal
GENSTKUND	Generation stack underflow	Fatal
HIBR200	HIBERNATE asked to wait for more than 200 seconds	Fatal
IGNFAIL	CMP_IGNORE called with inappropriate arguments	Fatal
ILLARGTYP	Illegal argument type or value	Fatal
ILLGENEXP	Illegal generation expression	Fatal
ILLVALUE	Illegal numeric value	Fatal
ILLOPER	Illegal buffer operator	Fatal
INCR200	TIM_INC asked to add more than 200 seconds	Fatal
INITFAIL	Failed to initialize control structures	Fatal

Name	Message	Severity
INVGENLRN	Generation block has invalid longest record length	Error
INVHISREC	Invalid history record	Fatal
INVSEVERITY	Invalid severity value	Fatal
INVSIGNAL	Invalid signal	Fatal
LENGTHERR	Illegal length value	Fatal
LIBLOCKED	Cannot lock the library; already locked	Fatal
LINTOOBIG	Line too large for CMS	Fatal
MAXFIXUP	Maximum fixup count exceeded	Fatal
MAXSIGVEC	Maximum signal vector size exceeded	Fatal
MEMDIFF	Dynamic memory error; allocated size is not the same as deallocated size	Informational
MEMINUSE	'number' bytes of dynamic memory not deallocated	Informational
MISSING_BLOCK	Mandatory argument block not passed	Fatal
MISSING_FIELD	Mandatory field was not initialized in argument block	Fatal
MRGBUG	MERGE failure	Fatal
MUSTBECLO	File must be closed to call this routine	Fatal
NEEDDIRIOB	A directory IOB was not passed	Fatal
NOASSIGN	\$ASSIGN failed	Fatal
NOBUFSPEC	No buffer specified	Fatal
NODATETIME	Unable to obtain system date and time	Fatal
NODEALLOC	Element data storage not deallocated	Fatal
NODEASSIGN	\$DEASSGN failed	Fatal
NODIRFOUND	No directory found	Fatal
NOFREELIST	Free space list does not exist	Fatal
NOMAP	Cannot map the appropriate control structure	Fatal
NOGETJPI	\$GETJPI failed	Fatal
NOOUTPUT	No output file is open	Fatal

Name	Message	Severity
NOPARSE	Parse not yet initialized	Fatal
NOPRI	No primary input specified	Fatal
NOQIO	\$QIO failed	Fatal
NORESR	Cannot remove reservation that does not exist	Fatal
NOSEC	No secondary input specified	Fatal
NOSETRFA	Unable to set RFA \$FIND	Fatal
NOTEXPRTYPE	Not a valid expr type	Fatal
NOTRANS	End transaction called with no write transaction in progress	Fatal
NOUNLOCK	Could not unlock library	Fatal
NOUPDSEC	\$UPDSEC failed	Fatal
NOWRITELOG	Could not write log record	Fatal
NULLACELEN	Access control list entry has null length	Fatal
NYI	Not yet implemented	Error
PARSEINIT	Parse already initialized	Fatal
QTEXT	Quoted text	Fatal
RBKINPROGRESS	Recursive call to rollback	Fatal
RBKINVBLK	Invalid rollback block	Fatal
RBKNOCLOSE	Rollback cannot close 'file'	Fatal
RBKNOENDTRAN	Rollback cannot end transaction	Fatal
RBKNODELETE	Rollback cannot delete 'file'	Fatal
RBKNOIOB	IOB passed to rollback routine is null	Fatal
RBKNOTOPEN	File passed to rollback is not open	Fatal
RBKNOTTRUNCATE	Rollback cannot truncate 'file'	Fatal
RBKNOUNARCHIVE	Rollback cannot unarchive library data base	Fatal
RBKNOUNLOCK	Rollback cannot unlock library	Fatal
RBKNULBLK	Rollback block is null	Fatal
RBKOVWRRT	Rollback cannot handle OPTIONS=OVERWRITE	Fatal
RBKTRAUNLOCK	Transaction in unlocked library	Fatal
RMSPAIR	RMS message not paired with STV value, STS follows	Fatal

Name	Message	Severity
STRTOOLONG	String too long	Fatal
SYSFAIL	System service request failed	Fatal
TEXT	'string'	Informational
TRANSONLY	Operation valid only during a transaction	Fatal
UNOPENEDIOB	Unopened IOB passed	Fatal
UNSET	Argument block passed to control file routine is not set	Fatal
UPDSECCNT	'count' calls to \$UPDSEC	Informational
USNAMFAIL	Failed to get user identification	Fatal
WRONUMARG	Wrong number of arguments	Fatal
WRONG_BLOCK	The wrong type of argument was passed to a control file routine	Fatal