

# **VAX Common Data Dictionary Utilities Reference Manual**

Order Number: AA-L629E-TE

---

**August 1988**

This document describes the Dictionary Management Utility (DMU) and the Dictionary Verify/Fix Utility (CDDV).

digital equipment corporation, maynard, massachusetts

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Copyright ©1981, 1982, 1983, 1984, 1985, 1988 by Digital Equipment Corporation. All rights reserved.

The postpaid Reader's Comments forms at the end of this document request the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

ACMS	PDP	VAXcluster
CDD	RALLY	VAXinfo
DATATRIEVE	Rdb/ELN	VAX Information Architecture
DEC	Rdb/VMS	VIDA
DECnet	ReGIS	VMS
DECreporter	TDMS	VT
DECUS	TEAMDATA	
MicroVAX	UNIBUS	
MicroVMS	VAX	<b>digital</b> ™

# Contents

	Page
<b>How to Use This Manual</b>	vii
<b>1 Introduction to the Common Data Dictionary</b>	
1.1 Sample Dictionary . . . . .	1-1
1.2 CDD Types . . . . .	1-3
1.3 History Lists . . . . .	1-3
1.4 Access Control Lists . . . . .	1-3
1.5 CDD Utilities . . . . .	1-4
1.5.1 Dictionary Management Utility (DMU) . . . . .	1-4
1.5.2 Dictionary Verify/Fix Utility (CDDV) . . . . .	1-4
<b>2 DMU Command Descriptions</b>	
2.1 BACKUP Command . . . . .	2-2
2.2 COPY Command . . . . .	2-7
2.3 CREATE Command . . . . .	2-13
2.4 DELETE Command . . . . .	2-17
2.5 DELETE/HISTORY Command . . . . .	2-21
2.6 DELETE/PROTECTION Command . . . . .	2-23
2.7 EXIT Command . . . . .	2-26
2.8 EXTRACT Command . . . . .	2-27
2.9 HELP Command . . . . .	2-32
2.10 LIST Command . . . . .	2-33
2.11 MEMO Command . . . . .	2-40
2.12 PURGE Command . . . . .	2-42
2.13 RENAME Command . . . . .	2-44
2.14 RENAME/SUBDICTIONARY Command . . . . .	2-49
2.15 RESTORE Command . . . . .	2-53
2.16 SET ABORT Command . . . . .	2-58
2.17 SET DEFAULT Command . . . . .	2-59
2.18 SET PROTECTION Command . . . . .	2-60
2.19 SET PROTECTION/EDIT Command . . . . .	2-66
2.19.1 VT100 and VT200 Keypad Functions . . . . .	2-69
2.19.2 VT52 Keypad Functions . . . . .	2-74
2.19.3 The SAMPLE USER SCREEN . . . . .	2-78
2.19.3.1 Entering passwords . . . . .	2-78
2.19.3.2 PASS_THRU privilege error . . . . .	2-78
2.20 SHOW DEFAULT Command . . . . .	2-81
2.21 SHOW PROTECTION Command . . . . .	2-82
2.22 SHOW VERSION Command . . . . .	2-84

### **3 Dictionary Verify/Fix Utility**

3.1	Restrictions . . . . .	3-2
3.2	COMPRESS Command . . . . .	3-3
3.3	EXIT Command (CTRL/Z) . . . . .	3-5
3.4	FIX Command . . . . .	3-6
3.5	HELP Command . . . . .	3-10
3.6	SHOW VERSION Command . . . . .	3-11
3.7	VERIFY Command . . . . .	3-12

### **A Sample CDDV VERIFY Listing of a Corrupt Dictionary File**

### **B Sample CDDV FIX Listing**

### **C Sample CDDV VERIFY Listing of an Uncorrupted Dictionary File**

### **D Error Messages**

D.1	DMU Error Messages . . . . .	D-3
D.2	CDDV Error Messages . . . . .	D-19
	D.2.1 Messages to the Terminal . . . . .	D-20
	D.2.2 Report Messages . . . . .	D-23
D.3	CDD Error Messages . . . . .	D-24

### **E Additional DMU Notes**

E.1	Subdictionary File Specifications Can Contain Any System Logical Name . . . . .	E-1
E.2	DECReporter Support . . . . .	E-1
E.3	Using Multiple Versions of CDD Objects with DATATRIEVE . . . . .	E-2
E.4	Installing CDD with SYSPRV . . . . .	E-2
E.5	Path Names Pointing to Subdictionary Files . . . . .	E-2
E.6	DMU Restrictions . . . . .	E-2
	E.6.1 Copying and Renaming a Dictionary Object Protected by a Password . . . . .	E-2
	E.6.2 Using /STAGE with DMU RESTORE . . . . .	E-3
	E.6.3 Using a Wildcard with the DELETE/SUBDICTIONARY Command . . . . .	E-3

### **Examples**

1-1	Listing of the Sample Dictionary . . . . .	1-3
-----	--	-----

## Figures

1-1	Sample Dictionary Hierarchy . . . . .	1-2
2-1	Sample ACL Editor Entry Display . . . . .	2-68
2-2	VT100 and VT200 Keypad Functions . . . . .	2-69
2-3	VT52 Keypad Functions . . . . .	2-74
2-4	SAMPLE USER Screen with First 10 Path Names Displayed . . . . .	2-79
2-5	SAMPLE USER Screen with Last 10 Path Names Displayed . . . . .	2-80

## Tables

2-1	/ITEM Options . . . . .	2-35
D-1	Explanation of Severity Codes . . . . .	D-2



## How to Use This Manual

This manual describes the Dictionary Management Utility (DMU) and the Dictionary Verify/Fix Utility (CDDV).

VAX CDD (Common Data Dictionary) software, also referred to in this document as CDD, is a central storage repository for shared data definitions. These definitions are stored hierarchically and accessed by reference to unique dictionary path names. The DMU and the CDDV help you organize and maintain the dictionary structure.

CDD is now a subset of the VAX CDD/Plus software, also referred to in this document as CDD/Plus. Many DIGITAL products, however, continue to function using Version 3.4 or earlier of CDD.

Version 3.4 and earlier of CDD use the DMU format for dictionary definitions, but not the CDO format. This manual is for dictionary users who need to use the DMU format for dictionary definitions.

If Version 3.4 or earlier of the VAX Common Data Dictionary is installed on your system, references in this manual to the “VAX Common Data Dictionary,” “Common Data Dictionary,” or “CDD” refer to the VAX Common Data Dictionary installed on your system.

If VAX CDD/Plus V4.0 or later is installed on your system, references in this manual to the “VAX Common Data Dictionary,” “Common Data Dictionary,” or “CDD” refer to the DMU format dictionary.

CDD/Plus supports dictionary definitions in two distinct formats:

- **DMU format** includes dictionary definitions that can be created and manipulated with the DMU, CDDL, and CDDV utilities, and other products that do not support the new features of CDD/Plus.

- **CDO format** includes dictionary definitions that can be created and manipulated with the CDO utility, the CDD/Plus call interface, and other supporting products.

## Intended Audience

The audience for this manual includes:

- The data administrator or system manager responsible for creating the directory hierarchy, setting up the security provisions, and maintaining the dictionary structure
- Programming supervisors responsible for maintaining portions of the dictionary hierarchy
- Programmers responsible for maintaining their own portions of the directory hierarchy, and for writing applications that use the data definitions stored in the CDD

Before you read this manual, you should be familiar with the terms and concepts described in Appendix A of the *VAX CDD/Plus User's Guide*.

## Operating System Information

For information on the compatibility of other software products with this version of CDD/Plus, refer to the System Support Addendum (SSA) that comes with the Software Product Description (SPD). You can use the SPD/SSA to verify which versions of your operating system are compatible with this version of CDD/Plus.

## Structure

This manual consists of three chapters, five appendixes, and an index.

- |            |  |
|------------|--|
| Chapter 1  | Provides a brief introduction to the CDD directory hierarchy and the DMU and CDDV utilities. |
| Chapter 2  | Explains and shows syntax for the Dictionary Management Utility (DMU).                       |
| Chapter 3  | Explains and shows syntax for the Dictionary Verify/Fix Utility (CDDV).                      |
| Appendix A | Contains a sample CDDV VERIFY listing of a corrupted dictionary.                             |
| Appendix B | Contains a sample CDDV FIX listing of a corrupted dictionary.                                |

- Appendix C Contains a sample CDDV VERIFY listing of an uncorrupted dictionary.
- Appendix D Documents the error messages generated by DMU, CDDV, and CDD.
- Appendix E Contains changes made with Version 3.4 of CDD that were not described earlier in this document.

## Related Documents

For up-to-date references to further information on the topics covered in this manual, see the prefaces of the VAX CDD/Plus manuals in this documentation set.

## Conventions

This section explains the conventions for the syntax and symbols used in this manual:

- WORD** An uppercase word in a syntax format is a keyword. You must include it in the statement if the clause is used.
- word** A lowercase word in a syntax format indicates a syntax element that you supply.
- [ ]** Square brackets enclose optional clauses from which you can choose one or none.
- { }** Braces enclose clauses from which you must choose one alternative.
- [ | | ]** Bars in square brackets indicate that you can choose any combination of the enclosed options, but you can use each option only once.
- <RET>** This symbol indicates the RETURN key.  
  
Unless otherwise indicated, end all user input lines in examples by pressing the RETURN key.
- <CTRL/x>** This symbol tells you to press the CTRL (control) key and hold it down while pressing a letter key.
- <GOLD-x>** This symbol indicates that you press the GOLD key and then a specified letter key consecutively.
- ...** A horizontal ellipsis means you can repeat the previous item.

. A vertical ellipsis in an example means that information not  
: directly related to the example has been omitted.  
Color Color in examples shows user input.

## References to Products

CDD is a member of the VAX Information Architecture, a group of products that work with each other and with VAX languages conforming to the VAX calling standard to provide flexible solutions for information management problems.

VAX Information Architecture documentation explaining how these products interrelate is included with VAX CDD/Plus documentation. VAX Information Architecture documentation is also available separately. Contact your DIGITAL representative.

The CDD documentation to which this manual belongs often refers to products that are part of the VAX Information Architecture by their abbreviated names:

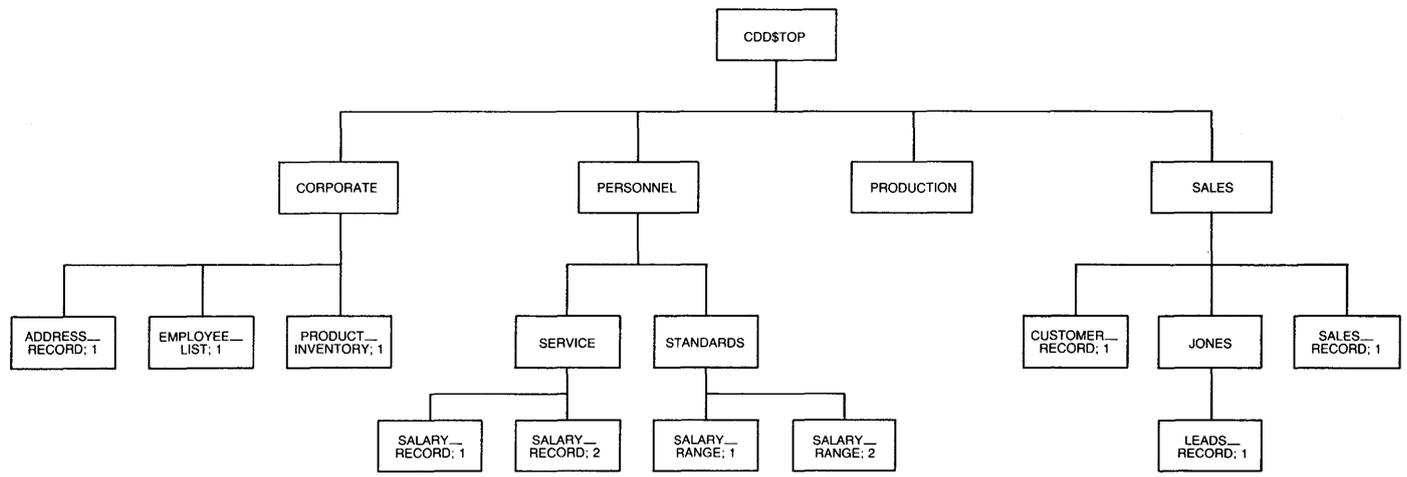
- VAX ACMS software is referred to as ACMS.
- VAX CDD/Plus software is referred to as CDD/Plus.
- VAX CDD software is referred to as CDD.
- VAX DATATRIEVE software is referred to as DATATRIEVE.
- VAX DBMS software is referred to as VAX DBMS.
- VAX Rdb/VMS software is referred to as Rdb/VMS.
- VAX TDMS software is referred to as TDMS.
- VIDA software is referred to as VIDA.

# Introduction to the Common Data Dictionary **1**

The VAX Common Data Dictionary is organized as a hierarchy of dictionary directories and dictionary objects. **Dictionary directories** are similar to VMS directories; they are used to organize information within the hierarchy. **Dictionary objects** are located at the ends of the branches in the hierarchy, and they contain the data definitions stored in the dictionary.

## **1.1 Sample Dictionary**

The sample dictionary in Figure 1-1 shows some of the relationships that can exist between dictionary directories and dictionary objects. All of the examples in this manual are drawn from this sample directory hierarchy and its associated data definitions.



ZK-8543-HC

Figure 1-1: Sample Dictionary Hierarchy

## 1.2 CDD Types

When you create dictionary directories and subdictionaries, and when you insert dictionary objects into the hierarchy, the CDD assigns dictionary **types**. Examples of CDD types include SUBDICTIONARY, CDD\$RECORD, and DTR\$DOMAIN.

Example 1-1 is a listing of the sample dictionary produced by the DMU LIST/BRIEF command. Notice that LIST/BRIEF produces no type specification for directories.

```
CDD$TOP
|  CORPORATE
|  |  ADDRESS_RECORD;1 <CDD$RECORD>
|  |  EMPLOYEE_LIST;1 <CDD$RECORD>
|  |  PRODUCT_INVENTORY;1 <CDD$RECORD>
|  PERSONNEL <SUBDICTIONARY> : DB3:[CASADAY,CDD]PERS.DIC
|  |  SERVICE
|  |  |  SALARY_RECORD;2 <CDD$RECORD>
|  |  |  SALARY_RECORD;1 <CDD$RECORD>
|  |  STANDARDS
|  |  |  SALARY_RANGE;2 <CDD$RECORD>
|  |  |  SALARY_RANGE;1 <CDD$RECORD>
|  PRODUCTION
|  SALES
|  |  CUSTOMER_RECORD;1 <CDD$RECORD>
|  |  JONES
|  |  |  LEADS_RECORD;1 <CDD$RECORD>
|  |  |  SALES_RECORD;1 <CDD$RECORD>
```

### Example 1-1: Listing of the Sample Dictionary

## 1.3 History Lists

The CDD's optional history list feature allows you to document any operation on a dictionary directory or object. You can store these documentary comments in the history list of the directory or object. You can use these history lists to monitor CDD processing and to plan additions or changes to the dictionary hierarchy.

Most DMU commands allow you to use the /AUDIT qualifier to make entries in the history list of a directory or object.

## 1.4 Access Control Lists

The CDD assures the security of the dictionary with access control lists associated with each dictionary directory or object. You can grant or deny nine CDD and four VAX DATATRIEVE access privileges to individual users or groups of users with CDD access control lists.

The DMU commands SET PROTECTION and SET PROTECTION/EDIT allow you to modify access control lists.

## 1.5 CDD Utilities

The CDD provides two utilities to help you create and maintain your dictionary hierarchy:

- The Dictionary Management Utility (DMU)
- The Dictionary Verify/Fix Utility (CDDV)

### 1.5.1 Dictionary Management Utility (DMU)

DMU commands allow you to create dictionary directories and subdictionaries, to manage the dictionary structure, and to control access to the dictionary. You can invoke the utility by typing:

```
# RUN SYS$SYSTEM:DMU
```

See Chapter 2 for a complete description of DMU commands, parameters, and qualifiers.

### 1.5.2 Dictionary Verify/Fix Utility (CDDV)

It is possible for dictionary files to become corrupted (for example, by a hardware failure during an input/output operation). The Dictionary Verify/Fix Utility (CDDV) validates and repairs corrupt dictionary files and compresses valid dictionary files.

You can use CDDV to maintain any dictionary file you own. You can also use CDDV if you have VMS SYSPRV or BYPASS privilege.

You can invoke the utility by typing:

```
# RUN SYS$SYSTEM:CDDV
```

See Chapter 3 for a complete description of the CDDV commands, parameters, and qualifiers.

## DMU Command Descriptions **2**

The Dictionary Management Utility enables you to create the directory hierarchy and to manage the dictionary's history lists and access control lists. DMU commands follow the general format:

### **COMMAND [/qualifiers] parameters**

The following sections contain complete descriptions of DMU commands, parameters, and qualifiers. These descriptions include:

- The syntax **format** you should use for each command
- The **parameters** of each command
- Command **qualifiers** that modify the functions of each command element
- **Restrictions** on the ways you can use commands
- **Usage notes** to show you how to use each command
- **Required privileges** for each command
- **Examples** from the sample dictionary (Figure 1-1) to illustrate the use of each command

# BACKUP

## 2.1 BACKUP Command

Copies portions of the directory hierarchy and their related data definitions into a backup file.

### Format

```
BACKUP [ path-name, path-name ] ... file-specification
```

<i>Qualifiers</i>	<i>Defaults</i>
/AUDIT [ = (quoted-string [ , quoted-string ] ...) ] /AUDIT = file-specification	
/NOAUDIT	/NOAUDIT
/ [ NO ] HISTORY	/NOHISTORY
/LIST [ = file-specification ] /NOLIST	/NOLIST
/LOG [ = file-specification ] /NOLOG	/NOLOG
/ [ NO ] PROTECTION	/NOPROTECTION

### Parameters

#### path-name

The dictionary directories, subdictionaries, and objects you want copied into the backup file. You can use the wildcards % and \* in the given name, and you can use either > or .> at the end of the path name. The path name is optional if /LIST is the only qualifier used. Otherwise, you must specify at least one path name. You cannot use @.

DMU assumes a > at the end of the path name as a default.

#### file-specification

The backup file. Use the standard VMS file specification. The default file type is .BAK.

## Qualifiers

`/AUDIT [= (quoted-string [, quoted-string]. . .)]`

`/AUDIT = file-specification`

Creates a history list entry auditing the backup of each dictionary directory, subdictionary, or object.

You can include explanatory text in history list entries in three ways:

- By simply specifying the `/AUDIT` qualifier. If you include no quoted string or file-specification, DMU provides a default history list entry describing your operation.
- By including quoted strings. Enclose each quoted string in double quotation marks, and enclose the series of strings in parentheses. The parentheses are optional if you specify only one quoted string. If you are using a terminal of the VT200 family, you can use 8-bit characters in quoted strings. Remember that other terminals cannot reproduce 8-bit characters.
- By specifying a file whose contents are to be included in the history list entry. The file specification is a standard VMS file specification, and the default file type is `.DAT`. You can include up to 64 input strings in a history list entry. DMU ignores any excess.

`/NOAUDIT`

Prevents the creation of a history list entry.

`/HISTORY`

Copies history lists into the backup file.

`/NOHISTORY`

Excludes history lists from the information copied into the backup file.

`/LIST [= listing-file-specification]`

Creates a listing of the dictionary directories and objects in a backup file. If you specify a path name and a backup file, `BACKUP/LIST` creates the backup file and the listing file. If you do not specify a path name, `BACKUP/LIST` creates a listing file from the specified existing backup file.

The listing file contains a sentence indicating whether or not protection and history list information are part of the backup file. The rest of the file resembles the output of a `LIST/BRIEF` command.

## BACKUP

Specify a standard VMS file name for the listing file. The default file type is .LIS. If you do not specify a listing file, DMU sends the output to SYS\$OUTPUT.

### /NOLIST

Suppresses the listing of the directories and objects in the backup file.

### /LOG [= file-specification]

Creates a list of the given names of all of the dictionary directories, subdictionaries, and objects backed up.

The file specification is a standard VMS file specification naming the file into which the log is written. The default file type is .LOG. If you use /LOG without specifying a file, DMU writes the log to SYS\$OUTPUT.

### /NOLOG

Suppresses the creation of a log file.

### /PROTECTION

Includes access control lists in the backup file.

### /NOPROTECTION

Excludes access control lists from the information copied into the backup file.

## Restrictions

- You can back up no more than 256 generations of descendants at any one time.
- Do not use /LOG with /LIST when the output for both qualifiers is directed to SYS\$OUTPUT. If you do, DMU produces confusing output.

## Usage Notes

- With the BACKUP command, you can keep a backup copy of your directory hierarchy, complete with history lists and access control lists. You specify whether or not you want to record the backup in their history lists.
- DMU BACKUP is useful for backing up portions of the directory hierarchy as a precaution before you make substantial changes to the directory hierarchy. Before you delete or change portions of the dictionary structure, you should consider making a DMU BACKUP.

## BACKUP

- You can use DMU BACKUP with DMU RESTORE to move portions of the directory hierarchy and their related data definitions from one VMS system to another.
- If you specify two opposing qualifiers, such as /HISTORY and /NOHISTORY, in the same command, then the qualifier that is not the default takes precedence. /HISTORY overrides /NOHISTORY in this case.
- BACKUP/LIST is most useful for examining the contents of backup files that were created without the /LOG or the /LIST qualifier.
- Enter CTRL/C to terminate execution of this command and return to DMU command level.

### Required Privileges

- You need PASS\_THRU and SEE at each dictionary directory, subdictionary, or object to be backed up.
- You need HISTORY at these directories and objects to use /AUDIT.
- You need CONTROL at these directories and objects to use /PROTECTION.

### Examples

The following command copies information, including history and access control lists, about the directories and objects in the directory SALES into a file called SALES.BAK. The use of the /AUDIT qualifier records the backup in the history list of each directory and object. A log of the backup appears on the terminal screen.

```
DMU> BACKUP/AUDIT/PROTECTION/HISTORY/LOG SALES SALES.BAK
"SALES" Backed up
"CUSTOMER_RECORD;1" Backed up
"JONES" Backed up
"LEADS_RECORD;1" Backed up
"SALES_RECORD;1" Backed up
```

The following command creates a file, SALES.LIS, listing the contents of SALES.BAK and reporting that the file contains history and ACL information.

```
DMU> BACKUP/LIST=SALES.LIS SALES.BAK
DMU> ^Z
$ TYPE SALES.LIS
```

## BACKUP

The backup file contains both history and ACL list information.

```
SALES
: CUSTOMER_RECORD;1 <CDD$RECORD>
: JONES
: | LEAD_RECORD;1 <CDD$RECORD>
: SALES_RECORD;1 <CDD$RECORD>
```

## 2.2 COPY Command

Copies portions of the directory hierarchy and their related data definitions from one area of the CDD to another.

### Format

```
COPY source-path-name [ ,source-path-name ] ... destination-path-name
```

#### *Qualifiers*

#### *Defaults*

/AUDIT [ = (quoted-string [ , quoted-string ] ... ) ]	
/AUDIT = file-specification	
/NOAUDIT	/NOAUDIT
/ [ NO ] HISTORY	/NOHISTORY
/LOG [ = file-specification ]	
/NOLOG	/NOLOG
/ [ NO ] PROTECTION	/NOPROTECTION
/ [ NO ] STAGE	/NOSTAGE
/ [ NO ] VERSION	/NOVERSION

### Parameters

#### source-path-name

The source of the copy. You can use the wildcards % and \* in the last given name of the source path name, and you can use > or .> at the end of the pathname. You cannot use @.

DMU assumes a > at the end of the path name by default.

#### destination-path-name

The dictionary directory or subdictionary into which the specified portions of the directory hierarchy are copied. You cannot use any wildcards in the destination path name. If you do not specify a destination path name, DMU uses your current default directory.

# COPY

## Qualifiers

`/AUDIT [= (quoted-string [, quoted-string]. . .)]`

`/AUDIT = file-specification`

Creates history list entries auditing the COPY operation in the history lists of both source and destination dictionary directories, subdictionaries, and objects.

You can include explanatory text in history list entries in three ways:

- By simply specifying the `/AUDIT` qualifier. If you include no quoted string or file-specification, DMU provides a default history list entry describing your operation.
- By including quoted strings. Enclose each quoted string in double quotation marks, and enclose the series of strings in parentheses. The parentheses are optional if you specify only one quoted string. If you are using a terminal of the VT200 family, you can use 8-bit characters in quoted strings. Remember that other terminals cannot reproduce 8-bit characters.
- By specifying a file whose contents are to be included in the history list entry. The file specification is a standard VMS file specification, and the default file type is `.DAT`. You can include up to 64 input strings in a history list entry. DMU ignores any excess.

`/NOAUDIT`

Prevents the creation of history list entries.

`/HISTORY`

Copies history lists.

If you use `/HISTORY` with `/VERSION`, the copied object retains the history list of the object in the source directory.

`/NOHISTORY`

Excludes history lists from the information that is copied.

If you use `/NOHISTORY` with `/VERSION`, the copied object receives the history list of the highest existing version of the object in the destination directory. If there is no object with the same name in the destination directory, no history list is created.

**/LOG [= file-specification]**

Creates a list of the given names of all of the dictionary directories, subdictionaries, and objects copied.

The file specification is a standard VMS file specification naming the file into which the log is written. The default file type is . LOG. If you use /LOG without specifying a file, DMU writes the log to SYS\$OUTPUT.

**/NOLOG**

Prevents the creation of a list.

**/PROTECTION**

Copies access control lists.

If you use /PROTECTION with /VERSION, the copied object retains the access control list of the object in the source directory.

**/NOPROTECTION**

Excludes access control lists from the information that is copied. If you use /NOPROTECTION with /VERSION, the copied object receives the access control list of the highest existing version of the object in the destination directory. If there is no object with the same name in the destination directory, no access control list is created.

**/STAGE**

Withholds committing changes in the CDD until the entire COPY operation is completed.

Using /STAGE assures that no changes remain in the dictionary if a command is interrupted. However, with /STAGE, execution time increases exponentially with the number of directories, subdictionaries, and objects copied.

**/NOSTAGE**

Makes changes in the CDD as each dictionary directory, subdictionary, or object is copied.

With /NOSTAGE, execution time increases linearly with the number of directories, subdictionaries, or objects copied. However, if you interrupt an unstaged COPY command, some of the changes remain in the dictionary.

# COPY

## /VERSION

Copies objects to a directory where an object with the same name exists. The new objects in the destination directory have the same version numbers as the original objects from which they were copied, regardless of whether or not you specified version numbers in the source path name. If an object to be copied has the same name *and version number* as an existing object in the destination directory, DMU does not copy it, *even* if you use the /VERSION qualifier.

Unless there is an object in the destination directory with the same name as an object you are copying, there is no need to use the /VERSION qualifier.

## /NOVERSION

Guarantees that the destination directory does not already contain an object with the same name as the object you are copying. /NOVERSION is the default.

## Restrictions

- You can copy up to 256 generations of descendants at any one time.
- You cannot copy a parent to its child.
- You cannot copy a child to its immediate parent.
- If a system, group, or process logical name CDD\$VERSION\_LIMIT has been defined, the dictionary will store only the number of versions allowed by the quota CDD\$VERSION\_LIMIT specifies.
- Cases may occur where you are granted UPDATE and CONTROL privileges for an object only through a password. If this is true, you cannot copy any version of another object with the same name as the protected object to the directory of the protected object, because DMU COPY provides no way to specify the protected object's password.

## Usage Notes

- In addition to portions of the directory hierarchy, you can copy history and access control lists.
- DMU copies subdictionaries as dictionary directories.

- If you specify two opposing options, such as /HISTORY and /NOHISTORY, in the same command line, then the one that is not the default takes precedence. /HISTORY overrides /NOHISTORY in this case.
- If a dictionary directory or subdictionary to be created in the destination area already exists, the COPY operation continues and uses the existing subdictionary or directory. No access control or history lists are copied for the existing subdictionary or directory.
- If an object to be copied has the same name as an existing object in the same directory, DMU does not copy that object unless you use /VERSION. If an object to be copied has the same name *and version number* as an existing object in the same directory, DMU does not copy that object *even* if you use /VERSION.
- To copy an object to a directory containing another object with the same name and version number, first use DMU RENAME to change the name or version number of one of the objects.
- If you specify an object with no version number in the source path name, DMU copies the highest version of the named object in the source directory. The object retains the same version number in the destination directory that it had in the source directory. If you specify a directory containing several versions of the same object, DMU copies all of them to the destination directory. Each object retains the version number it had in the source directory.
- VAX DATATRIEVE and VAX DBMS store certain types of definitions in the CDD under absolute rather than relative path names. After you use DMU to copy VAX DATATRIEVE definitions, you must use DATATRIEVE EDIT or DATATRIEVE EXTRACT to change a path name. To change path names in DBMS root files, use the DBMS command DBO/ALTER INTEGRATE.
- Enter CTRL/C to terminate execution of this command and return to DMU command level. If you have specified /STAGE, no changes are made in the CDD hierarchy.

### Required Privileges

- You need PASS\_THRU and SEE at the dictionary directories, subdictionaries, and objects you want to copy.
- You need PASS\_THRU and EXTEND at the destination directories and subdictionaries, and you need UPDATE at dictionary objects as you create them.

## COPY

- You need **HISTORY** at both the source and destination directories to use the **/AUDIT** qualifier.
- You need **CONTROL** at both the source and destination directories to use the **/PROTECTION** qualifier.
- You need **UPDATE**, **PASS\_THRU**, and **SEE** at the highest existing version of an object to use **/VERSION**.
- If you use the **/PROTECTION** qualifier with the **/VERSION** qualifier, you need **CONTROL** at the highest existing version of the object in the destination directory.

### Examples

The following command copies **ADDRESS\_RECORD;1** and its history list into the **SALES** directory. Because no destination path name is specified, **DMU** copies the information into the default directory. No version number of **ADDRESS\_RECORD** is specified, so **DMU** copies the highest existing version in the source directory. Because the **/VERSION** qualifier is not used, the destination directory cannot already contain a version of **ADDRESS\_RECORD**.

```
DMU> SET DEFAULT CDD$TOP.SALES
DMU> COPY/HISTORY CDD$TOP.CORPORATE.ADDRESS_RECORD
```

The following example copies **SALARY\_RANGE;2** into a directory already containing an object named **SALARY\_RANGE;3**. Because no destination directory is specified, **DMU** copies the object into the default directory. Note that the object retains its original version number even though the destination directory contains a higher version. Because the **/HISTORY** and **/PROTECTION** qualifiers are not specified, the newly created object receives the history list and access control list of the highest existing object with the same name, **SALARY\_RANGE;3**.

```
DMU> LIST >
    SALARY_RANGE;3 <CDD$RECORD>
DMU> COPY/VERSION CDD$TOP.PERSONNEL.STANDARDS.SALARY_RANGE;2
%DMU-I-HIGHVER, higher version of "SALARY_RANGE;2" already exists
DMU> LIST >
    SALARY_RANGE;3 <CDD$RECORD>
    SALARY_RANGE;2 <CDD$RECORD>
```

## 2.3 CREATE Command

Creates dictionary directories and subdictionaries.

### Format

CREATE path-name	
<i>Qualifiers</i>	<i>Defaults</i>
/[NO]ACL	/ACL
/AUDIT [= (quoted-string [, quoted-string]...)]	
/AUDIT = file-specification	
/NOAUDIT	/NOAUDIT
/[NO]HISTORY	/HISTORY
/SUBDICTIONARY = file-specification	
/NOSUBDICTIONARY	/NOSUBDICTIONARY

### Parameter

path-name

The name and ancestry of the directory or subdictionary you are creating.  
You cannot use wildcards in the path name.

### Qualifiers

/ACL

Creates a directory or subdictionary with the default access control list. /ACL is the default.

/NOACL

Creates a directory or subdictionary with no access control list.

/AUDIT [= (quoted-string [, quoted-string] . . .)]

/AUDIT = file-specification

Creates a history list entry auditing the creation of a dictionary directory or subdictionary.

## CREATE

You can include explanatory text in history list entries in two ways:

- By including quoted strings. Enclose each quoted string in double quotation marks, and enclose the series of strings in parentheses. The parentheses are optional if you specify only one quoted string. If you are using a terminal of the VT200 family, you can use 8-bit characters in quoted strings. Remember that other terminals cannot reproduce 8-bit characters.
- By specifying a file whose contents are to be included in the history list entry. The file specification is a standard VMS file specification, and the default file type is .DAT. You can include up to 64 input strings in a history list entry. DMU ignores any excess.

### `/NOAUDIT`

Prevents the creation of history list entries.

### `/HISTORY`

Facilitates the creation of history lists for dictionary directories and subdictionaries as you create them.

### `/NOHISTORY`

Saves storage by creating a dictionary directory or subdictionary with no history list. If you use `/NOHISTORY`, you can still create history list entries later, but the process will take slightly longer.

### `/SUBDICTIONARY = file-specification`

Creates the directory as a subdictionary. The file specification is a standard VMS file specification, and you *must* include device and directory names for this file. You can use system logical names to define the device, the directory, and the file name. You cannot use group or process logical names. For more information about creating logical names, see VMS documentation. The default file name is CDD.DIC.

Subdictionary files allow you to store portions of the dictionary on different devices and to use VMS file security mechanisms. However, you should consider the following before you use the `/SUBDICTIONARY` option:

- Subdictionary access requires more time than directory access.
- You can never have more than 255 subdictionaries open at any one time. However, other limitations or quotas may limit you to fewer open subdictionaries at any one time.

## **/NOSUBDICTIONARY**

Creates a dictionary directory in the main dictionary file.

**/NOSUBDICTIONARY** is the default.

## **Restriction**

In the entire logical dictionary, one and only one subdictionary directory can point to each subdictionary file.

## **Usage Notes**

- If any ancestors of the dictionary directory or subdictionary to be created do not exist, the CDD creates them automatically. For this reason, you do not receive an error message if you misspell a given name in a full path name specification; instead, the CDD creates a new directory. To avoid confusion, set your default to the desired parent directory before you use the **CREATE** command, and use relative path names.
- If you specify two opposing options, such as **/HISTORY** and **/NOHISTORY**, in the same command line, then the one that is not the default takes precedence. **/NOHISTORY** overrides **/HISTORY** in this case.
- A subdictionary file is not created if it already exists.
- When you create a new dictionary or subdictionary directory, you should also use the **SET PROTECTION** or **SET PROTECTION/EDIT** command to define the new directory's access control list (see Sections 2.18 and 2.19).

## **Required Privileges**

- You need **PASS\_THRU** and **EXTEND** at the parent of a dictionary directory to be created.
- You need **PASS\_THRU** and **FORWARD** at the parent of a subdictionary to be created.
- You need **PASS\_THRU** and **EXTEND** at the parent of any ancestors implicitly created for a new subdictionary or directory.

## **Examples**

The following **CREATE** command creates a dictionary directory named **SALES**.

```
DMU> CREATE CDD$TOP.SALES
```

## CREATE

The following command creates a subdictionary named PERSONNEL pointing to the subdictionary file PERS.DIC. Note that both the device and directory names must be specified in the file specification, and that once PERSONNEL is created, you cannot create another subdictionary pointing to DB3:[CASADAY.CDD]PERS.DIC.

```
DMU> CREATE/SUBDICTIONARY=DB3:[CASADAY.CDD]PERS.DIC PERSONNEL
```

In the following example, CDD\$DISK is a system logical name for DB3:[CASADAY.CDD]:

```
$ SHOW LOGICAL CDD$DISK
  CDD$DISK = "DB3:[CASADAY.CDD]"    (system)
$ DMU
DMU> CREATE/SUBDICTIONARY=CDD$DISK:PERS.DIC PERSONNEL
```

Wherever you can use a logical name, you can also specify a search list. A search list is a list of possible alternative translations for the location of a file you are creating. You can specify a search list, for example, when you do not know what devices are available. CDD searches the list, starting with the first entry, to see whether or not a file of that name already exists. If the file already exists in one of the listed locations, CREATE/SUBDICTIONARY opens the file. If no file is found, CDD creates the specified file in the first location named in the search list.

The following example defines CDD\$DISK as three alternative locations and then creates a subdictionary file in CDD\$DISK. CDD reads the list of possible translations. If no \$2\$DUA1:[CASADAY.CDD] is available, CDD searches for PERS.DIC on DIC\$DISK. DIC\$DISK is available and contains no PERS.DIC, but CDD checks NONCLUST\$DISK to be sure the file does not already exist there. Then CDD places PERS.DIC in the first available location. The DCL DIRECTORY command shows where CDD placed the subdictionary file:

```
$ DEFINE CDD$DISK $2$DUA1:[CASADAY.CDD], DIC$DISK, NONCLUST$DISK
DMU> CREATE/SUBDICTIONARY=CDD$DISK:PERS.DIC PERSONNEL
DMU> EXIT
$ DIRECTORY CDD$DISK:PERS.DIC
```

```
Directory DIC$DISK
```

```
PERS.DIC;1
```

```
Total of 1 file.
```

## 2.4 DELETE Command

Deletes dictionary directories, subdictionaries, and objects.

### Format

```
DELETE path-name [, path-name]...
```

#### *Qualifiers*

```
/ALL
```

```
/[NO]CHECK
```

```
/LOG [= file-specification]
```

```
/NOLOG
```

```
/[NO]SUBDICTIONARY
```

```
/TYPE = (type-specification  
        [, type-specification]...)
```

#### *Defaults*

```
/CHECK
```

```
/NOLOG
```

```
/NOSUBDICTIONARY
```

### Parameter

path-name

The subdictionaries, dictionary directories, or objects to be deleted. You can use the wildcards % and \* in the path name. You cannot use > or @.

### Qualifiers

/ALL

Deletes the specified dictionary directory or subdictionary and all its descendants. Note that /ALL and /CHECK are opposites. If you do not specify /ALL, then /CHECK is the default.

/CHECK

Deletes a dictionary directory or subdictionary only if it has no children. Note that /ALL and /CHECK are opposites.

/NOCHECK

Deletes the specified dictionary directory or subdictionary and all its descendants. /CHECK is the default.

## DELETE

`/LOG [= file-specification]`

Creates a list of the given names of the dictionary directories, subdictionaries, and objects deleted. However, if you specify `/ALL` or `/NOCHECK` to delete a dictionary directory or subdictionary with children, `/LOG` does not list the names of the deleted children.

The file specification is a standard VMS file specification naming the file into which the log is written. The default file type is `.LOG`. If you use `/LOG` without specifying a file, DMU writes the log to `SYS$OUTPUT`.

`/NOLOG`

Prevents the creation of a list of deleted directories and objects.

`/SUBDICTIONARY`

Deletes a subdictionary pointer and the directories and objects in the subdictionary file.

`/NOSUBDICTIONARY`

Deletes a subdictionary pointer but not the directories and objects in the file to which it points.

The subdictionary files themselves are never deleted. If you recreate a subdictionary you deleted earlier with the `/NOSUBDICTIONARY` qualifier, the new subdictionary again points to the file, and the dictionary directories and objects in the file again become part of the directory hierarchy.

`/TYPE = (type-specification [, type-specification]. . .)`

Specifies a particular type of dictionary directory or object (for example, `DTR$DOMAIN` or `CDD$RECORD`) to be deleted. Other types remain undeleted. You can use the wildcard characters `%` and `*` in a type specification. The parentheses are optional if you specify only one type.

Note that when you delete a subdictionary or dictionary directory, you also delete its descendants.

### Usage Notes

- The children of a dictionary directory or subdictionary are always deleted with their parent; therefore, `DELETE` has qualifiers to determine whether or not directories or subdictionaries with children should be deleted.

## DELETE

- If you enter two opposing qualifiers, such as /CHECK and /NOCHECK, on the same command line, the one that is not the default takes precedence. /NOCHECK overrides /CHECK in this case.
- Consider using the BACKUP command as a safeguard before you delete large portions of the dictionary.
- Enter CTRL/C to terminate execution of the command and return to DMU command level.

### Required Privileges

- To use the DELETE command, you need PASS\_THRU and EXTEND at the parents of the target dictionary directories, subdictionaries, or objects.
- To use /ALL or /NOCHECK, you need GLOBAL\_DELETE at the target dictionary directories or subdictionaries.
- If you do not use /ALL or /NOCHECK, you need either GLOBAL\_DELETE or LOCAL\_DELETE at the target dictionary directories, subdictionaries, and objects.

### Examples

The following command deletes PERSONNEL and all its descendants, including any directories and objects in the subdictionary files below it.

/LOG causes the name PERSONNEL to be listed as deleted, but the list does not contain the given names of PERSONNEL's descendants.

```
DMU> DELETE/LOG/ALL/SUBDICTIONARY PERSONNEL
"PERSONNEL" deleted
```

The following command deletes CUSTOMER\_RECORD;1 and SALES\_RECORD;1, the two children of SALES with type equal to CDD\$RECORD.

```
DMU> DELETE/ALL/TYPE=(CDD$RECORD) SALES.*
```

The following command deletes SALARY\_RANGE;2 but leaves SALARY\_RANGE;1. If you do not specify a version number for a specified object, DMU deletes the highest version of that object.

```
DMU> DELETE/LOG CDD$TOP.PERSONNEL.STANDARDS.SALARY_RANGE
"SALARY_RANGE;2" deleted
%DMU-I-LOWVER, lower version of "SALARY_RANGE;2" still exists
```

## DELETE

The following command deletes all the children of the STANDARDS directory. Note that the asterisk (\*) applies to *all* versions of the object SALARY\_RANGE. If a directory were a child of STANDARDS, it too would be deleted by this command.

```
DMU> DELETE/LOG CDD$TOP.PERSONNEL.*  
"SALARY_RANGE;2" deleted  
"SALARY_RANGE;1" deleted
```

## 2.5 DELETE/HISTORY Command

Purges history lists. You have the option of specifying the number of entries or the creation date of the earliest entry you wish to save.

### Format

```
DELETE/HISTORY path-name [, path-name]...
```

#### *Qualifiers*

/BEFORE = time-specification

/KEEP = number

/TYPE = (type-specification  
[, type-specification]...)

#### *Defaults*

/KEEP = 0

### Parameter

path-name

The dictionary directories, subdictionaries, or objects whose history lists you want to purge. You can use the wildcards %, >, @, and \* in the path name.

### Qualifiers

/BEFORE = time-specification

Specifies the creation time of the oldest history list entry you want to save. All entries made before that time are deleted.

Time is specified as the VAX standard time of day:

```
dd-mm-yy hh:mm:ss.ss
```

You can omit any of the fields, but you must include the punctuation (hyphen, colon, period) following any leading field you omit. Omitted fields default to the current time of day.

/KEEP = number

Specifies the number of history list entries you want to save. DMU then deletes all but that number of the most recent history list entries.

## DELETE/HISTORY

If you do not specify either /BEFORE or /KEEP, the default is /KEEP = 0, and all history list entries are deleted.

/TYPE = (type-specification [, type-specification]. . .)

Specifies a particular type of dictionary directory or object (for example, DTR\$DOMAIN or CDD\$RECORD) whose history lists you want to purge. You can use the wildcard characters % and \* in a type specification. The parentheses are optional if you specify only one type.

### Restriction

You cannot specify both /BEFORE and /KEEP in the same command line.

### Usage Note

Enter CTRL/C to terminate execution of the command and return to DMU command level.

### Required Privileges

You need PASS\_THRU, LOCAL\_DELETE, and SEE at the target dictionary directories, subdictionaries, or objects.

### Examples

The following DELETE/HISTORY command deletes all but the newest history list entry for CORPORATE.

```
DMU> DELETE /HISTORY/KEEP=1 CORPORATE
```

The following DELETE/HISTORY command deletes all history list entries for PERSONNEL created before 02-OCT-1983 at 12:30:00.00.

```
DMU> DELETE /HISTORY/BEFORE = 02-OCT-1983 12:30:00.00 PERSONNEL
```

## 2.6 DELETE/PROTECTION Command

Deletes access control list entries. You can delete the entire list or only those entries you specify.

You can also use the SET PROTECTION/EDIT command to delete access control list entries (see Section 2.19).

### Format

```
DELETE/PROTECTION path-name [, path-name]...
```

#### *Qualifiers*

```
/AUDIT [= (quoted-string
           [, quoted-string]...)]
/AUDIT = file-specification
/NOAUDIT
/POSITION = position-specification
/TYPE = (type-specification
         [, type-specification]...)
```

#### *Defaults*

```
/NOAUDIT
/POSITION = ALL
```

### Parameter

path-name

The dictionary directory, subdictionary, or object whose access control list entries you want to delete. You can use the wildcards %, >, @, and \* in the path name.

### Qualifiers

```
/AUDIT [= (quoted-string [, quoted-string]. . .)]
/AUDIT = file-specification
```

Creates history list entries auditing the purge of the access control lists.

You can include explanatory text in history list entries in two ways:

- By including quoted strings. Enclose each quoted string in double quotation marks, and enclose the series of strings in parentheses. The parentheses are optional if you specify only one quoted string. If you are using a terminal of the VT200 family, you can use 8-bit characters in quoted strings. Remember that other terminals cannot reproduce 8-bit characters.

## DELETE/PROTECTION

- By specifying a file whose contents are to be included in the history list entry. The file specification is a standard VMS file specification, and the default file type is .DAT. You can include up to 64 input strings in a history list entry. DMU ignores any excess.

### /NOAUDIT

Prevents the creation of history list entries.

### /POSITION = position-specification

Specifies the relative position or positions of the entries you want to delete.

The position specification can be any of the following:

ALL	Indicates all entries.
number	Indicates only one entry.
(number [,number]. . .)	Indicates several entries.

You can specify a maximum of 64 entries with the /POSITION qualifier.

### /TYPE = (type-specification [, type-specification]. . .)

Specifies a particular type of dictionary directory or object (for example, DTR\$DOMAIN or CDD\$RECORD) whose access control list entries are to be purged. You can use the wildcard characters % and \* in a type specification.

The parentheses are optional if you specify only one type.

## Restriction

You can explicitly specify up to 64 entries with the /POSITION qualifier.

## Usage Note

Enter CTRL/C to terminate execution of the command and return to DMU command level.

## Required Privileges

You need PASS\_THRU and CONTROL at the target dictionary directory, subdictionary, or object.

## DELETE/PROTECTION

### Example

The following DELETE/PROTECTION command deletes the second ACL entry for SALARY\_RECORD;2. The remaining entries are then renumbered.

```
DMU> SET DEFAULT CDD$TOP.PERSONNEL(SEMI_SECRET).SERVICE(SECRET)
DMU> LIST/PROTECTION SALARY_RECORD;2
CDD$TOP.PERSONNEL.SERVICE.SALARY_RECORD;2
  1:  [*,*], Username: "CASADAY"
      Grant - CDHPSX, Deny - none, Banish - none
  2:  [*,*], Username: "KELLERMAN"
      Grant - DEMRUW, Deny - none, Banish - none
  3:  [*,*], Username: "FOSTER"
      Grant - ER, Deny - none, Banish - none

DMU> DELETE/PROTECTION/POSITION=2 SALARY_RECORD
DMU> LIST/PROTECTION SALARY_RECORD;2
CDD$TOP.PERSONNEL.SERVICE.SALARY_RECORD;2
  1:  [*,*], Username: "CASADAY"
      Grant - CDHPSX, Deny - none, Banish - none
  2:  [*,*], Username: "FOSTER"
      Grant - ER, Deny - none, Banish - none
```

## EXIT

### 2.7 EXIT Command

Use the EXIT command or CTRL/Z to return to DCL command level.

#### Format

{ EXIT CTRL/Z }
--------------------------

## 2.8 EXTRACT Command

Copies the source text of dictionary objects from the CDD or creates source text from the CDD definition and places the source text into a file.

### Format

```
EXTRACT path-name [ ,path-name ] ... file-specification
```

#### *Qualifiers*

#### *Defaults*

```
/AUDIT [ = (quoted-string  
          [ , quoted-string ] ...) ]
```

```
/AUDIT = file-specification
```

```
/NOAUDIT
```

```
/NOAUDIT
```

```
/RECORD
```

```
/SOURCE
```

```
/SOURCE
```

```
/ [ NO ] TEMPLATE
```

```
/NOTEMPLATE
```

```
/TYPE = (type-specification  
         [ , type-specification ] ... )
```

### Parameters

path-name

The dictionary objects you want to extract. You can use the wildcards % and \* in the path name.

file-specification

The file into which the source text is copied. The file specification is a standard VMS file specification naming the output file. The default file type is .LIS.

### Qualifiers

```
/AUDIT [ = (quoted-string [ , quoted-string] . . . ) ]
```

```
/AUDIT = file-specification
```

Creates history list entries auditing the extraction of each dictionary object.

## EXTRACT

You can include explanatory text in history list entries in three ways:

- By simply specifying the /AUDIT qualifier. If you include no quoted string or file-specification, DMU provides a default history list entry describing your operation.
- By including quoted strings. Enclose each quoted string in double quotation marks, and enclose the series of strings in parentheses. The parentheses are optional if you specify only one quoted string. If you are using a terminal of the VT200 family, you can use 8-bit characters in quoted strings. Remember that other terminals cannot reproduce 8-bit characters.
- By specifying a file whose contents are to be included in the history list entry. The file specification is a standard VMS file specification, and the default file type is .DAT. You can include up to 64 input strings in a history list entry. DMU ignores any excess.

### /NOAUDIT

Prevents the creation of history list entries.

### /RECORD

Translates a record definition stored in the CDD into CDDL source text. Use /RECORD when the source text is not already part of the record definition or when you are using the /TEMPLATE qualifier. The opposite of /RECORD is /SOURCE. /SOURCE is the default.

### /SOURCE

Extracts source text from the record definition. If you use /SOURCE and the record definition does not include source text, DMU sends you an error message. The opposite of /SOURCE is /RECORD. /SOURCE is the default.

### /TEMPLATE

Includes the definitions of template records within the source text in the listing file. /TEMPLATE first extracts the source text of any template records in a record being extracted, and then extracts the designated record. If you use /TEMPLATE, you must also use /RECORD.

### /NOTEMPLATE

Copies the record definition without including the definitions of template records at the top of the listing. The default is /NOTEMPLATE.

`/TYPE = (type-specification [, type-specification]. . .)`

Specifies a particular type of dictionary object (for example, `DTR$DOMAIN` or `CDD$RECORD`) to be extracted. You can use the wildcard characters `%` and `*` in a type specification. The parentheses are optional if you specify only one type.

### **Usage Notes**

- Use `/RECORD` to create and extract CDDL source text from record definitions created by compilers other than CDDL (for example, the DBMS DDL compiler).
- You can use `/RECORD` only for objects whose type is `CDD$RECORD`.
- You can use `/RECORD` to extract objects whose type is `CDD$RECORD` even if they were created by VAX DBMS. The listing file contains a CDDL representation of the DBMS record definition but may lack some VAX DBMS attributes. Therefore, you should not attempt to recompile the extracted record definition into a DBMS schema.
- You must specify `/RECORD` if you specify `/TEMPLATE`.
- Enter `CTRL/C` to terminate execution of the command and return to DMU command level.

### **Required Privileges**

You need `PASS_THRU` and `SEE` at the target dictionary objects to extract them from the CDD. You must have `PASS_THRU` and `SEE` at any template record definitions to be extracted with the `/TEMPLATE` qualifier.

### **Example**

In the following example, the record definition contained in the object `CDD$TOP.CORPORATE.EMPLOYEE_LIST;1` is translated into source text and extracted into a file named `TEMP.LIS`. Because `/TEMPLATE` is used, DMU also extracts the template record `CDD$TOP.CORPORATE.ADDRESS_RECORD;1`. Note that the `COPY` field description `ADDRESS` in the record definition `EMPLOYEE_LIST` is expanded in `ADDRESS STRUCTURE`. Each line of source text in `ADDRESS STRUCTURE` is preceded by the exclamation point (!) comment delimiter.

## EXTRACT

The /AUDIT qualifier creates a history list entry noting that EMPLOYEE\_LIST was extracted.

```
DMU> EXTRACT/RECORD/TEMPLATE/AUDIT CDD$TOP.-
DMU>_CORPORATE,EMPLOYEE_LIST TEMP.LIS
DMU> ^Z
$ TYPE TEMP.LIS
DEFINE RECORD CDD$TOP.CORPORATE.ADDRESS_RECORD
  DESCRIPTION IS
    /* This record contains the standard format
    for addresses. It provides the source from which all
    address fields in other record descriptions are copied. */.
  ADDRESS STRUCTURE.
    STREET          DATATYPE IS TEXT
                   SIZE IS 30 CHARACTERS.
    CITY            DATATYPE IS TEXT
                   SIZE IS 30 CHARACTERS.
    STATE           DATATYPE IS TEXT
                   SIZE IS 2 CHARACTERS.
    ZIP_CODE STRUCTURE.
      NEW           DATATYPE IS UNSIGNED NUMERIC
                   SIZE IS 4 DIGITS
                   INITIAL_VALUE 0
                   BLANK WHEN ZERO.
      OLD           DATATYPE IS UNSIGNED NUMERIC
                   SIZE IS 5 DIGITS.
    END ZIP_CODE STRUCTURE.
  END ADDRESS STRUCTURE.
END ADDRESS_RECORD.

DEFINE RECORD CDD$TOP.CORPORATE.EMPLOYEE_LIST
  DESCRIPTION IS
    /* This record contains the master list of all
    employees */.
  EMPLOYEE STRUCTURE.
    /* An employee's ID number is his
    or her social security number */
    ID              DATATYPE IS UNSIGNED NUMERIC
                   SIZE IS 9 DIGITS.
    NAME STRUCTURE.
      LAST_NAME     DATATYPE IS TEXT
                   SIZE IS 15 CHARACTERS.
      FIRST_NAME    DATATYPE IS TEXT
                   SIZE IS 10 CHARACTERS.
      MIDDLE_INITIAL DATATYPE IS TEXT
                   SIZE IS 1 CHARACTER.
    END NAME STRUCTURE.
  ADDRESS          COPY FROM
                   CDD$TOP.CORPORATE.ADDRESS_RECORD.
```

(continued on next page)

## EXTRACT

```
! ADDRESS STRUCTURE,
!   STREET           DATATYPE IS TEXT
!                   SIZE IS 30 CHARACTERS.
!   CITY            DATATYPE IS TEXT
!                   SIZE IS 30 CHARACTERS.
!   STATE           DATATYPE IS TEXT
!                   SIZE IS 2 CHARACTERS.
!   ZIP_CODE STRUCTURE,
!     NEW           DATATYPE IS UNSIGNED NUMERIC
!                   SIZE IS 4 DIGITS
!                   INITIAL_VALUE 0
!                   BLANK WHEN ZERO.
!     OLD           DATATYPE IS UNSIGNED NUMERIC
!                   SIZE IS 5 DIGITS.
!   END ZIP_CODE STRUCTURE.
! END ADDRESS STRUCTURE.
! DEPT_CODE         DATATYPE IS UNSIGNED NUMERIC
!                   SIZE IS 3 DIGITS.
! END EMPLOYEE STRUCTURE.
END EMPLOYEE_LIST RECORD.
```

# HELP

## 2.9 HELP Command

Displays text describing DMU commands. DMU HELP follows the standard syntax of a DCL help command.

If you type HELP without specifying a key phrase, DMU displays a list of the key phrases for which text is available.

Type an asterisk (\*) in place of the key phrase for all first-level HELP text. Type ellipses (. . .) in place of the key phrase for all of the DMU HELP text in the library. Type ellipses (. . .) after a key phrase for all of the HELP text about the specified command, its parameters, and its qualifiers.

### Format

HELP [key-phrase]
-------------------

## 2.10 LIST Command

Displays information about dictionary directories, subdictionaries, and objects. This information includes sources, history lists, access control lists, and text.

### Format

LIST [path-name [, path-name]...]	
<i>Qualifiers</i>	<i>Defaults</i>
/ACCESS = (quoted-string [, quoted-string]...)	
/[NO]AUDIT_TRAIL	/NOAUDIT_TRAIL
/BRIEF	/BRIEF
/FULL	
/ITEM = (item-specification [, item-specification]...)	
/[NO]ORIGIN	/NOORIGIN
/LISTING [= file-specification]	
/OUTPUT [= file-specification]	
/PROTECTION [/POSITION = position-specification]	/POSITION = ALL
/NOPROTECTION	/NOPROTECTION
/TYPE = (type-specification [, type-specification]...)	

### Parameter

path-name

The dictionary directories, subdictionaries, and objects you want to list. You can use the wildcards %, \*, >, and @ in the path name. If you do not specify a path name, the default path name is \*.

### Qualifiers

/ACCESS = (quoted-string [, quoted-string]. . .)

Displays only those history list entries beginning with one of the quoted access strings.

## LIST

You can enclose the strings in either single or double quotation marks. DMU does not differentiate between uppercase and lowercase. The parentheses are optional if you specify only one string.

You need only enough characters in the quoted access string to avoid ambiguity. DMU compares the access string to the corresponding number of characters at the beginning of each history list entry.

You must specify `/AUDIT_TRAIL` or `/FULL` with `/ACCESS`.

### `/AUDIT_TRAIL`

Includes history list entries in the listing. `/AUDIT_TRAIL` displays the following seven items (see Table 2-1):

<code>ACCESS_TIME</code>	<code>PROGRAM</code>
<code>EXPLANATION</code>	<code>UIC</code>
<code>FACILITY</code>	<code>USER</code>
<code>PROCESS</code>	

### `/NOAUDIT_TRAIL`

Prevents the display of history list entries. `/NOAUDIT_TRAIL` is the default.

### `/BRIEF`

Displays only the given name and type of each dictionary directory, subdictionary, or object. In the case of subdictionaries, the file name is also listed. The type name "DIRECTORY" is not listed. The `LIST/BRIEF` command produces the shortest possible listing. There is no way to list only the given names of objects.

DMU indents ancestors and descendants to show the relationships within the family tree.

`/BRIEF` is the default.

### `/FULL`

Displays the name, type, description, source, and history list of each specified dictionary directory, subdictionary, and object.

`/FULL` lists all of the information that you can list with the `/ITEM` qualifier (see Table 2-1).

You can suppress the listing of any particular item by specifying the negation of that item in the `/ITEM` option. For example, the display of dictionary types is prevented by the use of `LIST/FULL/ITEM=NOTYPE`.

`/ITEM = (item-specification [, item-specification]. . .)`

Specifies particular items you want to list. The item specification can include one or more of the item options in Table 2-1. If you specify only one item, you can omit the parentheses.

The specification of "NO" for an item overrides the specification of the item. Therefore, if you specify `/ITEM=(TYPE,NOTYPE)`, NOTYPE takes precedence.

**Table 2-1: /ITEM Options**

Item	Description
ACCESS_TIME	Date and time at which the specified dictionary directory, subdictionary, or object was used
CREATOR	Name of the facility that created a dictionary object
DESCRIPTION	Text describing a dictionary object
EXPLANATION	Text added to history list entries
FACILITY	Compiler or language processor that accessed the specified dictionary directory, subdictionary, or object
PROCESS	Name of the process accessing the specified dictionary object
PROGRAM	Name of the program accessing the specified dictionary object
SOURCE	Source listing of the dictionary object
TIME_STAMP	Date and time of creation of a dictionary object
TYPE	Type, including directory, subdictionary, and several dictionary objects (CDD\$RECORD and DTR\$DOMAIN, for example)
UIC	User identification code of the user who accessed or created the specified dictionary directory, subdictionary, or object
USER	VMS username of the user that created or accessed the specified dictionary directory, subdictionary, or object
VERSION	The version of the CDD record protocol used when the dictionary object was created. DMU accepts <code>PROTOCOL_VERSION</code> as a synonym for <code>VERSION</code> in an <code>/ITEM</code> specification

# LIST

**/LISTING [= file-specification]**

Writes the listed information into an output file. The file specification is a standard VMS file specification naming the listing file. The default file type is .LIS. If you do not specify a file, DMU sends the listing to SYS\$OUTPUT.

**/ORIGIN**

Displays information about the creation of dictionary objects. /ORIGIN displays the following four items (see Table 2-1):

<b>CREATOR</b>	<b>SOURCE</b>
<b>VERSION</b>	<b>TIME_STAMP</b>

**/NOORIGIN**

Prevents the display of CREATOR, SOURCE, VERSION, and TIME\_STAMP information.

**/OUTPUT [= file-specification]**

DMU accepts /OUTPUT as a synonym for /LISTING.

**/POSITION = position-specification**

Indicates the access control list entries you want to list.

The position specification can be any of the following:

<b>ALL</b>	Indicates all entries.
<b>number</b>	Indicates only one entry.
<b>(number [,number]. . .)</b>	Indicates several entries.

The entries are listed in the order specified. /POSITION = ALL is the default if you do not specify the /POSITION qualifier.

**/PROTECTION**

Includes access control lists in the listing.

**/NOPROTECTION**

Excludes the access control lists from the listing.

**/TYPE = (type-specification [, type-specification]. . .)**

Specifies a particular type of dictionary directory or object (for example, DTR\$DOMAIN or CDD\$RECORD) to be listed. When you use LIST/BRIEF for specific types of dictionary directories and objects, DMU displays the

parents of these directories and objects as well. You can use the wildcard characters % and \* in a type specification. The parentheses are optional if you specify only one type.

## Restriction

The width of the listing on your terminal screen depends on the terminal width setting. For width settings between 80 and 132 characters, the terminal width and the listing width are the same. For a terminal width of fewer than 80 characters, the listing width defaults to 80. For a terminal width of more than 132 characters, the listing width defaults to 132.

## Usage Notes

- You cannot use the /ITEM=NOTYPE qualifier with LIST/BRIEF to produce a listing containing only the given name of the directories and objects. When no additional items are specified, the default, LIST/BRIEF, is used. LIST/BRIEF produces the shortest possible listing.
- You cannot use a semicolon with a directory name. If you use the semicolon with a wildcard name that matches both directories and objects, LIST produces only the objects.
- If you do not specify a version number for an object, LIST displays all versions of that object.
- Enter CTRL/C to terminate execution of the command and return to DMU command level.

## Required Privileges

- You need PASS\_THRU and SEE at the target dictionary directory, subdictionary, or object to list any information except the access control list.
- You need PASS\_THRU and CONTROL at the target dictionary directory, subdictionary, or object to list the access control list.

# LIST

## Examples

The following example is a listing of the `EMPLOYEE_LIST;1` record. The `/FULL` qualifier, modified by the negated items in the item specification, causes the `LIST` command to display everything about `EMPLOYEE_LIST;1` except the access control list, the descriptive text documenting the record definition, and the source code of the record definition. `LIST/FULL` does not display the access control list because `/NOPROTECTION` is the default.

```
DMU> SET DEFAULT CDD$TOP.CORPORATE
DMU> LIST/FULL/ITEM=(NODESCRIPTION,NOSOURCE) EMPLOYEE_LIST
CORPORATE.EMPLOYEE_LIST;1 <CDD$RECORD>
  Created by VAX CDD Data Definition Language Version 3.00
    on 8-JAN-1984 15:38:50.71 using Protocol version 4.
  Backed up by CASADAY (UIC [30,10]) in Process CASADAY
    using VAX CDD Dictionary Management Utility Version 3.00
    on 10-JAN-1984 11:36:18.47.
Explanation:
  EMPLOYEE LIST BACKUP
```

The following `LIST` command finds only those history list entries beginning with the words "BACKED UP". Other history list entries for `EMPLOYEE_LIST;1` are ignored.

```
DMU> LIST/AUDIT_TRAIL/ACCESS="BACKED UP" EMPLOYEE_LIST
CORPORATE.EMPLOYEE_LIST;1 <CDD$RECORD>
  Backed up by CASADAY (UIC [30,10]) in Process CASADAY
    using VAX CDD Dictionary Management Utility Version 3.1
    on 5-JUL-1984 11:36:18.47.
Explanation:
  CDD$TOP.CORPORATE.EMPLOYEE_LIST backed up to
  EMPLOYEE.BAK
```

The following command lists the record definitions whose type is `CDD$RECORD` and their parents in the `SALES` directory.

```
DMU> LIST/BRIEF/TYPE=CDD$RECORD SALES>
SALES
| CUSTOMER_RECORD;1 <CDD$RECORD>
| JONES
| | LEADS_RECORD;1 <CDD$RECORD>
| SALES_RECORD;1 <CDD$RECORD>
```

## LIST

The following example uses a wildcard (\*) in place of a path name and another wildcard (\*) in place of the version number. Because a semicolon follows the first wildcard, LIST displays only the objects that are children of CDD\$TOP.SALES.

```
DMU> SET DEFAULT CDD$TOP.SALES
DMU> LIST **
    CUSTOMER_RECORD;1
    SALES_RECORD;1
```

The following example displays a /BRIEF listing of the directory hierarchy under CDD\$TOP. /BRIEF is not specified because it is the default.

```
DMU> SET DEFAULT CDD$TOP
DMU> LIST >
CORPORATE
| ADDRESS_RECORD;1 <CDD$RECORD>
| EMPLOYEE_LIST;1 <CDD$RECORD>
| PRODUCT_INVENTORY;1 <CDD$RECORD>
PERSONNEL <SUBDICTIONARY> : DB3:ICASADAY.CDD|PERS.DIC
| SERVICE
| | SALARY_RECORD;2 <CDD$RECORD>
| | SALARY_RECORD;1 <CDD$RECORD>
| STANDARDS
| | SALARY_RANGE;2 <CDD$RECORD>
| | SALARY_RANGE;1 <CDD$RECORD>
PRODUCTION
SALES
| CUSTOMER_RECORD;1 <CDD$RECORD>
| JONES
| | LEADS_RECORD;1 <CDD$RECORD>
| SALES_RECORD;1 <CDD$RECORD>
```

# MEMO

## 2.11 MEMO Command

Creates new history list entries for specified dictionary directories, subdictionaries, and objects. You supply the explanation that is to be placed in the entry. DMU supplies all other information.

### Format

MEMO/AUDIT path-name [, path-name]...	
<i>Qualifiers</i>	<i>Defaults</i>
/AUDIT = (quoted-string [, quoted-string]...)	
/AUDIT = file-specification	None

### Parameter

path-name

The dictionary directory, subdictionary, or object for which you want to add a history list entry. You can use the wildcards % and \* in the path name. You cannot use > and @.

### Qualifier

/AUDIT = (quoted-string [, quoted-string]. . .)

/AUDIT = file-specification

Creates history list entries. You must use the /AUDIT qualifier.

You supply explanatory text in history list entries in either of two ways:

- By including quoted strings. Enclose each quoted string in double quotation marks, and enclose the series of strings in parentheses. The parentheses are optional if you specify only one quoted string. If you are using a terminal of the VT200 family, you can use 8-bit characters in quoted strings. Remember that other terminals cannot reproduce 8-bit characters.
- By specifying a file whose contents are to be included in the history list entry. The file specification is a standard VMS file specification, and the default file type is .DAT. You can include up to 64 input strings in a history list entry. DMU ignores any excess.

You cannot specify /NOAUDIT with the MEMO command.

**Usage Note**

Enter CTRL/C to terminate execution of the command and return to DMU command level.

**Required Privileges**

You need PASS\_THRU and HISTORY at the target dictionary directories, subdictionaries, and objects.

**Example**

The following MEMO command adds the explanation "Redefined protection" to the LEADS\_RECORD;1 history list.

```
DMU> SET DEFAULT CDD$TOP.SALES.JONES
DMU> MEMO/AUDIT="Redefined protection" LEADS_RECORD;1
DMU> LIST/AUDIT_TRAIL LEADS_RECORD;1
CDD$TOP.SALES.JONES.LEADS_RECORD;1 <CDD$RECORD>
Memo entered by CASADAY (UIC [30,10]) in process CASADAY
  using VAX CDD Dictionary Management Utility Version 3.00
  on 12-JAN-1984 14:02:02.40.
Explanation:
  Redefined protection
```

# PURGE

## 2.12 PURGE Command

Deletes all but the highest versions of specified dictionary objects.

### Format

```
PURGE [ path-name, path-name ] ...
```

<i>Qualifiers</i>	<i>Defaults</i>
/KEEP = n	/KEEP = 1
/LOG [ = file-specification ]	
/NOLOG	/NOLOG

### Parameter

path-name

The object you want to purge, or the directory or subdictionary that contains the object or objects you want to purge. You can use the wildcards \*, %, and > in the path name specification. The path name cannot contain a version number or a semicolon. If you do not specify a path name, DMU purges the objects in your default directory.

### Qualifiers

/KEEP = n

Specifies the number of versions of each object you want to keep. The number must be greater than zero. The default is /KEEP = 1.

/LOG [ = file-specification ]

Creates a listing of the given names of objects deleted by the PURGE command. The file specification is the standard VMS file name. If you do not specify a file, DMU sends the listing to SYS\$OUTPUT.

/NOLOG

Suppresses a listing of deleted objects. /NOLOG is the default.

## Usage Note

PURGE has no effect on directories and subdictionaries because they cannot have multiple versions.

## Required Privileges

You need `PASS_THRU` and either `LOCAL_DELETE` or `GLOBAL_DELETE` at each object to be deleted.

## Example

The following example deletes all but the highest version of the objects under `CDD$TOP.PERSONNEL`. Because the `/LOG` qualifier does not specify a file, the listing of deleted objects is sent to the terminal.

```
DMU> LIST CDD$TOP.PERSONNEL>
PERSONNEL
| SERVICE
| | SALARY_RECORD;2 <CDD$RECORD>
| | SALARY_RECORD;1 <CDD$RECORD>
| STANDARDS
| | SALARY_RANGE;2 <CDD$RECORD>
| | SALARY_RANGE;1 <CDD$RECORD>
DMU> PURGE/LOG CDD$TOP.PERSONNEL>
"SALARY_RECORD;1" deleted
"SALARY_RANGE;1" deleted
```

# RENAME

## 2.13 RENAME Command

Changes the name of any dictionary directory, subdictionary, or object in the CDD.

### Format

```
RENAME path-name given-name
```

#### *Qualifiers*

```
/AUDIT [ = (quoted-string  
            [ , quoted-string ] ...) ]  
/AUDIT = file-specification  
/[ NO ] VERSION
```

#### *Defaults*

```
/NOAUDIT  
/NOVERSION
```

### Parameters

path-name

The dictionary directory, subdictionary, or object you want to rename. You can use absolute or relative version numbers in the path name. You can also use \* in place of the version number. You cannot use any other wildcards in the path name.

given-name

The new name you give the dictionary directory, subdictionary, or object. You can use \* in place of the version number. You cannot use any other wildcards in the given name. You can use absolute version numbers, but not relative version numbers, in the given name.

### Qualifiers

```
AUDIT [ = (quoted-string [ , quoted-string] . . .) ]  
/AUDIT = file-specification
```

Creates a history list entry auditing the name change.

# RENAME

You can include explanatory text in history list entries in three ways:

- By simply specifying the /AUDIT qualifier. If you include no quoted string or file-specification, DMU provides a default history list entry describing your operation.
- By including quoted strings. Enclose each quoted string in double quotation marks, and enclose the series of strings in parentheses. The parentheses are optional if you specify only one quoted string. If you are using a terminal of the VT200 family, you can use 8-bit characters in quoted strings. Remember that other terminals cannot reproduce 8-bit characters.
- By specifying a file whose contents are to be included in the history list entry. The file specification is a standard VMS file specification, and the default file type is .DAT. You can include up to 64 input strings in a history list entry. DMU ignores any excess.

## /NOAUDIT

Prevents the creation of history list entries.

## /VERSION

Creates an additional version of an existing object in the same directory.

## /NOVERSION

Guarantees that you are not creating an object with the same name as an existing object in the same directory. /NOVERSION is the default.

## Restrictions

- You cannot rename CDD\$TOP.
- You cannot rename across directories. To move a directory or object to another directory, you must use COPY or BACKUP and RESTORE.
- You cannot rename an object if the directory already contains an object with the same name *and version number* even if you use /VERSION.
- The highest version number an object can have is 32,767.
- If a system, group, or process logical name CDD\$VERSION\_LIMIT has been defined, the dictionary will store only the number of versions allowed by the quota CDD\$VERSION\_LIMIT specifies.

## RENAME

- In a case where you are granted the UPDATE privilege for an object only through a password, you cannot rename another object as a new version of the protected object, because DMU RENAME provides no way to specify the protected object's password.

### Usage Notes

- You must update applications using dictionary directories, subdirectories, or objects that you rename. Such applications must use the new names.
- The results of using RENAME/VERSION vary depending on the way you specify the objects to be renamed. See the *VAX Common Data Dictionary User's Guide* for information about the various methods of specifying versions of dictionary objects.
- RENAME/VERSION successfully renames an object if:
  - You do not specify the version number of the new name of the object. For example, you want to rename CDD\$TOP.CORPORATE.ADDRESS\_RECORD;1 to EMPLOYEE\_LIST. If you do not specify a version number for EMPLOYEE\_LIST, DMU renames ADDRESS\_RECORD;1 to EMPLOYEE\_LIST;2, a version number one greater than the highest existing version of EMPLOYEE\_LIST.
  - You specify the version number of the new given name and the version number is higher than the version number of any object with the same name. For example, you specify EMPLOYEE\_LIST;2 as the new given name of CDD\$TOP.CORPORATE.ADDRESS\_RECORD;1. Because EMPLOYEE\_LIST;1 is the highest version of EMPLOYEE\_LIST in the CORPORATE directory, DMU renames ADDRESS\_RECORD;1 and gives it the version number you specified, EMPLOYEE\_LIST;2.
- RENAME/VERSION renames an object and gives a warning message if:
  - You specify the version number of the new name of the object, and the version number is lower than the highest version number of any object with the same name. There cannot be an object with the same name *and version number* in the directory. For example, you rename ADDRESS\_RECORD;1 to EMPLOYEE\_LIST;2 and the directory already contains an object EMPLOYEE\_LIST;3. It does not contain an object EMPLOYEE\_LIST;2, however. DMU renames the object, gives it the version number you specified, and issues a warning message.

- **RENAME/VERSION** does not rename an object if:
  - You specify the version number of the new name of the object, and the directory already contains an object with that name and version number. For example, you attempt to rename **ADDRESS\_RECORD;1** to **EMPLOYEE\_LIST;1** and an object **EMPLOYEE\_LIST;1** already exists in that directory.
- The renamed object always has the same access control list and history list as it had under its old name, even if the new name makes it a new version of an existing object in that directory.

### Required Privileges

- You need **PASS\_THRU** and **EXTEND** at the parent of the dictionary directory, subdictionary, or object you want to rename.
- You need **PASS\_THRU** and **LOCAL\_DELETE** at the target dictionary directory, subdictionary, or object.
- To use **/AUDIT**, you need **HISTORY** at the target dictionary directory, subdictionary, or object.
- You need **UPDATE**, **CONTROL**, **PASS\_THRU**, and **SEE** at the highest existing version of an object to use **/VERSION**.

---

### Note

---

Be careful when you use **RENAME/VERSION**. Unless you have **CONTROL** privilege at both the source and the highest existing version of the object with the same name, you may not be able to change the new name back to the old name.

---

## RENAME

### Examples

The following example changes the name of LEADS\_RECORD;1 to JONES\_RECORD;1.

```
DMU> LIST CDD$TOP,SALES,JONES>
JONES
| LEADS_RECORD;1
DMU> RENAME CDD$TOP,SALES,JONES,LEADS_RECORD JONES_RECORD
DMU> LIST CDD$TOP,SALES,JONES>
JONES
| JONES_RECORD;1
```

The following example changes the name of SALARY\_RECORD;1 to SALARY\_RECORD;3.

```
DMU> LIST CDD$TOP,PERSONNEL,SERVICE,SALARY_RECORD
SALARY_RECORD;2 <CDD$RECORD>
SALARY_RECORD;1 <CDD$RECORD>
DMU> RENAME/VERSION SALARY_RECORD;1 SALARY_RECORD
DMU> LIST CDD$TOP,PERSONNEL,SERVICE,SALARY_RECORD
SALARY_RECORD;3 <CDD$RECORD>
SALARY_RECORD;2 <CDD$RECORD>
```

# RENAME/SUBDICTIONARY

## 2.14 RENAME/SUBDICTIONARY Command

Changes the file specification to which a subdictionary directory points. You set the directory to point to a VMS subdictionary file that you are moving or renaming.

### Format

RENAME/SUBDICTIONARY = file-specification path-name	
<i>Qualifiers</i>	<i>Defaults</i>
/AUDIT [ = (quoted-string [, quoted-string ]...) ]	
/AUDIT = file-specification	
/NOAUDIT	/NOAUDIT

### Parameters

file-specification

The subdictionary file. Use the standard VMS file specification, and you *must* include device and directory names for this file. You can use system logical names to define the device, the directory, and the file name. You cannot use group or process logical names. For more information about creating logical names, see VMS documentation. The default file name is CDD.DIC.

path-name

The name of the subdictionary directory you want to point to a new VMS file. You can use a full or relative path name. You cannot use any wildcards in the path-name.

### Qualifiers

/AUDIT [= (quoted-string [, quoted-string] . . .)]

/AUDIT = file-specification

Creates a history list entry auditing the modification of the file specification.

## RENAME/SUBDICTIONARY

You can include explanatory text in history list entries in three ways:

- By including quoted strings. Enclose each quoted string in double quotation marks, and enclose the series of strings in parentheses. The parentheses are optional if you specify only one quoted string. If you are using a terminal of the VT200 family, you can use 8-bit characters in quoted strings. Remember that other terminals cannot reproduce 8-bit characters.
- By specifying a file whose contents are to be included in the history list entry. The file specification is a standard VMS file specification, and the default file type is .DAT. You can include up to 64 input strings in a history list entry. DMU ignores any excess.
- By allowing DMU to supply a default audit explanation including both the old and new names of the subdictionary file.

**/NOAUDIT**

Prevents the creation of a history list entry.

### Restrictions

- In the entire logical dictionary, one and only one subdictionary directory can point to each subdictionary file.
- No one else can have access to a subdictionary while you are renaming it.
- Use the DCL COPY command, not the DCL RENAME command, to change the location of the subdictionary file.
- If a system, group, or process logical name CDD\$VERSION\_LIMIT has been defined, the dictionary will store only the number of versions allowed by the quota CDD\$VERSION\_LIMIT specifies.

### Usage Notes

- Use RENAME/SUBDICTIONARY to avoid having to delete a subdictionary directory before you create a new one pointing to the file's new location. RENAME/SUBDICTIONARY both deletes the old subdictionary pointer and creates the new one.
- Before you use RENAME/SUBDICTIONARY to reset the directory pointer to the file, use the DCL COPY command to change the location of the subdictionary file.

## RENAME/SUBDICTIONARY

### Required Privileges

- You need `PASS_THRU` and `FORWARD` at the parent of the subdictionary you want to rename.
- You need `LOCAL_DELETE` or `GLOBAL_DELETE` at the target subdictionary directory.
- To use `/AUDIT`, you need `HISTORY` at the target subdictionary directory.

### Examples

The following example uses the `DCL COPY` command to copy the subdictionary file `CORP.DIC` from disk `DB2:` to `DR3:`. Then, with `RENAME/SUBDICTIONARY`, it resets the directory pointer `CORPORATE` to point to the file `CORP2.DIC` on `DR3:`. Because no user is granted `DELETE` rights by default, the example uses the `DCL SET PROTECTION` command to grant `DELETE` rights for the old file `CORP.DIC` and deletes it.

```
$ COPY CORP.DIC DR3:[CASADAY]CORP2.DIC
$ DMU
DMU> SHOW DEFAULT
CDD$TOP
DMU> LIST
CORPORATE <SUBDICTIONARY> : DB2:[CASADAY]CORP.DIC
PERSONNEL
PRODUCTION
SALES
DMU> RENAME/AUDIT/SUBDICTIONARY=DR3:[CASADAY]CORP2.DIC CORPORATE
DMU> LIST/AUDIT CORPORATE

CDD$TOP.CORPORATE
Subdictionary renamed by CASADAY (UIC [CASADAY]) in process _TTD0:
using VAX CDD Dictionary Management Utility Version 3.1
on 15-AUG-1984 11:44:59:49.
Explanation:
Subdictionary file DB2:[CASADAY]CORP.DIC renamed to
DR3:[CASADAY]CORP2.DIC

DMU> EXIT
$ SET PROTECTION=(S,O:RWED,G,W) CORP.DIC;1

$ DELETE CORP.DIC;1
```

## RENAME/SUBDICTIONARY

The following example moves the VMS subdictionary file CORP.DIC to CASADAY's directory CORPDATA. CASADAY does not change the file name:

```
$ COPY [CASADAY]CORP.DIC [CASADAY.CORPDATA]*
$ DMU
DMU> SHOW DEFAULT
CDD$TOP
DMU> RENAME/SUBDICTIONARY=DB3:[CASADAY.CORPDATA]CORP.DIC CORPORATE
DMU> LIST
CORPORATE <SUBDICTIONARY> : DB3:[CASADAY.CORPDATA]CORP.DIC
PERSONNEL
PRODUCTION
SALES
DMU> EXIT
$ DELETE [CASADAY]CORP.DIC;1
```

## 2.15 RESTORE Command

Copies portions of the directory hierarchy and their related data definitions from a backup file into the CDD.

You specify where to restore the information in the CDD. If the backup file contains history lists, access control lists, or both, DMU restores them also.

With the RESTORE command you can:

- Generate history list entries in the CDD to audit the restoration
- Generate a log of the restored dictionary directories, subdictionaries, and objects
- Restore files for subdictionaries
- Restore objects as new versions of existing objects

### Format

```
RESTORE file-specification [ path-name ]
```

#### *Qualifiers*

#### *Defaults*

```
/AUDIT [ = (quoted-string  
          [ , quoted-string ] ...) ]
```

```
/AUDIT = file-specification
```

```
/NOAUDIT
```

```
/NOAUDIT
```

```
/LOG [ = file-specification ]
```

```
/NOLOG
```

```
/NOLOG
```

```
/ [ NO ] STAGE
```

```
/NOSTAGE
```

```
/ [ NO ] SUBDICTIONARY
```

```
/NOSUBDICTIONARY
```

```
/ [ NO ] VERSION
```

```
/NOVERSION
```

### Parameters

file-specification

The backup file. Use the standard VMS file specification. The default file extension is .BAK.

## RESTORE

### path-name

The dictionary directory or subdictionary at which you are restoring the information. You cannot use wildcards in the path name.

If you omit the path name, the information is added to your current default directory.

### Qualifiers

`/AUDIT [= (quoted-string [, quoted-string] . . .)]`

`/AUDIT = file-specification`

Creates history list entries auditing the restoration of a portion of the CDD directory hierarchy. Entries are made in the history lists of each restored dictionary directory, subdictionary, and object.

You can include explanatory text in history list entries in three ways:

- By simply specifying the `/AUDIT` qualifier. If you include no quoted string or file-specification, DMU provides a default history list entry describing your operation.
- By including quoted strings. Enclose each quoted string in double quotation marks, and enclose the series of strings in parentheses. The parentheses are optional if you specify only one quoted string. If you are using a terminal of the VT200 family, you can use 8-bit characters in quoted strings. Remember that other terminals cannot reproduce 8-bit characters.
- By specifying a file whose contents are to be included in the history list entry. The file specification is a standard VMS file specification, and the default file type is `.DAT`. You can include up to 64 input strings in a history list entry. DMU ignores any excess.

`/NOAUDIT`

Prevents the creation of history list entries.

`/LOG [= file-specification]`

Creates a list of the given names of all restored dictionary directories, subdictionaries, and objects.

The file specification is a standard VMS file specification naming the file into which the log is written. The default file type is `.LOG`. If you use `/LOG` without specifying a file, DMU writes the log to `SYS$OUTPUT`.

# RESTORE

## /NOLOG

Prevents the creation of a log file.

## /STAGE

Withholds committing changes in the CDD until the entire restoration is completed. Using /STAGE assures that no changes remain in the dictionary if a command is interrupted. However, with /STAGE, execution time increases exponentially with the number of directories, subdictionaries, and objects restored.

## /NOSTAGE

Makes changes in the CDD as each dictionary directory, subdictionary, or object is restored. With /NOSTAGE, execution time increases linearly with the number of directories, subdictionaries, or objects restored. However, if you interrupt an unstaged RESTORE command, some of the changes remain in the dictionary.

## /SUBDICTIONARY

Restores subdictionaries as subdictionaries with their original file names.

## /NOSUBDICTIONARY

Restores subdictionaries as directories.

## /VERSION

Restores an object to a directory that contains an object with the same name.

## /NOVERSION

Guarantees that you are not creating a new version of an existing object when you do not intend to do so. If you use /NOVERSION, DMU does not restore an object to a directory that already contains an object with the same name. /NOVERSION is the default.

## Restrictions

- You can restore up to 256 generations of descendants at any one time.
- If a system, group, or process logical name CDD\$VERSION\_LIMIT has been defined, the dictionary will store only the number of versions allowed by the quota CDD\$VERSION\_LIMIT specifies.

# RESTORE

## Usage Notes

- If you enter two opposing qualifiers, such as `/AUDIT` and `/NOAUDIT`, on the same command line, the one that is not the default takes precedence. `/AUDIT` overrides `/NOAUDIT` in this case.
- An object has the same version number in the destination directory as it has in the backup file.
- You may not know which directories, subdictionaries, and objects the backup file contains. You may also not know whether or not the backup file contains history and access control lists. Use `BACKUP/LIST` to display the contents of the backup file and to report whether or not the backup file contains history and access control lists.
- If a directory to be restored already exists, the restoration continues, but no access control or history lists are restored for that directory.
- If an object to be restored has the same name as an existing object in the same directory, DMU restores that object only if you use `/VERSION`. If an object to be restored has the same name *and version number* as an existing object in the same directory, DMU does not restore that object *even* if you use `/VERSION`.
- Enter `CTRL/C` to terminate execution of the command and return to DMU command level. If you have specified `/STAGE`, no changes are made in the CDD.

## Required Privileges

- You need `PASS_THRU` and `EXTEND` at each dictionary directory and subdictionary as it is restored.
- You need `PASS_THRU` and `UPDATE` at each dictionary object as it is restored.
- You need `HISTORY` at each dictionary directory, subdictionary, or object for which you specify `/AUDIT`.
- To restore access control lists included in the backup file, you need `CONTROL` at the destination directory and at each restored dictionary directory, subdictionary, or object.
- You need `UPDATE`, `SEE`, and `PASS_THRU` at the highest existing version of an object to use `/VERSION`.

## RESTORE

- If you want to restore an object and its access control list to a directory already containing an object with the same name, you also need CONTROL at the highest existing version of the object in the destination directory.

### Examples

In the following example, DMU restores CDD\$TOP.PERSONNEL to the directory hierarchy as a subdictionary pointing to the original subdictionary file.

```
DMU> BACKUP CDD$TOP.PERSONNEL PERSONNEL.BAK
DMU> DELETE/ALL/SUBDICTIONARY CDD$TOP.PERSONNEL
DMU> LIST CDD$TOP.PERSONNEL
%DMU-E-CDDERROR, CDD error at "CDD$TOP.PERSONNEL"
-CDD-E-NODNOTFND, directory or object not found
DMU> RESTORE/SUBDICTIONARY PERSONNEL.BAK CDD$TOP
DMU> LIST CDD$TOP.PERSONNEL
PERSONNEL <SUBDICTIONARY> : _DB3:[CASADAY,CDD]PERS.DIC
```

In the following example, the /SUBDICTIONARY qualifier is missing, and DMU restores CDD\$TOP.PERSONNEL as a dictionary directory.

```
DMU> DELETE/ALL/SUBDICTIONARY CDD$TOP.PERSONNEL
DMU> RESTORE PERSONNEL.BAK CDD$TOP
DMU> LIST CDD$TOP.PERSONNEL
PERSONNEL
```

The following example restores a backup file that contains SALARY\_RANGE;2 and SALARY\_RANGE;1 to a directory that contains SALARY\_RANGE;3. Because the directory already contains a version of SALARY\_RANGE, the /VERSION qualifier is used. DMU issues an informational message indicating that the restored versions are not the highest existing versions of the object.

```
DMU> LIST CDD$TOP.PERSONNEL.STANDARDS.>
SALARY_RANGE;3 <CDD$RECORD>
DMU> RESTORE/VERSION/LOG SALARY.BAK CDD$TOP.PERSONNEL.STANDARDS
"SALARY_RANGE;2" restored
%DMU-I-HIGHVER, Higher version of "SALARY_RANGE;2" already exists
"SALARY_RANGE;1" restored
%DMU-I-HIGHVER, Higher version of "SALARY_RANGE;1" already exists
DMU>
```

# SET ABORT

## 2.16 SET ABORT Command

Halts a utility command procedure if DMU or the CDD issues a non-fatal error. Use SET NOABORT if you want a utility command procedure to continue when DMU or the CDD issues a non-fatal error.

### Format

```
SET [ NO ] ABORT
```

### Restriction

SET ABORT works only with command procedures begun within DMU. It does not work with DCL command procedures.

### Usage Note

When SET ABORT halts a command procedure, it returns you to DMU command level. DMU executes all the commands before the error. No command after the command containing the error is executed. If you use the /STAGE qualifier with the COPY or RESTORE command, SET ABORT aborts the entire operation when DMU or the CDD issues an error message.

### Example

In the following example, a command procedure contains a spelling error in a path name. DMU halts the command procedure before additional commands are executed.

```
$ TYPE DELJONES.COM
SET ABORT
SET DEFAULT CDD$TOP.SOLES.JONES
DELETE/ALL
EXIT
$ DMU
DMU> @DELJONES
%DMU-E-CDDERROR, CDD error at "CDD$TOP.SOLES"
-CDD-E-NODNOTFND, directory or object not found
DMU>
```

## 2.17 SET DEFAULT Command

Temporarily sets your default CDD directory. Only dictionary directories or subdictionaries can be used as default directories.

### Format

SET DEFAULT path-name
-----------------------

### Parameter

path-name

The default directory. You cannot use wildcards in the path name.

### Restrictions

- The default specified in the path name must be a dictionary directory or subdictionary.
- The temporary default directory setting remains in force until you set a new default or until you exit from DMU.
- If you set your default directory using a password, your privileges associated with that password remain in effect until you specify another password with that default directory.

### Required Privilege

You need `PASS_THRU` at the target dictionary directory or subdictionary and all of its ancestors in order to set the default directory.

### Example

The following command temporarily sets the default dictionary directory at `CDD$TOP.SALES.JONES`.

```
DMU> SET DEFAULT CDD$TOP.SALES.JONES
DMU> SHOW DEFAULT
CDD$TOP.SALES.JONES
```

# SET PROTECTION

## 2.18 SET PROTECTION Command

Adds entries to the access control list of each specified dictionary directory, subdictionary, or object.

You can also use the SET PROTECTION/EDIT command to add, delete, and modify access control list entries. For a complete description of this command, see Section 2.19.

### Format

```
SET PROTECTION path-name [, path-name]...
```

#### *Qualifiers*

```
/AUDIT[ = (quoted-string  
           [, quoted-string...])
```

```
/AUDIT = file-specification
```

```
/NOAUDIT
```

```
/BANISH = privileges
```

```
/NOBANISH = privileges
```

```
/DENY = privileges
```

```
/NODENY = privileges
```

```
/GRANT = privileges
```

```
/NOGRANT = privileges
```

```
/PASSWORD = quoted-string
```

```
/POSITION = number
```

```
/RIGHTS = uic- or rights-specification
```

```
/TERMINAL = terminal-specification
```

```
/UIC = uic- or rights-specification
```

```
/USERNAME = string
```

#### *Defaults*

```
/NOAUDIT
```

```
/BANISH = none
```

```
/DENY = none
```

```
/GRANT = none
```

```
/POSITION = 1
```

### Parameter

path-name

The dictionary directory, subdictionary, or object owning the access control list you want to change. You cannot use wildcards in the path name.

## Qualifiers

`/AUDIT [= (quoted-string [, quoted-string] . . .)]`

`/AUDIT = file-specification`

Creates history list entries auditing the creation of new access control list entries.

You can include explanatory text in history list entries in two ways:

- By including quoted strings. Enclose each quoted string in double quotation marks, and enclose the series of strings in parentheses. The parentheses are optional if you specify only one quoted string. If you are using a terminal of the VT200 family, you can use 8-bit characters in quoted strings. Note that other terminals cannot reproduce 8-bit characters.
- By specifying a file whose contents are to be included in the history list entry. The file specification is a standard VMS file specification, and the default file type is .DAT. You can include up to 64 input strings in a history list entry. DMU ignores any excess.

`/NOAUDIT`

Prevents the creation of history list entries.

`/BANISH = privileges`

`/DENY = privileges`

`/GRANT = privileges`

Specify the privileges in the new access control list entry. There are three formats for specifying particular privileges:

- By keyword. For example:

`/DENY = CONTROL.`

- By a group of keywords enclosed in parentheses and separated by commas. For example:

`/BANISH = (FORWARD,GLOBAL_DELETE).`

- By a group of key letters enclosed in angle brackets. For example:

`/GRANT = <HPS>.`

## SET PROTECTION

In addition, you can use the keyword "ALL" to specify all thirteen access privileges. For example:

```
/DENY = ALL.
```

With /GRANT, you specify the privileges you want to grant to the specified user or group of users.

With /DENY, you specify the privileges you want to deny to the specified user or group of users.

With /BANISH, you specify the privileges you want to deny permanently to a specified user or group of users, both at the current dictionary directory, subdictionary, or object and at its descendants. At the directory where you specify /BANISH, its effect is the same as /DENY; however, once you have banished particular privileges for a user, no one can grant those privileges to that user at descendants of the directory where the privileges were banished.

When you use /BANISH, /DENY, and /GRANT in combination with one another, the privilege specifications are subject to the following restrictions. In any one access control list entry:

- Privileges specified in /GRANT (but not in /NOGRANT) and in /DENY (but not in /NODENY) cannot be banished.
- Privileges specified in /GRANT (but not in /NOGRANT) cannot be denied.

```
/NOBANISH = privileges
```

```
/NODENY = privileges
```

```
/NOGRANT = privileges
```

Override privilege specifications. The specification of a privilege in /NOBANISH, /NODENY, or /NOGRANT negates the specification of the same privilege in /BANISH, /DENY, or /GRANT, respectively.

The following example illustrates the most common use for the combination of a privilege specification and its negation:

```
/GRANT=ALL/NOGRANT=(CONTROL,GLOBAL_DELETE)
```

Here, it is easier to list the ungranted privileges than the granted ones.

## SET PROTECTION

`/PASSWORD = quoted-string`

Includes a password in the user identification criteria of an access control list entry. The quoted string can be any string of printable characters other than open parenthesis `[`, close parenthesis `]`, or period `.`. If you are using a terminal of the VT200 family, you can use characters from the 8-bit character set. Note that other terminals cannot produce 8-bit characters. Enclose the string in double quotation marks.

`/POSITION = number`

Specifies the relative position within the access control list where you want to place the new entry. If you do not specify the `/POSITION` qualifier, the default position is 1.

`/RIGHTS = uic- or rights-specification`

Includes user identification codes with the user identification criteria of an ACL entry. You can specify any of three alternative types of user identification code: a numeric UIC, an alphanumeric UIC, or a rights identifier.

- A numeric UIC consists of an octal group number and an octal member number separated by a comma and enclosed by either square brackets `[]` or angle brackets `<>`.

You can use the wildcard `*` in place of the group number to identify all group numbers, and in place of the member number group to identify all member numbers. A `/UIC` specification of `[*, *]` matches all user identification codes.

- An alphanumeric UIC consists of a single text string within brackets.
- A rights identifier consists of a single text string that the system manager has defined in the rights database to indicate all members of a particular group.

`/RIGHTS` performs the same function as `/UIC`.

`/TERMINAL = terminal-specification`

Includes information about the terminal(s) in the user identification criteria of an access control list entry.

The terminal specification can be any of the following:

- `TTcn`, `TXcn`, or `WTcn` — a specific terminal number. For example:

`/TERMINAL = TTA7.`

## SET PROTECTION

- **LOCAL** — terminals hardwired to the system. For example:

`/TERMINAL = LOCAL.`

- **NON\_LOCAL** — dial-up and remote terminals, and terminals processing batch and network jobs. For example:

`/TERMINAL = NON_LOCAL.`

- **BATCH** — terminals processing batch jobs. For example:

`/TERMINAL = BATCH.`

- **NETWORK** — terminals processing network jobs. For example:

`/TERMINAL = NETWORK.`

`/UIC = uic-` or rights-specification

Includes user identification codes with the user identification criteria of an ACL entry. You can specify any of three alternative types of user identification code: a numeric UIC, an alphanumeric UIC, or a rights identifier.

- A numeric UIC consists of an octal group number and an octal member number separated by a comma and enclosed by either square brackets ([]) or angle brackets (<>).

You can use the wildcard `*` in place of the group number to identify all group numbers, and in place of the member number group to identify all member numbers. A `/UIC` specification of `[*,*]` matches all user identification codes.

- An alphanumeric UIC consists of a single text string within brackets.
- A rights identifier consists of a single text string that the system manager has defined in the rights database to indicate all members of a particular group.

`/UIC` performs the same function as `/RIGHTS`.

`/USERNAME = string`

Includes a specific VMS username with the user identification criteria of an access control list entry.

# SET PROTECTION

## Restrictions

- You must specify at least one of the user identification criteria: /PASSWORD, /RIGHTS, /TERMINAL, /UIC, or /USERNAME. You can use /RIGHTS and /UIC interchangeably, but you cannot use both in the same command.
- You cannot deny yourself CONTROL or PASS\_THRU privilege to a dictionary directory, subdictionary, or object.

## Usage Note

Enter CTRL/C to terminate execution of the command and return to DMU command level.

## Required Privileges

You need PASS\_THRU and CONTROL at the target dictionary directory, subdictionary, or object to set protection.

## Example

The following SET PROTECTION command creates access control list entry number 2, which grants privileges to user JONES.

```
DMU> SET DEFAULT CDD$TOP.SALES.JONES
DMU> SET PROTECTION/POSITION=2/USERNAME=JONES-
DMU>_/GRANT=<CDEMRW> LEADS_RECORD
DMU> LIST/PROTECTION LEADS_RECORD
JONES.LEADS_RECORD;1
  1:  [*,*], Username: "CASADAY"
      Grant - CDEHMRSUW, Deny - none, Banish - none
  2:  [*,*], Username: "JONES"
      Grant - CDEMRW, Deny - none, Banish - none
```

# SET PROTECTION/EDIT

## 2.19 SET PROTECTION/EDIT Command

Invokes the access control list keypad editor.

Instead of using the SET PROTECTION command and typing the command qualifiers for each access control list entry, you can edit the access control list with the SET PROTECTION/EDIT command. This allows you to see and test changes before you commit access control list modifications.

The functions of selected VT100 and VT200 keyboard and keypad keys are described in Section 2.19.1.

The functions of selected VT52 keyboard and keypad keys are described in Section 2.19.2.

### Format

```
SET PROTECTION/EDIT path-name
```

#### *Qualifiers*

```
/AUDIT [= (quoted-string  
           [, quoted-string]...)]  
/AUDIT = file-specification  
/NOAUDIT
```

#### *Defaults*

```
/NOAUDIT
```

### Parameter

path-name

The dictionary directory, subdictionary, or object whose access control list you wish to edit. You cannot use wildcards in the path name.

### Qualifiers

```
/AUDIT [= (quoted-string [, quoted-string] . . .)]  
/AUDIT = file-specification
```

Creates a history list entry auditing the modification of access control list entries.

You can include explanatory text in history list entries in two ways:

## SET PROTECTION/EDIT

- By including quoted strings. Enclose each quoted string in double quotation marks, and enclose the series of strings in parentheses. The parentheses are optional if you specify only one quoted string. If you are using a terminal of the VT200 family, you can use 8-bit characters in quoted strings. Remember that other terminals cannot reproduce 8-bit characters.
- By specifying a file whose contents are to be included in the history list entry. The file specification is a standard VMS file specification, and the default file type is .DAT. You can include up to 64 input strings in a history list entry. DMU ignores any excess.

`/NOAUDIT`

Prevents the creation of history list entries.

### Restriction

You can use SET PROTECTION/EDIT only from a VT200, VT100, or VT52 terminal. You cannot invoke the editor from a hard-copy terminal.

### Usage Notes

- When you invoke the keypad editor, DMU displays the current access control list for the specified dictionary directory, subdictionary, or object (see Figure 2-1).
- You use individual keypad keys to move the cursor and to add, delete, and modify access control list entries.
- You specify privileges by key letter in each entry, or you can use the asterisk (\*) to indicate all thirteen privileges.
- If you are using a terminal of the VT200 family, you can use 8-bit characters in passwords. Note that other terminals cannot produce 8-bit characters.
- Use CTRL/Z to end the editing session and commit the changes. Use GOLD Z to abort the editing session.

## SET PROTECTION/EDIT

### Required Privileges

You need PASS\_THRU and CONTROL at the target dictionary directory, subdictionary, or object.

```

                                CDD$TOP.SALES
Grant:  CDHPSX          Deny:
Username: CASADAY      Rights:  [*,*]      Banish:
Terminal:              Password:

Grant:  HS             Deny:
Username: JONES        Rights:  [*,*]      Banish:
Terminal:              Password:

Grant:                Deny:  CDEFGHMPRSUMX  Banish:
Username:              Rights:  [*,*]
Terminal:              Password:

[END]
```

Figure 2-1: Sample ACL Editor Entry Display

ZK-8633-HC

## 2.19.1 VT100 and VT200 Keypad Functions

Use selected keyboard and keypad keys to edit the access control list displayed on your VT100 or VT200 terminal.

^   Up Entry	Down Entry   v	← Left Char	→ Right Char
-----------------------	-------------------------	-------------------	--------------------

Delete	Rubout character	Move	Secl	Fndnxt	Del
Linefeed	Rubout field	Show Key		Find	Und
Tab	Move to next field	Advance	Backup	Cut	Entry
Back space	Backup to prior field	Bottom	Top	Paste	Field
CTRL/W	Refresh screen	Field	PW	Char	Enter
CTRL/Z	Return to DMU command level, modify access control list		Del PW		
GOLD Z	Return to DMU command level, do not modify ACL	Entry		Select	
		Open Entry		Reset	

Delete Rubout character

Linefeed Rubout field

Tab Move to next field

Back space Backup to prior field

CTRL/W Refresh screen

CTRL/Z Return to DMU command level,  
modify access control list

GOLD Z Return to DMU command level,  
do not modify ACL

Type a key for help on that key.  
To exit, type a space.

On VT200 keyboards,  
Type F17 for help on VT200 keys.

**Figure 2-2: VT100 and VT200 Keypad Functions**

ZK-8632-HC

### Usage Notes

You can use individual keypad keys to move the cursor, to delete access control list entries or portions of access control list entries, to move entries, and to display the entries that match the user identification criteria of sample users. The following are the ACL editor keys and their functions:

- **GOLD key (PF1).** The GOLD key functions as a shift key in the ACL editor and allows you to use the alternate function of two-function keys. Press GOLD followed by the key whose alternate function you wish to perform.
- You can also use GOLD as a repeat key. Press GOLD, use the keyboard numeric keys to enter the number of repeats, and press the key you want

## SET PROTECTION/EDIT

repeated. You can use **DELETE** or **CTRL/U** to change the number of repeats you specify, so you cannot use **GOLD** to repeat **DELETE** or **CTRL/U**.

- **HELP** key (PF2). **HELP** displays the keypad and its functions. While you are using interactive **HELP**, you can type any editing key for information about that key. To return to editing, type a space.
- **HELP** key (F17). **HELP** displays the auxiliary VT200 editing keypad and its functions.
- **SHOW** key (GOLD 7). The **SHOW** key displays the current definition of a specified key.
- **CTRL/W**. **CTRL/W** refreshes your screen if you receive any messages during an editing session.
- **CTRL/Z**. **CTRL/Z** returns you to DMU command level and modifies the access control list according to the changes you made during the editing session.
- **GOLD Z**. **GOLD Z** returns you to DMU command level, but it makes no modifications to the access control list. Use **GOLD Z** to abort an editing session.

Use the following keys to move the cursor:

- **ADVANCE** key (4). The **ADVANCE** key sets the direction for searches and cursor movement ahead and to the right, toward the end of the access control list.
- **BACKUP** key (5). The **BACKUP** key sets the direction for searches and cursor movement back and to the left, toward the beginning of the access control list.
- **BOTTOM** key (GOLD 4). The **BOTTOM** key advances the cursor to the end of the access control list.
- **TOP** key (GOLD 5). The **TOP** key moves the cursor to the first entry in the access control list.
- Up arrow key. The up arrow key moves the cursor up one entry.
- Down arrow key. The down arrow key moves the cursor down one entry.
- Left arrow key. The left arrow key moves the cursor one character to the left.
- Right arrow key. The right arrow key moves the cursor one character to the right.

## SET PROTECTION/EDIT

- **BACK SPACE** key. The keyboard **BACK SPACE** key moves the cursor back one field.
- **TAB** key. The keyboard **TAB** key advances the cursor one field to the right.
- **MOVE** key (7). The **MOVE** key moves the cursor to the first or last entry of the access control list. First, determine the direction with the **ADVANCE** or **BACKUP** key, and then press **MOVE**.
- **SECT** key (8). The **SECTION** key moves the cursor five entries ahead or five entries back. First, determine the direction with the **ADVANCE** or **BACKUP** key, and then press **SECTION**.

The **NEXT SCREEN** and **PREVIOUS SCREEN** keys on the VT200 auxiliary editing keypad perform the same functions.

- **FIELD** key (1). The **FIELD** key moves the cursor one field. First, determine the direction with the **ADVANCE** or **BACKUP** key, and then press **FIELD**.
- **PW** key (2). The **PASSWORD** key advances the cursor to the password field of the current entry.
- **CHAR** key (3). The **CHARACTER** key moves the cursor one character. First, determine the direction with the **ADVANCE** or **BACKUP** key, and then press **CHARACTER**.
- **ENTRY** key (0). The **ENTRY** key moves the cursor one ACL entry. First, determine the direction with the **ADVANCE** or **BACKUP** key, and then press **ENTRY**.
- **OPEN ENTRY** key (GOLD 0). The **OPEN ENTRY** key opens a position for a new ACL entry above the current location of the cursor.

The **INSERT HERE** key on the VT200 auxiliary editing keypad opens a position for a new ACL entry below the current location of the cursor.

- **FIND** key (GOLD PF3). The **FIND** key prompts you for the character string you want to find. When you type the string and press either **ADVANCE** or **BACKUP**, the editor stores the string in the search buffer, establishes the direction of the search, and moves the cursor to the first occurrence of the string. When you type the string and press the **ENTER** key, the editor stores

## SET PROTECTION/EDIT

the string in the search buffer and moves the cursor to the first occurrence of the string in the direction previously set.

The FIND key on the VT200 auxiliary editing keypad performs the same function.

- **FNDNXT key (PF3).** The FIND NEXT key moves the cursor to the next occurrence of the character or characters in the search buffer. First, determine the direction of the search with the ADVANCE or BACKUP key, and then press FIND NEXT.

Use the following keys to delete access control list entries or portions of access control list entries:

- **DELETE key.** The keyboard DELETE key erases the character to the left of the cursor.
- **LINE FEED key.** The keyboard LINE FEED key erases the column to the left of the cursor.
- **DEL ENTRY key (PF4).** The DELETE ENTRY key deletes all fields of the current access control list entry, regardless of which field the cursor is in.

The REMOVE key on the VT200 auxiliary editing keypad performs the same function.

- **UND ENTRY key (GOLD PF4).** The UNDELETE ENTRY key restores the last entry deleted from the access control list.
- **DEL FIELD key (-).** The DELETE FIELD key deletes the current field of the current access control list entry.
- **UND FIELD key (GOLD -).** The UNDELETE FIELD key restores the last field value deleted from the current field.
- **DEL CHAR key (,).** The DELETE CHARACTER key deletes the character at the current cursor position.
- **UND CHAR key (GOLD ,).** The UNDELETE CHARACTER key restores the last character deleted.
- **DEL PW key (GOLD 2).** The DELETE PASSWORD key deletes everything from the cursor's current position to the end of the current entry.

## SET PROTECTION/EDIT

Use the following keys to move access control list entries:

- **SELECT key (.)**. The **SELECT** key marks the beginning of the range you wish to **CUT** from the access control list. You then move the cursor to the end of the range to complete the selection. This selected range is ready to be **CUT**. You can **CUT** only entire ACL entries.

The **SELECT** key on the VT200 auxiliary editing keypad performs the same function.

- **RESET key (GOLD .)**. The **RESET** key cancels the current select range.
- **CUT key (6)**. The **CUT** key removes the access control list entries you select with the **SELECT** key and places them in the paste buffer.
- **PASTE key (GOLD 6)**. The **PASTE** key inserts the contents of the paste buffer into the access control list.

Use the following keys to display the access control list entries that match sample users. These keys can be useful when you are first setting up and testing the access control lists:

- **DEF USER (GOLD 9)**. The **DEFINE USER** prompts you for the user identification criteria of a sample user whose access privileges you want to display. You can identify the sample user by any of the following criteria: name, rights, terminal number, terminal type, or password at any level in the directory.
- **ENTER key (ENTER)**. Press **ENTER** when you have finished defining the sample user identification criteria. This returns you to the display of the entire access control list.
- **SHOW USER (9)**. The **SHOW USER** key moves the cursor to the first access control list entry matching the user identification criteria you defined with the **DEFINE USER** key. The **SHOW USER** key also displays a listing of the privileges granted at higher levels in the hierarchy, any banished privileges, and the net privileges at the named directory or object for that user.

# SET PROTECTION/EDIT

## 2.19.2 VT52 Keypad Functions

Use selected keyboard and keypad keys to edit the access control list displayed on your VT52 terminal.

LF	Delete prior field			DEL	UP
TAB	Move to next field	GOLD	HELP	ENTRY	ENTRY
BACK SP	Move to prior field			UND	SECTIONI
CTRL/W	Repaint screen	MOVE	FNDNXT	DEL	DOWN
CTRL/Z	Return to DMU command level, modify access control list			FIELD	ENTRY
GOLD Z	Return to DMU command level, do not modify ACL	SHOW	FIND	UND	SECTIONI
		ADVANCE	BACKUP	DEL	RIGHT
				CHAR	CHAR
		BOTTOM	TOP	UND	DEF USERI
		FIELD	PW	CUT	LEFT
			DEL PW	PASTE	CHAR
				SHOW USRI	
		MOVE	ENTRY	SELECT	ENTER
		OPEN	ENTRY	RESET	

\*\*\* For help on a key, press the key  
\*\*\* To return to editing, type a space.

Figure 2-3: VT52 Keypad Functions

ZK-8621-HC

### Usage Notes

You can use individual keypad keys to move the cursor, to delete access control list entries or portions of access control list entries, to move entries, and to display the entries that match the user identification criteria of sample users. The following are the ACL editor keys and their functions:

- **GOLD key (Blue).** The GOLD key functions as a shift key in the ACL editor and allows you to use the alternate function of two-function keys. Press GOLD followed by the key whose alternate function you wish to perform.

You can also use GOLD as a repeat key. Press GOLD, use the keyboard numeric keys to enter the number of repeats, and press the key you want

## SET PROTECTION/EDIT

repeated. You can use **DELETE** or **CTRL/U** to change the number of repeats you specify, so you cannot use **GOLD** to repeat **DELETE** or **CTRL/U**.

- **HELP** key (Red). **HELP** displays the keypad and its functions. While you are using interactive **HELP**, you can type any editing key for information about that key. To return to editing, type a space.
- **SHOW** key (GOLD 7). The **SHOW** key displays the current definition of a specified key.
- **CTRL/W**. **CTRL/W** refreshes your screen if you receive any messages during an editing session.
- **CTRL/Z**. **CTRL/Z** returns you to DMU command level and modifies the access control list according to the changes you made during the editing session.
- **GOLD Z**. **GOLD Z** returns you to DMU command level, but it makes no modifications to the access control list. Use **GOLD Z** to abort an editing session.

Use the following keys to move the cursor:

- **ADVANCE** key (4). The **ADVANCE** key sets the direction for searches and cursor movement ahead and to the right, toward the end of the access control list.
- **BACKUP** key (5). The **BACKUP** key sets the direction for searches and cursor movement back and to the left, toward the beginning of the access control list.
- **BOTTOM** key (GOLD 4). The **BOTTOM** key advances the cursor to the end of the access control list.
- **TOP** key (GOLD 5). The **TOP** key moves the cursor to the first entry in the access control list.
- **Up arrow** key. The up arrow key moves the cursor up one entry.
- **Down arrow** key. The down arrow key moves the cursor down one entry.
- **Left arrow** key. The left arrow key moves the cursor one character to the left.
- **Right arrow** key. The right arrow key moves the cursor one character to the right.
- **BACK SPACE** key. The keyboard **BACK SPACE** key moves the cursor backwards one field.

## SET PROTECTION/EDIT

- **TAB** key. The keyboard **TAB** key advances the cursor one column to the right.
- **MOVE** key (7). The **MOVE** key moves the cursor to the first or last entry of the access control list. First, determine the direction with the **ADVANCE** or **BACKUP** key, and then press **MOVE**.
- **SECT** key (GOLD down arrow). The **SECTION** key moves the cursor five entries ahead or five entries back. First, determine the direction with the **ADVANCE** or **BACKUP** key, and then press **SECTION**.
- **FIELD** key (1). The **FIELD** key moves the cursor one column. First, determine the direction with the **ADVANCE** or **BACKUP** key, and then press **FIELD**.
- **PW** key (2). The **PASSWORD** key advances the cursor to the password field of the current entry.
- **ENTRY** key (0). The **ENTRY** key moves the cursor one ACL entry. First, determine the direction with the **ADVANCE** or **BACKUP** key, and then press **ENTRY**.
- **OPEN ENTRY** key (GOLD 0). The **OPEN ENTRY** key opens a position for a new ACL entry on the line above the current location of the cursor.
- **FIND** key (GOLD 8). The **FIND** key prompts you for the character string you want to find. When you type the string and press either **ADVANCE** or **BACKUP**, the editor stores the string in the search buffer, establishes the direction of the search, and moves the cursor to the first occurrence of the string. When you type the string and press the **ENTER** key, the editor stores the string in the search buffer and moves the cursor to the first occurrence of the string in the direction previously set.
- **FNDNXT** key (8). The **FIND NEXT** key moves the cursor to the next occurrence of the character or characters in the search buffer. First, determine the direction of the search with the **ADVANCE** or **BACKUP** key, and then press **FIND NEXT**.

Use the following keys to delete access control list entries or portions of access control list entries:

- **LINE FEED** key. The keyboard **LINE FEED** key erases the field to the left of the cursor.
- **DEL ENTRY** key (Gray). The **DELETE ENTRY** key deletes all fields of the current access control list entry, regardless of which field the cursor is in.

## SET PROTECTION/EDIT

- **UND ENTRY** key (GOLD Gray). The **UNDELETE ENTRY** key restores the last entry deleted from the access control list.
- **DEL FIELD** key (9). The **DELETE FIELD** key deletes the current field of the current access control list entry.
- **UND FIELD** key (GOLD 9). The **UNDELETE FIELD** key restores the last field value deleted from the current field.
- **DEL CHAR** key (6). The **DELETE CHARACTER** key deletes the character at the current cursor position.
- **UND CHAR** key (GOLD 6). The **UNDELETE CHARACTER** key restores the last character deleted.
- **DEL PW** key (GOLD 2). The **DELETE PASSWORD** key deletes everything from the cursor's current field to the end of the current entry.

Use the following keys to move access control list entries:

- **SELECT** key (.). The **SELECT** key marks the beginning of the range you wish to **CUT** from the access control list. You then move the cursor to the end of the range to complete the selection. This selected range is ready to be **CUT**. You can **CUT** only entire ACL entries.
- **RESET** key (GOLD .). The **RESET** key cancels the current select range.
- **CUT** key (3). The **CUT** key removes the access control list entries you select with the **SELECT** key and places them in the paste buffer.
- **PASTE** key (GOLD 3). The **PASTE** key inserts the contents of the paste buffer into the access control list.

Use the following keys to display the access control list entries that match sample users. These keys can be useful when you set up and test the access control lists:

- **DEF USER** (GOLD right arrow). The **DEFINE USER** prompts you for the user identification criteria of a sample user whose access privileges you want to display. You can identify the sample user by any of the following criteria: name, UIC, terminal number, terminal type, or password at any level in the path name.

## SET PROTECTION/EDIT

- **ENTER** key (ENTER). Press **ENTER** when you have finished defining the sample user identification criteria. This returns you to the display of the entire access control list.
- **SHOW USER** (GOLD left arrow). The **SHOW USER** key moves the cursor to the first access control list entry matching the user identification criteria you defined with the **DEFINE USER** key. The **SHOW USER** key also displays a listing of the privileges granted at higher levels in the hierarchy, any banished privileges, and the net privileges at the named directory or object for that user.

### 2.19.3 The SAMPLE USER SCREEN

Pressing the **DEFINE USER** key (GOLD 9 on the VT100, GOLD right arrow on the VT52) while using the ACL editor produces a formatted screen used to identify a user whose ACL privileges you want to know.

**2.19.3.1 Entering passwords** — Entering necessary passwords on the **SAMPLE USER** screen is very important because the **SHOW USER** key (9 on the VT100, GOLD left arrow on the VT52) calculates the net ACL privileges of a sample user from **CDD\$TOP** to the target directory or object. If you omit a necessary password on the **SAMPLE USER** screen, the sample user's net privileges are different from what they would be if you had entered the password.

The **SAMPLE USER** screen can display all the path names from **CDD\$TOP** to the target directory or object and allows you to enter a password for the sample user at any level in that path.

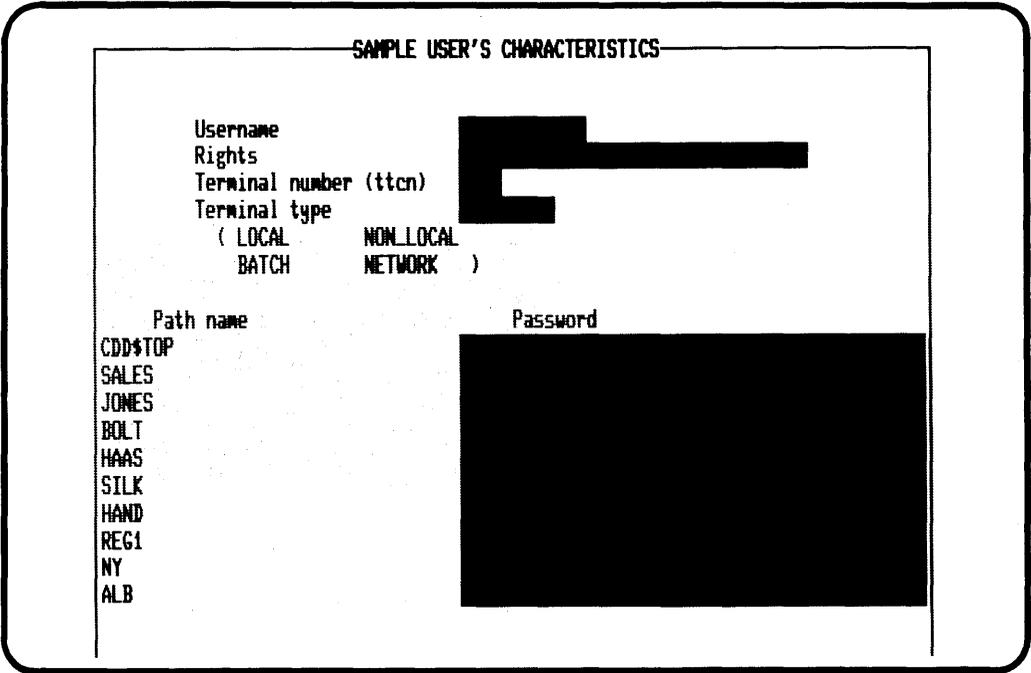
If the path is more than 10 levels deep, the **SAMPLE USER** screen cannot display the entire path at once. For example, the pathname for the object **CDD\$TOP.SALES.JONES.BOLT.HAAS.SILK.HAND.REG1.NY.ALB.AUTO.PARTS** is 12 levels deep. The **SAMPLE USER** screen for this object does not show the lowest two levels, **AUTO** and **PARTS** (see Figure 2-4). You can display those two levels by using the down arrow key or GOLD 4 to move to the bottom of the path name list (see Figure 2-5). You can use the up arrow key to move up the path name list and display **CDD\$TOP** again.

**2.19.3.2 PASS\_THRU privilege error** — A user must have **PASS\_THRU** privilege at every level in the path from **CDD\$TOP** to the target directory or object to reach the target directory or object. Therefore, when you check the

# SET PROTECTION/EDIT

privileges of a sample user who has been denied PASS\_THRU at a higher level in the path, the following error message appears on your screen:

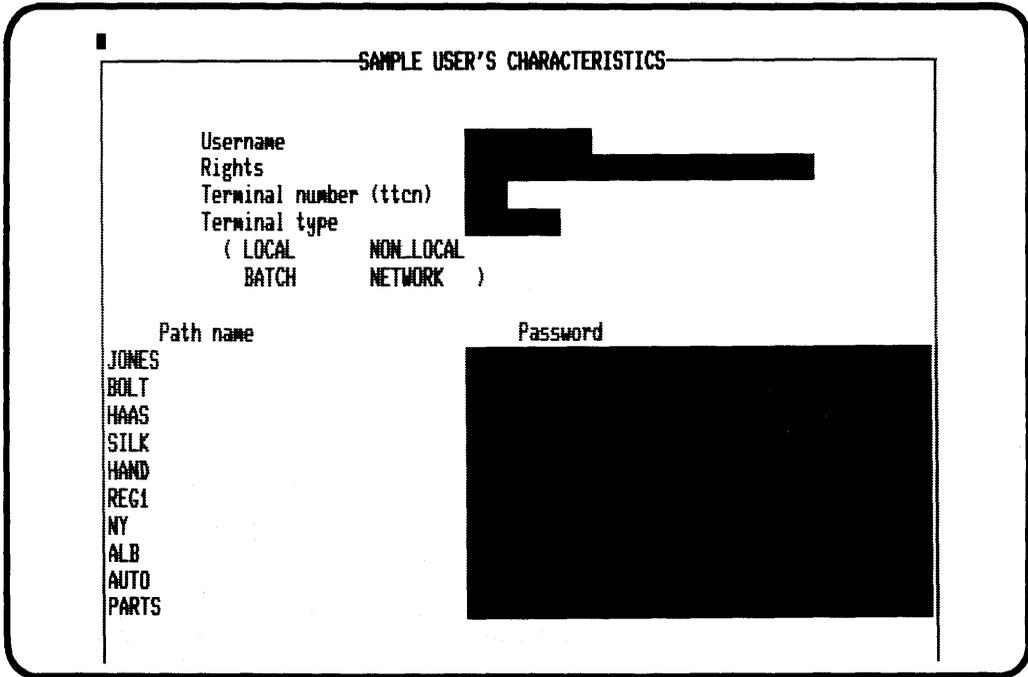
```
%DMU-W-NOSUPRIV, The sample user needs PASS_THRU at a higher directory.
```



ZK-8622-HC

Figure 2-4: SAMPLE USER Screen with First 10 Path Names Displayed

# SET PROTECTION/EDIT



ZK-8623-HC

Figure 2-5: SAMPLE USER Screen with Last 10 Path Names Displayed

# SHOW DEFAULT

## 2.20 SHOW DEFAULT Command

Displays your current default directory.

### Format

```
SHOW DEFAULT
```

### Required Privilege

You need PASS\_THRU at the default directory.

### Example

The following SHOW DEFAULT command displays CDD\$TOP.SALES, the current default directory.

```
DMU> SET DEFAULT CDD$TOP.SALES
DMU> SHOW DEFAULT
CDD$TOP.SALES
```

# SHOW PROTECTION

## 2.21 SHOW PROTECTION Command

Displays your access rights to specified dictionary directories, subdictionaries, and objects.

### Format

```
SHOW PROTECTION path name [, path name] . . .
```

### Parameter

path-name

The dictionary directory, subdictionary, or object to which you want to display your privileges. You can use any of the wildcard characters in the path name. If you do not specify a path name, DMU displays your privileges in your current default directory.

### Usage Note

Enter CTRL/C to terminate execution of the command and return to DMU command level.

### Required Privilege

You need PASS\_THRU at the target dictionary directory, subdictionary, or object.

## SHOW PROTECTION

### Example

The following SHOW PROTECTION command shows that the user has all 13 access privileges to CDD\$TOP.SALES.JONES.

```
DMU> SET DEFAULT CDD$TOP.SALES
DMU> SHOW PROTECTION JONES
CDD$TOP.SALES.JONES
Control (C)      -- may control access control list
Local Delete (D) -- may delete subdictionary, directory, or object
Global Delete (G) -- may delete subdictionary or directory and its
                  children
History (H)      -- may add entries to history list
Pass Thru (P)    -- may pass thru subdictionary or directory
See (S)          -- may see (read) dictionary object
Update (U)       -- may update dictionary object
Extend (X)       -- may create dictionary children
Forward (F)      -- may create subdictionaries
DTR Read (R)     -- may ready DATATRIEVE domain for read
DTR Write (W)    -- may ready DATATRIEVE domain for write
DTR Extend (E)   -- may extend DATATRIEVE table or procedure
DTR Modify (M)   -- may ready DATATRIEVE domain for modify
```

# SHOW VERSION

## 2.22 SHOW VERSION Command

Displays the current DMU version number.

### Format

```
SHOW VERSION
```

### Example

```
DMU> SHOW VERSION  
DMU Version: 3.2
```

## Dictionary Verify/Fix Utility **3**

Use the VAX Common Data Dictionary's Verify/Fix Utility (CDDV) to repair damaged dictionary files or to reorganize dictionary files to enhance performance and reduce size. CDDV contains three commands that allow you to examine and modify dictionary files:

- The **VERIFY** command scans a physical dictionary file and describes any inconsistencies it encounters.
- The **FIX** command scans a dictionary file, deletes damaged dictionary objects and history lists, and reconstructs damaged directories and free page lists.
- The **COMPRESS** command copies a dictionary file, eliminating free space and reorganizing the dictionary file.

CDDV commands follow the general format:

### **COMMAND [/qualifiers] parameters**

The following sections contain complete descriptions of these CDDV commands, qualifiers, and parameters. These descriptions include:

- The syntax **format** you should use for each command
- The **parameters** of each command
- Command **qualifiers** that modify the functions of each command
- **Restrictions** on the ways you can use commands
- **Usage notes** to show you how to use each command

- **Required privileges** for each command
- **Examples** from the sample dictionary (Figure 1-1) to illustrate the use of each command

### **3.1 Restrictions**

- To compress, verify, or fix a dictionary file, you need to be the owner of the file or have VMS SYSPRV or BYPASS privilege.
- The Dictionary Verify/Fix Utility opens a dictionary file for exclusive access. No other user can have access to a dictionary file while you are validating, repairing, or compressing it. If you are using CDDV on the main dictionary file (CDD, DIC), no other user can access any part of the dictionary. If you are using CDDV on a subdictionary file, however, other users can use other parts of the dictionary.

## 3.2 COMPRESS Command

Copies the contents of a dictionary file to a new dictionary file. Using COMPRESS usually produces a more compact file, because COMPRESS copies only currently active dictionary directories and objects, not the free space created previously by deleting portions of the directory hierarchy. COMPRESS also reorganizes the dictionary file to enhance dictionary performance.

### Format

```
COMPRESS input-file-specification output-file-specification
```

### Parameters

input-file-specification

The dictionary file you want to compress. It is a standard VMS file specification. Wildcards are not permitted. The default file extension is .DIC.

output-file-specification

The compressed dictionary file. It is a standard VMS file specification. Wildcards are not permitted. The default file extension is .DIC. You can use the same name as the input file specification.

### Restrictions

- No one else can have access to a dictionary file when you use the CDDV COMPRESS command. If you are compressing a subdictionary file, however, other users can use other parts of the dictionary at the same time.
- You cannot compress a corrupted file. If a file is corrupted, use the CDDV FIX command before using COMPRESS.

### Usage Notes

- Using COMPRESS reduces access time and improves performance by reducing fragmentation in a dictionary file.
- If you have deleted a large number of directories or objects from a dictionary file, there may be a great deal of free space available that the CDD might not fill for quite some time. Using COMPRESS frees that space immediately for use by other files.

## COMPRESS

- COMPRESS does not correct the problem of a full dictionary file. If your dictionary file is full, you must delete obsolete directories and objects or use subdictionaries to create more space.
- Because COMPRESS does not work on a corrupted file, it is a good idea to run CDDV VERIFY before compressing a file. If the file is corrupted, run CDDV FIX before running COMPRESS.
- You can use the same name for the output file and the input file. If you do so, the COMPRESS command creates an output file with the same name but the next higher version number. After using the VMS BACKUP command to save the old dictionary file, you can delete it.
- Compression can take as long as 3 CPU minutes per thousand blocks of dictionary file. If you are compressing the main dictionary file or a large subdictionary file, you should choose a time when the dictionary is not usually in use and use batch mode.
- If compression is interrupted or aborted, a new dictionary file can be created anyway. Delete the new, corrupted file. Because COMPRESS does not modify the input file, you can begin the compression process again.

### Example

The following command compresses a dictionary file, TEST.DIC, and reports the results.

```
CDDV> COMPRESS TEST.DIC TEST.DIC
%CDDV-S-CMPRSSUCC, Dictionary file compression completed successfully
-CDDV-S-BLKSFREED, 120 blocks were freed
```

## EXIT (CTRL/Z)

### 3.3 EXIT Command (CTRL/Z)

Returns you to DCL command level.

#### Format

$\left. \begin{array}{l} \text{EXIT} \\ \text{CTRL/Z} \end{array} \right\}$
---

# FIX

## 3.4 FIX Command

Repairs damaged dictionary files.

FIX describes any errors it finds, and it places this output in a listing file. FIX also modifies the dictionary to repair the damage in the following ways:

- By reconstructing corrupt dictionary directories and restoring their children to them
- By preserving internally consistent dictionary directories and objects whose parents are irreparably damaged
- By pruning off inconsistent dictionary objects, history lists, and directories that cannot be repaired
- By rebuilding the free page list

### Format

FIX file-specification

#### *Qualifiers*

*/[NO]BITMAP*  
*/[NO]COMPLETE*  
*/[NO]FAST*  
*/LISTING [= file-specification]*  
*/NOLISTING*  
*/OUTPUT [= file-specification]*  
*/NOOUTPUT*

#### *Defaults*

See Usage Notes  
*/NOCOMPLETE*  
*/NOFAST*  
*/NOLISTING*  
*/NOOUTPUT*

### Parameter

file-specification

The dictionary file you want to fix. It is a standard VMS file specification. Wildcards are not permitted. The default file extension is .DIC.

## Qualifiers

### /BITMAP

Tallies all of the pages in the directory hierarchy, and modifies the dictionary's free page list by marking all untallied pages as free.

/BITMAP also reconstructs a corrupt free page list.

### /NOBITMAP

Prevents the modification of the free page list.

### /COMPLETE

Scans a dictionary file for corrupt dictionary directories and makes the following repairs:

- When possible, rebuilds corrupt directories. The CDDV can rebuild a corrupt directory when its parent and at least one of its children are intact in the dictionary.
- Reinserts internally consistent directories and objects into repaired parent directories.
- Inserts internally consistent directories and objects whose parents cannot be repaired into CDD\$TOP.CDD\$LOST\_NODES, a special directory created for this purpose.

Corrupt dictionary objects and history lists cannot be repaired, so the CDDV deletes them from the dictionary.

If you have also specified /BITMAP, the CDDV modifies the free page list after it completes the repair of the hierarchy. You cannot specify /COMPLETE and /FAST in the same command line.

### /NOCOMPLETE

Prevents the scanning and repair of dictionary files.

### /FAST

Locates corrupt directories, objects, or history lists and deletes them from a dictionary file. With /FAST, you can also generate a listing of the corrupt directories and objects and of the action taken.

If you have also specified /BITMAP, the CDDV modifies the free page list after it finishes pruning the hierarchy. You cannot specify /COMPLETE and /FAST in the same command line.

## **FIX**

### **/NOFAST**

Prevents the deletion of corrupt directories, objects, and history lists from the dictionary file.

### **/LISTING [= file-specification]**

Specifies a file into which the results of the scan are written. The file specification is a standard VMS file specification. The default file type is .LIS.

If you do not include a file specification, the results are written to SYS\$OUTPUT.

### **/NOLISTING**

Prevents the creation of a listing file.

### **/OUTPUT [= file-specification]**

CDDV accepts /OUTPUT as a synonym for /LISTING.

### **/NOOUTPUT**

CDDV accepts /NOOUTPUT as a synonym for /NOLISTING.

## **Restrictions**

- No one else can access a dictionary file when you use the CDDV FIX command on that file. If you are fixing a subdictionary file, however, other users can access other parts of the dictionary.
- You cannot specify both /COMPLETE and /FAST in the same command line.

## **Usage Notes**

- As a precaution, make a copy of the dictionary file you want to repair before using CDDV FIX.
- /BITMAP is the default if you specify neither /COMPLETE nor /FAST.
- The free page list of a dictionary containing corrupt directories or objects remains unmodified unless you include /COMPLETE or /FAST as well as /BITMAP in the command line.
- Use /COMPLETE to repair damaged dictionary directories. Use /FAST to delete, or prune, damaged directories and their descendants from the dictionary file.

- Reconstructed directories lose their access control lists, history lists, and pointers to subdictionary files.

**Example**

The following command repairs a corrupted dictionary file named **SAMPLE.DIC** and writes a report of the repair into a file named **FIX.LIS** (see Appendix B for a complete listing of **FIX.LIS**).

```
CDDV> FIX/BITMAP/COMPLETE/LISTING=FIX.LIS SAMPLE.DIC
```

# HELP

## 3.5 HELP Command

Displays help text on any of the CDDV commands.

HELP follows the standard syntax of a DCL help command. If you type HELP without specifying a key phrase, the CDDV displays a list of the key phrases for which text is available.

Type an asterisk (\*) in place of the key phrase for all first-level HELP text. Type ellipses (. . .) in place of the key phrase for all of the CDDV HELP text in the library. Type ellipses after a key phrase for all of the HELP text about the specified command, its parameters, and its qualifiers.

### Format

HELP [key-phrase]
-------------------

# SHOW VERSION

## 3.6 SHOW VERSION Command

Displays the current CDDV version number.

### Format

```
SHOW VERSION
```

### Example

```
CDDV> SHOW VERSION  
CDDV Version: 3.2
```

# VERIFY

## 3.7 VERIFY Command

Scans a dictionary file for damaged dictionary directories and objects. You can have a report of the scan written to a listing file or to your terminal.

### Format

VERIFY file-specification

#### *Qualifiers*

*/[NO]COMPLETE*  
*/LISTING [= file-specification]*  
*/NOLISTING*  
*/OUTPUT [= file-specification]*  
*/NOOUTPUT*

#### *Defaults*

*/NOCOMPLETE*  
*/NOLISTING*  
*/NOOUTPUT*

### Parameter

file-specification

The dictionary file you want to verify. Wildcards are not permitted. The file specification is a standard VMS file specification, and the default file type is .DIC.

### Qualifiers

*/COMPLETE*

Scans a dictionary file for corrupt dictionary directories and displays a report including:

- Corrupt dictionary directories that can be reconstructed
- Dictionary directories or objects whose corrupt parent directories can be reconstructed
- Valid, uncorrupted dictionary directories or objects whose corrupt parent directories cannot be reconstructed

## **/NOCOMPLETE**

Generates a list of corrupt dictionary directories and objects.  
**/NOCOMPLETE** is the default.

## **/LISTING [= file-specification]**

Specifies a file into which the results of the scan are written. If you do not include a file specification, the results are written to **SYS\$OUTPUT**.

The file specification is a standard VMS file specification, and the default file type is **.LIS**.

## **/NOLISTING**

Prevents the creation of a listing file.

## **/OUTPUT [= file-specification]**

CDDV accepts **/OUTPUT** as a synonym for **/LISTING**.

## **/NOOUTPUT**

CDDV accepts **/NOOUTPUT** as a synonym for **/NOLISTING**.

## **Example**

The following **VERIFY** command scans a dictionary file named **SAMPLE.DIC** and writes a report into a listing file named **VERIFY.LIS** (see Appendix A for a complete listing of **VERIFY.LIS**).

```
CDDV> VERIFY/COMPLETE/LISTING=VERIFY.LIS  SAMPLE.DIC
```



## Sample CDDV VERIFY Listing of a Corrupt Dictionary File **A**

In the following example, the sample directory hierarchy (Figure 1-1) is stored in the file SAMPLE.DIC. CDDV VERIFY scans the dictionary file and writes a report of the scan into a listing file named VERIFY.LIS:

- Page 1 of the listing shows that the directory SALES is corrupt, but that it can be recreated and relinked to two of its children, SALES\_RECORD;1 and CUSTOMER\_RECORD;1.
- Page 2 reports that LEADS\_RECORD;1 can also be recovered, but that there is no parent to which it can be relinked.
- Page 3 displays the free page list.

VAX CDD Verify/Fix Utility Version 3.00 8-JAN-1984 14:15:11.86 Page 1  
 Command Line: VERIFY/BITMAP/COMPLETE/LISTING=VERIFY.LIS SAMPLE.DIC Dictionary File: \_DB3:[CASADAY.CDD]SAMPLE.DIC:1  
 \* \* \* \* \* Reconstructable Directory Report \* \* \* \* \*

Node Name <Protocol Name>	Descendants	Type of Corruption	Status
SALES <DIRECTORY>		Page has incorrect checksum	Can be recreated
	SALES_RECORD:1 <CDD\$RECORD>		Can be relinked
	CUSTOMER_RECORD:1 <CDD\$RECORD>		Can be relinked

Summary: 1 directory node could be recreated  
 2 descendants would be relinked

VAX CDD Verify/Fix Utility Version 3.00 8-JAN-1984 14:15:11.89 Page 2  
 Command Line: VERIFY/BITMAP/COMPLETE/LISTING=VERIFY.LIS SAMPLE.DIC Dictionary File: \_DB3:[CASADAY.CDD]SAMPLE.DIC:1  
 \* \* \* \* \* Recoverable Node Report \* \* \* \* \*

Node Name <Protocol Name>	Status
LEADS_RECORD:1 <CDD\$RECORD>	Can be recovered

Summary: 1 node can be recovered.

VAX CDD Verify/Fix Utility Version 3.00 8-JAN-1984 14:15:11.91 Page 3  
 Command Line: VERIFY/BITMAP/COMPLETE/LISTING=VERIFY.LIS SAMPLE.DIC Dictionary File: \_DB3:[CASADAY.CDD]SAMPLE.DIC:1  
 \* \* \* \* \* Dictionary Summary \* \* \* \* \*

	OLD	OPTIMAL	ACTUAL
Free Pages	32	36	32
Pages In Use	31	27	31
Pages Recovered		4	0
Total Pages	63	63	63
Corrupt Nodes	1		1
Reconstructable Nodes	1		1
Relinkable Nodes	2		2
Stranded Nodes	1		1

Free Page List Status	Type of Corruption
----- Okay	-----

Dictionary not modified

## Sample CDDV FIX Listing **B**

The CDDV FIX command modifies and repairs corrupt dictionary files, and it lists reports of the actions it has taken. The following example, FIX.LIS, shows the repairs made on the corrupt file SAMPLE.DIC:

- Page 1 of FIX.LIS is similar to page 1 of VERIFY.LIS; it reports that SALES is corrupt. The status, however, has changed because the FIX command has recreated the SALES directory along with its two recoverable children.
- Page 2 shows that LEADS\_RECORD;1 has been recovered. Because its parent directory has been lost, LEADS\_RECORD;1 now resides in CDD\$LOST\_NODES.
- Page 3 displays a modified free page list.

VAX CDD Verify/Fix Utility Version 3.00 8-JAN-1984 14:16:04.88 Page 1  
 Command Line: FIX/BITMAP/COMPLETE/LISTING=FIX,LIS SAMPLE.DIC Dictionary File: \_LDB3:ICASADAY.CDD\$SAMPLE.DIC;1  
 \*\*\*\*\* Reconstructable Directory Report \*\*\*\*\*

Node Name <Protocol Name>	Descendants	Type of Corruption	Status
SALES <DIRECTORY>		Page has incorrect checksum	Recreated
	SALES_RECORD;1 <CDD\$RECORD>		Relinked
	CUSTOMER_RECORD;1 <CDD\$RECORD>		Relinked

Summary: 1 directory node recreated  
 2 descendants relinked  
 0 descendants had a duplicate name and could not be relinked

VAX CDD Verify/Fix Utility Version 3/00 8-JAN-1984 14:16:04.90 Page 2  
 Command Line: FIX/BITMAP/COMPLETE/LISTING=FIX,LIS SAMPLE.DIC Dictionary File: \_LDB3:ICASADAY.CDD\$SAMPLE.DIC;1  
 \*\*\*\*\* Recoverable Node Report \*\*\*\*\*

Node Name <Protocol Name>	Status
LEADS_RECORD;1 <CDD\$RECORD>	Recovered

Summary: 1 node recovered.  
 0 nodes had a duplicate name and could not be recovered.

VAX CDD Verify/Fix Utility Version 3.00 8-JAN-1984 14:16:04.92 Page 3  
 Command Line: FIX/BITMAP/COMPLETE/LISTING=FIX,LIS SAMPLE.DIC Dictionary File: \_LDB3:ICASADAY.CDD\$SAMPLE.DIC;1  
 \*\*\*\*\* Dictionary Summary \*\*\*\*\*

	OLD	OPTIMAL	ACTUAL
Free Pages	32	34	34
Pages In Use	31	29	29
Pages Recovered		2	2
Total Pages	63	63	63
Corrupt Nodes	1	0	0
Reconstructable Nodes	1	0	0
Relinkable Nodes	2	0	0
Stranded Nodes	1	0	0

Free Page List Status	Type of Corruption
----- Okay	-----

Free Page List was modified

## **Sample CDDV VERIFY Listing of an Uncorrupted Dictionary File**

**C**

The following listing file, REVERIFY.LIS, is the report produced by a CDDV VERIFY scan of SAMPLE.DIC after modification by CDDV FIX.

VAX CDD Verify/Fix Utility Version 3.00 8-JAN-1984 14:16:55.81 Page 1  
Command Line: VERIFY/BITMAP/COMPLETE/LISTING=REVERIFY,LIS SAMPLE.DIC Dictionary File: \_DB3:[CASADAY,CDD]SAMPLE.DIC:1  
\*\*\*\*\* Dictionary Summary \*\*\*\*\*

	OLD	OPTIMAL	ACTUAL
Free Pages	34	34	34
Pages In Use	29	29	29
Pages Recovered		0	0
Total Pages	63	63	63

Free Page List Status	Type of Corruption
----- Okay	-----

Dictionary not modified

## Error Messages **D**

This appendix lists error messages generated by DMU, CDDV, and the CDD program interface. After each message, there is an explanation of the message and of the action you should take to correct the problem.

For example, if you misspell the given name of a directory in a DMU command, you receive the following error messages:

```
DMU> LIST PORDUCTION
-ZDMU-E-CDDERROR, CDD error at "PORDUCTION"
-CDD-E-NODNOTFND, directory or object not found
```

The CDD cannot find a directory named "PORDUCTION" because none exists. To correct the problem, reenter the command:

```
DMU> LIST PRODUCTION
```

DMU, CDDV, and CDD error messages contain the following elements:

- A facility name preceded by a percent sign (%) or a hyphen (-) and followed by a hyphen (-), such as %DMU- or -CDD-.
- The severity code followed by a hyphen (-). Table D-1 lists severity codes in order of increasing severity.

**Table D-1: Explanation of Severity Codes**

Code	Severity	Explanation
S	Success	Indicates that your command has executed successfully.
I	Information	Reports on actions taken by the software.
W	Warning	Indicates conditions that do not abort execution of a command, but that might require checking to ensure that the result is the one you expected.
E	Error	Signals conditions that are not fatal, but that do abort the execution of commands. Most of the messages documented in this appendix are of severity E.
F	Fatal	Indicates severe errors; the software cannot continue. Most commonly, the severe error messages you receive from DMU, CDDV, or CDD are followed by messages explaining the problem from other facilities such as RMS. In some cases, however, fatal errors indicate an internal problem with the software.

- The diagnostic error message name followed by a comma (,). This name identifies the message. In the sections that follow, messages are alphabetized by diagnostic error message name.
- The diagnostic error message. The message is a brief description of the problem. Error messages may contain string substitutions identifying the particular file names or path names in question. In the documentation, these string substitutions are indicated by angle brackets (< >) within a message. For example:

```
I/O error while closing file <file name>
```

If you received this message, the facility generating it, in this case CDD, would have substituted the name of the file for <file name>.

---

**Note**

---

You may occasionally receive a CDDL error message when using DMU EXTRACT. CDDL error messages are documented in the *VAX Common Data Dictionary Data Definition Language Reference Manual*.

---

**WHEN YOU RECEIVE AN ERROR MESSAGE**, you have several choices of action to take. At the simplest level, you can correct syntax errors by retyping the command lines that contain them. If you receive a severe error indicating a

bug in CDD or one of the utilities, or if you receive a CDD or utilities error message that is not documented, you should immediately notify your system manager or the person responsible for maintaining the dictionary. If your facility has a service contract with DIGITAL, your system manager should submit a Software Performance Report (SPR) on the forms provided by DIGITAL. Be sure to include a VMS BACKUP copy of the dictionary file and a listing of the commands that produced the error.

## **D.1 DMU Error Messages**

### **ACLNOTFND, Access control list entry at position <number> not found**

**Explanation:** In the DELETE/PROTECTION command, you specified a nonexistent entry with the /POSITION = <number> qualifier.

**User Action:** Use the LIST/PROTECTION command for a listing of current access control list entries.

### **ACTSELNAN, Select range is already active**

**Explanation:** You have pressed the SELECT key when the select range was already active.

**User Action:** When the select range is active, you must either “cut” an ACL entry or group of entries, or deactivate the select range by pressing GOLD SELECT.

### **ALLORSPEC, POSITION cannot be specified both explicitly and by ALL**

**Explanation:** Within one command line, you have used /POSITION = <number> to specify one ACL entry and /POSITION = ALL to specify all ACL entries; you cannot use both specifications in the same command line.

**User Action:** Choose the appropriate option and reenter the command.

### **AMBIG, Ambiguous command**

**Explanation:** You have entered an ambiguous abbreviation of a DMU command. For example, EX could mean EXIT or EXTRACT.

**User Action:** Make sure your command line uniquely identifies a command.

**BADCMD, Command syntax error**

Explanation: You have entered an invalid sequence of keypad commands.

User Action: If necessary, refer to Appendix A of the *VAX CDD/Plus User's Guide* or press PF2 for HELP. Then reenter the command.

**BADDEFLT, Command aborted because CDD\$DEFAULT cannot be used**

Explanation: CDD tried unsuccessfully to set your default dictionary directory using the CDD\$DEFAULT logical name.

User Action: When you exit from DMU, edit your login command file and define CDD\$DEFAULT as an existing path name.

**BADESC, Escape sequence syntax error**

Explanation: The command sequence you have entered has produced an invalid escape sequence; the command cannot be processed. For instance, hitting the <ESC> key while in the keypad editing mode produces this error.

User Action: Check the syntax and reenter the command.

**BADKEEP, The PURGE/KEEP specification must be greater than zero**

Explanation: You cannot specify KEEP=0 with the PURGE command.

User Action: Specify a positive number with the /KEEP qualifier.

**BADKEY, You cannot use that key in this mode**

Explanation: You have used some illegal keys (for example, CUT and PASTE) while defining a sample user in the access control list editor.

User Action: Press PF2 while you are in the DEFINE USER mode for a display of legal keys.

**BADNODTYP, A dictionary object cannot have children**

Explanation: You have tried to create children for a dictionary object; only dictionary directories or subdictionaries can have children.

User Action: None.

**BADOPTION, Specified combination of qualifiers is not allowed**

**Explanation:** You have entered a combination of qualifiers that cannot be used in the same command line (for example, the qualifiers /KEEP and /BEFORE cannot both be options for the same DELETE/HISTORY command).

**User Action:** Check the *VAX Common Data Dictionary Utilities Reference Manual* for legal qualifier combinations.

**BADOBJNAM, Object name does not have version number appended**

**Explanation:** In certain cases, DMU requires you to specify the version number of an object.

**User Action:** Retype the command, specifying the version number of the object.

**BADTEMPLT, Template record at <quoted string> could not be extracted**

**Explanation:** You have tried to extract a template record with the EXTRACT/RECORD/TEMPLATE command, but DMU could not extract the specified template record.

**User Action:** Check the CDD to determine whether the template record still exists. It may have been deleted or renamed.

**BEGOFACL, Beginning of ACL hit**

**Explanation:** You have incorrectly tried to move the cursor above the first entry in the access control list. The cursor has remained at the first entry.

**User Action:** None.

**CANTTRANS, Unsupported language feature in record <quoted string>**

**Explanation:** You have used the EXTRACT/RECORD command to extract a record definition, such as those created by DBMS DDL, that CDDL did not compile. It contains features that cannot be expressed in CDDL.

**User Action:** None.

**CDDERROR, CDD error at <quoted string>**

Explanation: An error occurred when DMU accessed CDD for the specified dictionary directory, subdictionary, or object. An accompanying message explains the problem.

User Action: Refer to the documentation for the accompanying error message for explanation and user action.

**CDDTOPDEF, Your default directory is now CDD\$TOP**

Explanation: DMU has set your default directory to CDD\$TOP.

User Action: None.

**CHGSYNTAX, Syntax error in change mode command near <quoted string>**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

**CHLDTOPRN, You cannot copy a directory or object to its immediate parent**

Explanation: You have tried to copy a directory or object to its current parent directory.

User Action: None.

**CMDENDSYN, Syntax error — you did not end the command correctly**

Explanation: You have entered a command line with too many parameters.

User Action: Check the syntax and reenter the command.

**CMDTERM, Syntax error — you entered an incomplete command**

Explanation: You have entered a command line with too few parameters.

User Action: Check the syntax and reenter the command.

**CMMANDSYN, Syntax error in command name**

Explanation: You have mistyped a DMU command.

User Action: Check the spelling and reenter the command.

### **CTRLCAST, Execution terminated by operator**

**Explanation:** You have stopped the execution of the command by typing CTRL/C.

**User Action:** You should check whether you have made any changes to the dictionary file during the partial execution. For example, the COPY/NOSTAGE command copies directories and objects as it reads them; if you interrupt the command, some directories and objects may have been copied and some may not.

### **DELHSTOPT, Syntax error — you must specify the /HISTORY qualifier**

**Explanation:** You have used a qualifier (/BEFORE, for example) with the DELETE command that requires the /HISTORY qualifier.

**User Action:** Reenter the command using the /HISTORY qualifier.

### **DELPROOPT, Syntax error — you must specify the /PROTECTION qualifier**

**Explanation:** You have used a qualifier (/POSITION, for example) with the DELETE command that requires the /PROTECTION qualifier.

**User Action:** Reenter the command using the /PROTECTION qualifier.

### **DENIED, Node temporarily locked by another user — try again later**

**Explanation:** Another user's update has locked a directory or object that you have tried to access. Completion of that update will release the lock.

**User Action:** Wait a few moments and try again.

### **ENDOFACL, End of ACL hit**

**Explanation:** You have incorrectly tried to move the cursor beyond the last entry in the access control list. The cursor has remained at the last entry.

**User Action:** None.

### **FILESYN, Syntax error in file specification**

Explanation: You have entered an illegal VMS file specification.

User Action: Check the VMS documentation set for legal file names and file types, and then reenter the file specification.

### **HIGHVER, Higher version of object "<object name>" already exists**

Explanation: You have created a version of this object for which a higher-numbered version already exists.

User Action: Unless you have specified a lower version number for an express purpose, avoid future confusion by changing the version number to the highest existing number.

### **HLPLIBERR, Unable to open HELP library**

Explanation: An accompanying message from the librarian explains why the HELP file cannot be opened.

User Action: Refer to the documentation for the accompanying error message for explanation and user action.

### **HLPNOTAVL, Help not available**

Explanation: You have typed HELP after receiving the HLPLIBERR message.

User Action: Correct the problem identified in the librarian message that was issued with HLPLIBERR.

### **HLPNOTGVN, Failed to obtain text from HELP library**

Explanation: You have accessed the HELP library, but the librarian has not returned the text. The librarian message following HLPNOTGVN describes the problem.

User Action: Refer to the documentation for the accompanying error message for explanation and user action.

### **ILLENT, Illegal entity type**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

### **ILLPASWOR, Illegal password string**

**Explanation:** You have specified a password containing one or more illegal characters. Legal passwords are strings of printable characters except parentheses and the period.

**User Action:** Check the *VAX Common Data Dictionary Utilities Reference Manual* for proper password syntax and try again.

### **ILLPRIV, Illegal privilege specification**

**Explanation:** You have entered an illegal privilege specification in the SET PROTECTION command. Legal privilege specifications are strings of key letters for the CDD and DATATRIEVE access privileges.

**User Action:** Refer to Appendix A in the *VAX CDD/Plus User's Guide* and Section 2.18 of this manual or type HELP SPECIFY PRIVILEGES for more information. Also, you cannot deny yourself CONTROL or PASS\_THRU access, even if your privilege specification is otherwise legal.

### **ILLRELVER, Relative version numbers are not allowed in this context**

**Explanation:** You cannot specify a relative version number for the given name in the RENAME command.

**User Action:** Enter the given name with an absolute version number or no version number at all.

### **ILLTERM, Illegal terminal name or class**

**Explanation:** You have entered an illegal terminal specification. Legal terminal specifications include: LOCAL, TTnn, NON\_LOCAL, BATCH, and NETWORK.

**User Action:** Check Appendix A of the *VAX CDD/Plus User's Guide* for legal terminal specifications.

### **ILLTRMCLS, Illegal value for terminal class field**

**Explanation:** You have entered an illegal terminal class during an access control list editing session. Legal terminal classes include: LOCAL, TTnn, NON\_LOCAL, BATCH, and NETWORK.

**User Action:** Check Appendix A of the *VAX CDD/Plus User's Guide* for more information on terminal classes.

### **ILLUIC, Illegal UIC**

**Explanation:** You have entered an illegal user identification code as an access control list identification criterion. A legal UIC consists of a group octal number and a member octal number, separated by commas and enclosed in square brackets ([ ]). You can use the \* wildcard symbol to replace either the group number, the member number, or both.

**User Action:** Check the syntax and reenter the command.

### **ILLUSRNAM, Illegal username**

**Explanation:** You have entered an illegal VMS username as an access control list identification criterion. The user name you specify must be a legal VMS username.

**User Action:** Check the syntax and reenter the command.

### **ILLVERFLD, STRUCTUREs cannot be virtual fields — UNSPECIFIED assumed**

**Explanation:** In a listing file you have tried to expand a template record that illegally specifies a virtual field for a STRUCTURE field.

**User Action:** Modify and recompile the illegal STRUCTURE field description.

### **INVERTYP, Unsupported terminal type**

**Explanation:** To use the ACL editor, you must have a VT52, a terminal from the VT100 family, or a terminal from the VT200 family.

**User Action:** None.

### **IOREADERR, I/O error during read from terminal**

**Explanation:** DMU was unable to read your command from the terminal.

**User Action:** Reenter the command.

### **ITEMSYN, Syntax error in item specification**

**Explanation:** You have entered an incorrect item specification for the LIST command.

**User Action:** Check Section 2.10 of the *VAX Common Data Dictionary Utilities Reference Manual* for the correct syntax for LIST/ITEM.

**LOWVER, Lower version of object “<object name>” still exists**

Explanation: You have explicitly deleted a particular version of an object, but a lower version still exists.

User Action: If you intended to delete all versions of the object, use the wildcard (;\*) or no version number at all.

**NOACL, Directory’s or object’s access control list is empty**

Explanation: You have tried to process or display an access control list that has no ACL entries.

User Action: To create an access control list entry, use the SET PROTECTION or the SET PROTECTION/EDIT command.

**NOCTLACC, You do not have CONTROL access to this directory or object**

Explanation: You have tried to read, write, or modify the access control list for a dictionary directory, subdictionary, or object to which you do not have CONTROL privilege. You must have CONTROL to process an access control list.

User Action: None.

**NODEFKEY, That key may not be redefined**

Explanation: You have tried to redefine a key that cannot be redefined.

User Action: Select another key for your definition.

**NODNOTOBJ, <quoted string> is a directory and cannot be extracted**

Explanation: You tried to extract the source text of a dictionary directory. This is not possible; dictionary directories do not have source text. You can only use the EXTRACT command with dictionary objects.

User Action: None.

**NOEDIT, directory or object <given name> was not readied for editing**

Explanation: An accompanying message will explain the problem. For instance, you may have attempted to edit an access control list to which you do not have CONTROL privilege.

User Action: Refer to the documentation for the accompanying error message for explanation and user action.

**NOFINISH, Edit session was not terminated successfully**

Explanation: The editing session ended abnormally. An accompanying message will explain the problem.

User Action: Refer to the documentation for the accompanying error message for explanation and user action. Reinvoke the ACL editor and make necessary corrections.

**NOMATCH, No ACL entry matches the sample user's characteristics**

Explanation: While in the ACL editor, you pressed the SHOW USER key (keypad 9) to display the sample user's access privileges; none of the access control list entries matched the sample user's characteristics.

User Action: None.

**NONODFND, No directories or objects found**

Explanation: You have specified a path name using wildcards, but DMU cannot find any directories, subdictionaries, or objects that match the specification.

User Action: Make sure you are in the correct default directory.

**NOOBJPURG, No objects purged for "<object name>"**

Explanation: You entered a PURGE command for an object that does not exist.

User Action: None.

**NOPRIV, You do not have sufficient privilege to perform this operation**

**Explanation:** You do not have the privilege(s) you need to execute the command you have entered.

**User Action:** You can enter the SHOW PROTECTION command to learn what privileges you have, and check the *VAX Common Data Dictionary Utilities Reference Manual* for the privileges required for the operation you want to perform. If you have CONTROL privilege, change the protection; otherwise, see your system manager.

**NOPSWDALL, Syntax error — password not allowed with wildcards**

**Explanation:** You cannot associate a password with a given name that contains a wildcard character. You must use the complete given name along with a password specification.

**User Action:** Reenter the command, eliminating any wildcard in a directory or object name that has a password associated with it.

**NOROOTATT, Does not conform to record description protocol version 4**

**Explanation:** This is an internal software error in the CDD or in another product using the CDD.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

**NOSELRAN, Select range is not active**

**Explanation:** You have tried to use the CUT key of the access control list editor, but you have not selected a range of entries to cut.

**User Action:** Activate the select range, and then CUT.

**NOSRCTXT, Source text for object <quoted string> is not available**

**Explanation:** You used the EXTRACT/SOURCE command with a dictionary object for which there is no source text available. For example, you cannot extract the source text from dictionary objects entered in the CDD through DBMS because DBMS does not create source text.

**User Action:** Use the EXTRACT/RECORD command to translate the record definition into CDDL source text.

**NOSUPRIV, The sample user needs PASS\_THRU at a higher directory**

Explanation: The sample user cannot reach the directory or object where you are checking his privileges because he lacks PASS\_THRU privilege at a higher directory in the path.

User Action: You must find the directory where he lacks PASS\_THRU and grant PASS\_THRU to him before he can reach the target directory or object.

**NOTFOUND, String not found**

Explanation: The string you have placed in the search buffer does not occur in the current access control list.

User Action: Make sure the string specification is accurate and reenter the command.

**NOTRECORD, object "<object name>" is not a record definition**

Explanation: You have tried to use the EXTRACT/RECORD command with a CDD directory or with an object that is not a record definition. You can use the /RECORD qualifier only with objects of the type CDD\$RECORD.

User Action: None.

**NULSELNAN, Select range does not contain any entries**

Explanation: The select range is active, but you have not moved the cursor to select any entries. When you pressed the CUT key, no entries were placed in the PASTE buffer.

User Action: None.

**OBJALREXI, Object <quoted string> already exists; not superseded**

Explanation: You have tried to create a new version of a dictionary object in a directory that already contains an object with the same name and version number.

User Action: Delete the object already in the directory, if you no longer need it. Otherwise, use DMU RENAME to change the name or version number of one of the conflicting objects.

**OPTIONSYN, Syntax error in option specification**

**Explanation:** You have mistyped a command qualifier, and DMU cannot process your command.

**User Action:** Correct the error and reenter the command line.

**OPTREQUIR, Command is missing a required qualifier**

**Explanation:** You have entered a command line that is missing a required qualifier, such as the /AUDIT qualifier with the MEMO command.

**User Action:** Type HELP or refer to the *VAX Common Data Dictionary Utilities Reference Manual* for more information.

**PATHNMSYN, Syntax error in path name**

**Explanation:** You have entered an illegal path name.

**User Action:** Make sure the path name you specify contains only given names separated by periods, and that the path name contains legal characters (A-Z, 0-9, \$, and \_).

**POSNSYN, Syntax error in position specification**

**Explanation:** You have entered an illegal position specification. Legal specifications are: /POSITION=ALL, /POSITION=integer, or /POSITION=(integer, integer,...).

**User Action:** Check the syntax and reenter the command.

**PRIVSYN, Syntax error in privilege specification**

**Explanation:** You have entered an incorrect privilege specification. Correct specifications include keywords separated by commas and enclosed in parentheses, or key letters enclosed in angle brackets.

**User Action:** Check Section 2.18 of this manual for privilege specification syntax.

**PRNTOCHILD, You cannot copy a parent directory to a child**

**Explanation:** You have tried to copy a directory to one of its child directories. DMU does not allow this operation.

**User Action:** None.

**PSSWDSYN, Syntax error in password specification**

**Explanation:** You have entered an illegal password. Legal passwords are strings of printable characters other than the parentheses and the period. Enclose passwords in quotation marks with the /PASSWORD qualifier, or in parentheses in path names.

**User Action:** Check the syntax and reenter the command.

**PSWDIGNOR, Password ignored in this context — use SET PROTECTION command**

**Explanation:** You associated a password with the given name of a directory to be created with the CREATE command. DMU CREATE ignores passwords when creating new directories.

**User Action:** Use the SET PROTECTION or SET PROTECTION/EDIT command to associate the password with the new directory.

**REQPROT, PROTECTION qualifier is required when you specify /POSITION**

**Explanation:** You specified the /POSITION qualifier without specifying the /PROTECTION qualifier in the LIST command. The /POSITION qualifier of the LIST command refers to the relative position of ACL entries.

**User Action:** You must specify the /PROTECTION qualifier to display access control lists.

**REQRECORD, /RECORD qualifier is required when you specify /TEMPLATE**

**Explanation:** You cannot use the /TEMPLATE qualifier with EXTRACT unless you also specify /RECORD.

**User Action:** Reenter the command adding the /RECORD qualifier.

**REQUIRUIC, Syntax error — you must specify user identification criteria**

**Explanation:** You must specify at least one user identification criterion when you create an access control list entry.

**User Action:** Reenter the command using at least one of the user identification criteria. For a description of user identification criteria, see Section 2.18 of this manual.

**RMSBUFOVF, Attempted entry exceeds maximum RMS buffer size**

Explanation: You illegally tried to back up a string or attribute that was longer than 65,535 characters.

User Action: None.

**RMSCREERR, Unsuccessful attempt to create file <quoted string>**

Explanation: DMU was unable to create an output file. The RMS message that follows explains the problem.

User Action: Refer to VMS documentation for explanations of RMS errors.

**RMSERROR, RMS error at <quoted string>**

Explanation: An RMS error occurred while DMU was trying to read a subdictionary directory; an accompanying RMS message explains the problem.

User Action: Refer to VMS documentation for explanations of RMS errors.

**RMSOPNERR, Unsuccessful attempt to open file <quoted string>**

Explanation: DMU was unable to open an input file. An accompanying RMS message explains the problem.

User Action: Refer to VMS documentation for explanations of RMS errors.

**STRSYNTAX, Syntax error in string specification**

Explanation: You have entered an incorrect string specification.

User Action: Check for syntax errors like the failure to open or close parentheses or quotation marks, or the use of illegal characters.

**SYNTAX, Parsing error near <quoted string>**

Explanation: This message tells you where in the command line you entered a syntax error. A second, more specific message follows.

User Action: Refer to the documentation for the accompanying error message for explanation and user action.

### **TERMSPEC, Syntax error in terminal specification**

Explanation: You have entered an incorrect terminal specification.

User Action: Check Section 2.18 of this manual for legal terminal specifications.

### **TERNOTSUP, Your terminal is not supported in change mode**

Explanation: You are attempting to use a terminal other than a VT52, a VT100 family terminal, or a VT200 family terminal with the access control list editor. The ACL editor supports only VT52, VT100 family, and VT200 family terminals.

User Action: None.

### **TIMANDNUM, /KEEP and /BEFORE options cannot be used together**

Explanation: You have used the /KEEP and the /BEFORE qualifiers in the same command line. /KEEP specifies the number of most recent history list entries to be saved in a history list purge; /BEFORE specifies the date of the oldest history list entry to be saved.

User Action: Specify one or the other.

### **TOOFEWCRI, ACL entry has no user identification criteria**

Explanation: You have tried to create an access control list entry without any user identification criteria.

User Action: Include at least one of the following qualifiers in an access control list entry: /PASSWORD, /TERMINAL, /UIC, or /USERNAME.

### **UICSYNTAX, Syntax error in UIC specification**

Explanation: You have specified an incorrect user identification code (UIC).

User Action: Check Section 2.18 of this manual for UIC syntax.

### **UNEXPEOF, BACKUP file is corrupt or incomplete**

Explanation: Your backup file is corrupt.

User Action: If the dictionary is also corrupt, then restore an earlier version of the backup file. If the dictionary is not corrupt, then make a new backup file.

### **USRNMESYN, Syntax error in username specification**

**Explanation:** You entered an invalid VMS username as a user identification criterion for an ACL entry.

**User Action:** Refer to Section 2.18 of this manual for proper VMS username syntax. Correct the error and reenter the command.

### **VERALREXI, Other versions already exist — object “<object name>” not created**

**Explanation:** You have tried to place an object in a directory that already contains an object with the same name, and you have not used the /VERSION qualifier.

**User Action:** If you want to create an additional version of an existing object, reenter the command using the /VERSION qualifier. If you do not want to create an additional version of an existing object, delete any objects in the directory with the same name as the object you are inserting in the directory, then reenter the command.

### **VERNOTALL, Version numbers cannot be used on a directory name**

**Explanation:** You have used a version number as part of the given name of a CDD directory. Only CDD objects can have multiple versions.

**User Action:** Reenter the command without specifying a version number for the directory.

### **WLDRELVER, Relative version numbers cannot contain wildcard characters**

**Explanation:** You cannot use a relative version number with an object when you have used a wildcard character in the object's given name.

**User Action:** Reenter the command using the complete given name of the object.

## **D.2 CDDV Error Messages**

CDDV displays messages to your terminal, and it writes messages to the reports that list the status of dictionary files as they are validated. The following sections describe both types of messages.

## **D.2.1 Messages to the Terminal**

### **AMBIG, Ambiguous command**

**Explanation:** You have entered an ambiguous abbreviation of a CDDV qualifier. For example, /NO could refer to any of several qualifiers.

**User Action:** Enter a longer abbreviation of the qualifier.

### **CMDTERM, Syntax error — you entered an incomplete command**

**Explanation:** You have entered a command line with too few parameters.

**User Action:** Check the syntax and reenter the command.

### **CMMANDSYN, Syntax error in command name**

**Explanation:** You have mistyped a CDDV command.

**User Action:** Check the spelling and reenter the command.

### **CMPRSFAIL, Compression failed on file <file name>, page <page number>**

**Explanation:** CDDV was unable to compress the specified file.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

### **DICNOTCRE, Unable to create new dictionary file**

**Explanation:** CDDV is unable to create a new dictionary file. An accompanying RMS error message explains the problem.

**User Action:** Refer to the VMS documentation for an explanation of the RMS error.

### **DICNOTOPN, Unable to open dictionary file**

**Explanation:** CDDV cannot open the dictionary file to verify it. An accompanying RMS message explains the problem.

**User Action:** Refer to VMS documentation for explanations of RMS error messages.

### **DICTCRPT, Dictionary is corrupt**

**Explanation:** The dictionary file you specified contains damaged directories or objects.

**User Action:** Use the CDDV FIX command to repair the damaged file.

### **DICTNOMOD, Dictionary not modified**

**Explanation:** The dictionary was not modified.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

### **EXTENDERR, I/O error while extending file <file name>**

**Explanation:** CDDV tried unsuccessfully to extend a dictionary file during a FIX operation. An accompanying message explains the problem.

**User Action:** If the second message is a system message, refer to VMS documentation. If the second message is a CDD message, refer to the CDD message section in this appendix. Otherwise, use the instructions at the beginning of the error message listing of this appendix.

### **FILESYN, Syntax error in file specification**

**Explanation:** You have entered an illegal VMS file specification.

**User Action:** Check the VMS documentation set for legal file names and file types, then reenter the file specification.

### **INVCMMND, Invalid command and option combination**

**Explanation:** You have entered an invalid combination of commands and command qualifiers.

**User Action:** Type HELP or refer to the *VAX Common Data Dictionary Utilities Reference Manual* for valid commands and the qualifiers that modify them.

### **NOFREEPAG, Dictionary file is full**

**Explanation:** Your dictionary file is full.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

### **NOPRVMSG, Insufficient privilege to run CDDV on this file**

**Explanation:** To use CDDV on a dictionary file, you must be the owner of the file or have VMS SYSPRV or BYPASS privilege.

**User Action:** None.

### **OPTIONSYN, Syntax error in option specification**

**Explanation:** You have mistyped a command qualifier, and CDDV cannot process your command.

**User Action:** Correct the error and reenter the command line.

### **REPNOTCRE, Unable to create listing file**

**Explanation:** CDDV is unable to open an output file in which to list its report. An accompanying message explains the problem.

**User Action:** Refer to the documentation for the accompanying error message for explanation and user action.

### **SYNTAX, Parsing error near "<error>"**

**Explanation:** This message tells you where in the command line you entered a syntax error. A second, more specific message follows.

**User Action:** Refer to the documentation for the accompanying error message for explanation and user action.

### **WRITERERROR, I/O error writing VBN <number> in file <file name>**

**Explanation:** CDDV was unable to modify part of the dictionary file being FIXED. An accompanying message explains the problem.

**User Action:** If the second message is a system message, refer to VMS documentation. If the second message is a CDD message, refer to the CDD message section in this appendix. Otherwise, use the instructions at the beginning of the error message listing of this appendix.

## **D.2.2 Report Messages**

These messages, appearing in CDDV listings, have different meanings, depending on whether they are in a CDDV VERIFY or a CDDV FIX listing:

- CDDV VERIFY listing — The message tells you that you should use CDDV FIX to correct this error.
- CDDV FIX listing — The message tells you that CDDV FIX found this error in your dictionary; the steps CDDV FIX took to correct the error appear elsewhere in the listing.

### **BADCHKSUM, Page has incorrect checksum**

Explanation: An I/O error has corrupted your dictionary file.

User Action: Follow the instructions at the beginning of this section.

### **BADCLUSEQ, Page has incorrect cluster sequence number**

Explanation: An I/O error has corrupted your dictionary file.

User Action: Follow the instructions at the beginning of this section.

### **BADGRPSEQ, Page has incorrect group sequence number**

Explanation: A system crash or power failure has corrupted your dictionary.

User Action: Follow the instructions at the beginning of this section.

### **INVBCKPTR, Invalid back pointer to free page block**

Explanation: CDDV has encountered an error in the free page list.

User Action: Follow the instructions at the beginning of this section.

### **INVBLKPTR, Dictionary's name chain is invalid**

Explanation: The dictionary directory hierarchy has been corrupted.

User Action: Follow the instructions at the beginning of this section.

### **INVMAPSIZ, Invalid free page list size**

Explanation: The dictionary directory hierarchy has been corrupted.

User Action: Follow the instructions at the beginning of this section.

### **PAGALRUSE, Page is already in use by another cluster**

Explanation: There is a dictionary pointer to the wrong page, possibly indicating that the page has been overwritten in error.

User Action: Follow the instructions at the beginning of this section.

### **WRONGPAGE, Page has wrong page number**

Explanation: An I/O error has corrupted your dictionary.

User Action: Follow the instructions at the beginning of this section.

## **D.3 CDD Error Messages**

### **ACLNOTFND, Access control list entry not found**

Explanation: You have specified a non-existent ACL entry, perhaps by including a POSITION number greater than the number of entries in the list.

User Action: Reenter your command line.

### **ATTALREXI, Attribute already exists**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

### **ATTNOTFND, Attribute not found**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

### **BADCHKSUM, Page has incorrect checksum**

Explanation: Your dictionary has become corrupt.

User Action: Run CDDV VERIFY to determine the extent of the damage.

**BADCLUSEQ, Page has incorrect cluster sequence number — cluster is invalid**

Explanation: Your dictionary has become corrupt.

User Action: Run CDDV VERIFY to determine the extent of the damage.

**BADGRPSEQ, Page has incorrect group sequence number — cluster is invalid**

Explanation: Your dictionary has become corrupt.

User Action: Run CDDV VERIFY to determine the extent of the damage.

**BADPATH, Syntax error in pathname**

Explanation: You have entered an invalid path name.

User Action: Check Appendix A of the *VAX CDD/Plus User's Guide* for legal path name specifications.

**BADSTREAM, Context stream is corrupt and cannot be used**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

**BADUIC, Syntax error in UIC parameter**

Explanation: You have entered a user identification code (UIC) incorrectly.

User Action: Check Appendix A of the *VAX CDD/Plus User's Guide* for UIC syntax.

**BADUSRCNX, User provided invalid context pointer**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

**BADVERPRO, Protocol of new object does not match that of prior versions**

**Explanation:** You have tried to create an additional version of a CDD object that does not match the CDD type of existing version. Objects with the same name in the same directory must be the same CDD type.

**User Action:** Choose a different name for the new object.

**BADVERNUM, Version number is out of range**

**Explanation:** You have entered a version number greater than 32,767 or less than -32,766. The largest possible object version number is 32,767.

**User Action:** Reenter the command, using the correct version number of the object.

**CDDNOTINS, VAX CDD is not installed on your system**

**Explanation:** You do not have VAX CDD installed on your system.

**User Action:** Have your system manager install VAX CDD, then retry the operation.

**CELNOTEMP, Cell is not empty**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

**CELTOOBIG, Cell number is greater than the list size**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

**CLOSEPER, Character following the close parenthesis is not a period**

**Explanation:** You have omitted a period in a path name specification. The period belongs to the right of a protected directory specification.

**User Action:** Check Appendix A of the *VAX CDD/Plus User's Guide* for proper use of passwords in dictionary path names.

**CLOSERROR, I/O error while closing file <file name>**

**Explanation:** CDD was unable to close a file. An accompanying message may explain the problem.

**User Action:** If the second message is a system message, refer to VMS documentation. If the second message is a CDD message, refer to the explanation elsewhere in this appendix. If no other message appears, use the instructions at the beginning of the error message listing of this appendix.

**CONSECPER, Path name contains two or more consecutive periods**

**Explanation:** Each given name in a path name must be separated from adjacent given names by a single period. The path name you entered contains two or more consecutive periods.

**User Action:** Correct the error and reenter the command.

**DEADLOCK, Node temporarily locked by another user — try again later**

**Explanation:** You and another user are attempting to access the same dictionary directory, subdictionary, or object and are blocking each other.

**User Action:** Try again. If you still get a DEADLOCK error, try again later.

**DENIED, Node temporarily locked by another user — try again later**

**Explanation:** Another user has locked the dictionary directory, subdictionary, or object that you are trying to access. This lock will be released when that user finishes the operation.

**User Action:** Wait a few moments and try again.

**EMPTYCELL, Cell is empty**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

**ENDOFATT, End of attributes**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

**ENDOFDIR, End of directory**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

**EXCENQLIM, Your process has exceeded its authorized ENQLM**

Explanation: The operation you are attempting requires more lock resources than you have been allocated in your process enqueue limit (ENQLM).

User Action: The DCL command SHOW PROCESS/QUOTA displays your current limit. If it is insufficient, check with your system manager.

**EXCVERLIM, Exceeded version limit for object**

Explanation: You have tried to create more versions of an object than CDD\$VERSION\_LIMIT, as currently defined, allows.

User Action: Purge the object if you want to create a newer version.

**EXTENDERR, I/O error while extending file <file name>**

Explanation: CDD tried unsuccessfully to extend a file. An accompanying message explains the problem.

User Action: If the second message is a system message, refer to VMS documentation. If the second message is a CDD message, refer to the explanation elsewhere in this appendix. Otherwise, use the instructions at the beginning of the error message listing of this appendix.

**FILALRACC, Another subdictionary directory already points to this file**

**Explanation:** You have illegally attempted to create two subdictionary directories that point to the same subdictionary file; or you have tried to rename a subdictionary whose file is already open.

**User Action:** Specify a unique subdictionary file for each subdictionary directory you create; or, if you are trying to rename a subdictionary whose file is already open, exit from DMU to close the file and then reinvoke DMU to rename the subdictionary file.

**FNDWRGVER, Requested version of object not found — found another version**

**Explanation:** You specified a version of an object that does not exist; however, another version of that object does exist.

**User Action:** None.

**ILLATTOPR, Illegal attribute operator**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

**ILLCHRVER, Illegal character in version number**

**Explanation:** You have used a character in a version number that is not allowed. You can use only a number, an asterisk (\*), or a minus sign (-) in a version number specification.

**User Action:** Correct the error and reenter the command.

**ILLDATYP, Illegal data type**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

**ILLOCCON, Illegal lock conflict**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

**ILLNAMCHR, A given name contains a character other than A-Z, 0-9, \$, or \_**

**Explanation:** You have used a given name containing illegal characters. Only the characters A-Z, 0-9, \$, and \_ are legal in a CDD given name.

**User Action:** Check the syntax and reenter the command.

**ILLNAMSIZ, A given name's length is zero or greater than 31**

**Explanation:** You have attempted to use a given name containing more than 31 characters. Valid CDD names cannot exceed this limit.

**User Action:** Reenter the command.

**ILLSEMICO, Illegal use of a semicolon (;) in path name**

**Explanation:** A semicolon is only legal when used after the given name of an object as part of the version number. It cannot be used elsewhere in the path name.

**User Action:** Reenter the command without the illegal semicolon.

**ILLUIC, This is not a valid UIC**

**Explanation:** You have entered an illegal user identification code as an access control list identification criterion. A legal UIC consists of a group octal number and a member octal number, separated by commas and enclosed in square brackets ([ ]). You can use the \* wildcard symbol to replace either the group number, the member number, or both.

**User Action:** Check the syntax and reenter the command.

**ILLVALSIZ, Illegal value size parameter**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

**ILLVERLIM, CDD\$VERSION\_LIMIT cannot be converted to a valid number**

**Explanation:** You have defined CDD\$VERSION\_LIMIT to be an illegal number.

**User Action:** Redefine the logical name CDD\$VERSION\_LIMIT or deassign the logical name.

**ILLVERNUM, Illegally used or specified version number**

**Explanation:** You specified a relative version number (or a version number of 0) with a command that does not allow such specification.

**User Action:** Check the reference material in Chapter 2 for the command you used to determine the legal ways to specify a version number.

**INCOMPVER, Object has incompatible version number: <number>**

**Explanation:** You have a dictionary file that is incompatible with the version of CDD running on your system.

**User Action:** Install the latest version of CDD on your system.

**INSFILNAM, Subdictionary file spec is missing device or directory name**

**Explanation:** Subdictionary file specifications must include both a device and a directory. You can use system logical names for the specification, but they must translate to a valid device and directory. You have tried to create a subdictionary file without specifying either a device, a directory, or both.

**User Action:** Correct the error and reenter the command.

**INV1STCHR, The first character of a given name is not A-Z**

**Explanation:** You have illegally attempted to use a given name whose first character is not a letter from A to Z.

**User Action:** Correct the error and reenter the command.

**INVATTNAM, Invalid attribute name**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

**INVBUFSIZ, Invalid buffer size**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

**INVCELNUM, Invalid cell number**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

**INVCLOPAR, Path name contains an invalid close parenthesis**

Explanation: You have entered a path name containing an invalid close parenthesis. Parentheses are legal in path names only when they are used to enclose passwords.

User Action: Correct the error and reenter the command.

**INVHISCIB, Invalid history list cluster pointer**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

**INVHYPHEN, Path name contains an invalid hyphen**

Explanation: A hyphen in a path name indicates a directory one level up. If you use one or more hyphens in a path name, they must be the first elements in the name, and they must precede all specified given names. You have specified a path name with a hyphen following a given name.

User Action: Correct the error and reenter the command.

**INVLSTCHR, The last character of given name is not A-Z or 0-9**

Explanation: You have illegally attempted to use a given name whose final character is not A-Z or 0-9.

User Action: Correct the error and reenter the command.

**INVNODNAM, You cannot specify both a location code and a full pathname**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

**INVOPNPAR, Path name contains an invalid open parenthesis**

Explanation: You have entered a path name containing an open parenthesis in an illegal position. Parentheses are valid in path names only when they enclose a password associated with a particular directory, subdictionary, or object. Parentheses cannot be nested, and each open parenthesis must be matched by a corresponding close parenthesis.

User Action: Correct the syntax and reenter the command.

**INVPARAM, Invalid parameter**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

**INVPERIOD, Path name contains an invalid period**

Explanation: You have entered a path name containing a period where one is not allowed. Separate given names by single periods.

User Action: Make sure you do not have consecutive periods, or periods within given names, in your path specification.

**INVRLSOPT, Invalid release locks options**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

**LCCNOTAVL, Location code not available**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

### **LCCNOTFND, Location code not found in hash table**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

### **LCKNOTGRA, Lock not granted**

**Explanation:** CDD tried unsuccessfully to use the VMS lock manager to fulfill your request. An accompanying message explains the problem.

**User Action:** If the second message is a system message, refer to VMS documentation. If the second message is a CDD message, refer to the explanation elsewhere in this appendix. Otherwise, use the instructions at the beginning of the error message listing of this appendix.

### **LCKNOTREL, Lock not released**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

### **LISTOOBIG, List size too big**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

### **LOCNOTREL, Lock not released**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

### **NODALREXI, Directory or object already exists**

**Explanation:** You have tried to give the same name to two directory children of the same dictionary directory or subdictionary, or you have tried to give the same name and version number to two object children of the same directory or subdictionary.

**User Action:** Give each dictionary directory or subdictionary a unique name. Two or more objects in the same directory can have the same name, provided they have unique version numbers.

### **NODHASCHI, Directory not deleted because it has descendants**

**Explanation:** Your attempt to delete a dictionary directory or subdictionary has failed because that directory or subdictionary has children. To delete a directory and all of its descendants, you must have the GLOBAL\_DELETE privilege, and you must specify either the /ALL or the /NOCHECK qualifier. If you try to delete a directory with children without meeting these criteria, CDD displays this message.

**User Action:** None.

### **NODICT, CDD\$DICTIONARY not defined in system logical name table**

**Explanation:** CDD\$DICTIONARY is not defined in your system logical name table, so CDD cannot find your dictionary file.

**User Action:** Have your system manager run the startup procedure, CDDSTRUP.COM.

### **NODNOTFND, Directory or object not found**

**Explanation:** CDD cannot find the directory or object you specified, either because it does not exist or because your default directory is different from what you think it is.

**User Action:** Check spelling and the default setting; then reenter the command.

### **NODNOTLOC, Directory or object not locked as required**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

### **NOFREEPAG, Dictionary file is full**

**Explanation:** You have used all available dictionary disk blocks ((2\*\*17)-1).

**User Action:** Delete obsolete directories and objects, then use subdictionaries to make more dictionary space available.

### **NOLCKTBL, CDD is unable to use the VMS lock manager**

**Explanation:** CDD tried unsuccessfully to use the VMS lock manager to fulfill your request. An accompanying message explains the problem.

**User Action:** If the second message is a system message, refer to VMS documentation. If the second message is a CDD message, refer to the explanation elsewhere in this appendix. Otherwise, use the instructions at the beginning of the error message listing of this appendix.

### **NOPRIV, You do not have sufficient privilege to perform this operation**

**Explanation:** You do not have the privilege(s) required for the operation you wish to perform.

**User Action:** Refer to the *VAX Common Data Dictionary Utilities Reference Manual* for the required privileges, and check with your system manager if you need privileges that you do not currently have.

### **NOSTORDIR, User tried to store user attribute on directory**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

### **NOTASTREE, CDD is not AST reentrant**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

**NOTDICFIL, Existing file was found, but it is not a CDD dictionary file**

**Explanation:** You tried to create or use a subdictionary file, but the file you named already exists and is not a CDD dictionary file.

**User Action:** Make sure you have the proper file specification and reenter the command.

**NOTDIRNOD, Tried to use an object as a directory**

**Explanation:** You have attempted to use a dictionary object in a place where only a dictionary directory can be used.

**User Action:** Specify the name of a dictionary directory or subdictionary when you reenter the command.

**NOTLIST, Location code does not reference a list**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

**NOTNODE, Location code does not belong to a directory or object**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

**NOTOWNER, Another subdictionary directory already points to this file**

**Explanation:** You have incorrectly tried to create two subdictionaries pointing to the same subdictionary file.

**User Action:** Specify a unique file for the subdictionary you are creating, and then reenter the command.

**NOTREQACC, You cannot deny yourself CONTROL or PASS\_THRU access**

**Explanation:** As a safeguard against inadvertent denial of privileges, CDD does not permit you to add any access control list entries that deny you CONTROL or PASS\_THRU. The ACL entry you have entered denies you one or both of these privileges, so CDD refused to accept the privilege specification and generated this error message.

**User Action:** None.

**NOTSUBDIC, Directory is not a subdictionary pointer**

**Explanation:** You have attempted a RENAME/SUBDICTIONARY operation on a dictionary directory. Only subdictionary directories point to subdictionaries.

**User Action:** None.

**NOTSYSLOG, Dictionary filespec must use system logical names**

**Explanation:** You have used a process or group logical in a dictionary file specification. Only system logical names are allowed.

**User Action:** None.

**NOTSYSNAM, Not system defined attribute**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

**NOUSRCNTX, User context not established**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

**NULNODNAM, Given name is null**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

**PASTTOP, You used too many hyphens in a path name**

**Explanation:** You have specified too many hyphens at the front of a path name and caused CDD to try to back up past CDD\$TOP in the directory hierarchy.

**User Action:** Check the syntax of the specified path name and reenter the command.

**PERINPASS, Password contains a period**

**Explanation:** You have specified a password containing a period, but periods and parentheses are not legal characters in passwords.

**User Action:** Change the password and reenter the command.

**READERERROR, Error reading or validating VBN <number> in file <file name>**

**Explanation:** A dictionary file is corrupt. An accompanying message explains the problem.

**User Action:** If the second message is a system message, refer to VMS documentation and restore the file from a VMS BACKUP copy. If the second message is a CDD message, run CDDV.

**READONLY, Dictionary file is open for read-only access**

**Explanation:** You do not have WRITE access to the dictionary file you want to update.

**User Action:** See your system manager.

**RENINCTYP, Object being renamed is a different type from previous version**

**Explanation:** You have tried to rename an object, but another object with the same name and a different type already exists in that directory. You can only create a new version of an object if the CDD type of the new version is the same as the CDD type of the existing version.

**User Action:** Choose a different given name for the new object.

**REAONLATT, User tried to create a read only attribute**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

**RENTOTOP, You cannot rename a directory or object to CDD\$TOP**

Explanation: You have illegally tried to change the name of a directory, subdictionary, or object to CDD\$TOP.

User Action: None.

**TOOFEWPAR, Too few parameters**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

**TOOMANLCC, Too many location codes in one stream**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

**TOOMANPAR, Too many conflicting parameters**

Explanation: This is an internal software error.

User Action: Use the instructions at the beginning of the error message listing of this appendix.

**TOPNOTDEL, You cannot delete CDD\$TOP**

Explanation: You have illegally tried to delete the root dictionary directory.

User Action: None.

**TOPNOTREN, You cannot rename CDD\$TOP**

Explanation: You have illegally tried to rename the root dictionary directory.

User Action: None.

**VERNOTALL, Version number not allowed with a directory name**

**Explanation:** You have used a version number as part of the given name of a CDD directory. Only CDD objects can have multiple versions.

**User Action:** Reenter the command without specifying a version number for the directory.

**WRITERERROR, I/O error writing VBN <number> in file <file name>**

**Explanation:** CDD was unable to update the named dictionary file. An accompanying message explains the problem.

**User Action:** If the second message is a system message, refer to VMS documentation. If the second message is a CDD message, refer to the explanation elsewhere in this appendix. Otherwise, use the instructions at the beginning of the error message listing of this appendix.

**WRODATYP, Attribute has the wrong data type**

**Explanation:** This is an internal software error.

**User Action:** Use the instructions at the beginning of the error message listing of this appendix.

**WRONGPAGE, Copy of page in dictionary file has the wrong page number**

**Explanation:** Your dictionary file is corrupt.

**User Action:** Run CDDV VERIFY to determine the extent of the problem.



### **E.1 Subdictionary File Specifications Can Contain Any System Logical Name**

In previous versions of CDD, within a dictionary file specification, you could use only logical names from the LNM\$SYSTEM\_TABLE table. With this release, you can use a logical name from any system logical name table defined in LNM\$SYSTEM.

### **E.2 DECReporter Support**

If DECReporter is installed on your system, you can use it to create reports using definitions stored in the CDD. You can use either DATATRIEVE or DECReporter's Report Dictionary Management Utility to manipulate dictionary objects. With DATATRIEVE, DECReporter lets you create reports using Rdb/VMS and DBMS definitions in the CDD.

DECReporter supports two ways to define a report:

- A menu interface, which helps you navigate through the CDD to specify the report criteria.
- The DECReporter program interface, which lets you use DECReporter calls in a command file.

To run DECReporter, type REPORT at DCL level. You can type HELP REPORT for information on DECReporter commands and qualifiers.

## **E.3 Using Multiple Versions of CDD Objects with DATATRIEVE**

With CDD Versions 3.0 and later, you can create multiple versions of **dictionary objects**. If you are using DATATRIEVE, however, you need DATATRIEVE Version 3.x to access multiple versions of CDD objects. Earlier versions of DATATRIEVE do not support multiple versions of dictionary objects.

## **E.4 Installing CDD with SYSPRV**

If you use the INSTALL utility to install a known image with the qualifier /PRIVILEGED=(SYSPRV), you must install the associated message file with the qualifier /SHARED. Otherwise, the image cannot access its own message file. For example, if you install DMU with SYSPRV, you should install DMUEXC.EXE, DMU's message file, with the qualifier /SHARED.

## **E.5 Path Names Pointing to Subdictionary Files**

When you move a subdictionary file from one system to another, make sure that the subdictionary's path name on the new system is the same as it was on the old system. Because both DATATRIEVE and DBMS use full path names by default to locate data definitions, changing the path name of a subdictionary will produce errors when DATATRIEVE or DBMS tries to access definitions in that subdictionary.

Currently, CDD does not check to ensure that subdictionary path names remain unchanged.

## **E.6 DMU Restrictions**

You should be aware of several restrictions in the CDD Data Management Utility (DMU).

### **E.6.1 Copying and Renaming a Dictionary Object Protected by a Password**

Cases may occur where you are granted UPDATE and CONTROL privileges for an object only through a password. If this is true, you cannot copy any version of another object with the same name as the protected object to the directory containing the protected object, because DMU COPY provides no way to specify the protected object's password.

In a case where you are granted the UPDATE privilege for an object only through a password, you cannot rename another object as a new version of the protected object, because DMU RENAME provides no way to specify the protected object's password.

### **E.6.2 Using /STAGE with DMU RESTORE**

If you specify the /STAGE qualifier with the DMU RESTORE command, all changes remain in virtual memory until the restoration is completed, to assure that either all or none of the changes are made. Therefore, the size of the portion you back up is limited by the amount of virtual memory. By contrast, /NOSTAGE (the default) frees virtual memory in stages, as each directory is restored.

The DMU BACKUP and RESTORE commands should be used to back up only portions of the dictionary hierarchy. Use the DCL BACKUP command to back up an entire dictionary file.

### **E.6.3 Using a Wildcard with the DELETE/SUBDICTIONARY Command**

With one DELETE/SUBDICTIONARY command you can delete only 255 or fewer subdictionaries.



## Index

In this index, a page number followed by a “t” indicates a table reference. A page number followed by an “e” indicates an example reference. A page number followed by an “f” indicates a figure reference.

### A

- Access control list editor, 2-66 to 2-80
  - sample entry display, 2-68
  - SAMPLE USER screen, 2-78 to 2-80
  - VT100 keypad functions, 2-69 to 2-73
  - VT52 keypad functions, 2-74 to 2-78
- Access control lists, 1-3
- Access privileges, 1-4
  - required for BACKUP, 2-5
  - required for COPY, 2-11
  - required for CREATE, 2-15
  - required for DELETE, 2-19
  - required for DELETE/HISTORY, 2-22
  - required for DELETE/PROTECTION, 2-24
  - required for EXTRACT, 2-29
  - required for LIST, 2-37
  - required for MEMO, 2-41
  - required for PURGE, 2-43
  - required for RENAME, 2-47
  - required for RESTORE, 2-56
  - required for SET DEFAULT, 2-59
  - required for SET PROTECTION, 2-65
  - required for SET PROTECTION/EDIT, 2-68
  - required for SHOW DEFAULT, 2-81
- Access privileges (Cont.)
  - required for SHOW PROTECTION, 2-82
- /ACCESS qualifier
  - with LIST, 2-33
- /ACL qualifier
  - with CREATE, 2-13
- ADVANCE key
  - with VT100 ACL editor, 2-70
  - with VT52 ACL editor, 2-75
- /ALL qualifier
  - with DELETE, 2-17
- Arrow keys
  - See down arrow key
  - See left arrow key
  - See right arrow key
  - See up arrow key
- /AUDIT qualifier
  - with BACKUP, 2-3
  - with COPY, 2-8
  - with CREATE, 2-13
  - with DELETE/PROTECTION, 2-23
  - with EXTRACT, 2-27
  - with MEMO, 2-40
  - with RENAME, 2-44
  - with RENAME/SUBDICTIONARY, 2-49
  - with RESTORE, 2-54
  - with SET PROTECTION, 2-61
  - with SET PROTECTION/EDIT, 2-66 to 2-67
- /AUDIT \_ TRAIL qualifier
  - with LIST, 2-34

## **B**

- BACK SPACE key
  - with VT100 ACL editor, 2-71
  - with VT52 ACL editor, 2-75
- BACKUP command, 2-2 to 2-6
- BACKUP key
  - with VT100 ACL editor, 2-70
  - with VT52 ACL editor, 2-75
- /BANISH qualifier
  - with SET PROTECTION, 2-61 to 2-62
- /BEFORE qualifier
  - with DELETE/HISTORY, 2-21
- /BITMAP qualifier
  - with FIX, 3-7
- BOTTOM key
  - with VT100 ACL editor, 2-70
  - with VT52 ACL editor, 2-75
- /BRIEF qualifier
  - with LIST, 2-34

## **C**

- CDD\$VERSION \_ LIMIT
  - with COPY, 2-10
  - with RENAME, 2-45
  - with RENAME/SUBDICTIONARY, 2-50
  - with RESTORE, 2-55
- CDDV, 1-4
  - COMPRESS command, 3-3 to 3-4
  - EXIT command, 3-5
  - FIX command, 3-6 to 3-9
    - general command format, 3-1
  - HELP command, 3-10
  - SHOW VERSION command, 3-11
  - VERIFY command, 3-12 to 3-13
- CHARACTER key
  - with VT100 ACL editor, 2-71
- /CHECK qualifier
  - with DELETE, 2-17
- /COMPLETE qualifier
  - with FIX, 3-7
  - with VERIFY, 3-12
- COMPRESS command, 3-3 to 3-4
- COPY command, 2-7 to 2-12
- CREATE command, 2-13 to 2-16
- CTRL/W
  - with VT100 ACL editor, 2-70
  - with VT52 ACL editor, 2-75

## **CTRL/Z**

- with VT100 ACL editor, 2-70
- with VT52 ACL editor, 2-75

## **CUT key**

- with VT100 ACL editor, 2-73
- with VT52 ACL editor, 2-77

## **D**

- DEFINE USER key
  - with VT100 ACL editor, 2-73
  - with VT52 ACL editor, 2-77
- DELETE CHARACTER key
  - with VT100 ACL editor, 2-72
  - with VT52 ACL editor, 2-77
- DELETE command, 2-17 to 2-20
- DELETE ENTRY key
  - with VT100 ACL editor, 2-72
  - with VT52 ACL editor, 2-76
- DELETE FIELD key
  - with VT100 ACL editor, 2-72
  - with VT52 ACL editor, 2-77
- DELETE key
  - with VT100 ACL editor, 2-72
- DELETE PASSWORD key
  - with VT100 ACL editor, 2-72
  - with VT52 ACL editor, 2-77
- DELETE/HISTORY command, 2-21 to 2-22
- DELETE/PROTECTION command, 2-23 to 2-25
- /DENY qualifier
  - with SET PROTECTION, 2-61 to 2-62
- Dictionary directory, 1-1
- Dictionary Management Utility
  - See DMU
- Dictionary objects, 1-1
- Dictionary types, 1-3
- Dictionary Verify/Fix Utility
  - See CDDV
- Dictionary, sample, 1-2f
  - listing, 1-3e
- Directory
  - dictionary, 1-1
- Directory hierarchy, 1-2f
- DMU, 1-4
  - BACKUP command, 2-2 to 2-6
  - COPY command, 2-7 to 2-12
  - CREATE command, 2-13 to 2-16
  - DELETE command, 2-17 to 2-20

## DMU (Cont.)

DELETE/HISTORY command, 2-21 to 2-22

DELETE/PROTECTION command, 2-2B to 2-25

EXIT command, 2-26

EXTRACT command, 2-27 to 2-31

general command format, 2-1

HELP command, 2-32

LIST command, 2-33 to 2-39

MEMO command, 2-40 to 2-41

PURGE command, 2-42 to 2-43

RENAME command, 2-44 to 2-48

RENAME/SUBDICTIONARY command, 2-49 to 2-52

RESTORE command, 2-53 to 2-57

SET ABORT command, 2-58

SET DEFAULT command, 2-59

SET PROTECTION command, 2-60 to 2-65

SET PROTECTION/EDIT command, 2-66 to 2-80

SHOW DEFAULT command, 2-81

SHOW PROTECTION command, 2-82

SHOW VERSION command, 2-84

## Down arrow key

with VT100 ACL editor, 2-70

with VT52 ACL editor, 2-75

## E

### ENTER key

with VT100 ACL editor, 2-73

with VT52 ACL editor, 2-78

### ENTRY key

with VT100 ACL editor, 2-71

with VT52 ACL editor, 2-76

### EXIT command

with CDDV, 3-5

with DMU, 2-26

EXTRACT command, 2-27 to 2-31

## F

### /FAST qualifier

with FIX, 3-7

### FIELD key

with VT100 ACL editor, 2-71

with VT52 ACL editor, 2-76

## FIND key

with VT100 ACL editor, 2-71

with VT52 ACL editor, 2-76

## FIND NEXT key

with VT100 ACL editor, 2-72

with VT52 ACL editor, 2-76

## FIX command, 3-6 to 3-9

### /FULL qualifier

with LIST, 2-34

## G

### GOLD key

with VT100 ACL editor, 2-69

with VT52 ACL editor, 2-74 to 2-75

### GOLD Z

with VT100 ACL editor, 2-70

with VT52 ACL editor, 2-75

### /GRANT qualifier

with SET PROTECTION, 2-61 to 2-62

## H

### HELP command

with CDDV, 3-10

with DMU, 2-32

### HELP key

with VT100 ACL editor, 2-70

with VT100/VT200 ACL editor, 2-70

with VT52 ACL editor, 2-75

### Hierarchy, directory, 1-2f

### History lists, 1-3

### /HISTORY qualifier

with BACKUP, 2-3

with COPY, 2-8

with CREATE, 2-14

## I

### /ITEM qualifier

options, 2-35t

with LIST, 2-35

## K

### /KEEP qualifier

with DELETE/HISTORY, 2-21

with PURGE, 2-42

## L

- Left arrow key
  - with VT100 ACL editor, 2-70
  - with VT52 ACL editor, 2-75
- LINE FEED key
  - with VT100 ACL editor, 2-72
  - with VT52 ACL editor, 2-76
- LIST command, 2-33 to 2-39
- /LIST qualifier
  - with BACKUP, 2-3
- /LISTING qualifier
  - with FIX, 3-8
  - with LIST, 2-36
  - with VERIFY, 3-13
- /LOG qualifier
  - with BACKUP, 2-4
  - with COPY, 2-9
  - with DELETE, 2-18
  - with PURGE, 2-42
  - with RESTORE, 2-54
- Logical names
  - in subdictionary file specifications, 2-16
  - search lists, 2-16

## M

- MEMO command, 2-40 to 2-41
- MOVE key
  - with VT100 ACL editor, 2-71
  - with VT52 ACL editor, 2-76

## N

- /NOBANISH qualifier
  - with SET PROTECTION, 2-62
- /NODENY qualifier
  - with SET PROTECTION, 2-62
- /NOGRANT qualifier
  - with SET PROTECTION, 2-62

## O

- Objects
  - dictionary, 1-1
- OPEN ENTRY key
  - with VT100 ACL editor, 2-71
  - with VT52 ACL editor, 2-76
- /ORIGIN qualifier
  - with LIST, 2-36

- /OUTPUT qualifier
  - with FIX, 3-8
  - with LIST, 2-36
  - with VERIFY, 3-13

## P

- PASS\_THRU (P) privilege
  - inheritance error, 2-78
- PASSWORD key
  - with VT100 ACL editor, 2-71
  - with VT52 ACL editor, 2-76
- /PASSWORD qualifier
  - with SET PROTECTION, 2-63
- Passwords
  - entered on SAMPLE USER screen, 2-79f
- PASTE key
  - with VT100 ACL editor, 2-73
  - with VT52 ACL editor, 2-77
- Path names
  - displayed on SAMPLE USER screen, 2-79f
- /POSITION qualifier
  - with DELETE/PROTECTION, 2-24
  - with LIST, 2-36
  - with SET PROTECTION, 2-63
- Privileges, access, 1-4
- /PROTECTION qualifier
  - with BACKUP, 2-4
  - with COPY, 2-9
  - with LIST, 2-36
- PURGE command, 2-42 to 2-43

## R

- /RECORD qualifier
  - with EXTRACT, 2-28
- RENAME command, 2-44 to 2-48
- RENAME/SUBDICTIONARY command, 2-49 to 2-52
- RESET key
  - with VT100 ACL editor, 2-73
  - with VT52 ACL editor, 2-77
- RESTORE command, 2-53 to 2-57
- Right arrow key
  - with VT100 ACL editor, 2-70
  - with VT52 ACL editor, 2-75
- /RIGHTS qualifier
  - with SET PROTECTION, 2-63

## S

- Sample dictionary, 1-2f
  - listing, 1-3e
- SAMPLE USER screen, 2-78 to 2-80
  - path name display, 2-79f
- Search lists, 2-16
- SECTION key
  - with VT100 ACL editor, 2-71
  - with VT52 ACL editor, 2-76
- SELECT key
  - with VT100 ACL editor, 2-73
  - with VT52 ACL editor, 2-77
- SET ABORT command, 2-58
- SET DEFAULT command, 2-59
- SET PROTECTION command, 2-60 to 2-65
- SET PROTECTION/EDIT command, 2-66 to 2-80
  - sample entry display, 2-68
- SAMPLE USER screen, 2-78 to 2-80
- VT100 keypad functions, 2-69 to 2-73
- VT52 keypad functions, 2-74 to 2-78
- SHOW DEFAULT command, 2-81
- SHOW key
  - with VT100 ACL editor, 2-70
  - with VT52 ACL editor, 2-75
- SHOW PROTECTION command, 2-82
- SHOW USER key
  - with VT100 ACL editor, 2-73
  - with VT52 ACL editor, 2-78
- SHOW VERSION command
  - with CDDV, 3-11
  - with DMU, 2-84
- /SOURCE qualifier
  - with EXTRACT, 2-28
- /STAGE qualifier
  - with COPY, 2-9
  - with RESTORE, 2-55
- /SUBDICTIONARY qualifier
  - with CREATE, 2-14
  - with DELETE, 2-18
  - with RESTORE, 2-55

## T

- TAB key
  - with VT100 ACL editor, 2-71

## TAB key (Cont.)

- with VT52 ACL editor, 2-76
- /TEMPLATE qualifier
  - with EXTRACT, 2-28
- /TERMINAL qualifier
  - with SET PROTECTION, 2-63 to 2-64
- TOP key
  - with VT100 ACL editor, 2-70
  - with VT52 ACL editor, 2-75
- /TYPE qualifier
  - with DELETE, 2-18
  - with DELETE/HISTORY, 2-22
  - with DELETE/PROTECTION, 2-24
  - with EXTRACT, 2-29
  - with LIST, 2-36
- Types, 1-3

## U

- /UIC qualifier
  - with SET PROTECTION, 2-64
- UNDELETE CHARACTER key
  - with VT100 ACL editor, 2-72
  - with VT52 ACL editor, 2-77
- UNDELETE ENTRY key
  - with VT100 ACL editor, 2-72
  - with VT52 ACL editor, 2-77
- UNDELETE FIELD key
  - with VT100 ACL editor, 2-72
  - with VT52 ACL editor, 2-77
- Up arrow key
  - with VT100 ACL editor, 2-70
  - with VT52 ACL editor, 2-75
- /USERNAME qualifier
  - with SET PROTECTION, 2-64

## V

- VERIFY command, 3-12 to 3-13
- /VERSION qualifier
  - with COPY, 2-10
  - with RENAME, 2-45
  - with RESTORE, 2-55
- VT100 keypad
  - with SET PROTECTION/EDIT, 2-69 to 2-73
- VT52 keypad
  - with SET PROTECTION/EDIT, 2-74 to 2-78



# How to Order Additional Documentation

---

## Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

## Electronic Orders

To place an order at the Electronic Store, dial 800-DEC-DEMO (800-332-3366) using a 1200- or 2400-baud modem. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

## Telephone and Direct Mail Orders

<b>Your Location</b>	<b>Call</b>	<b>Contact</b>
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local DIGITAL subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	-----	Local DIGITAL subsidiary or approved distributor
Internal <sup>1</sup>	-----	SDC Order Processing - WMO/E15 <i>or</i> Software Distribution Center Digital Equipment Corporation Westminster, Massachusetts 01473

---

<sup>1</sup>For internal orders, you must submit an Internal Software Order Form (EN-01740-07).



# Reader's Comments

VAX Common Data Dictionary  
Utilities Reference Manual  
AA-L629E-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completeness (enough information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clarity (easy to understand)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization (structure of subject matter)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Figures (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Examples (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index (ability to find topic)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page layout (easy to find information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

I would like to see more/less \_\_\_\_\_

What I like best about this manual is \_\_\_\_\_

What I like least about this manual is \_\_\_\_\_

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

I am using **Version** \_\_\_\_\_ of the software this manual describes.

Name/Title \_\_\_\_\_ Dept. \_\_\_\_\_

Company \_\_\_\_\_ Date \_\_\_\_\_

Mailing Address \_\_\_\_\_

Phone \_\_\_\_\_

Do Not Tear - Fold Here and Tape

**digital**™



No Postage  
Necessary  
if Mailed  
in the  
United States



**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION  
Corporate User Publications—Spit Brook  
ZK01-3/J35  
110 SPIT BROOK ROAD  
NASHUA, NH 03062-9987



Do Not Tear - Fold Here

Cut Along Dotted Line

# Reader's Survey

VAX Common Data Dictionary  
Utilities Reference Manual  
AA-L629E-TE

1. How useful are the following methods for finding information in this manual?

	Most	Very	Moderately	Not Very	Not at All
Table of contents	<input type="checkbox"/>				
Divider pages (if applicable)	<input type="checkbox"/>				
Index (circle: book or master)	<input type="checkbox"/>				
Other (specify) _____	<input type="checkbox"/>				

2. What feature do you most want to see improved in this manual? Why?

\_\_\_\_\_

3. How helpful are these sources when you use the software this manual describes?

	Most	Very	Moderately	Not Very	Not at All
Handbook or user's guide	<input type="checkbox"/>				
Introduction or overview	<input type="checkbox"/>				
Reference manual	<input type="checkbox"/>				
Quick reference guide	<input type="checkbox"/>				
Online help	<input type="checkbox"/>				
Online tutorial (if available)	<input type="checkbox"/>				
Other: colleague, telephone support services (specify) _____	<input type="checkbox"/>				

4. What business tasks are you using the software described by this manual to solve (for example: billing, funds transfer, report writing)?

\_\_\_\_\_

5. Please estimate, if you can, how long the following VAX Information Architecture products have been used at your site:

VAX ACMS _____	VAX CDD/Plus _____	VAX DATATRIEVE _____
VAX Data Distributor _____	VAX DBMS _____	VAX RALLY _____
VAX Rdb/VMS _____	VAX SQL _____	VAX TEAMDATA _____
VAX TDMS _____	VIDA with IDMS/R _____	

6. This release of VAX Information Architecture documentation uses a 7x9 format for quick reference guides. Do you prefer such books in a 7x9 or a 4x8 pocket guide format? \_\_\_\_\_

Thank you for your assistance.

May we contact you at work for further information?  Yes  No

Name \_\_\_\_\_ Dept. \_\_\_\_\_

Company \_\_\_\_\_ Date \_\_\_\_\_

Mailing Address \_\_\_\_\_

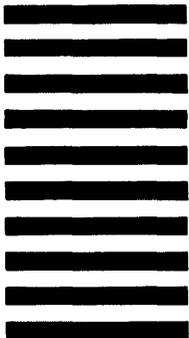
\_\_\_\_\_ Phone \_\_\_\_\_

Do Not Tear - Fold Here and Tape

**digital**™



No Postage  
Necessary  
if Mailed  
in the  
United States



**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION  
Corporate User Publications—Spit Brook  
ZK02-2/N53  
110 SPIT BROOK ROAD  
NASHUA, NH 03062-9987



- Do Not Tear - Fold Here