



Data Facility Data Set Services

SC26-4389-1

Reference

Version 2 Release 5.0

Second Edition (December 1989)

This is a major revision of SC26-4389-0, which is now obsolete. Changes from the previous edition are marked with a vertical bar in the left margin. Significant changes are summarized under "Summary of Changes."

This edition applies to Version 2, Release 5, Modification Level 0 of Data Facility Data Set Services, Program Number 5665-327, and to any subsequent releases until otherwise indicated in new editions or technical newsletters.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's program may be used. Any functionally equivalent program may be used instead.

Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality. If you request publications from the address given below, your order will be delayed because publications are not stocked there.

A Reader's Comment Form is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, General Products Division, Department 61C, Tucson, Arizona, U.S.A. 85744. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1984, 1985, 1986, 1987, 1988, 1989. All Rights Reserved.

Note to U.S. Government Users—Documentation related to restricted rights—Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corporation.



Reference

Version 2 Release 5.0

Technical Newsletter

Base Publication Number: SC26-4389-01

File Number: S370-32

Date: 01 Sept 1990

Previous Newsletters: (None)

This Technical Newsletter provides replacement pages and additional information for the subject manual. The replacement pages and additional information remain in effect for subsequent editions unless specifically altered. Pages to be inserted or replaced are:

ii.1 to vi

ix to xii

17, 18

79 to 80.2

219 to 226

235 to 244

If you are inserting pages from different Newsletters and identical numbers are involved, always use the page with the latest date (shown in the change-page notice at the top of the page). The page with the latest date contains the most complete information.

A change to the text is indicated by a vertical line to the left of the change. Absence of a vertical line on a page bearing an "Updated" notice means only that previously existing text has been moved or rearranged.

Summary of Amendments

This Technical Newsletter contains revisions to text from the subject manual.

Note: Please file this cover letter at the back of the manual to provide a record of changes.

Comments on this Technical Newsletter can be addressed to:

IBM Corporation
Department 61C
Tucson, AZ 85744



Special Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights or other legally protectible rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

This book is intended to help the customer perform various storage management tasks. It contains commands to use the Data Facility Data Set Services (DFDSS) program. This book documents no programming interface for use by customers in writing programs that request or receive the services of DFDSS, except as noted below.

This book also documents *general-use programming interfaces and associated guidance information*, *product-sensitive programming interfaces associated guidance information*, and *product implementation information*.

General-use programming interfaces allow the customer to write programs that request or receive the services of DFDSS.

General-use programming interface and associated guidance information is explicitly identified where it occurs, either by an introductory statement to a chapter or section that is entirely *general-use programming interface and associated guidance information*, or by the following marking:

General-Use Programming Interface

Description of *general-use programming interface and associated guidance information*...

End of General-Use Programming Interface

Product-sensitive programming interfaces are provided to allow the customer installation to perform tasks such as tailoring, monitoring, modification, tuning, or diagnosis of this IBM product. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM product. Product-sensitive interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Product-sensitive programming interface and associated guidance information is explicitly identified where it occurs, either by an introductory statement to a chapter or

section that is entirely *product-sensitive programming interface and associated guidance information*, or by the following marking:

Product-Sensitive Programming Interface

Description of *product-sensitive programming interface and associated guidance information*...

End of Product-Sensitive Programming Interface

Product implementation information is detailed design and implementation information that is provided to assist customers in performing tasks such as tailoring, monitoring, modifying, tuning, and diagnosing which by their nature require the use of information that depends on the detailed design and implementation of DFDSS. *Product implementation information* identified in this book documents no programming interfaces for use by customers in writing programs that request or receive the services DFDSS.

Warning: Do not use *product implementation information* for programming interface purposes.

Product implementation information is explicitly identified where it occurs, either by an introductory statement to a chapter or section that is entirely *product implementation information*, or by the following marking:

Product Implementation Information

Description of *product implementation information*...

End of Product Implementation Information

Note: Examples provided in this book contain sets of program statements offered to users as an illustration of how to obtain product services and are not to be used as programming interfaces.

Trademarks

The following terms are trademarks of IBM and are used in this publication:

DB2

DFSMS

ESA/370

MVS/DFP

MVS/ESA

MVS/SP

MVS/XA

System/370

Preface

This manual describes the commands to use Data Facility Data Set Services (DFDSS) and is intended for the storage administrator and the system programmer. Its purpose is to explain the command syntax to perform various storage management tasks.

Major Divisions of This Book

This book is organized as follows:

Chapter 1, Specifying DFDSS Commands describes syntax requirements and their conventions and summarizes the JCL you might need to use DFDSS.

Chapter 2, Choosing the Data Sets You Want Processed—Filtering describes the filtering process, filtering by data set name or by data set characteristics, and the standard catalog search order.

Chapter 3, Syntax—Function Commands describes the DFDSS commands, their keywords, and examples of how they are used.

Chapter 4, Syntax—Auxiliary Commands describes the commands that allow you to control DFDSS processing in the SYSIN data set.

Chapter 5, Stand-alone DFDSS Restore Program describes how to use the stand-alone DFDSS restore program and provides examples of job streams to process stand-alone DFDSS restore.

Chapter 6, Customizing DFDSS and Its Interfaces explains how you can customize DFDSS by coding exit routines and tailoring Interactive Storage Management Facility (ISMF) panels.

Appendix A, Compatibility Considerations for Prior Releases provides information regarding compatibility of prior releases with DFDSS Version 2 Release 5.

Appendix B, Serialization and Authorization Checking discusses how serialization is accomplished, the purpose of serialization, and how DFDSS checks user authorization.

Appendix C, User Interaction Module Exit Options discusses the various parts of the Application Interface.

Appendix D, Examples of the Application Program with the User Interaction Module (UIM) provides examples and explanations of the UIM exit functions.

Appendix E, Data Area Descriptions describes two data area descriptions.

Appendix F, Macro Library as Part of Programming Interfaces for Customers identifies product-sensitive and general-use programming interfaces.

Required Product Knowledge

To use this manual you need some knowledge of Data Facility Product (DFP), Data Facility Hierarchical Storage Manager (DFHSM), Resource Access Control Facility (RACF), and job control language (JCL).

DFDSS Machine Requirements

DFDSS Version 2 Release 5 is designed to operate on any IBM processor that is supported by MVS/System Product Version 1, Version 2, or Version 3.

DFDSS Books

- *Data Facility Data Set Services: General Information*, GC26-4123, introduces you to DFDSS and helps in evaluating its use.
- *Data Facility Data Set Services: User's Guide*, SC26-4388, gives guidance on how to use DFDSS.
- *Data Facility Data Set Services: Messages*, SC26-4390, lists DFDSS messages.
- *Data Facility Data Set Services: Diagnosis Guide*, LY27-9538, tells how to diagnose errors you may encounter while using DFDSS.
- *Data Facility Data Set Services/Interactive Storage Management Facility: Installation Planning Guide*, SC26-4129, tells what to consider when planning for and installing DFDSS/ISMF.

Referenced Books

Within the text, the following books are referenced:

- *MVS/ESA Application Development Guide*, GC28-1821
- *MVS/DFP Version 3 Release 2: Customization*, SC26-4560
- *MVS/DFP Version 3 Release 2: Utilities*, SC26-4559
- *MVS/DFP Version 3 Release 2: Access Method Services for the Integrated Catalog Facility*, SC26-4562
- *MVS/DFP Version 3 Release 2: Managing VSAM Data Sets*, SC26-4568
- *MVS/ESA JCL User's Guide*, GC28-1830
- *MVS/ESA JCL Reference*, GC28-1829
- *Resource Access Control Facility (RACF) General Information Manual*, GC28-0722
- *Resource Access Control Facility (RACF) Security Administrator's Guide*, SC28-1340.

3: A Tracks RESTORE Operation Using Generic Device Names	141
Requirements	142
Customizing DFDSS and Its Interfaces	143
Installation Exit Routines	143
Accession Authorization Exit Routine (ADRUPSWD)	143
Accession Enqueue Exit Routine	149
Accession Options Exit Routine	152
Accession Reblock Exit Routine	158
Accession ISMF	160
Libraries That Can Be Customized	160
Options to Customizing ISMF	161
4. Compatibility Considerations for Prior Releases	162
4.1 Serialization and Authorization Checking	165
Security—Serialization	165
Accession Serialization	165
Accession Serialization	166
Accession Option	169
Example of RESERVE-ENQUEUE Processing	171
Accession Authorization Checking	171
Accession Authorization Checking	173
Accession Authorization Checking	173
Accession Scheduled for Moving Data Sets	176
Accession Logging during Authorization Checking	177
Accession Flow of Security Processing	177
4.2 User Interaction Module Exit Options	179
Accession Link Structure	179
Accession Options	182
Accession Programming Information	182
Accession Interface Descriptions	187
Accession Startup (Eioption 0)	187
Accession SYSIN Record (Eioption 1)	187
Accession SYSPRINT Record (Eioption 2)	187
Accession Physical Tape Record (Eioption 3)	188
Accession Logical Tape Record (Eioption 4)	188
Accession Logical Tape Record (Eioption 5)	188
Accession Physical Tape Record (Eioption 6)	188
Accession Disk Track (Eioption 7)	189
Accession Disk Track (Eioption 8)	189
Accession Utility SYSPRINT (Eioption 9)	189
Accession WT (Eioption 10)	189
Accession WTO Message (Eioption 11)	189
Accession WTOR Message (Eioption 12)	190
Accession Accession Log ADRUFO Record (Eioption 13)	190
Accession Accession Ending (Eioption 14)	190
Accession Accession Log WTOR Response (Eioption 15)	190
Accession Accession EOVS Tape Volume Security and Verification Exit (Eioption 16)	190
Accession Accession EOVS Nonspecific Tape Volume Mount (Eioption 17)	191
Accession Accession Logical VSAM Record During Restore (Eioption 18)	191
Accession Accession Tape I/O Error (Eioption 19)	191
Accession Accession Notification (Eioption 20)	191
Accession Accession Tape Verification (Eioption 21)	192
Accession Accession Tape Verification Exit (Eioption 22)	193

Data Set Processed Notification Exit (Eioption 23)	194
Avoiding Lockout	196
Application Interface Summary	196
ADREID0 Data Area	197
An Example of Invoking DFDSS by Using an Application Program	201
How to Determine DFDSS Version, Release, and Modification Level	201
Appendix D. Examples of the Application Program with the User Interaction	
Module (UIM)	203
Appendix E. Data Area Descriptions	220
ADRBMB Data Area	220
ADRTAPB Data Area	221
Appendix F. Macro Library as Part of Programming Interfaces for Customers	224.2
Glossary of Terms and Abbreviations	226
Index	235

Figures

1.	Example of a Syntax Diagram for the COPYDump Command	4
2.	BY Keywords	14
3.	Defining Profiles When RACF 1.5 or Later Is Installed	43
4.	Results of Copying Non-VSAM Data Sets	56
5.	Printed Output Resulting from a Successful DEFrag Run	63
6.	Output Resulting from a PRINT Command	91
7.	Physical Data Set Restore Actions on SMS-Managed Data Sets	118
8.	RESTORE Actions on Non-VSAM Data Sets	119
9.	DFDSS Target Class Selection	120
10.	Return Codes From the Installation Authorization Exit Routine	147
11.	ADRUPB Mapping Macro	148
12.	ADRUNQB Mapping Macro	151
13.	Task Alterations Made with Installation Options Exit	152
14.	ADRUF0 Mapping Macro	155
15.	ADRRBLKB Mapping Macro	159
16.	System DFDSS RESTORE Compatibility among Releases	163
17.	Stand-alone DFDSS RESTORE Compatibility Among Releases	163
18.	Data Set Enqueue Options for Non-VSAM Data Sets Specified on DFDSS Commands	167
19.	Read/Write Access Serialization Scheme	169
20.	Resource Serialization	170
21.	Access Authority Requirement—RACF-Protected Data Sets and DASDVOL	175
22.	Data Mover Selection Matrix for Data Set COPY	176
23.	DFDSS Application Interface Structure	181
24.	Invoking DFDSS with Its Application Interface	182
25.	DFDSS Exit Interface Structure	183
26.	DFDSS Exit Identification Block	183
27.	ADREID0 Mapping Macro	197
28.	Constants	199
29.	Expansion of Macro ADRMCLVL	202
30.	The Application Program Process	203
31.	Application Interface Program Example	204
32.	User Interaction Module Example	206
33.	Output Resulting from Use of the UIM Exits	218
34.	ADRBMB Mapping Macro	220
35.	ADRTAPB Mapping Macro	221
36.	Product-Sensitive Programming Interfaces	224.2
37.	General-Use Programming Interfaces	224.2

Summary of Changes

Version 2 Release 5, December 1989

DFDSS Version 2 Release 5 provides additional storage management facilities and enhancements to help you:

- Take advantage of a new type of data set.

MVS/DFP Version 3 Release 2 now provides a specially designed data set, called a partitioned data set extended (PDSE). DFDSS supports the PDSE, which offers significant advantages over a partitioned data set (PDS). DFDSS also provides new keywords to convert a PDS to a PDSE and a PDSE to a PDS.

This support requires MVS/DFP Version 3 Release 2 or a later release.
- Use new application interface exit points.

New options for application interface exits further enhance the interaction between DFDSS and its invoking program at the data set level. DFDSS passes control to the user interaction module (UIM) when a data set is selected and when the processing of that data set is completed. At these exit points, the UIM can perform a number of tasks. For example, you can customize the UIM to end processing of the data set or to bypass options such as serialization and security verification.
- Process data sets with a high-level qualifier of SYS1. By using the PROCESS(SYS1) keyword, you can:
 - Release unused space in partitioned or sequential data sets with a high-level qualifier of SYS1.
 - Move data sets with a high-level qualifier of SYS1.
 - Specify the DELETE keyword with the DUMP command to dump and delete data sets with a high-level qualifier of SYS1.
 - Replace preallocated data sets that have a high-level qualifier of SYS1.
- Process data sets that have undefined DSORG. By using the PROCESS(UNDEFINEDSORG) keyword, you can copy or restore undefined DSORG data sets to an unlike device that has a larger track capacity.
- Specify both the OPTIMIZE and COMPRESS keywords for dumps.

You no longer must choose between performance (by specifying OPTIMIZE) or saving space (by specifying COMPRESS). You can back up data by using both keywords.
- Reduce CPU time by improving the data compression algorithm during DUMP.
- Use stand-alone RESTORE to restore compressed dump data.
- Reduce I O to the VVDS during logical data set DUMP/RESTORE, data set COPY and CONVERTV.
- Declare a larger block size for a SYSPRINT data set or the output data set from the PRINT command.

DFDSS now allows a block size greater than 141.

Chapter 1. Specifying DFDSS Commands

This chapter is divided into the following sections:

- “Command Syntax” describes syntax requirements.
- “How Many Subkeywords Are Allowed?” on page 2 lists possible combinations of subkeywords that you can specify in the input (command) stream.
- “Specifying Subkeywords in a Command Data Set” on page 3 explains how to use keywords in a command data set.
- “How to Read Syntax Diagrams” on page 3 explains the syntax conventions used in this book.
- “The JCL You Need” on page 5 summarizes the job control language (JCL) that you might need to use DFDSS.

Command Syntax

You can write DFDSS commands in free form in card columns 2 through 72 (inclusive). Any character in column 1 and past column 72 is ignored thus causing unpredictable results when the task is processed. Syntax requirements are:

- A *command* must appear first, followed by its keywords.
A command is separated from its keywords by one or more blanks, a comment, or both. For example:
`DUMP FULL INDD(DASD1) OUTDD(TAPE1)`
- A *comment* is a string of characters preceded by `/*` and followed by `*/`. A comment must be started and finished on the same line. For example:
`/* THIS IS A COMMENT */`
- A *separator* consists of a comma (`,`), or one or more blanks, or a comment.
- *Keywords* are separated from one another by one or more separators.
- *Subkeywords* are separated from one another by one or more separators. Subkeywords follow their associated keywords and are separated from them by a pair of enclosing parentheses. One or more blanks can, optionally, precede and follow each parenthesis in the pair. For example:

```
REBLOCK( DATASET1 )
```

or

```
REBLOCK(DATASET1)
```

If two or more subkeywords are permissible for a single keyword, these are separated from one another by at least one blank. Subkeywords can also be separated from one another by commas, where each comma can, optionally, be preceded and followed by one or more blanks. For example:

```
REBLOCK(DATASET1 , DATASET2)
```

or

```
REBLOCK(DATASET1,DATASET2)
```

or

```
REBLOCK(DATASET1 DATASET2)
```

- *Continuation* of a command is specified by a hyphen (-) as the right-most non-blank character, preceded by one or more blanks. For example:

```
COPY DATASET (INCLUDE(DATASET1)) ALLDATA(*) -  
          CATALOG REBLOCK(DATASET1)
```

The absence of such a hyphen indicates the end of the command. If a keyword or subkeyword cannot fit on the remainder of a line, it can be started on that line, followed immediately by a plus sign (+) in column 72, and continued on the next line.

- *The end* of a command can also be specified by a semicolon. Everything to the right of the semicolon is ignored.

How Many Subkeywords Are Allowed?

The following is not a comprehensive list of keywords with subkeywords, but rather a list of keywords that can have a maximum of 255 subkeywords specified in the inline command stream.

```
ALLDATA(subkeyword)
BY(subkeyword)
BY(LIST(subkeyword))
BYPASSACS(subkeyword)
CONVERT(PDS(subkeyword))
CONVERT(PDSE(subkeyword))
DDNAME(subkeyword)
DYNAM(subkeyword)
EXCLUDE(subkeyword)
EXCLUDE(LIST(subkeyword))
INCAT(subkeyword)
INCLUDE(subkeyword)
INDDNAME(subkeyword)
INDYNAM(subkeyword)
LOGICALVOLUME(subkeyword)
LOGINDDNAME(subkeyword)
LOGINDYNAM(subkeyword)
NOPACKING(dsn)
OUTDDNAME(subkeyword)
OUTDYNAM(subkeyword)
OUTTRACKS(subkeyword)
PASSWORD(subkeyword)
PERCENTUTILIZED(subkeyword)
REBLOCK(subkeyword)
REBLOCKADDRESS(subkeyword)
RENAME(subkeyword)
RENAMEUNCONDITIONAL(subkeyword)
TRACKOVERFLOW(subkeyword)
TRACKS(subkeyword)
```

Specifying Subkeywords in a Command Data Set

As an alternative to specifying the data set names in the inline (command) stream, you can specify the ddname of a sequential or data set member of a partitioned data set that contains the list of data set names. This allows you to specify more than 255 data set names. The keywords that allow this ddname specification are:

```

DATASET(FILTERDD(ddn))
EXCLUDE(DDNAME(ddn))
FILTERDD(ddn)
PASSWORD(ddn)

```

How to Read Syntax Diagrams

There is one basic rule for reading the syntax diagrams: Follow only one line at a time from the beginning to the end, and code everything you encounter on that line.

The following rules apply to the conventions used in the syntax diagrams for all the DFDSS commands:

- Read the syntax diagrams from left to right and from top to bottom.
- Each syntax diagram begins with a double arrowhead (➤) and ends with opposing arrows (↔).
- An arrow (→) at the end of a line indicates that the syntax continues on the next line. A continuation line begins with an arrow (←).
- Commands and keywords can contain uppercase and lowercase letters. The uppercase portion is the minimum needed to code the command properly; the lowercase portion is optional. For example, **COPYDump** can be coded in any of the following ways: **COPYD**, **COPYDU**, **COPYDUM**, or **COPYDUMP**.

Note: All commands *must* be entered in uppercase. Lowercase is not recognized.

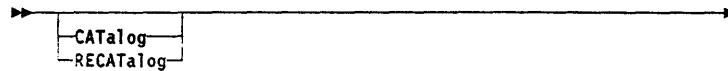
- Some commands and keywords have alternative abbreviations; these appear as part of the stack for that command or keyword. For example, the alternative abbreviation for **COPYDump** is **CPYD**.



- Words in all lowercase letters represent information *you* supply. For example, *volser* or *ddn*.
- You must provide all items enclosed in parentheses, (). You must include the parentheses.
- Where you can choose from two or more keywords, the choices are stacked one above the other. If one choice within the stack lies on the main path, the keyword is required, and you must choose one or none. In the following example you must choose either **BLK**, **TRK**, or **SOURCE**.



- If a stack is placed below the main path, the keyword is optional, and you can choose one or none. In the following example **CATaLog** and **RECATaLog** are optional keywords. You can choose either one or the other, but not both.



- The repeat symbol (\downarrow) shows that you can specify a keyword (or keywords) and variables more than once. The repeat symbol appears above the keywords and variables that can be repeated. When a comma appears in the repeat symbol, you must separate repeated keywords with a comma.

The following example illustrates all the rules described above. The numbers in the diagram are keyed to the explanations that follow.

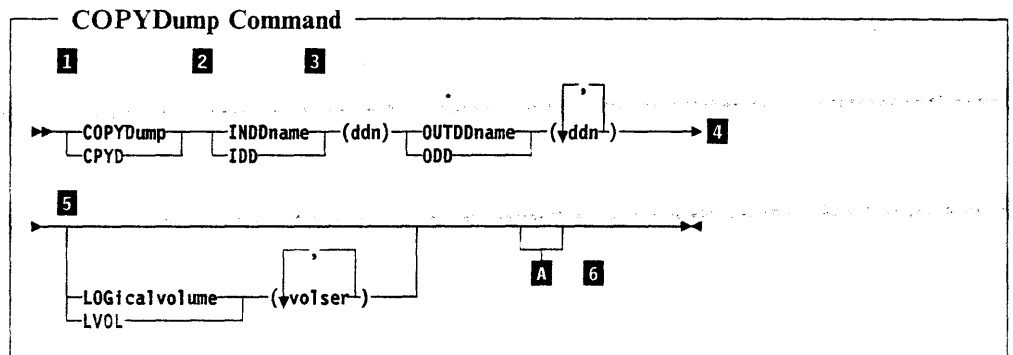


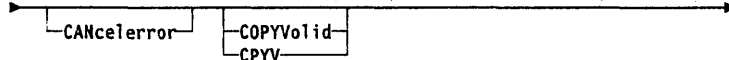
Figure 1. Example of a Syntax Diagram for the COPYDump Command

- 1 Either the command **COPYD** (or any other acceptable abbreviation of **COPYDump**) or the alternate abbreviation **CPYD** must be specified.
- 2 Either the keyword **INDD** (or any acceptable abbreviation of **INDDname**) or the alternate abbreviation **IDD** must be specified, followed by one *ddn*. You must provide the *ddn* enclosed in parentheses.
- 3 Either the keyword **OUTDD** (or any acceptable abbreviation of **OUTDDname**) or the alternate abbreviation **ODD** must be specified, followed by up to 255 *ddns*. Each list of *ddns* must be enclosed in parentheses and each *ddn* must be separated by a comma.
- 4 When there is a single arrow head at the end of a line, the syntax continues on the next line.
- 5 The **LOG** and **LVOL** keywords are optional, because they fall below the mainpath. You can specify as many volume serial numbers (*volser*) as needed on either keyword with each *volser* separated by a comma and enclosed in parentheses.
- 6 The substitution block **A** is used to simplify the syntax diagram. It indicates that a block of the syntax diagram is located outside of the main diagram. You insert the keywords for that block where the symbol appears. You then return to

the main diagram to continue with the command. For example, with the **RESTORE** command there are optional keywords for both **FULL** and **TRACKS**



A Optional Keywords used with FULL



The following example shows how the **COPYDump** command might be used (the JCL is included):

```

//JOB2    JOB  accounting information,REGION=nnnnK
//STEP1   EXEC PGM=ADRDUSSU
//SYSPRINT DD  SYSOUT=A
//TAPE2   DD  UNIT=3480,VOL=SER=TAPE20,
//         LABEL=(1,SL),DISP=(OLD,CATALOG),DSN=USER.BACKUP.REL3A
//OUTT2   DD  UNIT=3480,VOL=SER=TAPE21,
//         LABEL=(1,SL),DISP=(NEW,CATALOG),DSN=USER.BACKUP.REL3A.A
//SYSIN   DD  *
COPYDUMP -
  INDD(TAPE2)          /* DUMP TAPE TO BE COPIED    */ -
  OUTDD(OUTT2)         /* NEW DUMP TAPE           */ -
  LVOL(338001)        /* SER NO OF VOL TO BE COPIED */
/*
  
```

The JCL You Need

The examples in this manual are shown with the necessary JCL. You may need the following JCL to use DFDSS:

Statement	Usage
JOB (required)	Initiates your job.
EXEC (required)	Specifies the program name (PGM = ADRDUSSU) or, if the job control statements reside in a procedure library, the procedure name. See "How to Control DFDSS via PARM Information in the EXEC Statement" on page 6 for additional information that can be entered in the PARM parameter of the EXEC statement.
SYSPRINT DD (required)	Defines a sequential message data set. The data set can be written to a system output device, a tape volume, or a direct-access device. If the DCB keyword LRECL is specified on the DD statement, it must be in the ranges of 84 to 137 inclusive. If BLKSIZE is specified, it must be at least four greater than the LRECL. If an LRECL less than 84 is chosen, the return code is 8 and an error message is issued. If the specified LRECL is greater than 137, then LRECL and BLKSIZE are set to 137 and 141, respectively.
SYSIN DD (required)	Defines a command data set consisting of cards or card-image records that contains your DFDSS commands. It usually resides in the input stream, however, it can be defined as a blocked or unblocked sequential data set or

as a member of a partitioned data set. Records must be fixed format, LRECL = 80.

input DD (optional)

Defines the input (also called the source). The ddname, *input*, is supplied by you and is referred to by your DFDSS commands. This DD statement is not required for some operations. Do not specify BUFNO.

output DD (optional)

Defines the output (also called the target). The ddname, *output*, is supplied by you and is referred to by your DFDSS commands. This DD statement is not required for some DFDSS operations. Do not specify BUFNO. Code a volume count when a new data set will reside on six or more volumes. (Note that DISP = MOD is not supported for an output DD statement.)

filter DD (optional)

Defines a data set consisting of cards or card-image records that contains the filtering criteria (INCLUDE, EXCLUDE, and BY) to be used in a data set command. The ddname, *filter*, is supplied by you and is referred to by your DFDSS commands.

Note: You cannot concatenate DD statements to the *filter* DD statement.

password DD (optional)

Defines a data set consisting of card-image records that contains data set names and their passwords. The ddname, *password*, is supplied by you and is referred to by your DFDSS commands.

For more information on coding JCL, see *MVS/ESA JCL Reference*.

How to Control DFDSS via PARM Information in the EXEC Statement

The EXEC statement for DFDSS can contain PARM information that is used by the program. You can use the following keyword parameters:

TYPRUN = SCAN

Only control card syntax checking is done. No tasks are processed.

TYPRUN = NORUN

For COPY, DUMP, RESTORE, COMPRESS, CONVERTV, and RELEASE operations, only data set filtering is to be done without actually processing data sets. Printed output for the run indicates the data sets selected. For a DEFrag operation, the initial volume statistics are to be printed without actually relocating any extents. For a CONVERTV operation, a full report is produced, but no volumes or data sets are actually converted.

PAGENO = nnnn

The nnnn is a 1- to 4-digit number indicating the starting page number to be used for the SYSPRINT data set. The default is 1.

LINECNT = nnnn

The nnnn is a 1- to 4-digit number indicating the number of lines to be printed per page for the SYSPRINT data set. The default is 60. Specify LINECNT = 9999 to prevent a page ejection.

SIZE = nnnnK

The nnnn is a 1- to 4-digit number indicating the number of K bytes (1K equals 1024 bytes) of main storage to be used by DFDSS. This amount must be less than or equal to that specified by the REGION keyword. The default value is the region size.

ABEND = nnn

The **nnn** is a 3-digit decimal message number (ADRnnnx). If this is specified, and this message is to be issued, DFDSS performs a user 0001 ABEND dump after issuing the message, and the task stops. To get a dump, include a DD statement for SYSABEND or SYSUDUMP. This keyword is provided for debugging purposes only.

AMSGCNT = nnnn

The abend message occurrence count that tells DFDSS to end abnormally on the nth occurrence of the message specified in ABEND = nnn. The **nnnn** for AMSGCNT can be a number between 1 and 9999. The default is 1 (first occurrence).

RACFLOG = YES

Resource Access Control Facility (RACF) logging is done for all function tasks. This parameter can be overridden or set by the user installation options exit. See "RACF Logging during Authorization Checking" on page 177 for a description of DFDSS RACF logging.

TRACE = YES

When used during a DEFrag operation, this results in printing messages that indicate the relocated extents. This is a debugging tool.

UTILMSG = YES

Messages from auxiliary programs, such as IEBCOPY and ICKDSF, are copied into the DFDSS SYSPRINT output.

XABUFF = ABOVE16|BELOW16

The I O buffers used by DFDSS for COPY, COPYDUMP, DEFrag, DUMP, PRINT, and RESTORE operations are to be above or below 16 megabytes virtual storage. The default, when DFDSS runs under MVS/XA, is ABOVE16. This parameter is ignored when DFDSS runs under MVS/370.

WORKUNIT = workunit

You can supply an esoteric DASD unit name (for example, SYSDA), generic DASD unit name (for example, 3380), or a specific DASD address. DFDSS passes this unit to dynamic allocation when temporary data sets are allocated.

WORKVOL = volser

You can supply a volser on which DFDSS should allocate temporary data set. DFDSS passes the volser to dynamic allocation.

Notes:

1. WORKUNIT, WORKVOL, or both parameters can be specified when invoking DFDSS.
2. When DFDSS invokes the IDCAMS utility to copy an ICF user catalog, the export data set is allocated as a permanent data set. The permanent data set cannot be placed on the volume specified by WORKUNIT and/or WORKVOL.
3. An esoteric unit name that requests virtual I/O can be used in the WORKUNIT parameter, but when DFDSS invokes the IEHMOVE utility during data set copy, the default dynamic allocation unit name is used (SYSALLDA).

Examples of PARM information are:

```
// EXEC PGM=ADRSSU,  
//   PARM='PAGENO=8,LINECNT=57,SIZE=500K,UTILMSG=YES'
```

```
// EXEC PGM=ADRSSU,PARM='TYPRUN=SCAN,XABUFF=ABOVE16'
```

For further information on the PARM parameter of an EXEC statement, see *MVS/ESA Application Development Guide*.

Chapter 2. Choosing the Data Sets You Want Processed—Filtering

This chapter is divided into the following sections:

- “How DFDSS Filters Data Sets” describes how DFDSS filters data sets by data set name and by data set characteristic.
- “Filtering by Data Set Names” on page 10 explains how to specify fully- and partially-qualified data set names for filtering. This section includes rules for spelling out qualifiers and examples of qualified data set names.
- “Filtering by Data Set Characteristics” on page 12 lists the data set characteristics on which filtering can be done.
- “Standard Catalog Search Order” on page 16 identifies the standard order in which DFDSS searches for a data set name.

How DFDSS Filters Data Sets

DFDSS filters data sets by:

1. Applying inclusion and/or exclusion criteria to fully- or partially-qualified data set names.
 - a. If you specify inclusion (INCLUDE) criteria, DFDSS *tentatively* selects all data sets that fit any of the criteria.

Note: If inclusion (INCLUDE) criteria are not specified, then EXCLUDE or BY must be. In this case, DFDSS tentatively selects all data sets (equivalent to *INCLUDE(**)*). For the DEFrag command, *INCLUDE(**)* is always implied.
 - b. If you specify exclusion (EXCLUDE) criteria, DFDSS surveys the data sets selected in Step 1a and rejects the data sets that fit any of the criteria.
2. Applying data set characteristics (BY) criteria to the data sets tentatively selected in Step 1. DFDSS selects only those data sets that satisfy *all* BY criteria.

DFDSS lets you find out which data sets are selected by the filtering process without actually performing the requested operation(s). You can do this by specifying the TYPRUN=NORUN option on the JCL EXEC PARM field.

Considerations for VSAM Data Sets:

- INCLUDE and/or EXCLUDE filtering is performed on the cluster names of the data sets.
- If the BY criterion DSORG, MULTI, CATLG, or FSIZE is specified, filtering is done at the cluster level. If you specify other BY criteria, the data components of the selected clusters are further filtered on those BY criteria by using information from the VTOC. If a data component is selected, the index component for the cluster is also selected.
- DFDSS supports the BY criterion EXPDT as it exists in the VTOC only. For releases prior to DFP Version 3 Release 1, when a VSAM data set is defined by IDCAMS, the expiration date is in the catalog instead of the VTOC. Hence, the EXPDT is not used by DFDSS.

- A fully-qualified name is required in order to dump a VSAM data set that is in open-for-update status.
- For physical data set RESTORE: To prevent index components from being restored inadvertently, you must specify the fully-qualified name of the cluster.
- For data set PRINT: The fully-qualified name of the data set to be printed is required. Support is at the component, not the cluster, level.

Filtering by Data Set Names

A *fully-qualified* data set name is one in which all qualifiers are completely spelled out. A *partially-qualified* data set name is one in which asterisks (*) and percent signs (%) are used to represent qualifiers or parts of qualifiers.

Using an * in Partially-Qualified Data Set Names

The **single asterisk**, *, is used in place of exactly *one* qualifier. In addition, it can be used to indicate to DFDSS that only *part* of a qualifier has been specified; for example, just the first, last, middle, or first and last parts.

When used with other qualifiers, the **double asterisk**, **, indicates either the nonexistence of leading, trailing, or middle qualifiers, or the fact that they play no role in the selection process.

The following rules regarding asterisks in a qualifier must be observed:

- Two asterisks are the maximum permissible in a qualifier.
- If there are two asterisks in a qualifier, then the asterisks must be the first and last characters.

Consequently, the following are permissible qualifiers:

**
A

The following are **not** permissible qualifiers:

**A*
*A*B*
*A*B
A*B*C

Using a % in Partially-Qualified Data Set Names

The % acts as a place holder for a single character during data set name filtering. The rules for using % are:

- Each % corresponds to exactly one character.
- % can be specified more than once, consecutively or in any level of the qualifier.
- A % cannot match a null (' '), blank (' '), or period ('.').
- Use of a % in filtering does not change any of the other filtering specifications for data set names.

Consequently, specifying

I%M.A.%AT%%ET

will match the data set names

IAM.A.DATASET
IBM.A.BATTYET

but not

IAM.A.DATASE
IAM.AA.DATASET
IAM.A.ATASET
IM.A.DATASET

Examples of Fully- and Partially-Qualified Data Set Names

Examples of fully-qualified data set names:

USER2LD	The first and only qualifier is <i>USER2LD</i> .
SYS1.UTIL3.LOAD	The first qualifier is <i>SYS1</i> , the second, <i>UTIL3</i> , and the third, <i>LOAD</i> .
USER2.PROGRAM1.LIST	The first qualifier is <i>USER2</i> , the second, <i>PROGRAM1</i> , and the third, <i>LIST</i> .

Examples of partially-qualified data set names using ******:

** .LOAD	All data sets whose last, or only, qualifier is <i>LOAD</i> .
SYS1.**	All data sets whose first, or only, qualifier is <i>SYS1</i> .
USER2.**.LIST	All data sets whose first and last qualifiers are <i>USER2</i> and <i>LIST</i> , respectively, including <i>USER2.LIST</i> .

Additional examples of partially-qualified data set names:

*.LOAD	All data sets with two qualifiers whose last qualifier is <i>LOAD</i> .
SYS1.*	All data sets with two qualifiers whose first qualifier is <i>SYS1</i> .
SYS1.*.LOAD	All data sets with three qualifiers whose first and last qualifiers are <i>SYS1</i> and <i>LOAD</i> , respectively.
SYS1.UT*.LIST	All data sets with three qualifiers whose first and last qualifiers are <i>SYS1</i> and <i>LIST</i> , respectively, and whose second qualifier begins with <i>UT</i> .
**.*LIB	All data sets whose last, or only, qualifier ends with <i>LIB</i> .
**.*LIB*	All data sets whose last, or only, qualifier has <i>LIB</i> in it.
*.*PLI*.*	All data sets with three qualifiers whose second qualifier has <i>PLI</i> in it.
*.*P*M	All data sets with three qualifiers whose last qualifier begins with <i>P</i> and ends with <i>M</i> .
**	All data sets.
% .LIST	All data sets with one character in the first qualifier and <i>LIST</i> in the last qualifier.

Filtering by Data Set Characteristics

USER%.*	All data sets with two qualifiers whose first qualifier is <i>USER</i> followed by some other character.
%%*	All data sets whose single qualifier consists of two or more characters.

Note: Single quotation marks around either a fully- or partially-qualified data set name indicates the name is fully-qualified. Do not use single quotation marks to enclose a data set name that will appear in a DFDSS control statement.

Relative Generation Filtering

DFDSS allows filtering on relative generations of a generation data group (GDG) data set in the INCLUDE, EXCLUDE, and REBLOCK data set name lists. A GDG is specified as *dsn(n)* where *dsn* is a fully- or partially-qualified base name without the last qualifier (GggggVvv), and *n* is the relative generation number or * for all generations.

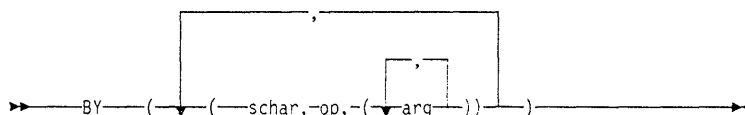
Examples of relative generation filtering are:

dsn(0)	For the current generation
dsn(-x)	For the xth prior generation
dsn(+x)	For the xth future generation
dsn(*)	For all generations.

For logical operations, you must use either the fully-qualified data set name, the partially-qualified base name with the last qualifier (GggggVvv), or the above relative generation filtering. Partial qualification of the last qualifier (GggggVvv) is not supported.

Filtering by Data Set Characteristics

After DFDSS has tentatively selected data sets by applying INCLUDE and EXCLUDE criteria, you can apply BY criteria to further restrict the data sets finally chosen. You can, for example, use BY to select data sets by creation date, storage class, and a wide variety of other criteria. The BY keyword takes this form:



Where **Represents**

schar The selection characteristics:

Keyword	Criteria
ALLOC	Allocation type (cylinder, track, block, absolute track, or movable).
CATLG	Whether the data set is currently cataloged or not (using the standard catalog search order).
CREDIT	Creation date (absolute or relative).

	DSCHA	Whether the data-set-changed flag is on or off.
	DSORG	Data set organization (SAM, PAM, PDS, PDSE, BDAM, EXCP, ISAM, or VSAM).
	EXPDT	Expiration date (absolute or relative) ¹
	EXTNT	Number of allocated or used extents for the entire data set on all the volumes on which it resides.
	FSIZE	Number of allocated or used tracks for the entire data set on all the volumes on which it resides (data set size).
	MULTI	Whether the data set is single- or multivolume.
	REFDT	Last-referenced date (absolute or relative) ² .
	DATACLAS	Data class for SMS
	MGMTCLAS	Management class for SMS
	STORCLAS	Storage class for SMS
op	The operator:	
	Operator	Meaning
	EQ or =	Equal to
	LE or <=	Less than or equal to
	LT or <	Less than
	GT or >	Greater than
	GE or >=	Greater than or equal to
	NE or -=	Not equal to.
arg	An argument that qualifies the selection characteristic (<i>schar</i>)	

Figure 2 on page 14 summarizes the permissible combinations of *schar*, *op*, and *arg*.

¹ Data sets without expiration dates explicitly assigned to them are considered to have an expiration date of zero. If you wish to exclude these data sets from expiration date processing, you must specifically exclude them in your filtering list, that is, BY EXPDT NE 00000.

² To filter on the last-referenced date for VSAM data sets, if Data Facility Hierarchical Storage Manager (DFHSM) is not installed, module IDATMSTP (in load module IDA0192A) must be modified to send a return code of 4 to its caller.

Filtering by Data Set Characteristics

Figure 2 (Page 1 of 2). BY Keywords

schar	op	arg	Notes
ALLOC	EQ NE	CYL (cylinder allocation) TRK (track allocation) BLK (block length allocation) ABSTR (absolute track allocation) MOV (movable data sets)	<p>If MOV is picked, the data sets to be processed cannot be allocated as any of the following:</p> <ul style="list-style-type: none"> • PSU (physical sequential unmovable) • POU (partitioned organization unmovable) • DAU (BDAM unmovable) • ABSTR (absolute tracks) • ISAM (indexed sequential access method). <p>For the COMPRESS command, if MOV is picked, only data sets allocated as POU cannot be compressed.</p>
CATLG	EQ	YES (or 1) NO (or 0)	<p>If YES is specified, only data sets that are currently cataloged are processed. If NO is specified, only uncataloged data sets are processed.</p> <p>Note: A data set is considered not cataloged if it is not cataloged in the standard order of search or if the catalog that the data set is cataloged in is not available.</p> <p>For the DUMP command, specifying YES is valid only for a physical dump.</p> <p>For the COPY command, specifying yes is valid only for a COPY with INDD or LOGINDD.</p> <p>For the RESTORE command, the target data set is tested to determine if it is cataloged or not.</p>
CREDIT EXPDT REFDT	LT GT EQ NE GE LE	yyddd[n]: Year and day, modified by a 1- to 4-digit positive or negative number, n. For example, 86100 means the 100th day of 1986. *[n]: Run date, modified by the 1- to 4-digit positive or negative number, n. For example, *,-5 means five days before this job is run.	<p>If you specify a yy of 00 through 50, DFDSS assumes you want years 2000 through 2050; if you specify a yy of 51 through 99, DFDSS assumes 1951 through 1999.</p> <p>CREDIT and EXPDT: For a multivolume data set, the date from its first volume's VTOC is used for checking.</p> <p>REFDT: For a multivolume data set, the latest date from all its VTOCs is used for checking.</p> <p>For the RELEASE and DEFRAG commands, the date in the VTOC of the volume being processed is used for filtering both single- and multivolume data sets.</p>
DSCHA	EQ NE	YES (or 1) NO (or 0)	<p>For a multivolume data set, the value used for checking is 1 if any of the indicators from all of its VTOCs is 1. Otherwise, the value used is 0.</p>
DSORG	EQ NE	SAM PAM PDS PDSE BDAM ISAM VSAM EXCP	<p>EXCP applies to data sets not allocated as any of the listed organizations and not accessed by using any of the listed access methods.</p> <p>For the COMPRESS command, because DFDSS assumes data set organization, you do not need to specify it.</p>

Figure 2 (Page 2 of 2). BY Keywords

schar	op	arg	Notes
EXTNT FSIZE	LT GT EQ NE GE LE	nnnn: where nnnn is a 1- to 5-digit number (in the range 0 to 99999)	This is the number of used or allocated extents (EXTNT) or used or allocated tracks (FSIZE) for an entire data set on all the volume(s) on which it resides. For the RESTORE command, the dump data set determines the number of used or allocated extents or tracks. For the DUMP and COPY commands, if ALLDATA or ALLEXCP is specified, FSIZE is equal to the allocated tracks and EXTNT is equal to the allocated extents.
MULTI	EQ	YES (or 1) NO (or 0)	If YES is specified, DFDSS processes only multivolume data sets. If NO is specified, DFDSS processes only single-volume data sets. For the DEFrag command, if a data set's volume sequence number is greater than one in the VTOC, DFDSS assumes it is multivolume. If this sequence number is 1, DFDSS assumes it is a single-volume data set. For the COMPRESS command, because DFDSS assumes the data set is single-volume, you do not need to specify MULTI.
DATACLAS MGMTCLAS STORCLAS	EQ NE	An appropriate SMS class name.	

Notes to Figure 2:

- When multiple arguments are specified for an NE operation, DFDSS selects only those data sets not matching any of the arguments.
- When multiple arguments are specified for an EQ operation, DFDSS selects those data sets matching any of the arguments.
- BY criteria do not apply for the CONVERTV or COPYDUMP commands.

Some Examples of the BY Keywords

If you code

```
BY((ALLOC EQ CYL) (CATLG EQ YES))
```

you will receive all cataloged data sets with cylinder allocation.

If you code

```
BY(FSIZE GE 100)
```

you will receive all data sets whose size is greater than or equal to 100 tracks.

If you code

```
BY(DSORG EQ (PAM,SAM))
```

DFDSS will select all PAM and SAM data sets.

Standard Catalog Search Order

For a complete description of the standard catalog search order, see the appropriate *MVS/DFP Version 3 Release 2: Access Method Services for the Integrated Catalog Facility*.

When catalogs are searched for a data set name, the standard order of the search is:

1. For DFP Version 3 Release 1, or subsequent releases, DFDSS first searches the catalogs specified with INCAT, if any.
2. DFDSS searches the user catalog(s) specified in the current job step (STEPCAT), or if none is specified for the job step, the user catalog(s) specified for the current job (JOB CAT). If more than one catalog is specified for the job step or job, the catalogs are searched in order of concatenation.

Note: If data sets with the same name are cataloged in more than one catalog within the standard search order, DFDSS selects only the first one encountered. Also, STEPCAT and JOBCAT can be used only in a non-SMS environment.
3. If the data set name is a qualified name and is the name of a user catalog, or the data set name is the same as the alias of a user catalog or the alias of a CVOL, DFDSS searches that user catalog or CVOL.
4. DFDSS searches the master catalog.

Chapter 3. Syntax—Function Commands

This chapter describes *function* commands. Function commands specify tasks, which are operations to be performed. The commands are:

COMPRESS
 CONVERTV
 COPY
 COPYDUMP
 DEFRAG
 DUMP
 PRINT
 RELEASE
 RESTORE.

The *Data Facility Data Set Services: User's Guide* contains usage notes, including DFDSS restrictions. Refer to these notes as needed.

This chapter begins with a task-oriented overview of each command, followed by detailed descriptions in alphabetical order. Each description contains:

- An overview of the function command
- Syntax for the command**
- Explanation of keywords and subkeywords for the command
- Data integrity considerations and format of the output data set
- Examples of command operations using keywords.

Product implementation information is contained in this chapter.

What DFDSS Commands Can Do

Backup and Recovery

With DFDSS, you can back up your data in case of hardware failure or user error. You can use DFDSS to back up data on DASD and to restore the backup if the original is lost, damaged, or changed inadvertently. In addition to providing for routine backup requirements, DFDSS can be used to back up application data for disaster recovery and vital records purposes.

You can use DFDSS to back up and recover data sets, entire volumes, or specific tracks. The DFDSS DUMP command is used to back up tracks, volumes, and data sets. The RESTORE command is used to recover them.

Data Movement

Moving data is an essential part of storage management. In order to replace storage devices, add storage capacity, and meet storage requirements, you must move data. The data movement support that DFDSS provides allows you to:

- Move data sets from old to new direct-access storage devices (DASD)
- Move data sets between SMS- and non-SMS-managed volumes
- Move data sets off a volume when hardware maintenance is required
- Move or copy data sets for other purposes.

You can move data sets from one DASD volume to another volume of like or unlike device type³. Or, you can copy a full volume or ranges of tracks, in which case the DASD must be of like device type.

The DFDSS COPY command is used to perform data set, volume, and track movement.

Conversion to and from Storage Management Subsystem (SMS)

SMS allows you to match the needs of users' data (like data set organization, size, and format) to the characteristics of storage devices without requiring users to know or understand your installation's hardware configuration. With SMS, users can store and retrieve data without being aware of space limitations, device characteristics, or volume serial numbers.

The DFDSS CONVERTV command is used to convert existing volumes to and from SMS management without data movement.

Space Management

With DFDSS, you can effectively manage space. DFDSS provides features to consolidate free space on volumes, compress partitioned data sets, and release unused space in data sets.

The DFDSS DEFrag command is used to consolidate the free space on a volume to help prevent out of space abends on new allocations. The DFDSS COMPRESS command is used to compress partitioned data sets by taking unused space and consolidating it at the end of the data set. The DFDSS RELEASE command is used to release unused space in sequential and partitioned data sets for use by other data sets.

Conversion from a PDS to a PDSE and from a PDSE to a PDS

The DFDSS data set COPY command can be used to move a PDS and convert it to a PDSE, or vice versa.

Printing for Diagnostic Purposes

You can print DASD data to SYSPRINT or to a sequential data set in print format. For data set printing, tracks are printed in logical sequence of the data set on the volume, not in the physical cylinder and head sequence.

The DFDSS PRINT command is used to print a single-volume non-VSAM data set, a single-volume VSAM data set component, all or part of the VTOC, or ranges of tracks.

³ Like devices have the same track capacity and number of tracks per cylinder (for example, 3380 Standard and 3380 Model D or E). Unlike DASD devices have different track capacities, a different number of tracks per cylinder (for example, 3350 and 3380), or both.

COMPRESS Command

The COMPRESS command compresses partitioned data sets on a specified volume. Compressing removes unused space between members in a partitioned data set. This recovered space is then available for reuse at the end of the data set. Depending on the filtering criteria you specify, you can compress either all or some of the partitioned data sets. This command is useful for compressing system partitioned data sets before you apply maintenance (thus avoiding certain space-related abends).

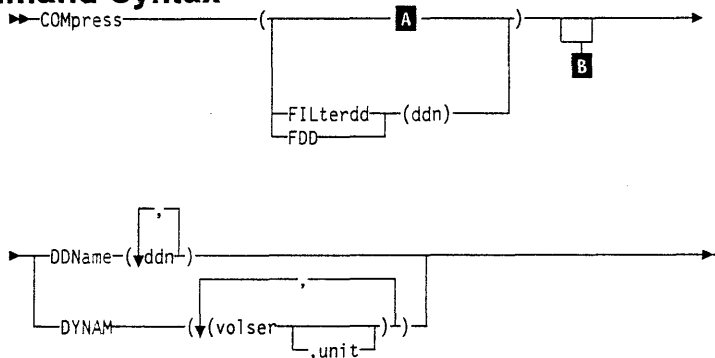
Note: You must not compress the data sets that contain DFDSS or IEBCOPY executable code.

The actual PDS compression is done on the existing volume using the IEBCOPY utility. To prevent loss of data if the system or IEBCOPY abnormally ends during processing, back up volumes or data sets that meet the filtering criteria before using the COMPRESS command.

COMPRESS cannot process partitioned data sets that:

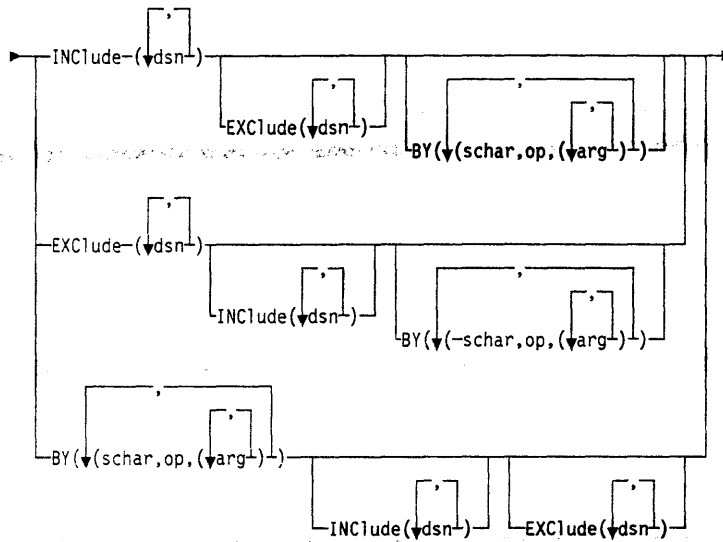
- Are unmovable
- Contain track overflow records
- Have no directory.

COMPRESS Command Syntax



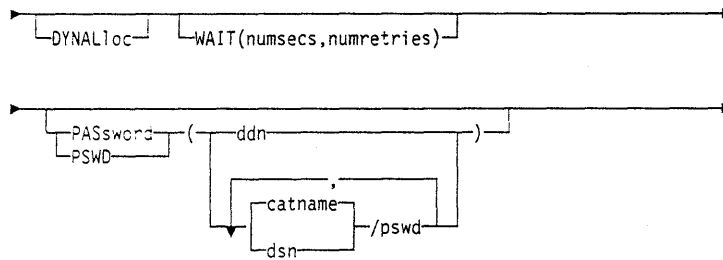
Additional Keywords Used with COMPRESS

A:



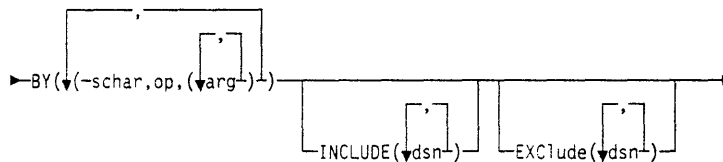
Optional Keywords

B:



Explanation of COMPRESS Command Keywords

BY



specifies that the data sets selected up to this point, by the processing of the INCLUDE and EXCLUDE keywords, are to be further filtered. To select the data set, all BY criteria must be met. See the separate discussions of INCLUDE and EXCLUDE for information on how these keywords are specified.

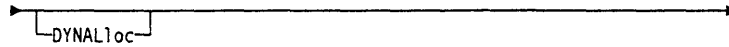
DDName



ddn

specifies the name of the DD statement that identifies a volume whose partitioned data sets, if selected, are to be compressed. To assure correct processing, each of the DD statements corresponding to a DDNAME (ddn) must identify only one volume serial number.

DYNALloc

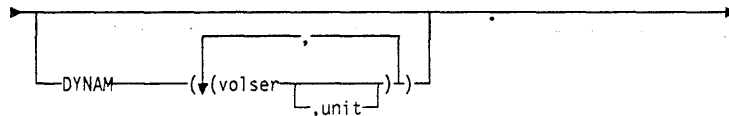


specifies dynamic allocation, instead of enqueue, to serialize the use of selected partitioned data sets. The data sets whose extents are to be relocated are serialized throughout COMPRESS. This allows cross-system serialization in a JES3/MVS environment.

Consider:

- The serialization is of value only when the dynamic allocation/JES3 interface is not disabled.
- SVC 99 is used to serialize data sets.
- Run time increases when DYNALLOC is used to serialize data sets (as opposed to enqueue) because overhead is involved in dynamic allocation and serialization across multiple processors.

DYNAM



specifies a dynamically allocated volume whose partitioned data sets, if selected, are to be compressed. If the volume is real (not virtual), it must be mounted and online. Consider using DYNAM instead of DD statements to allocate DASD volumes. This does not appreciably increase run time and permits easier coding of JCL and command input.

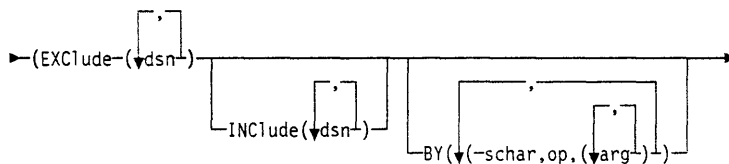
volser

specifies the volume serial number of a DASD volume to be processed.

unit

specifies the device type of a DASD volume to be processed. This keyword is optional.

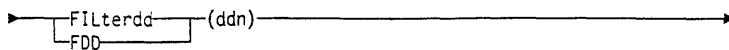
EXClude



dsn

specifies the name of a data set to be excluded from the data sets selected by INCLUDE. Either a fully- or a partially-qualified data set name can be used. See the separate discussions of INCLUDE and BY for information on how these keywords are specified.

FILterdd

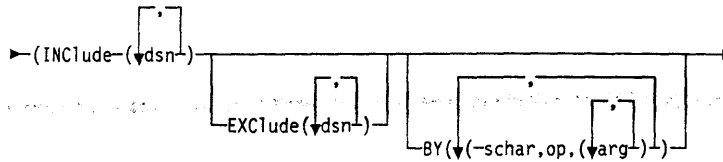


ddn

specifies the name of the DD statement that identifies the sequential data set or member of a partitioned data set that contains the filtering criteria to use. This is in the form of card-image records, in DFDSS command syntax, that contain the

INCLUDE, EXCLUDE, and BY keywords that complete the syntax of the COMPRESS command.

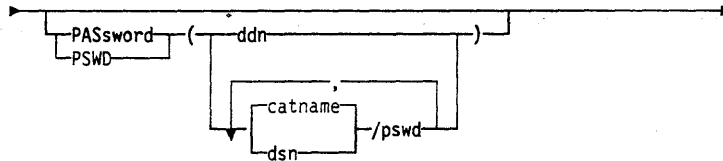
INClude



dsn

specifies the name of a data set eligible to be compressed. Either a fully- or a partially-qualified data set name can be used. See “Filtering by Data Set Names” on page 10. If INCLUDE is omitted (but EXCLUDE or BY is specified) or if INCLUDE(**) is specified, *all* partitioned data sets are eligible to be selected for compressing. See the separate discussions of EXCLUDE or BY for information on how these keywords are specified.

PASsword



specifies the passwords DFDSS is to use for selected password-protected data sets. (Password checking is bypassed for RACF-protected data sets.) This must be specified only if:

- You do not have the required RACF DASDVOL or RACF DATASET access.
- The installation authorization exit does not bypass the checks.
- You do not want to be prompted for the password.

ddn

specifies the name of the DD statement that identifies the sequential data set or member of a partitioned data set that contains data set names and their passwords. This data set must contain card-image records in DFDSS command syntax format.

dsn/pswd

dsn is a fully-qualified data set name. *pswd* is its password. If no password follows the slash (/), *dsn* is treated as though it were *ddn*.

Printing of actual data set passwords specified in your input command stream is suppressed in the SYSPRINT output.

Note: When a system utility (such as IEBCOPY) is being used to perform the COMPRESS, the user must supply the password for each password-protected data set selected.

WAIT



specifies to DFDSS the length of a wait in seconds and the number of retries to obtain control of a volume or data set.

numsecs

is a number from 1 to 255 that specifies the interval, in seconds, between retries.

numretries

is a 1- or 2-digit number that specifies the number of times an attempt to gain control of a volume or data set is to be retried.

The default for *numsecs,numretries* is WAIT(2,2), which specifies two retries at 2-second intervals. If you do not want to wait for a resource, specify 0 for either *numsecs* or *numretries*.

Example of a COMPRESS Operation

Compressing Selected Partitioned Data Sets

```
//JOB1      JOB  accounting information,REGION=nnnnK
//STEP1     EXEC PGM=ADRDUSSU
//SYSPRINT DD  SYSOUT=A
//SYSIN     DD  *
COMPRESS
  DYNAM(338000)      /* DYNAM ALLOC VOL 338000      */ -
  EXCLUDE(SYS1.***) /* EXCL 'SYS1....' DATA SETS  */ -
                    /* IF THEY MEET THIS CRITERION */ -
  BY((DSCHA EQ 0)) /* DATA SET WAS BACKED UP     */ /
/*
```

Compress partitioned data sets on volume 338000 if:

- They are not system data sets (EXCLUDE(SYS1.**)), and
- They have not been updated since the last time they were backed up (dumped). This ensures that the data set can be recovered if the system fails while COMPRESS is running (DSCHA EQ 0).

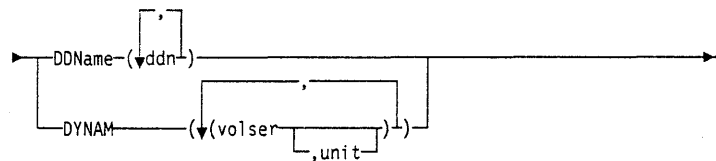
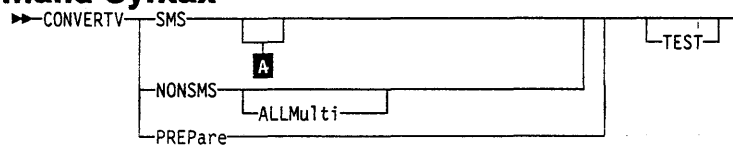
CONVERTV Command

The CONVERTV command is used to convert existing volumes to and from SMS management without data movement. The CONVERTV command performs three functions:

1. Lock volumes that are ready for conversion to prevent new data set allocations (**PREPARE** keyword).
2. Examine volumes identified by SMS to determine if they can be converted to SMS management (**TEST** keyword). No conversion is actually performed, but DFDSS identifies any data sets that cannot be converted to SMS management and why they cannot be converted.
3. Perform conversion of volumes into or out of SMS management. Any conditions that prevent conversion are identified.

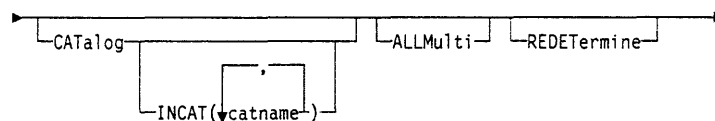
Note: DFP Version 3 Release 1 or subsequent releases and RACF Facility Class authorization are required for CONVERTV.

CONVERTV Command Syntax



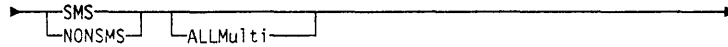
Optional Keywords with SMS

A :



Explanation of CONVERTV Command Keywords

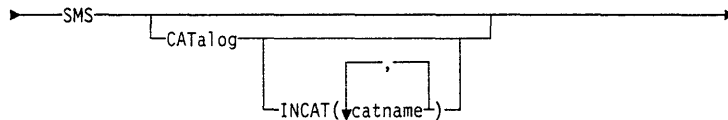
ALLMulti



specifies that cataloged, multi-volume data sets found on volumes identified in the **DDNAME** or **DYNAM** volume lists that have extents residing on volumes not in the list will be processed successfully. If **ALLMULTI** is not specified, such data sets are not eligible for conversion.

Note: For SMS processing, the first extent of a multivolume data set must reside on one of the volumes in the input volume list or that data set is not processed. This restriction does not apply to NONSMS processing.

CATalog



specifies that if a data set's catalog entry is not found in the standard order of search, the data set is to be cataloged during the conversion.

If the **CATALOG** keyword has not been specified, and a data set's catalog entry is not found in the standard order of search, the data set is not converted.

INCAT

specifies input catalog(s) that are not in the standard search order. This allows non-VSAM data sets cataloged outside the standard order of search to be processed.

catname

specifies a fully-qualified data set name.

If **CATALOG** is specified without **INCAT**, a single volume, non-VSAM data set cataloged outside the standard order of search might be cataloged in more than one place.

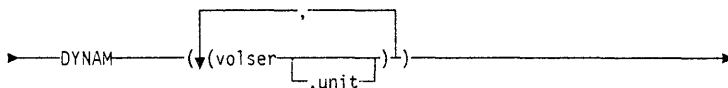
DDName



ddn

specifies the name of the DD statement that identifies a volume to be processed. This keyword is required when **DYNAM** is not specified.

DYNAM



specifies that a volume being processed must be dynamically allocated. If the volume is real (not virtual), it must be mounted and online. This keyword is required if **DDNAME** is not specified.

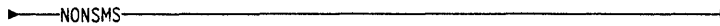
volser

specifies the volume serial number of a DASD volume to be processed.

unit

specifies the device type of a DASD volume to be processed. This keyword is optional.

NONSMS



specifies that a volume and all of the data sets on that volume are to be converted from SMS management.

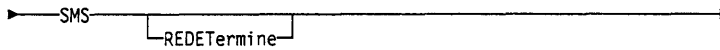
PREPare



specifies that a volume is to be prepared for SMS without conversion of data sets. This prevents the volume from changing prior to performing the full SMS conversion. After PREPARE is requested, the volume is placed in initial status and you cannot allocate new data sets. However, you can delete existing data sets.

NONSMS must be specified to return the volume to non-SMS management.

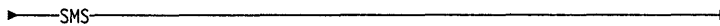
REDETermine



specifies that the SMS class information is to be reset for data sets previously converted to SMS management. REDETERMINE allows management class and storage class to be reset, but does not update the data class.

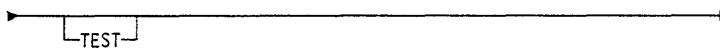
If REDETERMINE is used with the TEST keyword, a report is produced specifying all data sets eligible for conversion, including those already converted.

SMS



specifies that a volume and all of the data sets on that volume are to be converted to SMS management. SMS is the default when SMS, NONSMS, or PREPARE is not specified.

TEST



specifies that DFDSS is to verify that a volume and its data sets are eligible for conversion or for preparation. The TEST keyword functions just as if TYPRUN=NORUN had been specified on the JCL EXEC PARM field.

Note: It is also possible to use TEST to verify that the ACS algorithms would process correctly because the resulting report indicates the classes associated with the various data sets on the volume.

Examples of CONVERTV Operations

Example 1: Using CONVERTV to simulate conversion

```
//JOB1     JOB    accounting information,REGION=nnnnK
//STEP1    EXEC   PGM=ADDRSSU
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD     *
```

```
CONVERTV SMS
DYNAM((VOL001,3380),(VOL002,3380)(VOL003))-
TEST
/*
```

The preceding example uses the TEST keyword to simulate conversion. The TEST keyword produces a report that indicates whether the three volumes (VOL001, VOL002, and VOL003) can be converted to SMS management.

Example 2: Using CONVERTV to go to SMS

```
//JOB1     JOB    accounting information,REGION=nnnnK
//STEP1    EXEC   PGM=ADDRSSU
//SYSPRINT DD    SYSOUT=*
//DVOL1    DD     UNIT=SYSDA,VOL=SER=338001,DISP=OLD
//DVOL2    DD     UNIT=SYSDA,VOL=SER=338002,DISP=OLD
//SYSIN    DD     *
```

```
CONVERTV -
DDNAME(DVOL1,DVOL2) -
SMS -
INCAT(SYS1.ICFCAT.V338002) -
ALLMULTI -
CATALOG
/*
```

The non-SMS volume 338002 and the SMS volume (in INITIAL state) 338001 are converted to SMS. The volume 338001 was placed in the initial state by the storage administrator. Regardless of where the data sets reside, all multivolume data sets whose first extent is on volume 338001 or 338002 are processed. In addition, there are some data sets on volume 338002 cataloged in the user catalog SYS1.ICFCAT.V338002. These data sets are uncataloged from the user catalog and cataloged in the standard order of search. The INCAT keyword provides access to the user catalog because STEPCAT is not allowed.

Example 3: Using CONVERTV to go from SMS

```
//JOB1     JOB    accounting information,nnnnK
//STEP1    EXEC   PGM=ADDRSSU
//SYSPRINT DD    SYSOUT=*
//SYSIN    DD     *
```

```
CONVERTV -
DYNAM(338003) -
NONSMS
/*
```

This example removes a volume that is SMS-managed to non-SMS control.

COPY Command

With the COPY command, you can copy data from one DASD volume to another. You can copy data sets between DASD of like or unlike device type or a full volume or ranges of tracks between DASD of like device type.

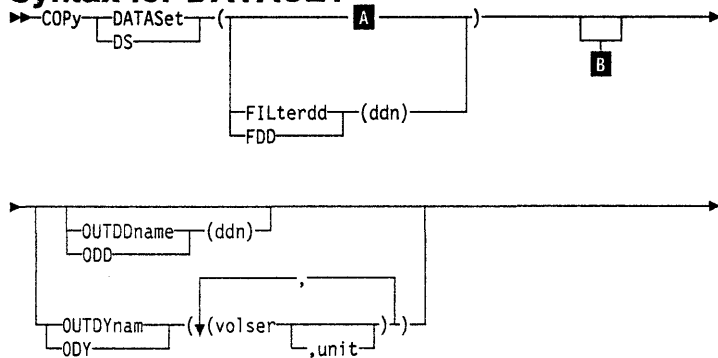
DFDSS offers two ways to process COPY commands:

- *Logical processing* is data-set oriented, which means it operates against data sets and volumes independently of physical device format.
- *Physical processing* can operate against data sets, volumes, and tracks, but is oriented toward moving data at the track-image level.

The processing method is determined by the keywords specified on the command.

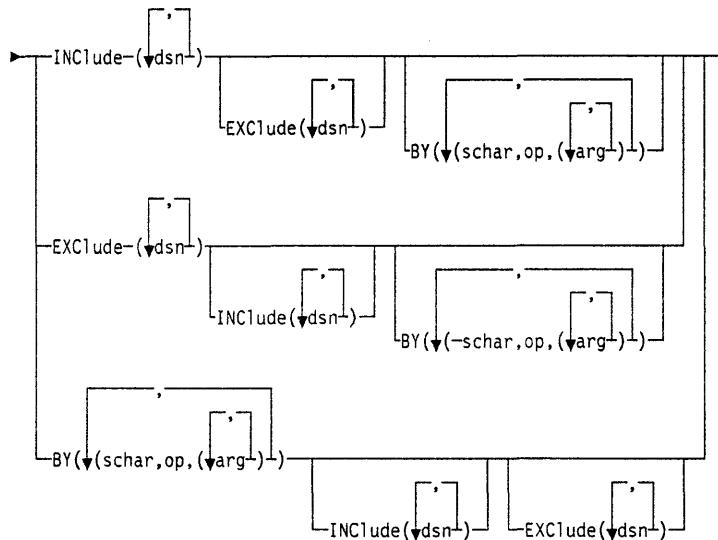
For information on how to use the COPY command, see *Data Facility Data Set Services: User's Guide*.

COPY Command Syntax for DATASET



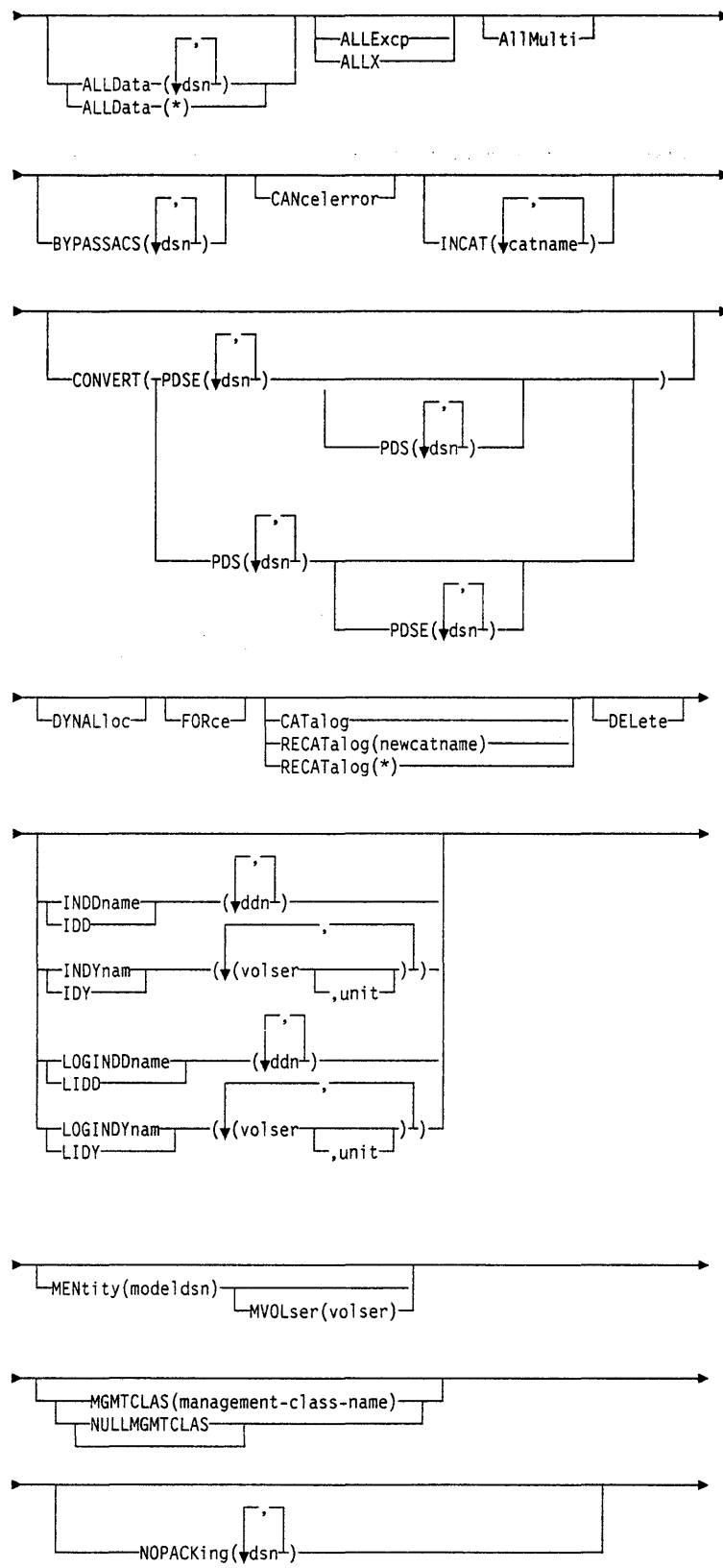
Additional Keywords Used with DATASET

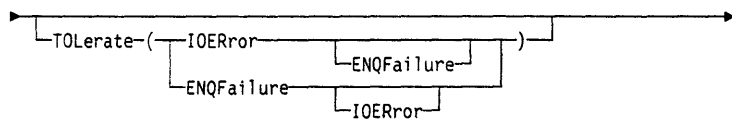
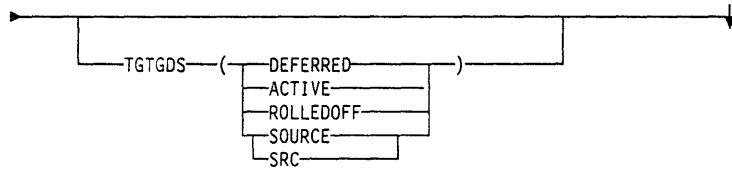
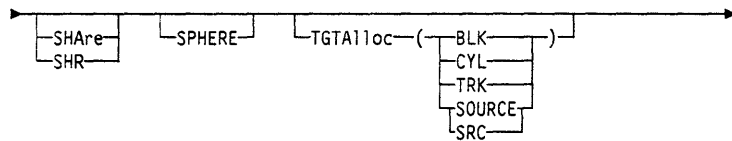
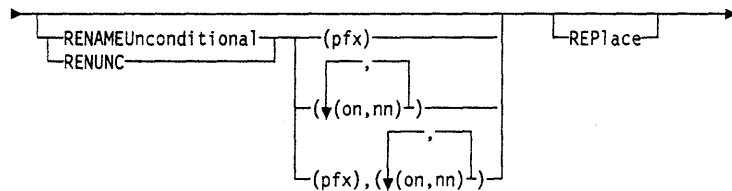
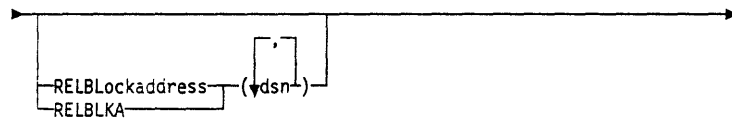
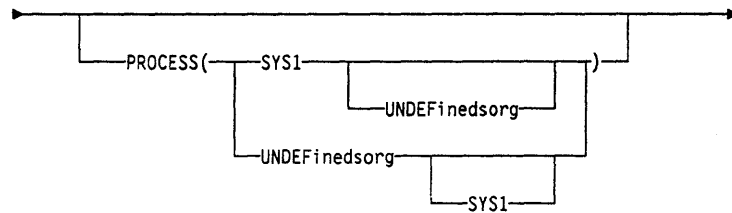
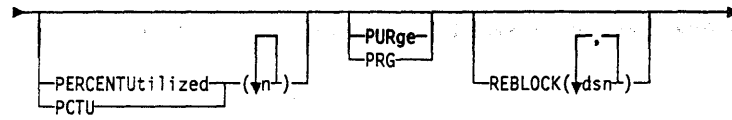
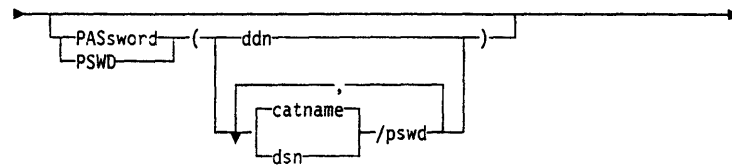
A:

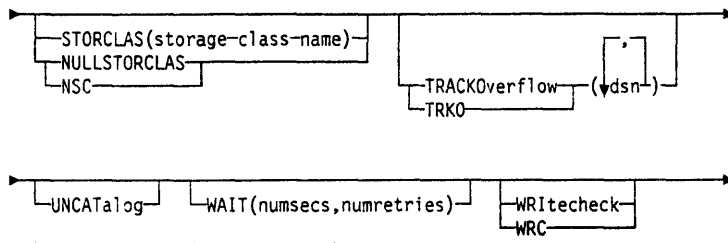


Optional Keywords Used with DATASet

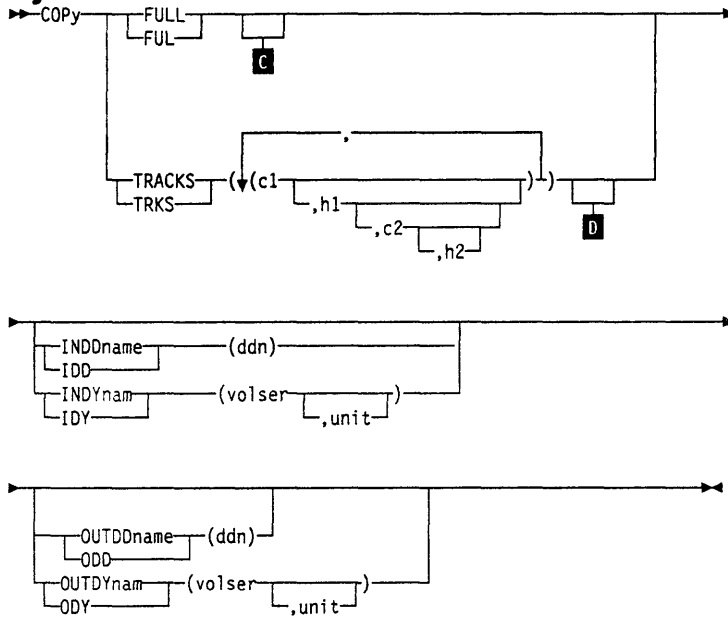
B:





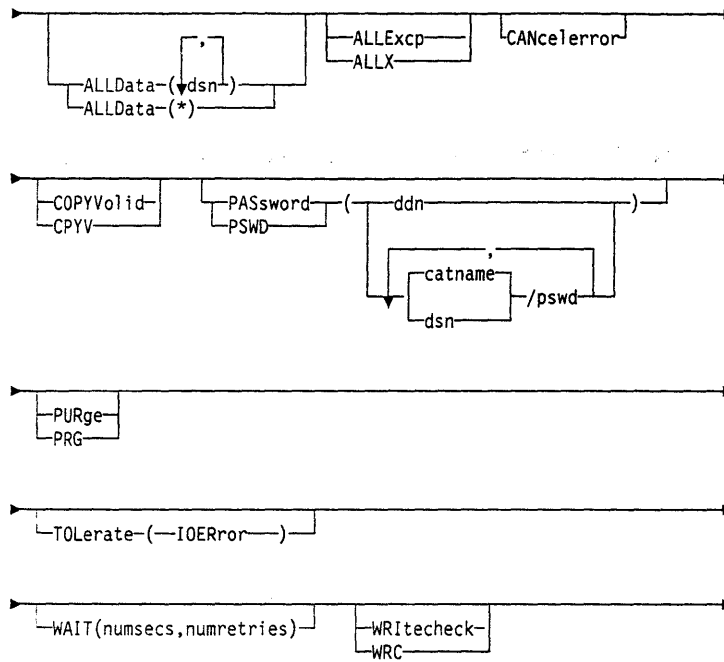


COPY Command Syntax for FULL or TRACKS



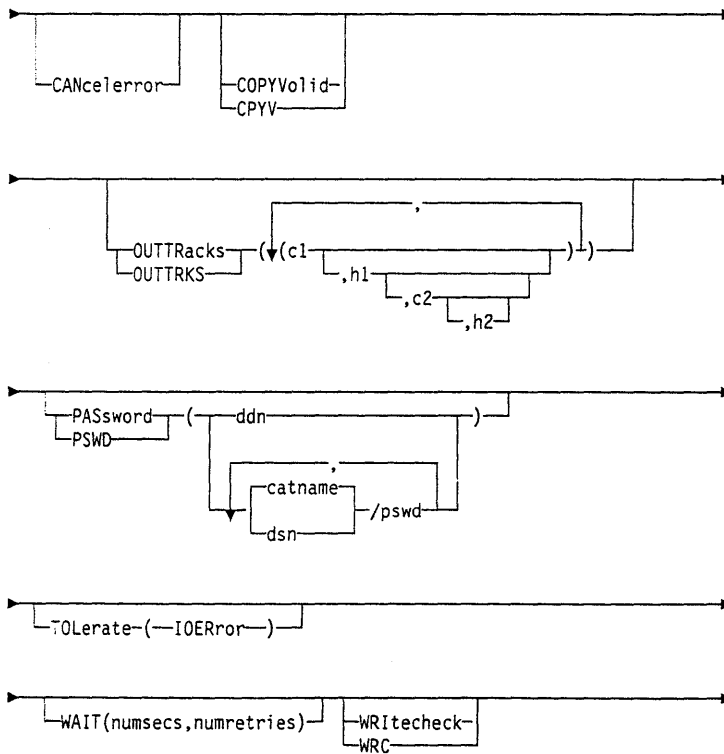
Optional Keywords Used with FULL

C :



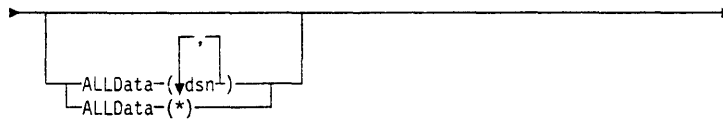
Optional Keywords Used with TRACKS

D :



Explanation of COPY Command Keywords

ALLData



dsn

specifies the name of a SAM (DSORG=PS or PSU) or PAM (DSORG=PO or POU) data set or a data set with a null DSORG field, for which:

- When the source data set and the target data set are on the same device type, all allocated space on the source is to be allocated and copied to the target for SAM data sets or for PDS load modules without REBLOCK specified. When DFDSS copies a PDS that is not a load module or when DFDSS copies a PDS load module with REBLOCK specified, all allocated space on the source is to be allocated on the target. However, only the used space is copied.
- When the source data set and the target data set are different device types, all allocated space on the source is to be allocated on the target but only used space is to be copied.

See Figure 4 on page 56 for results when you use ALLEXCP/ALLDATA.

Note: Any time a data set can have data beyond the limits described by the DSCB DS1LSTAR field, or contains a null DS1LSTAR field, or is a first or intermediate volume of a multivolume data set that contains a null DS1LSTAR field, you must specify ALLDATA(*dsn*) to copy all its data. JES2/JES3 data sets can have these characteristics, as can CICS journal and MSVCJRNL data sets.

* (*asterisk*)

specifies that, for *all* SAM and PAM data sets and data sets with a null DSORG field whose DS1LSTAR field is nonzero:

- When the source data set and the target data set are on the same device type, all allocated space on the source is to be allocated and copied to the target for SAM data sets or for PDS load modules without REBLOCK specified. When DFDSS copies a PDS that is not a load module or when DFDSS copies a PDS load module with REBLOCK specified, all allocated space on the source is to be allocated on the target. However, only the used space is copied.
- When the source data set and the target data set are on different device types, all allocated space on the source is to be allocated on the target, but only used space is to be copied.

ALLEXcp

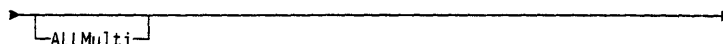


specifies that, for data sets whose DSCB DS1LSTAR field is zero (SAM or PAM data sets or data sets with a null DSORG field):

- When the source data set has an EOF as its first record, all allocated space on the source is to be allocated and an EOF is copied.
- When the source data set does not have an EOF as its first record, the data set is not processed.

See Figure 4 on page 56 for results when you use ALLEXCP/ALLDATA.

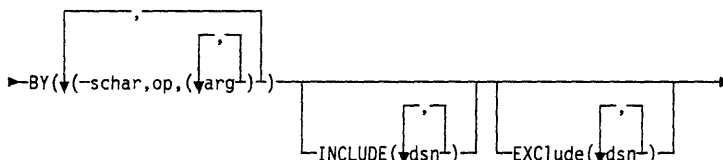
ALLMulti



specifies that cataloged, multivolume data sets that have volumes missing from the LOGINDDNAME or LOGINDYNAM volume list will be copied in their entirety. To qualify for ALLMULTI processing, a data set must have at least one primary volume specified in the LOGINDDNAME or LOGINDYNAM list, and it must be cataloged. Data from the missing volume(s) will be copied for applicable data sets.

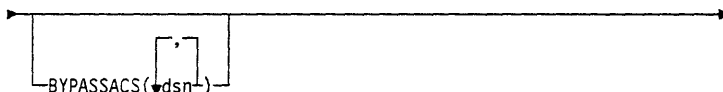
Note: ALLMULTI is valid only when LOGINDDNAME or LOGINDYNAM is specified.

BY



specifies that the data sets selected up to this point, by the processing of the INCLUDE and EXCLUDE keywords, are to be further filtered. To select the data set, *all* BY criteria must be met. See "Filtering by Data Set Characteristics" on page 12 for further information on BY filtering.

BYPASSACS



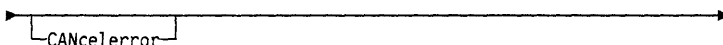
specifies that the Automatic Class Selection (ACS) routines are not invoked to determine the target data set class names. In order to bypass ACS filtering, the user must have RACF Facility Class authorization.

See "Assignment of Class Names by Using RESTORE and COPY" on page 120 for information on the assignment of class names by using COPY.

dsn

specifies a fully-qualified data set name.

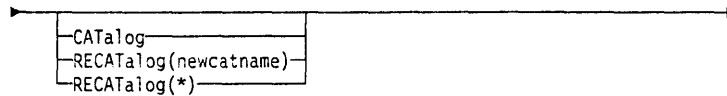
CANcelerror



specifies that the COPY is to be ended if a permanent read error, such as a data check, occurs. If this keyword is *not* specified and a permanent read error occurs, the track in error is not copied and the COPY continues. On COPY operations in which a utility performs the copy, DFDSS ignores this keyword. CANCELERROR has no effect on the following types of errors on a DASD volume:

- Equipment check
- Command reject
- Intervention required
- Busout parity.

CATalog



specifies that on a data set COPY, DFDSS is to catalog data sets that it allocates.

CATALOG

catalogs the target data set as determined by the standard catalog search order. This is the default for VSAM, multivolume data sets, and SMS-managed data sets.

RECATALOG(*)

catalogs the target data set in the same catalog that points to the source data set. If the source data set was not cataloged, the new data set is not cataloged either.

RECATALOG(*newcatname*)

catalogs the target data set in the *newcatname* catalog.

Notes:

1. Be careful when using RECATALOG(*newcatname*) because the target data set may be cataloged outside of the standard order of search.
2. CATALOG or RECATALOG fails if the target data set is already cataloged in the same catalog and DELETE, RENAMEU, or UNCATALOG is not specified. RECATALOG is ignored for SMS targets.
3. An alternate index (AIX) is always cataloged in the same catalog as its associated base cluster. If the base cluster is recataloged, the AIX is recataloged.
4. If you omit the CATALOG or RECATALOG keyword for a non-VSAM data set, the target data set will be uncataloged.

COPYVolid

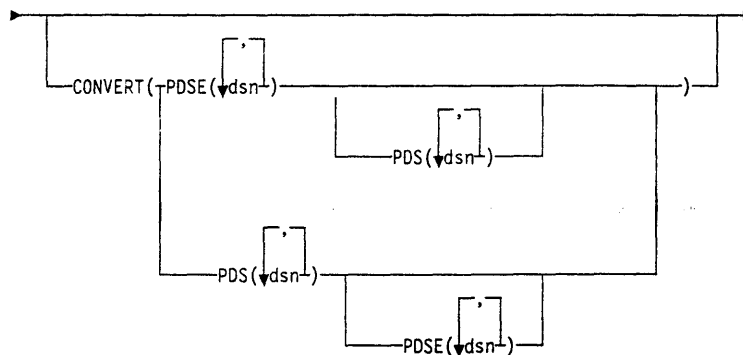


specifies that the volume serial number (VOLID) from the input DASD volume is to be copied to the output DASD volume. This applies to full copy operations and to tracks copy operations if track 0 is copied.

Note: COPYVOLID is required for a full-volume COPY of an SMS-managed input volume.

When the volume serial number is changed by using a COPYVOLID keyword or if the volume serial of the dumped volume and the volume to be restored to are different, profiles are not built for the RACF-protected data sets on the restored volume or for the RACF DASDVOL for the RACF-protected DASD volume.

When the volume serial number on a DASD volume is changed, the volume serial field in the UCB of the receiving volume is cleared and the operator is notified. The operating system then initiates a demount of the volume.

CONVERT**CONVERT(PDSE(dsn))**

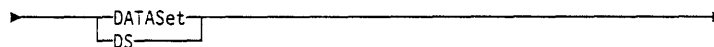
specifies that the PDS(s) listed in the *dsn* are to be converted to PDSE

CONVERT (PDS(dsn))

specifies that the PDSE listed in the *dsn* are to be converted to PDS(s).

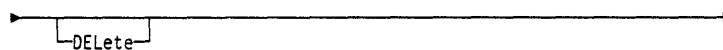
Notes:

1. The CONVERT keyword requires at least MVS/DFP Version 3 Release 2 and MVS/SP Version 3 Release 1.0E.
2. If the target data set is a PDSE, it must be SMS-managed; and it must be a data library (non-load modules).

DATASet

specifies a data set COPY operation using filtering. See Chapter 2, "Choosing the Data Sets You Want Processed—Filtering" on page 9 for an explanation of the filtering process used. Unless DATASET is specified with ALLDATA or ALLEXCP, only *used tracks* are copied for SAM and PAM data sets and for data sets with a null DSORG field (for example, JES2/JES3 data sets). If the free space map in the VTOC is invalid, all tracks are copied. The FILTERDD or INCLUDE keyword must be substituted when this choice is selected.

Note: VSAM catalogs are not supported for data set processing. If VSAM catalogs are processed, the results can be unpredictable.

DELeTe

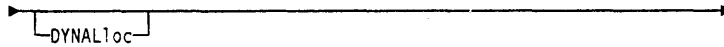
deletes VSAM and non-VSAM data sets from the source volume after a successful copy. This moves, in effect, a data set from one volume to another. The data sets are scratched and uncataloged.

Notes:

1. If you do not specify DELETE when copying cataloged data sets, the target data set must either be cataloged (with RECATALOG) in a different catalog or renamed (with RENAMEU).
2. *Unexpired* source data sets are deleted only if you also specify PURGE.
3. Even if PROCESS (SYS1) is specified, SYS1.VVDS and SYS1.VTOCIX data sets cannot be copied and deleted.
4. Do not specify SHARE if you specify DELETE.

5. If DFDSS encounters a damaged PDS during logical data set COPY, it displays messages indicating the nature and relative location of the problem. In order to maintain complete data integrity, DFDSS does not delete the source data set. The COPY of the data set fails, and the target is deleted. In order to copy and delete a damaged PDS, use the NOPACKING keyword.

DYNALloc

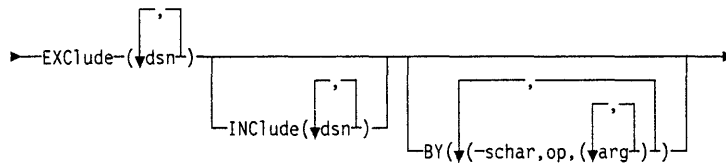


specifies dynamic allocation, instead of enqueue, to serialize the use of data sets. The data sets whose extents are to be relocated are serialized throughout COPY. This allows cross-system serialization in a JES3/MVS environment.

Consider:

- The serialization is of value only when the dynamic allocation/JES3 interface is not disabled.
- SVC 99 is used to serialize data sets.
- Run time increases when you use DYNALLOC to serialize data sets (as opposed to enqueue) because overhead is involved in dynamic allocation and serialization across multiple processors.

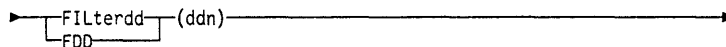
EXClude



dsn

specifies the name of a data set to be excluded from the data sets selected by INCLUDE. Either a fully- or a partially-qualified data set name can be used. See the separate discussions of INCLUDE and BY for information on how these keywords are specified.

FILterdd

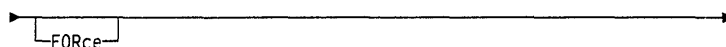


ddn

specifies the name of the DD statement that identifies the sequential data set or member of a partitioned data set that contains the filtering criteria to be used. This is in the form of card-image records, in DFDSS command syntax, that contain the INCLUDE, EXCLUDE, and BY keywords described below.

Note: You must use FILTERDD when you have more than 255 entries in INCLUDE, EXCLUDE, or BY list keywords.

FORce



specifies that DFDSS copy one or more unmovable data sets to a like or unlike device type. Unmovable data sets are those allocated as absolute track (ABSTR) or as unmovable (PSU, POU, DAU, or ISU). FORCE applies to ISAM data sets only when they are allocated with the unmovable attribute. The allocation attribute, unmovable or ABSTR, is carried over to the output volume.

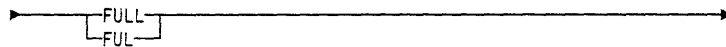
When copying to like devices, DFDSS copies the data sets to the same track locations on the target volume. In this case, FORCE is not required if all the following conditions are true:

- The operating system is MVS/XA DFP Version 2 Release 1 or a later release.
- The target volume uses an indexed VTOC.
- The space where the unmovable data set is to reside is available.

If any of these conditions is not true, DFDSS does not copy any unmovable data sets unless FORCE is specified. In this case, DFDSS places the unmovable data sets in any available location.

When copying to unlike devices, you must specify FORCE. The unmovable data sets are placed in any available location.

FULL



specifies that an entire DASD volume is to be copied. This is the default. Unallocated tracks are not copied. Unless specified by ALLDATA or ALLEXCP, only the used (rather than allocated) tracks are copied for SAM data sets, PAM data sets, and for data sets with unknown data set organization (for example, JES2/JES3 data sets with a null DSORG field). If the VTOC has errors, all tracks are copied. Used tracks consist of the tracks from the beginning of the data set to the last-used track (as indicated in the DS1LSTAR field of the Format 1 DSCB).

Notes:

1. You cannot specify SHARE or TOL(ENQF) for FULL operations.
2. VSAM catalogs are not supported for full volume processing. If VSAM catalogs are processed, the results can be unpredictable.

INCAT

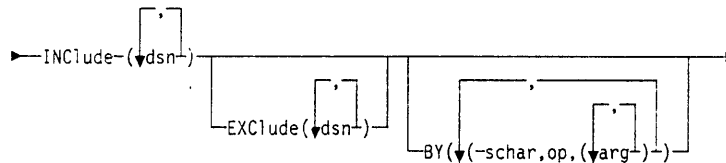


specifies one or more user catalogs to be searched before the standard order of search for locating data sets. STEPCAT and JOBCAT are not allowed for processing SMS-managed data sets. This keyword allows you to identify specific source catalogs. To specify INCAT, you must have the proper RACF Facility Class authorization. Refer to *Resource Access Control Facility (RACF) Security Administrator's Guide* for additional information.

catname

specifies a fully-qualified catalog name.

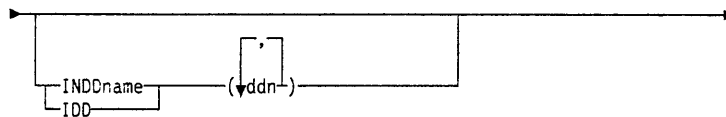
INCLude



dsn

specifies the name of a data set eligible to be copied. Either a fully- or a partially-qualified data set name can be used. See “Filtering by Data Set Names” on page 10. If INCLUDE is omitted (but EXCLUDE or BY is specified) or INCLUDE(**) is specified, *all* data sets are eligible to be selected for copying.

INDDname



ddn

specifies the name of the DD statement that identifies a volume to be copied. For a data set COPY, you can specify multiple names (that is, multiple volumes), separated by commas. For single-volume data sets, each of the DD statements corresponding to a DDNAME (*ddn*) must identify only one volume serial number

Note: If no input volume(s) is specified for a data set COPY, DFDSS selects from all data sets cataloged in the catalogs accessible through the standard search order. If either INDDNAME or INDYNAM is specified, DFDSS still uses the standard catalog search order, but it selects data sets only from the specified volume(s). For multivolume data sets, use LOGINDDNAME or LOGINDDYNAM with ALLMULTI.

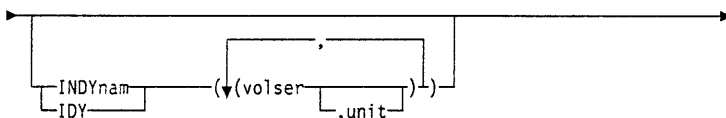
When DFDSS invokes IEHMOVE to copy multivolume non-VSAM data sets, IEHMOVE requires that the first DD statement in the job stream must identify all input volumes. DFDSS requires separate DD statements for the input volumes. To accommodate both requirements, you must code your JCL as follows:

```
//INDD1 DD UNIT=(SYSDA,2),VOL=SER=(VOL1,VOL2),DISP=SHR
//INDD2 DD UNIT=SYSDA,VOL=SER=VOL2,DISP=SHR
```

and code your DFDSS control statement:

```
COPY LOGINDD(INDD1,INDD2) ...
```

INDYnam



specifies that the volume(s) to be copied is to be dynamically allocated. If the volume is real (not virtual), it must be mounted and online. Only one volume is allowed for a full or tracks COPY; one or more volumes are allowed for a data set COPY. Consider using INDYNAM instead of DD statements to allocate DASD volumes. This does not appreciably increase run time and permits easier coding of JCL and command input.

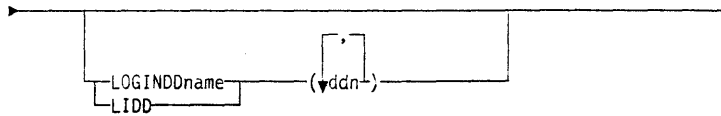
volser

specifies the volume serial number of a DASD volume to be copied.

unit

specifies the device type of a DASD volume to be copied. This keyword is optional.

LOGINDDname

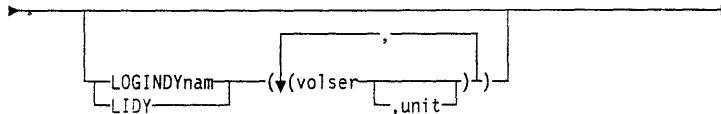


specifies that data sets be selected from the specified volume or volumes (which you can also do by specifying INDDNAME or INDYNAM) and also allows you to specify ALLMULTI (which cannot be done with INDDNAME or INDYNAM).

ddn

specifies the name of the DD statement that identifies a volume that contains the data set(s) to be copied. For single-volume data sets, each of the DD statements corresponding to a DDNAME (*ddn*) must identify only one volume serial number. See the note under LOGINDYNAM.

LOGINDYnam



specifies that the volume(s) that contains the data set(s) to be copied is dynamically allocated. If the volume is real (not virtual), it must be mounted and online.

volser

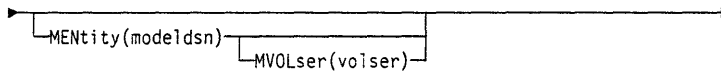
specifies the volume serial number of a DASD volume to be copied.

unit

specifies the device type of a DASD volume to be copied. This keyword is optional.

Note: If neither LOGINDDNAME nor LOGINDYNAM is specified, DFDSS selects from all data sets cataloged in the catalogs accessible through the standard search order. If either LOGINDDNAME or LOGINDYNAM is specified, DFDSS still uses the standard catalog search order, but it selects data sets only from the specified volume(s). A multivolume data set that has extents on volumes not specified with LOGINDDNAME or LOGINDYNAM is copied only if you specify the ALLMULTI keyword.

MENTity



specifies, for RACF-protected data sets, an entity (*modeldsn*) and, optionally, the serial number of the volume containing that entity (*volser*). These keywords are used to define the data sets to RACF. Specification of MVOLSER is optional for one of the following:

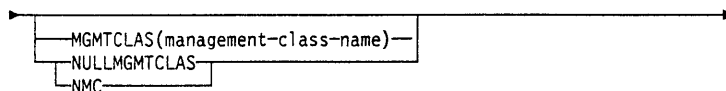
- When the model entity (MENTITY) is cataloged in an integrated catalog facility catalog.
- When a non-VSAM data set is cataloged in the standard catalog search order.

When MVOLSER is specified for a VSAM model entity, *volser* must be the volume serial number of the catalog in which the model entity is cataloged. If these keywords are not specified, DFDSS defines the data set to RACF by modeling the target profile after the source data set, if the source is discretely protected (for details, see Figure 3 on page 43). Target data sets (preallocated or nonpreallocated)

are RACF-protected only if the corresponding source data sets were so protected. If a source data set is protected with a generic profile, RACF generic profile checking must be activated prior to invoking COPY.

If you do not specify MENTITY and you have RACF Release 1.5 or later installed, Figure 3 on page 43 shows how profiles must be defined.

MGMTCLAS



specifies the user-desired management class that will replace the source management class as input to the ACS routines. You must have the proper RACF authority for the management class specified. The keyword itself does not require RACF authorization.

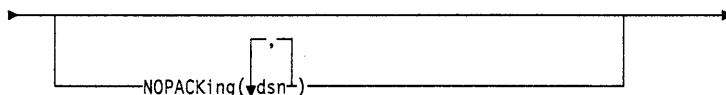
NULLMGMTCLAS specifies that the input to the ACS routines will be a null management class rather than the source data set's management class.

MGMTCLAS and NULLMGMTCLAS are mutually exclusive; you cannot specify these keywords together.

Notes:

1. All SMS-managed data sets specified in the BYPASSACS keyword will be assigned the specified management class because the ACS routines will not be invoked. Non-SMS-managed data sets do not have a management class.
2. See "Assignment of Class Names by Using RESTORE and COPY" on page 120 for information on the assignment of class names using COPY.

NOPACKing



specifies that DFDSS is to allocate the target data set only to the same or like device types as the source data set is allocated on and that DFDSS is to use track level I/O to perform data movement. This results in an exact track-for-track image of the source data set on the target volume.

dsn

specifies the fully- or partially-qualified name(s) of a PDS to be processed.

NOPACKing is only valid with a PDS. If REBLOCK is specified, REBLOCK is ignored for the data set. If the data set is specified with CONVERT(PDSE()), NOPACKing is ignored for the data set.

A PDS copied or restored by using NOPACKing will not be compressed during data movement.

NOPACKing can be used for a damaged PDS that is currently usable by an application but would be made unusable by compression or other rearrangement of the physical layout of the data.

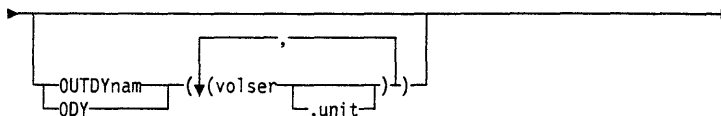
OUTDDname



ddn

specifies the name of the DD statement that identifies the output DASD volume. To assure correct processing, each of the DD statements corresponding to a DDNAME (*ddn*) must identify only one volume serial number. For a data set COPY, you can specify multiple names separated by commas.

OUTDYnam



specifies that the output DASD volume is to be dynamically allocated. If the volume is real (not virtual), it must be mounted and online. Only one volume is allowed for a full or tracks COPY; one or more volumes are allowed for a data set COPY.

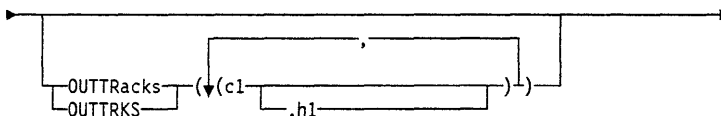
volser

specifies the volume serial number of the volume.

unit

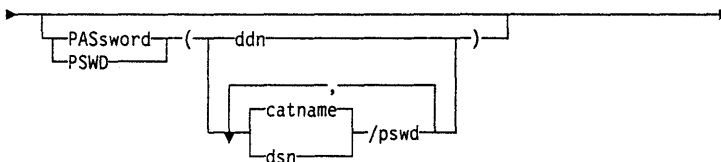
specifies the device type of the volume. This keyword is optional.

OUTTRacks



specifies, for a tracks COPY, the cylinder (cc) and head number (hh) on the output volume to which the tracks from the input volume are to be copied. If you do not specify OUTTRACKS, the tracks are copied to the place on the output volume where they were on the input volume.

PASsword



specifies the passwords that DFDSS is to use for password-protected data sets. (Password checking is bypassed for RACF-protected data sets.) This is required only when:

- You do not have the required RACF DASDVOL or RACF DATASET access.
- The installation authorization exit does not bypass the checks.
- You do not want to be prompted for the password.

Passwords must be specified for data sets as follows:

- Read passwords for source data sets when DELETE is not specified.
- Alter passwords for target data sets when DELETE is specified.
- Master passwords for catalogs.

Password checking is done at the cluster level for VSAM data sets. Passwords for a tracks COPY are required only for the data sets on which the requested ranges fall.

ddn

specifies the name of the DD statement that identifies the sequential data set, or member of a partitioned data set, that contains data set names and their passwords. This data set must contain card-image records in DFDSS command syntax format.

dsn/pswd

dsn is a fully-qualified data set name. *pswd* is its password. If no password follows the slash (/), *dsn* is treated as though it were *ddn*.

catname/pswd

catname is a fully-qualified catalog data set name. *pswd* is its password. This is supported for VSAM data sets only.

Printing of actual data set passwords specified in the input command stream is suppressed in the SYSPRINT output.

When a system utility is being used to perform the DFDSS COPY, the user must supply the password for each password-protected data set selected or have the proper RACF data set access authority. Figure 3 on page 43 shows how profiles must be defined.

	No RENAME Keyword No DELETE Keyword	No RENAME Keyword DELETE Keyword	RENAME Keyword No DELETE Keyword	RENAME Keyword DELETE Keyword
Unprotected source	No profile to define.	No profile to define.	No profile to define.	No profile to define.
Generic profile for source	Leave source profile. No target profile to define.	Leave source profile. No target profile to define.	Leave source profile. ⁴	Leave source profile. ⁴
Discrete profile for source	Leave source profile. Define discrete target profile. ⁵	Delete source profile. Define discrete target profile. ⁵	Leave source profile. Define a discrete target profile. ⁵	Delete source profile. Define a discrete target profile. ⁵

Figure 3. Defining Profiles When RACF 1.5 or Later Is Installed

⁴ Issue RACHECK to determine if a generic or discrete profile already exists for the renamed target. If so, there is no profile to define. If not, a discrete profile will be defined. DFDSS gets the *volser* from the data set serial number field in the Format 1 DSCB. If, for instance, the volume has been clipped and renamed, ensure that this field is correct.

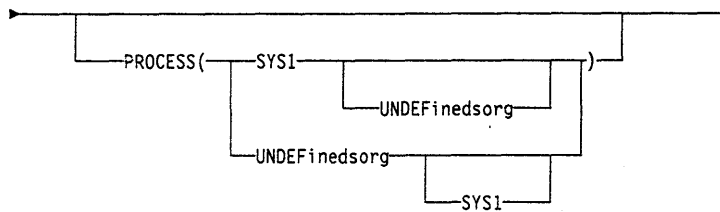
⁵ When a discrete profile exists for a preallocated target data set, that profile is not deleted and no new profile is defined.

PERCENTUtilized



specifies that DFDSS must stop allocating data sets to the target volume(s) when the allocated space reaches *n* percent of the total space on the target volume. The default value is 100. Specify more than one *n* if you have more than one target volume (for instance, a volume for overflow). This keyword is used as a guide only and might not be precise for all situations. This keyword might be ignored by DFDSS for VSAM data sets allocated to multiple target volumes.

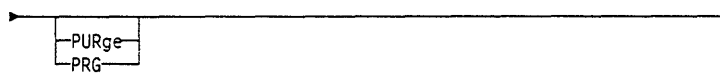
PROCESS



specifies that DFDSS is to allow data sets with a high-level qualifier of SYS1 to be copied to a preallocated target and that SYS1 data sets can be deleted. However, even if DELETE is specified, SYS1.VVDSs and SYS1.VTOCIXs are not copied and deleted. To specify PROCESS, you must have the proper RACF Facility Class authorization. Refer to *Resource Access Control Facility (RACF) Security Administrator's Guide* for additional information.

PROCESS also specifies that DFDSS allows undefined DSORG data sets to be copied to an unlike target with a larger capacity. Refer to Figure 4 on page 56 to discern the action taken by DFDSS.

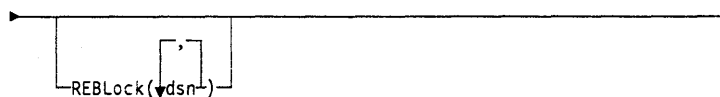
PURge



specifies, for a full or tracks COPY, that unexpired data sets on the target volume can be overlaid. If PURGE is not specified and unexpired data sets reside on the target tracks, the COPY operation is ended.

For a data set COPY, PURGE specifies that unexpired source data sets are to be deleted after they are successfully copied. This keyword is only valid when DELETE is also specified.

REBLock



specifies that DFDSS is to reblock one or more of the selected SAM or PAM data sets unless otherwise overridden by the user reblock exit routine.

dsn

specifies the fully- or partially-qualified name(s) of a SAM or PAM data set to be copied and reblocked.

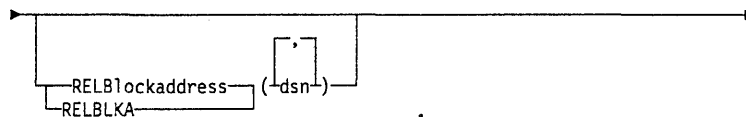
If a PDS is also specified in NOPACKing, the PDS is not reblocked. The following data sets will not be eligible for reblocking:

- Non-load module PAM data sets with NOTE lists
- SAM or PAM data sets with RECFM = U.

When copying PAM load modules to an unlike device, DFDSS uses IEBCOPY with COPYMOD specified. This may result in a reblocked data set. When copying to a like device, IEBCOPY is used.

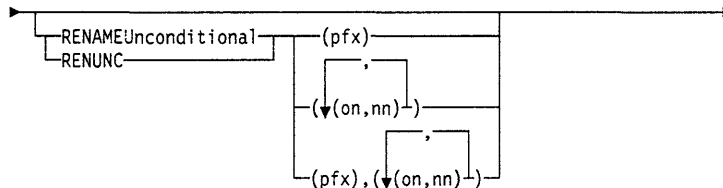
Under MVS/XA DFP Version 2 Release 2 or later release, DFDSS uses the DASDCALC macro to determine the optimal block size for the target. The reblocking method used, DFDSS or DASDCALC, is presented to the installation reblock exit. Track overflow data sets can be reblocked by using either method.

RELBBlockaddress



identifies the BDAM data sets whose names match the fully- or partially-qualified names specified (*dsn*). These BDAM data sets are organized by relative block address instead of TTR and are to be copied block by block without maintaining the relative track and record number for each record, provided the blocks fit on the tracks of the target volume. This keyword is ignored for TTR-organized BDAM data sets and for BDAM data sets with standard user labels.

RENAMEUnconditional



specifies that the data set must be copied with the new name, regardless of whether the data set exists on DASD with the old name. If the data set exists on the target volume with the new name, an allocation error occurs, and the data set is not copied, regardless of whether REPLACE is specified.

pfx

specifies the prefix used to replace the first-level qualifier of the data set name. It is optional, but if specified, must be the first parameter in the list of subkeywords. The prefix is used only if the (*on,nn*) parameters are not specified or the old name filters do not match the data set name.

on

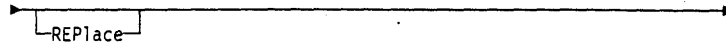
specifies the old name to be used as a filtering criterion to check if it matches the data set name.

nn

specifies the new name to be used to derive the new data set name when the data set name matches the corresponding old name filtering criterion.

The syntax rules for *pfx* (prefix), *on* (old name), and *nn* (new name) are the same as in the RENAME keyword in a RESTORE operation. See “REName” on page 112 under the RESTORE command for more information.

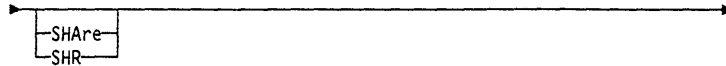
REPlace



specifies that DFDSS is to search the target volumes for usable preallocated data sets. If no preallocated target is found, DFDSS attempts to allocate a data set.

Note: If REPLACE is specified with COPY, the classes already associated with the preallocated data set remain the same.

SHAre

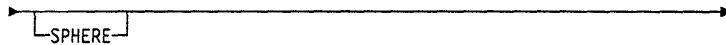


specifies that DFDSS is to share the data sets to be copied for read access with other programs.

SHARE and FULL are mutually exclusive; you cannot specify these keywords together.

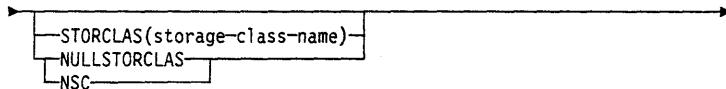
Do not specify DELETE if you specify SHARE. You must have exclusive control over data sets that are to be deleted; SHARE does not require such exclusive control.

SPHERE



specifies that, for any VSAM cluster copied, all associated AIX clusters and paths are to be copied. Individual names of sphere components do not need to be specified. Only the base cluster name is required.

STORCLAS

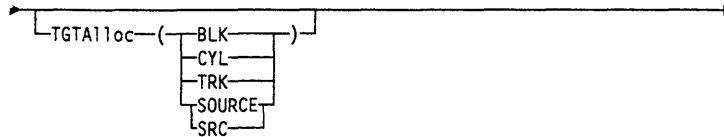


specifies the storage class that you want to replace the source storage class as input to the ACS routines. You must have the proper RACF authorization for the storage class specified. The keyword itself does not require authorization.

NULLSTORCLAS specifies that the input to the ACS routines is to be a null storage class rather than the source data set's storage class.

STORCLAS and NULLSTORCLAS are mutually exclusive; you cannot specify both keywords simultaneously. See "Assignment of Class Names by Using RESTORE and COPY" on page 120 for information on assigning class names by using COPY.

Note: If BYPASSACS(dsn) is specified, also all data sets that pass the BYPASSACS selection criteria are guaranteed the specified storage class. The combination of NULLSTORCLAS and BYPASSACS(dsn) forces the selected data sets to be non-SMS.

TGTALloc

specifies how DFDSS is to allocate the target data set.

BLK

specifies to allocate by blocks.

CYL

specifies to allocate by cylinders.

TRK

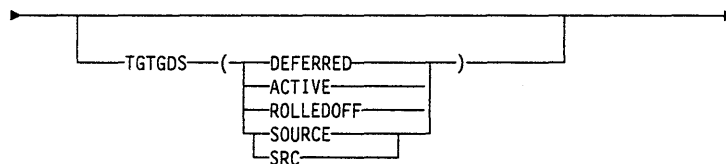
specifies to allocate by tracks.

SOURCE/SRC

specifies to allocate with the same space allocation type as that of the source data set.

Notes:

1. If the TGTALLOC keyword is omitted, the target allocation defaults to SOURCE.
2. If SRC is specified and if the source data set is allocated by cylinder or if CYL is specified, then the final VSAM allocation might be different from the requested one because of VSAM allocation rules.
3. If BLK is specified for VSAM data sets, TRK is used instead. The final VSAM allocation might be different from the requested one because of VSAM allocation rules.

TGTGDS

specifies in what status, during a data set operation, that DFDSS is to place non-preallocated SMS-managed GDG data sets:

DEFERRED

specifies that the target data set is to be assigned the DEFERRED status.

ACTIVE

specifies that the target data set is to be assigned the ACTIVE status, for example, rolled into the GDG base.

ROLLEDOFF

specifies that the target data set is to be assigned the rolled-off status.

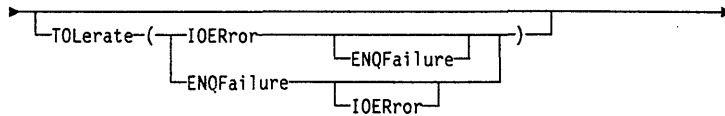
SOURCE/SRC

specifies that the target data set is to be assigned the same status as that of the source data set

Notes:

1. If DELETE is specified without RENAMEUNCONDITIONAL while processing SMS-managed generation data sets, the TGTGDS keyword is ignored and the source GDS status is copied to the target.
2. The requested target status of generation data sets must not violate generation data group rules.

TOLerate



specifies that DFDSS tolerates certain error conditions. For a COPY (except of a user catalog) in which a utility performs the copy, this keyword is ignored.

ENQFailure

specifies that data sets are to be processed even though shared or exclusive access fails.

ENQF and FULL

are mutually exclusive; you cannot specify these keywords together.

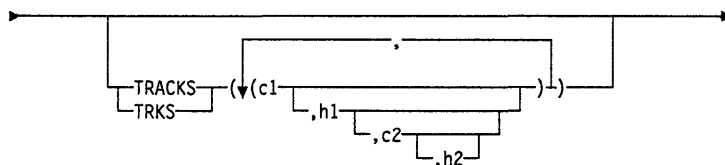
For more information on TOL(ENQF), see Appendix B, "Serialization and Authorization Checking" on page 165.

IOERROR

specifies that if the input volume can be opened, DFDSS is to continue copying even though permanent input errors (busout parity and equipment checks only) occur. DFDSS will end after 100 errors when this keyword is specified. The default is to end on permanent input errors. On a data set COPY in which a utility performs the copy, DFDSS ignores this keyword.

Note: TOL(IOERror) is ignored if CANCELerror is specified.

TRACKS



specifies ranges of tracks to be copied (that is, a tracks COPY).

c1,h1

specifies the cylinder and head number of the beginning of the range. Specify hexadecimal numbers as X'c1' or X'h1'.

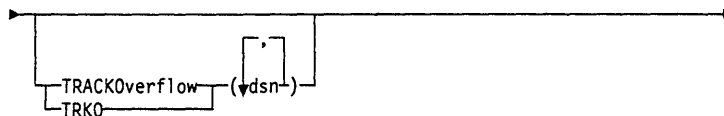
c2,h2

specifies the cylinder and head number of the end of the range. Specify hexadecimal numbers as X'c2' or X'h2'. The c2 must be greater than or equal to c1. If c2 equals c1, h2 must be greater than or equal to h1.

DFDSS verifies that the range is within the limits of the device. If you do not specify all four values for a range, DFDSS provides the missing values unless the omitted value causes a syntax error. No intervening values can be omitted. For example:

Specified	Results
<i>None</i>	Syntax error
<i>c1</i>	<i>c1,0,c1,maximum head number</i>
<i>c1,h1</i>	<i>c1,h1,c1,maximum head number</i>
<i>c1,h1,c2</i>	<i>c1,h1,c2,maximum head number</i>
<i>c1,,c2</i>	Syntax error
<i>,h1</i>	Syntax error.

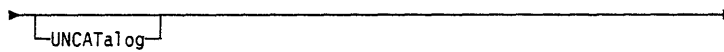
TRACKOverflow



dsn

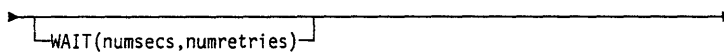
specifies the name of data set(s) in which track overflow records are present but are not indicated by the Format 1 DSCB. The TRACKOVERFLOW keyword is not needed for data sets indicated as having overflow records. When you perform a COPY to an unlike device and the target device supports track overflow, specifying TRACKOVERFLOW allows DFDSS to correctly track-pack data sets. It sets the target's Format 1 DSCB to indicate that overflow records are present.

UNCATalog



specifies that DFDSS is to uncatalog but not scratch successfully copied non-VSAM data sets that are currently cataloged on the source volume. Any data set that is not a user catalog and has a name beginning with SYS1. cannot be uncataloged. UNCATALOG is ignored for VSAM data sets and SMS non-VSAM data sets.

WAIT



For a volume or tracks COPY, WAIT specifies to DFDSS the length of a wait in seconds and the number of retries to obtain control of a volume or data set.

numsecs

is a number from 1 to 255 that specifies the interval, in seconds, between retries.

numretries

is a 1- or 2-digit number that specifies the number of times an attempt to gain control of a volume is to be retried.

For a data set COPY, WAIT specifies to DFDSS the length of the wait in seconds, and the number of passes to be made through the list of selected data sets to obtain control of a data set.

numsecs

is a number from 1 to 255 that specifies the interval, in seconds, to wait before attempting another pass through the entire list of selected data sets.

numretries

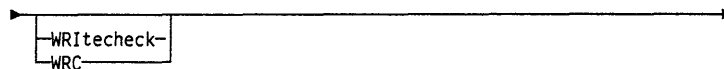
is a 1- or 2-digit number that specifies the number of passes to make through the list of selected data sets in an attempt to obtain control of a data set.

The default for *numsecs,numretries* is (2,2), which specifies two retries at 2-second intervals. If you do not want to wait for a resource, specify 0 for either *numsecs* or *numretries*.

For a data set COPY or logical data set DUMP, the WAIT keyword has a different meaning when: (1) data sets are being serialized, (2) multiple data sets are being processed, and (3) WAIT(0,0) is not specified. In this case, DFDSS makes multiple passes through the list of data sets. On each pass, DFDSS processes the data sets that (1) can be serialized without waiting for the resource and (2) were not processed before. At the end of a pass, if none of the data sets could be processed without waiting for a resource, then, in the next pass, at the first occurrence of a data set that was not processed, a WAIT is issued. That data set and the remainder of the list are processed if possible.

The above procedure is repeated until all data sets are processed or the WAIT limits are reached. For example, if WAIT(3,10) is specified and five data sets are left to be processed, up to ten passes are made. On each pass, an unprocessed data set is waited upon for 3 seconds. Thus, only a 30-second maximum will ever be WAITed, not 150 (5 times 3 times 10).

WRIttecheck



specifies that the data copied is to be verified for successful completion. This keyword increases the overall elapsed time. The default is no WRITECHECK. On a data set COPY in which a utility performs the copy, DFDSS ignores this keyword.

Examples of Full and Tracks COPY Operations

Example 1:

```
//JOB1    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADRDSU
//SYSPRINT DD   SYSOUT=A
//DASD1   DD     UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//DASD2   DD     UNIT=3380,VOL=(PRIVATE,SER=222222),DISP=OLD
//SYSIN   DD     *
          command input (see Examples 1A and 1B below)
/*
```

Example 1A: A Full COPY

```
COPY INDDNAME(DASD1) OUTDDNAME(DASD2) -
     ALLDATA(*) ALLEXCP CANCELERROR WAIT(30,20)
```

Example 1B: A Tracks COPY

```
COPY TRACKS(1,0,1,15) INDDNAME(DASD1) -
     OUTDDNAME(DASD2) CANCELERROR WAIT(30,20)
```

The data from DASD volume 111111 is to be copied to DASD volume 222222. For the full COPY, all allocated space in SAM and PAM data sets and data sets with null DSORG field is copied (ALLDATA(*)), when the DS1LSTAR field in the Format 1 DSCB is nonzero. For these data sets, DFDSS copies all allocated space (ALLEXCP) when the DS1LSTAR field in the Format 1 DSCB in the VTOC is zero. The COPY operation is to be ended if a permanent read error occurs (CANCELERROR). Retry 20 times at 30-second intervals (30,20) if the input or output volume is busy.

Example 2: A Tracks COPY With Track Relocation

```
//JOB2    JOB  accounting information,REGION=nnnnK
//STEP1   EXEC PGM=ADRSSU
//SYSPRINT DD  SYSOUT=A
//SYSIN   DD   *
COPY TRACKS(1,0,1,14) /* SOURCE TRACKS */ -
  OUTTRACKS(3,0) /* TARGET TRACKS */ -
  INDYNAM(338000) /* ALLOC VOL 338000 DYNAMICALLY */ -
  OUTDYNAM(338001) /* ALLOC VOL 338001 DYNAMICALLY */ -
  CANCELERROR /* STOP ON INPUT ERROR */ -
  WRITECHECK /* VERIFY DATA WRITTEN TO OUT VOL */
/*
```

This example shows a tracks COPY operation in which the contents of cylinder 1, tracks 0 through 14, on source volume 338000 are copied to cylinder 3, tracks 0 through 14, on target volume 338001. The operation stops if a permanent error occurs on the source volume (CANCELERROR). The data written to the target volume is to be verified (WRITECHECK).

Examples of Data Set COPY Operations**Example 1: A Data Set Move—Only Single Volume Data Sets**

```
//JOB3    JOB  accounting information,REGION=nnnnK
//STEP1   EXEC PGM=ADRSSU
//SYSPRINT DD  SYSOUT=A
//SYSIN   DD   *
COPY DATASET(
  INCLUDE(USER1.**)) /* FILTER ON DS W/1ST LEV Q USER1 */ -
  BY(MULTI,=,NO)) /* FILTER ON SINGLE VOLUME */ -
  INDYNAM (338000) /* ALLOC VOL 338000 DYNAMICALLY */ -
  OUTDYNAM(338001) /* ALLOC VOL 338001 DYNAMICALLY */ -
  DELETE
/*
```

This example shows a data set COPY in which all single-volume data sets with the first-level qualifier USER1 on the source volume labeled 338000 are copied to the target volume labeled 338001. All source data sets that are selected are deleted. The copied non-VSAM data sets are not cataloged.

Example 2: A Data Set COPY to Move Data Sets to a Single Volume—Device Conversion

```
//JOB4    JOB  accounting information,REGION=nnnnK
//STEP1   EXEC PGM=ADRSSU
//SYSPRINT DD  SYSOUT=A
//SYSIN   DD   *
COPY DATASET(
  INCLUDE(USER1.**)) /* FILTER ON DS W/1ST LEV Q USER1 */ -
  OUTDYNAM(338001) /* ALLOC VOL 338001 DYNAMICALLY */ -
  DELETE CATALOG FORCE -
  TGTALLOC(SOURCE)
/*
```

This example shows a data set COPY in which all cataloged data sets with a first-level qualifier of USER1 (USER1.***) data sets are to be consolidated on a single target volume labeled 338001. The data sets are scattered on multiple source volumes. The target volume can be of unlike device type to any of the volumes on which the source data sets reside. A few of the data sets might already be on volume 338001. The data sets are to be cataloged (CATALOG) either in the master catalog or in a user catalog that is pointed to by the master catalog. The source data sets are scratched and uncataloged (DELETE) after successfully moving them to volume

338001, where they will have the same allocation type (BLK, TRK, or CYL) that they had on the source volumes. FORCE is specified to include unmovable data sets.

Example 3: A Data Set COPY of a Multivolume Data Set

```
//JOB5     JOB  accounting information,REGION=nnnnK
//STEP1    EXEC PGM=ADRDSSU
//IVOL1    DD  UNIT=(SYSDA,2),VOL=SER=(VOL111,VOL222),DISP=SHR
//IVOL2    DD  UNIT=SYSDA,VOL=SER=VOL222,DISP=SHR
//SYSPRINT DD  SYSOUT=A
//SYSIN    DD  *
COPY DATASET(
    INC(USER.MULTI.VOLUME1)) /* SELECT THIS DATA SET    */ -
    INDD(IVOL1,IVOL2)       /* IDENTIFY INPUT VOLUMES */ -
    OUTDYNAM((338001),(338002),(338003)) /* DYNAM ALLOC VOLS */ -
    PCTU(80,80,80)         /* PERCENTUTIL = 80 PERCENT */ -
    RECATALOG(USERCAT2)
/*
```

This example shows a data set COPY in which a multivolume data set is copied to a set of target volumes labeled 338001, 338002, and 338003. The source data set is not deleted. The copied data set is cataloged in a new catalog, USERCAT2. The data set currently exists on multiple source volumes. Multiple output volumes are specified for overflow purposes. These target volumes may already have some data sets on them. Some space is left on these volumes to allow expansion of the data sets (PCTU) remaining on the volumes.

Change this example as shown below to include ALLMULTI processing on missing volumes. The INCLUDE keyword is changed to select all data sets on input volumes VOL111 and VOL222. ALLMULTI specifies that a cataloged data set residing on volumes VOL111, VOL444, and VOL555 will be copied even though VOL444 and VOL555 are omitted from the LOGINDD volume list.

```
COPY DATASET(
    INC(**)) /* SELECT ALL DATA SETS */ -
    LOGINDD(IVOL1,IVOL2) /* IDENTIFY INPUT VOLUMES */ -
    OUTDYNAM((338001),(338002),(338003)) /* DYNAM ALLOC VOLS */ -
    ALLMULTI /* PROCESS MISSING VOLUMES */ -
    PCTU(80,80,80) /* PERCENTUTIL = 80 PERCENT */ -
    RECATALOG(USERCAT2)
```

Example 4: A Data Set COPY With DELETE and RENAMEU Options

```
//JOB6     JOB  accounting information,REGION=nnnnK
//STEP1    EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=A
//SYSIN    DD  *
COPY DATASET(
    INCLUDE(USER1.**)) /* FILTER ON DS W/1ST LEV Q USER1 */ -
    OUTDYNAM((338001),(338002),(338003)) /* DYNAM ALLOC VOLS */ -
    DELETE           -
    RENAMEU(USER2)   -
    RECATALOG(USERCAT2)
/*
```

This example shows a data set COPY in which all data sets with the first-level qualifier USER1 that are in the master catalog are copied to the target volumes labeled 338001, 338002, and 338003. The copied data sets are renamed (RENAMEU) with the first-level qualifier USER2, followed by the second through last qualifiers of the old names. If data sets with the same name as the new names are on the target volumes or if they are already cataloged in USERCAT2, they are not copied. The

copied data sets are deleted (DELETE) from the source volumes, uncataloged, and cataloged (RECATALOG) in the catalog USERCAT2. This provides movement of data sets from one set of volumes to another set of volumes, from one catalog to another catalog, and the renaming of the data sets.

Example 5: A Data Set COPY With REBLOCK Option

```
//JOB7      JOB  accounting information,REGION=nnnnK
//STEP1     EXEC PGM=ADRDSSU,PARM='UTILMSG=YES'
//SYSPRINT  DD  SYSOUT=A
//DISK      DD  UNIT=3350,VOL=(PRIVATE,,,,SER=335001),DISP=SHR
//DISK2     DD  UNIT=3380,VOL=(PRIVATE,,,,SER=338001),DISP=SHR
//SYSIN     DD  *
COPY DATASET(
  INCLUDE(**) )          /* INCLUDE ALL DATA SETS */ -
  INDDNAME(DISK)         /* INPUT VOLUME           */ -
  OUTDDNAME(DISK2)      /* OUTPUT VOLUME          */ -
  REBLOCK(**.USER1.**))
/*
```

This example shows a data set COPY in which all data sets on the 3350 source volume labeled 335001 are copied to the 3380 target volume labeled 338001. If data sets with the same name are on the target volume, they are not copied. The SAM and PAM data sets that meet the filtering criteria specified in the REBLOCK keyword are reblocked on the target volume. The block size is selected by DFDSS, unless it is modified by the user reblock exit routine.

Example 6: A Data Set COPY to a Preallocated Target Data Set

```
//JOBn      JOB  accounting information,REGION=nnnnK
//STEP1     EXEC PGM=ADRDSSU
//STEP1CAT  DD  DSN=USERCAT,DISP=SHR
//SYSPRINT  DD  SYSOUT=A
//DASD1     DD  UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//DASD2     DD  UNIT=3380,VOL=(PRIVATE,SER=222222),DISP=OLD
//SYSIN     DD  *
COPY DATASET(
  INCLUDE(USER.TEST.DATA) ) -
  INDDNAME(DASD1) -
  OUTDDNAME(DASD2) -
  REPLACE -
  DELETE
/*
```

This example shows a data set COPY in which a source data set (USER.TEST.DATA) allocated on volume 111111 and cataloged in catalog USERCAT is copied to a preallocated target data set (with the same name as the source) on volume 222222. The REPLACE keyword specifies that DFDSS is to search the target volume for a usable preallocated data set. The copied data set is deleted (DELETE) from the source volume.

Example 7: Using COPY to Convert to SMS

```
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COPY -
  DS(INC(**)) -
    LOGINDYNAM ( -
      (338001) -
      (338002) -
    ) -
    STORCLAS(DB2PERF) -
    MGMTCLAS(DBBACKUP) -
    BYPASSACS(**) -
    DELETE
/*
```

In the first step of this example, all of the data sets on the non-SMS volumes 338001 and 338002 are copied to SMS volumes on the system. DELETE is used to avoid duplicate catalog entries. The ACS routines will not be invoked for this COPY. The storage and management classes are provided by the users using the STORCLAS and MGMTCLAS keywords. In addition, BYPASSACS(**) is specified in order to suppress calls made to the ACS routines. All data sets that are supported by SMS will be given the new storage and management classes. All data sets that cannot be SMS managed (for example, ISAM) will not be copied.

```
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COPY -
  DS(INC(**)) -
    LOGINDYNAM ( -
      (338001) -
      (338002) -
    ) -
    RENUNC(AUG0387)
/*
```

In this next step all data sets on the non-SMS volumes 338001 and 338002 are copied to SMS volumes on the system. RENUNC is used to avoid duplicate catalog entries. The ACS routines will select a target storage and management class for each data set. Those data sets that cannot be SMS managed (storage class ACS routine returns a null storage class) will not be copied because no output volume is specified. Each data set that is copied will be given a new high-level qualifier (AUG0387) and automatically cataloged.

Example 8: A Data Set Copy Using CONVERT PDSE

```

//SYSPRINT DD  SYSOUT=*
//SYSIN     DD  *
COPY -
  DS(INC(USER.**)) -
  LOGINDYNAM ( -
    (338001) -
    (338002) -
  ) -
  CONVERT (PDSE(**)) -
  RENCUNC (USER.PDS.**, USER.PDSE.**
/*

```

In this step all data sets on non-SMS volumes 338001 and 338002 are copied to SMS volumes on the system. CONVERT PDSE is used to convert data sets to PDSE. RENCUNC is used to avoid duplicate catalog entries. The ACS routines will select a target storage and management class for each data set. Each data set that is copied and converted will be given a new secondary qualifier (PDSE) and automatically cataloged.

In Figure 4, read each column from top to bottom, reading vertically down the list of criteria, to discern the action taken by DFDSS.

Results of Copying Non-VSAM Data Sets																						
Conditions																						
Non-empty data sets?	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Empty data sets (DS1LSTAR=0)?	-	-	-	-	-	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-
BLKSIZE=0 data sets?	N	N	N	N	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	-	-
Undefined DSORG data sets?	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	Y
To LIKE target device?	Y	Y	-	-	Y	Y	-	Y	Y	Y	Y	-	-	-	-	Y	Y	Y	Y	Y	-	-
To UNLIKE target device?	-	-	Y	Y	-	-	Y	-	-	-	-	Y	Y	Y	Y	-	-	-	-	-	Y	Y
To UNLIKE target device (greater track capacity)?	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	N	Y
PROCESS(UNDEF) specified?	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y
ALLDATA specified?	N	Y	N	Y	N	Y	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ALLEXCP specified?	-	-	-	-	-	-	-	N	N	Y	Y	N	N	Y	Y	N	N	Y	Y	-	-	-
Data set has EOF as first record?	-	-	-	-	-	-	-	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	-	-	-
Actions																						
Allocates and copies all allocated space		X				X																X
Allocates and copies only used space	X		X		X																	
Allocates all allocated space but copies only used space				X																		
Allocates all allocated space and copies only EOF										X				X				X				
Allocates only used space and copies only EOF								X				X				X						
Data set is not processed							X		X		X		X		X		X		X	X	X	X

Figure 4. Results of Copying Non-VSAM Data Sets

Legend to Figure 4:

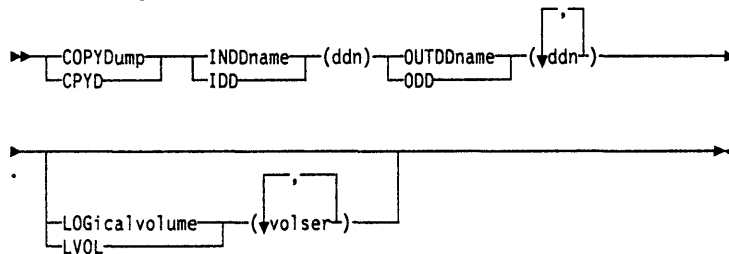
- Y Yes
- N No
- N/A
- X Action taken.

COPYDUMP Command

With the COPYDUMP command, you can make from 1 to 255 copies of DFDSS-produced dump data. The data to be copied, a sequential data set, can be on a tape or a DASD volume, and copies can be written to a tape or a DASD volume. If the dump data is produced from multiple DASD volumes by using a data set DUMP, you can selectively copy the data from one or more of those volumes.

Extra dump tapes can be used for such things as disaster recovery backup or distribution of dumped data (for example, a newly generated system).

COPYDUMP Command Syntax



Explanation of COPYDUMP Command Keywords

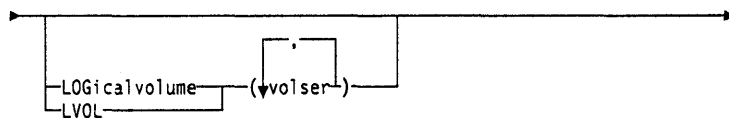
INDDname



ddn

specifies the name of the DD statement that identifies the sequential data set to be copied. This data set can reside on one or more tapes, or on DASD volume(s).

LOGicalvolume



volser

specifies the source DASD volume serial number from which dumped data is to be copied. Omission of the LOGICALVOLUME keyword causes DFDSS to copy data from all logical volumes in the dump data set. This keyword is useful only if the data being copied was created by a physical data set DUMP operation from multiple DASD volumes. When copying a *logical* dump, LOGICALVOLUME is ignored.

OUTDDname



ddn

specifies the name of the DD statement that identifies the output sequential data set. This data set can be on a tape or a DASD volume.

Examples of COPYDUMP Operations

Example 1: Making Two Copies of a Dump

```
//JOB1    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADRDSSU
//SYSPRINT DD   SYSOUT=A
//BACKUP  DD     UNIT=3480,VOL=SER=TAPE05,DISP=OLD,
//         DSNAME=V111111.BACKUP
//COPY1   DD     UNIT=3480,VOL=SER=TAPE06,
//         DISP=(NEW,CATLG),DSNAME=V111111.BACKUP1
//COPY2   DD     UNIT=3480,VOL=SER=TAPE07,
//         DISP=(NEW,CATLG),DSNAME=V111111.BACKUP2
//SYSIN   DD     *
COPYDUMP INDD(BACKUP) -
         OUTDD(COPY1,COPY2)
/*
```

In this example, two copies are to be made from a DFDSS dump tape (OUTDD(COPY1,COPY2)).

Example 2: Copying a Dump Created by Using Physical Data Set Processing

```
//JOB2    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADRDSSU
//SYSPRINT DD   SYSOUT=A
//TAPE2   DD     UNIT=3480,VOL=SER=TAPE20,
//         LABEL=(1,SL),DISP=(OLD,KEEP),DSN=USER.BACKUP.REL3A
//OUTT2   DD     UNIT=3480,VOL=SER=TAPE21,
//         LABEL=(1,SL),DISP=(NEW,CATLG),DSN=USER.BACKUP.REL3A.A
//SYSIN   DD     *
COPYDUMP  -
  INDD(TAPE2)           /* DUMP TAPE TO BE COPIED   */ -
  OUTDD(OUTT2)         /* NEW DUMP TAPE             */ -
  LVOL(338001)         /* SER NO OF VOL TO BE COPIED */
/*
```

Assume that a *physical* data set DUMP was used to create a dump tape, volume TAPE20. Also assume that source DASD volumes 338000, 338001, and so on were specified, resulting in VTOCs being used for data set selection. Only the dump data from DASD volume 338001 is to be copied (LVOL(338001)).

Example 3: Copying a Dump Created by Using Logical Data Set Processing

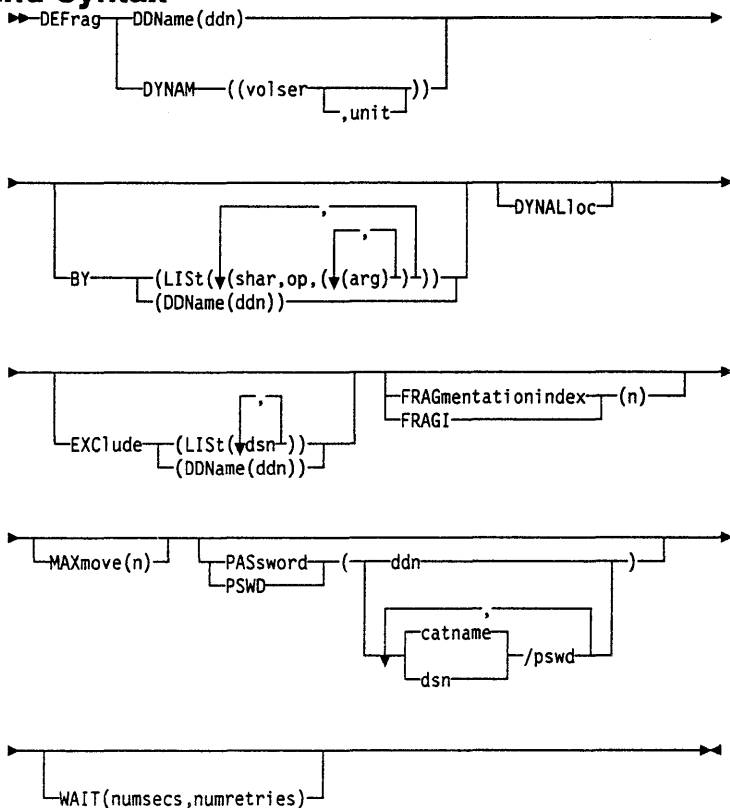
```
//JOB3    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADRDSSU
//SYSPRINT DD   SYSOUT=A
//TAPE3   DD     UNIT=3480,VOL=SER=TAPE30,
//         LABEL=(1,SL),DISP=(OLD,KEEP),DSN=USER.BACKUP.REL3B
//OUTT3   DD     UNIT=3480,VOL=SER=TAPE31,
//         LABEL=(1,SL),DISP=(NEW,CATLG),DSN=USER.BACKUP.REL3B.A
//SYSIN   DD     *
COPYDUMP  -
  INDD(TAPE3)           /* DUMP TAPE TO BE COPIED   */ -
  OUTDD(OUTT3)         /* NEW DUMP TAPE             */
/*
```

Assume that a *logical* data set DUMP was used to create a dump tape, volume TAPE30. All dump data is copied.

DEFRAG Command

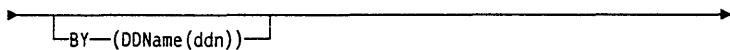
With the DEFRAG command, you can relocate data set extents on a DASD volume (including IBM 3330 virtual volumes) to reduce or eliminate free-space fragmentation, and print a report about free space and other volume statistics. Also, you can specify which data sets, if any, are to be excluded from data-set-extent relocation.

DEFRAG Command Syntax



Explanation of DEFRAG Command Keywords

BY(DDName)

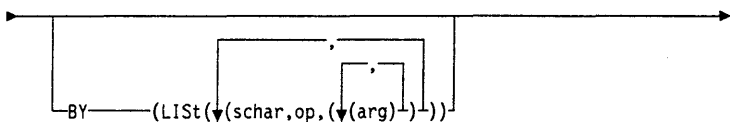


specifies data set filtering criteria.

ddn

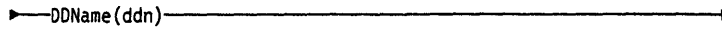
specifies the name of the DD statement that identifies a sequential data set or member of a partitioned data set that contains the filtering criteria to use. This is in the form of card-image records, in DFSS command syntax, that contain the BY keywords described below.

BY(LIST)



specifies data set filtering. To select the data set for inclusion in the DEFRAG operation, *all* BY criteria must be met. See “Filtering by Data Set Characteristics” on page 12 for further information on BY filtering.

DDName



ddn

specifies the name of the DD statement that describes the volume to be processed.

DYNALloc

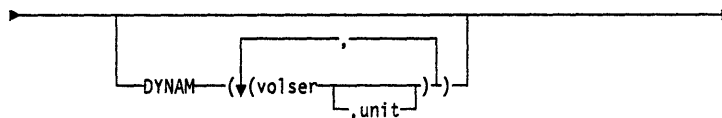


specifies dynamic allocation, instead of enqueue, to serialize the use of data sets. The data sets whose extents are to be relocated are serialized throughout DEFRAG. This allows cross-system serialization in a JES3/MVS environment.

Consider:

- The serialization is of value only when the dynamic allocation/JES3 interface is not disabled.
- SVC 99 is used to serialize data sets.
- Run time increases when you use DYNALLOC to serialize data sets (as opposed to enqueue) because overhead is involved in dynamic allocation and serialization across multiple processors.
- Using INDYNAM instead of DD statements to allocate DASD volumes does not appreciably increase run time and permits easier coding of JCL and command input.

DYNAM



specifies that the volume to be processed is to be dynamically allocated. If the volume is real (not virtual), it must be mounted and online.

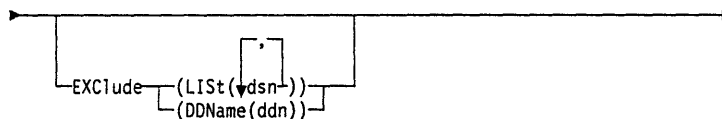
volser

specifies the volume serial number of a DASD volume to be processed.

unit

specifies the device type of a DASD volume to be processed. This keyword is optional.

EXClude



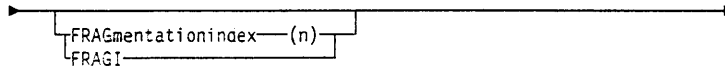
dsn

specifies a fully- or partially-qualified name of a data set to be excluded from the DEFRAG operation. You can specify either a cluster or component name for VSAM data sets.

ddn

specifies the name of the DD statement that identifies a sequential data set or member of a partitioned data set.

FRAGmentationindex



specifies that DEFRAG is to end if the fragmentation index is less than *n*, where *n* is a 1- to 3-digit number. DFDSS prefixes your number, *n*, with a decimal point. For example, *1* becomes *.1*, *999* becomes *.999*, *001* becomes *.001*, and so forth. For more information, see *Data Facility Data Set Services: User's Guide*.

MAXmove

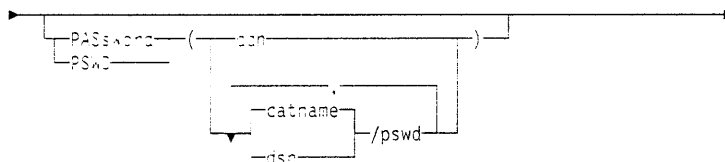


n is a 1- to 4-digit number that specifies that DFDSS is to attempt to assemble up to *n* tracks in a contiguous free area. If DEFRAG algorithms cannot assemble *n* contiguous free tracks without moving more than *n* tracks, or if *n* contiguous free tracks already exist, DFDSS issues a message and ends the DEFRAG operation. If MAXMOVE is not specified, DFDSS tries to assemble a contiguous free area of a size equal to the total number of free tracks on the volume.

The first pass attempts to assemble the largest contiguous amount of free space with a minimum amount of data movement. If you want the DEFRAG operation to end after the first pass, specify MAXMOVE.

The second pass attempts to assemble multiple groups of large areas of contiguous free space and generally moves more data around than the first pass. This technique usually refines the results of the first pass, but might result in a situation in which the largest individual free space is smaller than existed on the DASD originally or after the first pass.

PASsword



specifies the passwords DFDSS is to use for password-protected data sets. (Password checking is bypassed for RACF-protected data sets.) This must be specified only if:

- You do not have the required RACF DASDVOL or RACF DATASET access.
- The installation authorization exit does not bypass the checks.
- You do not want to be prompted for the password.

For VSAM data sets, password checking is done only at the cluster level.

ddn

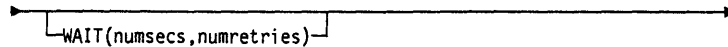
specifies the name of the DD statement that identifies the sequential data set, or member of a partitioned data set that contains data set names and their passwords. This data set must contain card-image records in DFDSS command syntax format.

dsn/pswd

dsn is a fully-qualified data set name. *pswd* is its password. If no password follows the slash (/), *dsn* is treated as though it were *ddn*.

Printing of actual data set passwords specified in the input command stream is suppressed in the SYSPRINT output.

WAIT



specifies to DFDSS the length of a wait in seconds and the number of retries to obtain control of a volume or data set.

numsecs

is a number from 1 to 255 that specifies the interval, in seconds, between retries.

numretries

is a 1- or 2-digit number that specifies the number of times an attempt to gain control of a volume or data set is to be retried.

The default for *numsecs,numretries* is (2,2), which specifies two retries at 2-second intervals. If you do not want to wait for a resource, specify 0 for either *numsecs* or *numretries*.

Examples of DEFRAG Operations

Example 1: A DEFRAG With Excluded Data Sets

```
//JOB1    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADDRSSU
//SYSPRINT DD   SYSOUT=A
//DASD    DD     UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//A1      DD     DSN=USER2.EXCLUDE,DISP=SHR
//SYSIN   DD     *
command input (see examples 1A and 1B below)
/*
```

Example 1A: With the Names of Excluded Data Sets in the Input Stream

```
DEFRAG DDNAME(DASD) -
      EXCLUDE(LIST(USER2.**.LIST,*.LOAD))
```

Example 1B: With the Names of Excluded Data Sets in a Data Set

```
DEFRAG DDNAME(DASD) -
      EXCLUDE(DDNAME(A1))
```

In examples 1A and 1B, DASD volume 111111 is defragmented. All data sets whose first and last qualifiers are USER2 and LIST, respectively, are to be excluded from this operation, as are data sets with two qualifiers whose second qualifier is LOAD. In example 1B, cataloged data set USER2.EXCLUDE contains a single card-image record with the following in columns 2 through 72:

```
USER2.**.LIST,*.LOAD
```

Example 2: A DEFRAG Using a BY Criterion

```
//JOB2    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADDRSSU
//SYSPRINT DD   SYSOUT=A
//DASD    DD     UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//SYSIN   DD     *
DEFRAG DDNAME(DASD) /* VOLUME TO BE PROCESSED          */ -
      BY(LIST(REFDT LT *,-1)) /* DATE LAST REF LT RUN DATE -1 */
/*
```

Only data sets last referenced more than one day before the run date are included in this DEFRAG operation. That is, those that were last referenced one day before or on the run date are excluded.

Results of a Successful DEFRAG Operation

Figure 5 on page 63 is the printout from a DEFRAG run for a DASD volume. It gives an indication of the free space fragmentation before and after a DEFRAG operation, as well as the distribution of data set extents by size.

The following JCL was used for this job:

```
//SAMPLE1 JOB   accounting information,REGION=nnnnK
//STEP4  EXEC   PGM=ADRDSSU
//SYSPRINT DD   SYSOUT=*
//DASD1   DD     VOL=SER=DUMPDS,UNIT=3380,DISP=OLD
//SYSIN   DD     *
           DEFRAG DDNAME(DASD1)
/*
```

```
PAGE 0001 DFDSS 5665-327 DATA FACILITY DATA SET SERVICES V2R5.0 89.067 14:55
DEFRAG DDNAME(DASD1) 150000
ADR1011 RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DEFRAG '
ADR109I RI01 (01), 89067 14:55:32 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR006I (001)-SETUP(01), 89067 14:55:32 PROCESSING BEGINS
ADR208I (001)-DANAL(01), BEGINNING STATISTICS ON DUMPDS:
      FREE CYLINDERS 0027
      FREE TRACKS 0021
      FREE EXTENTS 0004
      LARGEST FREE EXTENT (CYL,TRK) 0027,0000
      FRAGMENTATION INDEX 0.046
      PERCENT FREE SPACE 23
ADR213I (001)-DANAL(01), ENDING STATISTICS ON DUMPDS:
      DATA SET EXTENTS RELOCATED 0002
      TRACKS RELOCATED 0004
      FREE CYLINDERS 0027
      FREE TRACKS 0021
      FREE EXTENTS 0002
      LARGEST FREE EXTENT (CYL,TRK) 0026,0011
      FRAGMENTATION INDEX 0.044
PAGE 0002 DFDSS 5665-327 DATA FACILITY DATA SET SERVICES V2R5.0 89.067 14:55
ADR212I (001)-DANAL(01), EXTENT DISTRIBUTION MAP FOR DUMPDS:
      EXTENT **INPUT FREE** **OUTPUT FREE** ** OCCUPIED **
      SIZE
      IN NO. CUM. NO. CUM. NO. CUM.
      TRACKS EXTS PCT/100 EXTS PCT/100 EXTS PCT/100
      1 0.000 0.000 10 0.007
      2 1 0.004 6 0.016
      3 1 0.011 1 0.018
      4 0.011 0.000 1 0.021
      10 0.011 0.000 1 0.028
      15 0.011 0.000 4 0.072
      16 1 0.049 0.000 0.072
      25 0.049 1 0.058 0.072
      30 0.049 0.058 3 0.137
      75 0.049 0.058 3 0.301
      105 0.049 0.058 2 0.454
      375 0.049 0.058 2 1.000
      401 0.049 1 1.000 1.000
      405 1 1.000 1.000 1.000
ADR013I (001)-CLTSK(01), 89067 14:55:44 TASK COMPLETED WITH RETURN CODE 0000
ADR012I DSSU (01), 89067 14:55:44 DFDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
```

Figure 5. Printed Output Resulting from a Successful DEFRAG Run

The value in the first column of message ADR212I is the size of the extent in tracks (free space or data set) and is printed only if an extent of that size occurs. The second and third columns show the number of free space extents existing *before* processing that were of the size shown in the first column, along with their cumulative percentage divided by 100. The fourth and fifth columns give the same information for free space, but *after* processing. The sixth and seventh columns give the distribution of occupied data set extents, which do not change during a run.

DUMP Command

With the DUMP command, you can dump DASD data to a sequential data set. The storage medium for the sequential data set can be a tape, DASD, or a mass storage virtual volume. You can dump data sets, an entire volume, or ranges of tracks.

Note: The FULL keyword for the DUMP command specifies that an entire DASD volume is to be dumped. The TRACKS keyword for DUMP specifies ranges of tracks to be dumped.

DFDSS offers two ways to process DUMP commands:

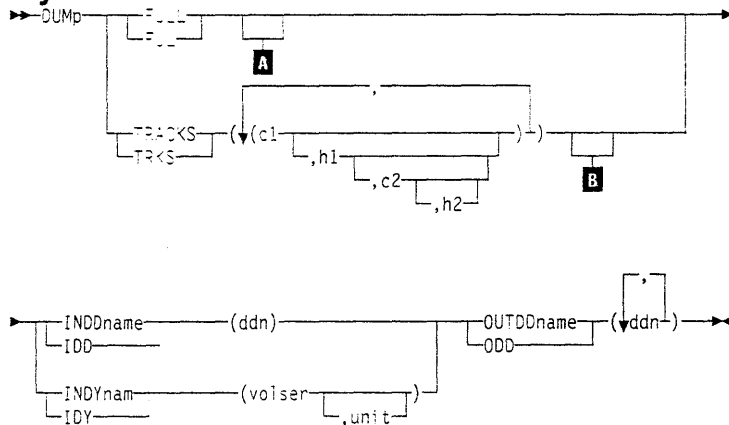
- *Logical processing* is data-set oriented, which means it operates against data sets and volumes independently of physical device format.
- *Physical processing* can operate against data sets, volumes, and tracks, but is oriented toward moving data at the track-image level.

The processing method is determined by the keywords specified on the command.

DFDSS logical DUMP processing cannot be used to process partitioned data sets containing location-dependent information that does not reside in note lists or in the directory. For example, PDS load libraries that contain overlay modules are not supported during logical DUMP processing.

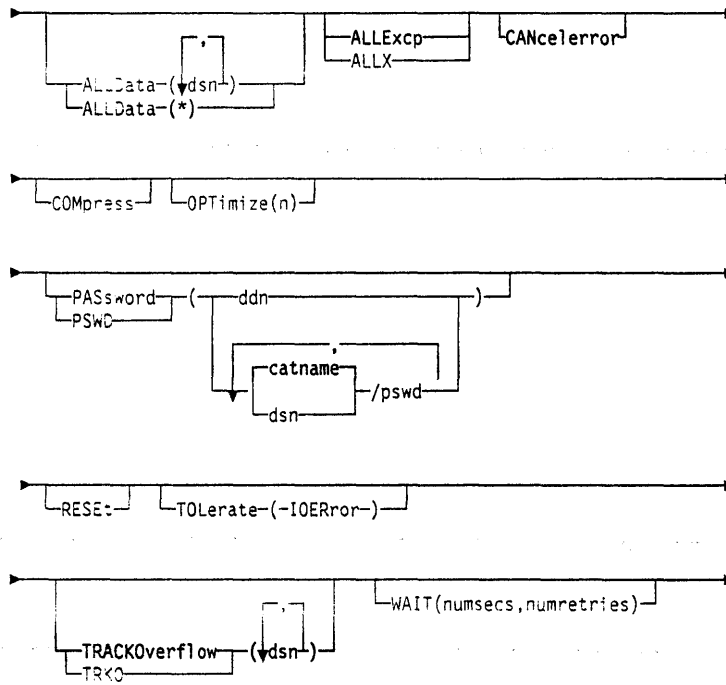
For information on using the DUMP command, see *Data Facility Data Set Services: User's Guide*.

DUMP Command Syntax for FULL or TRACKS



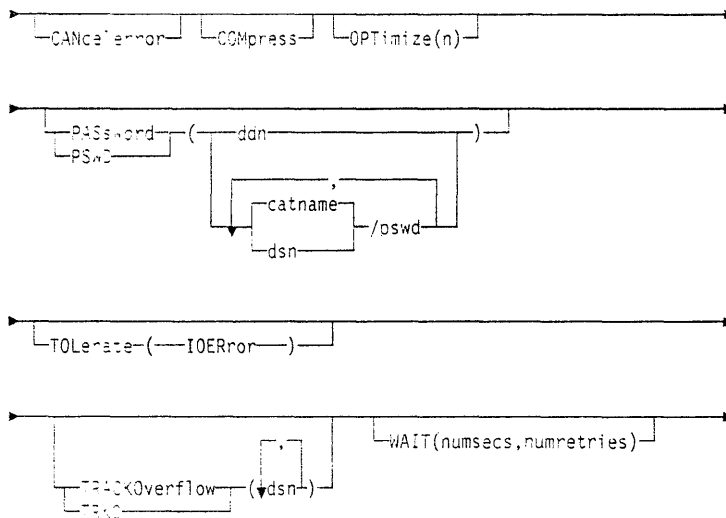
Optional Keywords Used with FULL

A :

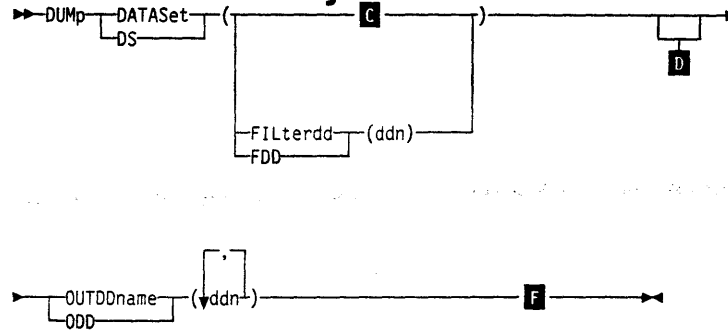


Optional Keywords Used with TRACKS

B :

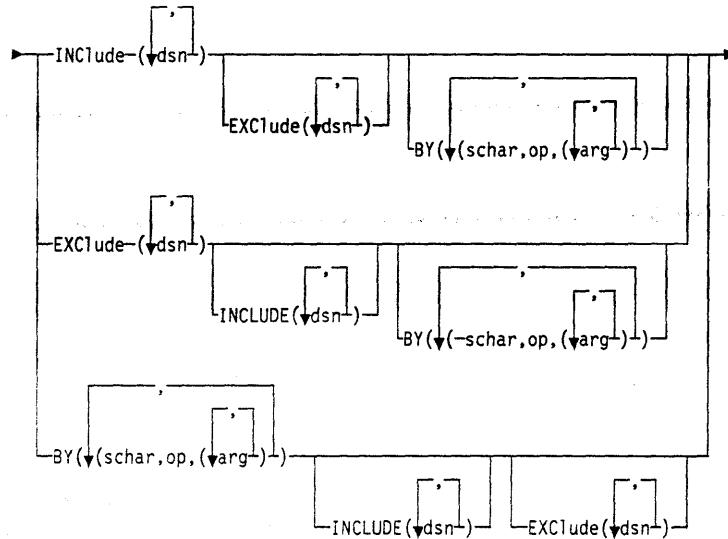


Logical Data Set DUMP Command Syntax



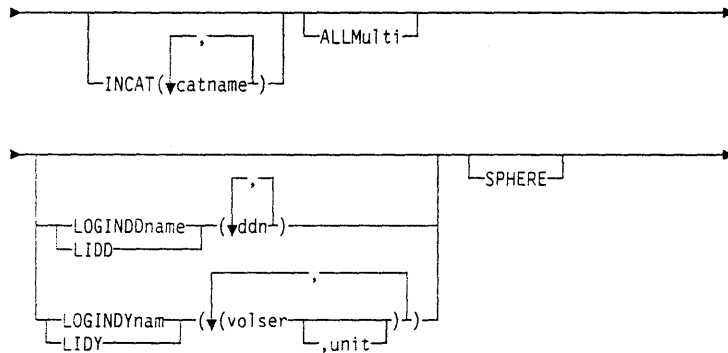
Additional Keywords Used with DATASet

C:

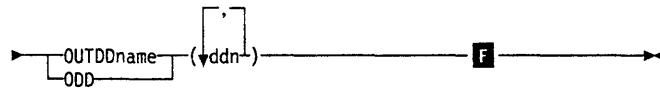
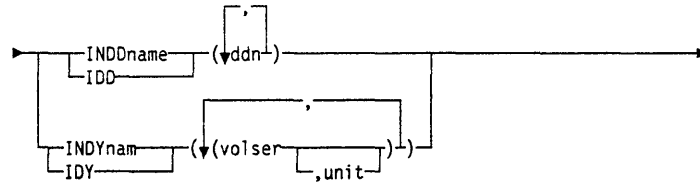
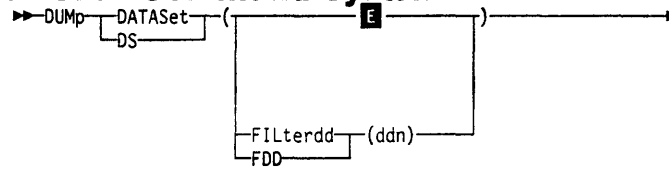


Optional Logical Data Set DUMP Keywords

D:

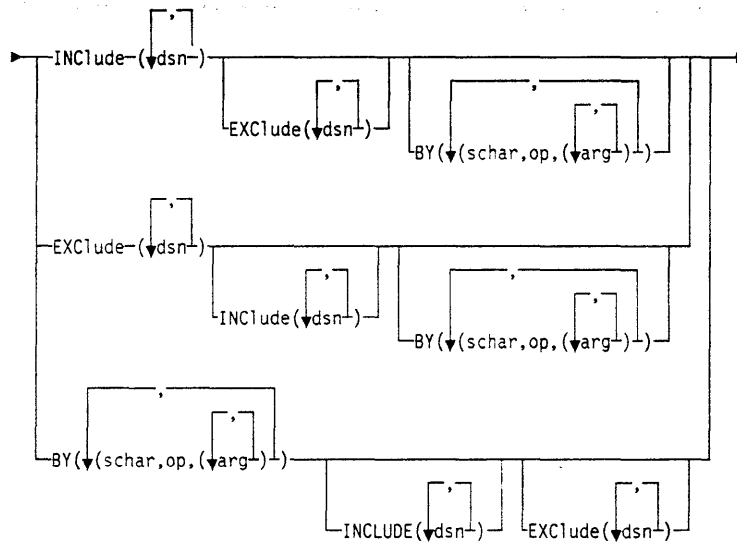


Physical Data Set DUMP Command Syntax



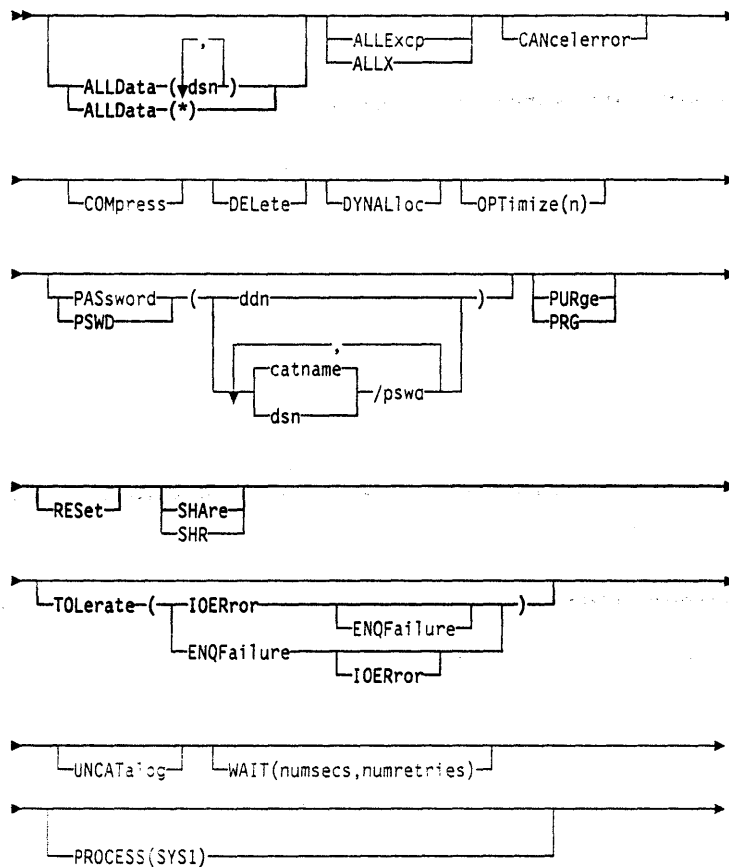
Additional Keywords Used with DATASet

E :



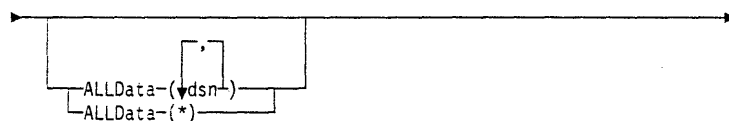
Optional Keywords for Logical and Physical Data Set DUMP

F



Explanation of DUMP Command Keywords

ALLData



applies to full and data set DUMP operations.

dsn

specifies the name of a SAM (DSORG = PS or PSU) or PAM (DSORG = PO or POU) data set or a data set with a null DSORG field for which all allocated space is to be dumped.⁶

specifies that all allocated space is to be dumped for:

- all SAM and PAM data sets

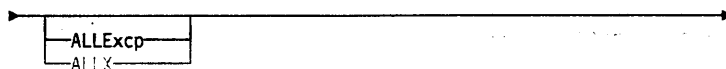
and

⁶ A data set can have data beyond the limits described by the DSCB DS1LSTAR field or it can contain a null DS1LSTAR field. For example, JES2/JES3 data sets can have these characteristics, as can CICS journal and MSVCJRNL data sets.

- data sets with a null DSORG field that are selected in the DUMP operation and whose DS1LSTAR field is nonzero.

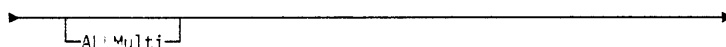
The DS1LSTAR field contains the last-used track and last block on that track in the form of a TTR.

ALLExcp



is an option for full and data set DUMP operations. It instructs DFDSS to dump all space allocated for data sets whose DSCB DS1LSTAR field is zero. This includes SAM and PAM data sets, and data sets with a null DSORG field.⁶

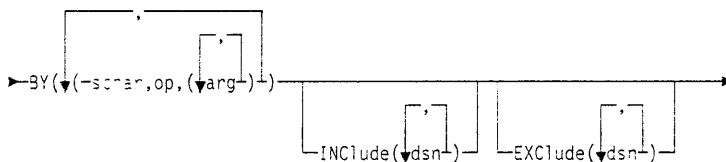
ALLMulti



specifies that cataloged, multivolume data sets residing on volumes missing from the LOGINDDNAME or LOGINDYNAM volume list are to be dumped in their entirety. To qualify for ALLMULTI processing, a data set must be cataloged and have at least one primary volume specified in the LOGINDDNAME or LOGINDYNAM list. Data from the missing volume(s) will be dumped for applicable data sets.

Note: The ALLMULTI keyword is valid only when it is specified in conjunction with LOGINDDNAME or LOGINDYNAM.

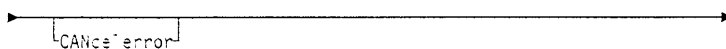
BY



specifies that the data sets selected up to this point by the processing of the INCLUDE and EXCLUDE keywords are to be further filtered. To select the data set, *all* BY criteria must be met. See the separate discussions of INCLUDE and EXCLUDE for information on how these keywords are specified.

Note: A data set is considered not cataloged if it is not cataloged in the standard order of search or if the catalog that the data set is cataloged in is not available.

CANcelerror

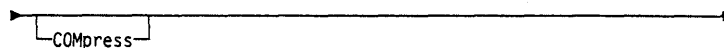


specifies that the DUMP is to be ended if a track-related permanent read error occurs. If this keyword is *not* specified and a permanent read error occurs, the track image record is flagged on the output volume as having an I/O error, and the DUMP continues. This track is cleared in a RESTORE operation. When CANCELERROR is specified, TOLERATE(IOERROR) is ignored.

Note: CANCELERROR has no effect on the following types of errors on a DASD volume:

- Equipment check
- Command reject
- Intervention required
- Busout parity.

COMpress

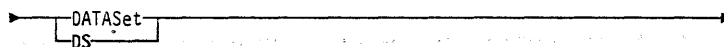


specifies that the dumped data is to be written in compressed form to the output medium. This decreases the space occupied by the dump data at the expense of increased processor and elapsed times.

Notes:

1. The DFDSS stand-alone restore program can restore any physical dump made by DFDSS as long as the physical dump is the first data set on the tape volume. The dump tapes may contain standard user labels.
2. If you have a tape drive with the compaction feature and you want to use hardware data compaction, you do not need to specify the **COMPRESS** keyword. If software data compression is desired, you do not need to specify **DCB=TRTCH=COMP** in the JCL.

DATASet

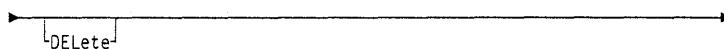


specifies a data set DUMP operation, using filtering. See Chapter 2, "Choosing the Data Sets You Want Processed—Filtering" on page 9 for an explanation of the filtering process used. Unless you use **DATASET** along with **ALLDATA** or **ALLEXCP**, only *used tracks* are dumped for SAM and PAM data sets and for data sets with a null **DSORG** field (for example, JES2/JES3 data sets). If the free space map in the VTOC is invalid, all tracks are dumped.

Notes:

1. Either the **FILTERDD** or the **INCLUDE** keyword must be substituted when **DATASET** is selected.
2. VSAM catalogs are not supported for data set processing. If VSAM catalogs are processed, the results can be unpredictable.

DELeTe

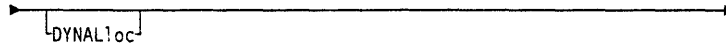


applies to physical and logical data set DUMP. For a *physical* data set DUMP, **DELETE** instructs DFDSS to delete expired single-volume, non-VSAM data sets that are successfully serialized and dumped. In addition, DFDSS is to uncatalog successfully deleted data sets. **DELETE** is ignored for VSAM data sets. Even if **PROCESS(SYS1)** is specified, **SYS1.VVDS** and **SYS1.VTOCIX** data sets cannot be dumped and deleted.

For a *logical* data set DUMP, **DELETE** can be used to delete expired single- and multivolume VSAM and non-VSAM data sets. Unmovable data sets can also be deleted. User catalogs and VSAM alternate indexes cannot be deleted. *Unexpired* source data sets are deleted only if you also specify **PURGE**.

To delete or scratch a password-protected data set, the operator must supply the password for DADSM scratch password checking.

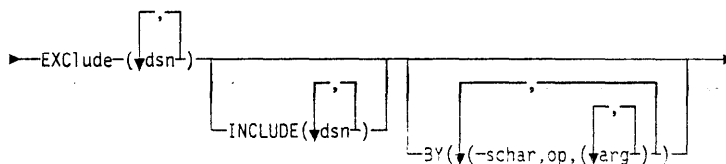
Note: Do not specify **SHARE** if you specify **DELETE**.

DYNALloc

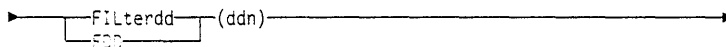
specifies dynamic allocation, instead of enqueue, to serialize the use of data sets. This allows cross-system serialization in a JES3/MVS environment.

Consider:

- The serialization is of value only when the dynamic allocation/JES3 interface is not disabled.
- SVC 99 is used to serialize data sets.
- Run time increases when you use DYNALLOC to serialize data sets (as opposed to enqueue) because overhead is involved in dynamic allocation and serialization across multiple processors.

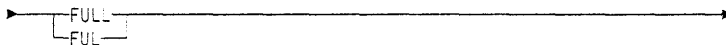
EXClude**dsn**

specifies the name of a data set to be excluded from the data sets selected by INCLUDE. Either a fully- or a partially-qualified data set name can be used. See the separate discussions of INCLUDE and BY for information on how these keywords are specified.

FILterdd

specifies the name of the DD statement that identifies the sequential data set or member of a partitioned data set that contains the filtering criteria to use. The filtering criteria are in the form of card-image records, in DFDSS command syntax, containing the INCLUDE, EXCLUDE, and BY keywords that complete the DUMP command syntax.

Note: You must use FILTERDD when you have more than 255 entries in the INCLUDE, EXCLUDE, or BY list of subkeywords.

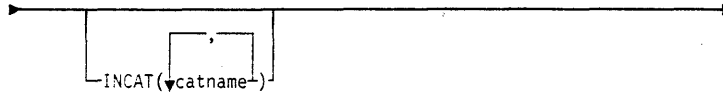
FULL

specifies that an entire DASD volume is to be dumped. This is the default. Unallocated tracks are not dumped. Unless FULL is specified with ALLDATA or ALLEXCP, only *used tracks* are dumped for SAM and PAM data sets and for data sets with a null DSORG field (for example, JES2/JES3 data sets). If the free space map in the VTOC is invalid, all tracks are dumped. Used tracks consist of the tracks from the beginning of the data set to the last-used track (as indicated in the DS1LSTAR field of the Format 1 DSCB).

Notes:

1. You cannot specify SHARE or TOL(ENQF) for FULL operations.
2. VSAM catalogs are not supported for full volume processing. If VSAM catalogs are processed, the results can be unpredictable.

INCAT

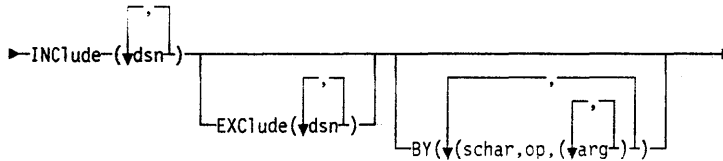


specifies one or more user catalogs to be searched before the standard order of search for locating data sets. STEPCAT and JOBCAT are not allowed for processing SMS-managed data sets. INCAT allows you to identify specific source catalogs. To specify INCAT you must have the proper RACF facility class authorization (refer to *Resource Access Control Facility (RACF) Security Administrator's Guide* for additional information).

catname

specifies a fully-qualified catalog name.

INCLUDE



dsn

specifies the name of a data set eligible to be dumped. Either a fully- or a partially-qualified data set name can be used. See "Filtering by Data Set Names" on page 10. If INCLUDE is omitted (but EXCLUDE or BY is specified) or if INCLUDE(**) is specified, *all* data sets are eligible to be selected for dumping. See the separate discussions of EXCLUDE and BY for information on how these keywords are specified.

INDDname

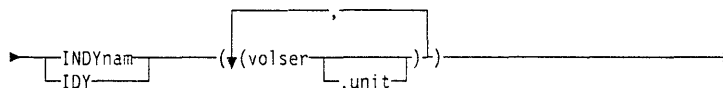


ddn

specifies the name of the DD statement that identifies the input volume to be dumped. To assure correct processing, each of the DD statements corresponding to a DDNAME (*ddn*) must identify only one volume serial number.

Note: Only one *ddn* can be specified for INDDname when you use full or tracks DUMP. One or more are allowed for a physical data set DUMP.

INDYnam



specifies that volume(s) to be dumped are to be dynamically allocated for a full, tracks, or physical data set dump.

Notes:

1. If the volume is real (not virtual), it must be mounted and online.
2. Only one volume is allowed for a full or tracks DUMP; one or more volumes are allowed for a physical data set DUMP.
3. Using INDYNAM instead of DD statements to allocate DASD volumes does not appreciably increase run time and permits easier coding of JCL and command input.

volser

specifies the volume serial number of a DASD volume to be dumped.

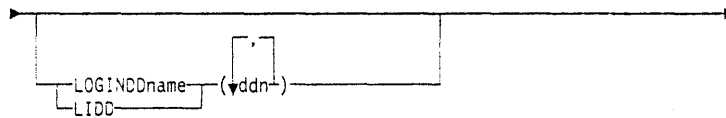
unit

specifies the device type of a DASD volume to be dumped. This keyword is optional.

Notes:

1. If either INDDNAME or INDYNAM is specified, *physical* processing is used to perform the dump.
2. If both INDDNAME and INDYNAM are omitted, a *logical* data set dump is performed.
3. A logical data set dump is also performed if you specify LOGINDDNAME or LOGINDYNAM.

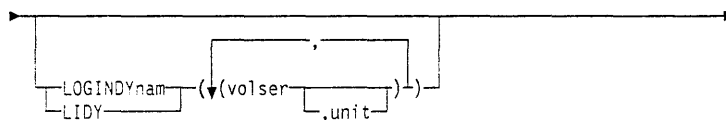
LOGINDDname



ddn

specifies the name of the DD statement that identifies the input volume that contains the data set(s) for a logical dump. To ensure correct processing, each of the DD statements corresponding to a DDNAME (*ddn*) must identify only one volume serial number.

LOGINDYnam



specifies that volume(s) that contain the data set(s) to be dumped using logical processing are to be dynamically allocated.

Note: If the volume is real (not virtual), it must be mounted and online.

volser

specifies the volume serial number of a DASD volume to be dumped.

unit

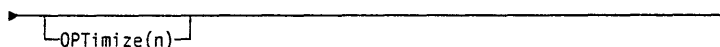
specifies the device type of a DASD volume to be dumped. This keyword is optional.

DUMP Command

Notes:

1. If **LOGINDDNAME** or **LOGINDYNAM** is specified, DFDSS uses logical processing to perform the dump. Logical processing is also used if no input volume is specified.
2. A multivolume data set that has extents on volumes not specified with **LOGINDDNAME** or **LOGINDYNAM** will not be dumped if you omit **ALLMULTI**.

OPTimize



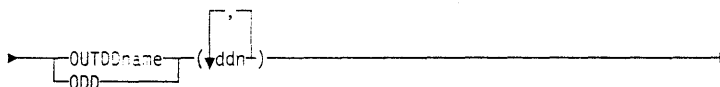
specifies the number of tracks to be read at a time, as follows:

- If *n* is 1, DFDSS reads one track at a time.
- If *n* is 2, DFDSS reads two tracks at a time.
- If *n* is 3, DFDSS reads five tracks at a time.
- If *n* is 4, DFDSS reads one cylinder at a time.

If **OPTIMIZE** is not specified, **OPTIMIZE(1)** is the default. **OPTIMIZE(2)**, (3), and (4) can be processed efficiently for data sets with no track overflow records. Specifying **OPTIMIZE (2)**, (3), or (4) reduces the time for a dump. Notice that this keyword uses more real and virtual storage. It also keeps the channel busy for longer blocks of time.

Note: There is no performance gain on mass storage virtual volumes by specifying **OPTIMIZE**.

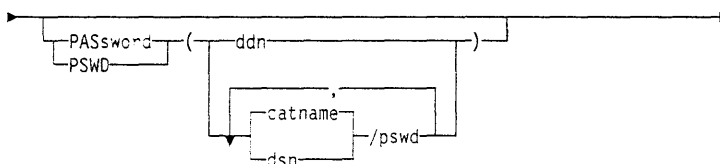
OUTDDname



ddn

specifies the name of the DD statement that identifies the (output) dump data set. This data set can be on a tape or a DASD volume. Up to 255 DDNAMEs can be specified; that is, up to 255 dump copies can be made.

PASsword



specifies the passwords DFDSS is to use for password-protected data sets for all **DUMP** operations. (Password checking is bypassed for RACF-protected data sets.) This is required only if:

- You do not have the required volume-level RACF **DASDVOL** or RACF **DATASET** access.
- The installation authorization exit does not bypass the checks.
- You do not want to be prompted for the password. For DFDSS authorization checking, see the **DELETE** keyword for the **DUMP** command.

For VSAM data sets, password checking can be done only at the cluster level.

ddn

specifies the name of the DD statement that identifies the sequential data set or member of a partitioned data set that contains data set names and their passwords. This data set must contain card-image records in DFDSS command syntax format.

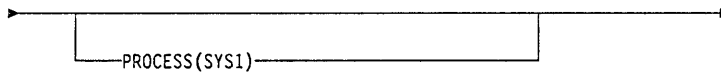
dsn/pswd

dsn is a fully-qualified data set name. *pswd* is its password. If no password follows the slash (/), *dsn* is treated as though it were *ddn*.

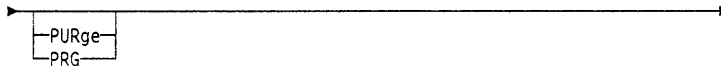
catname/pswd

catname is a fully-qualified catalog data set name. *pswd* is its password. This is supported for VSAM data sets only.

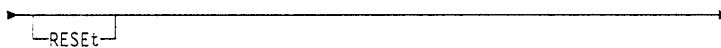
Note: Printing of actual data set passwords specified in your input command stream is suppressed in the SYSPRINT output.

PROCESS

specifies that when the DELETE keyword is also specified, DFDSS is to dump and delete data sets with the high-level qualifier of SYS1. However, even if PROCESS(SYS1) is specified, SYS1.VVDSs and SYS1.VTOCIXs are not processed. Refer to *Resource Access Control Facility (RACF) Security Administrator's Guide* for additional information.

PURge

for a data set DUMP, instructs DFDSS to delete unexpired data sets that are dumped successfully. This keyword is valid only when DELETE has been specified.

RESEt

specifies that the data-set-changed flag (DS1DSCHA field) is to be reset for all data sets serialized and successfully dumped. This applies to both a full DUMP and a data set DUMP.

Note: Do not specify SHARE if you specify RESET. You might not want to specify RESET if you use a storage management program, such as the Data Facility Hierarchical Storage Manager (DFHSM).

SHAre

specifies that DFDSS is to share the data sets to be dumped for read access with other programs. Do not specify DELETE, RESET, or UNCATALOG if you specify SHARE. Use SHARE carefully to ensure that the contents of the dumped copy of the data set are valid.

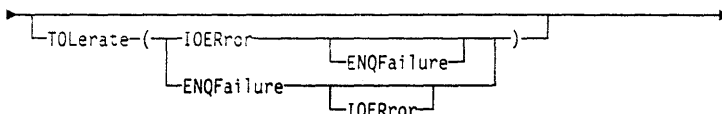
SHARE and FULL are mutually exclusive; you cannot specify these keywords together.

SPHERE



is an option for a logical data set DUMP. **SPHERE** specifies that for any VSAM cluster dumped DFDSS must also dump all associated AIX clusters and paths. Individual sphere components need not be specified, only the base cluster name.

TOLerate



ENQFailure

specifies that data sets are to be processed even though shared or exclusive access fails.

TOL(ENQF) and FULL are mutually exclusive; you cannot specify these keywords together.

For more information on TOL(ENQF), see Appendix B, "Serialization and Authorization Checking" on page 165.

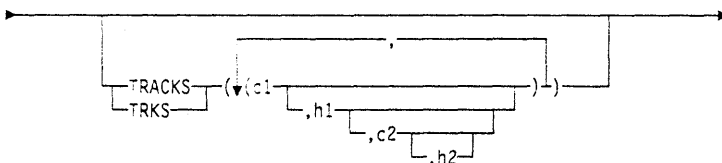
IOERror

specifies that DFDSS is to continue processing even though input errors occur, but is to end after 100 errors. This applies only to input errors and only to equipment check and busout parity.

Notes:

1. TOL(IOERror) is ignored if CANcelerror is specified.
2. If a permanent read error occurs, the track image record is flagged on output as having an I/O error and the DUMP continues.
3. This track is cleared in a RESTORE operation.

TRACKS



specifies ranges of tracks to be dumped. When you restore the data, this entire range or its subset must be specified with the RESTORE command.

c1,h1

specifies the cylinder and head number of the beginning of the range. Specify hexadecimal numbers as X'c1' or X'h1'.

c2,h2

specifies the cylinder and head number of the end of the range. Specify hexadecimal numbers as X'c2' or X'h2'.

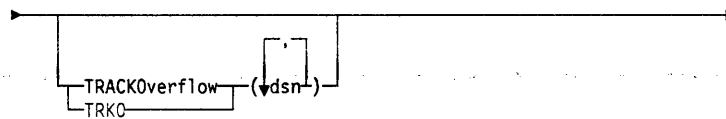
Notes:

1. The c2 must be greater than or equal to c1.
2. If c2 equals c1, h2 must be greater than or equal to h1.

DFDSS verifies that the range is within the limits of the device. If you do not specify all four values for a range, DFDSS provides the missing values unless the omitted value causes a syntax error. No intervening values can be omitted. For example:

Specified	Results
<i>None</i>	Syntax error
<i>c1</i>	c1,0,c1,maximum head number
<i>c1,h1</i>	c1,h1,c1,maximum head number
<i>c1,h1,c2</i>	c1,h1,c2,maximum head number
<i>c1,,c2</i>	Syntax error
<i>,h1</i>	Syntax error

TRACKOverflow

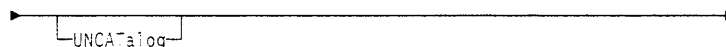


dsn

specifies the name of data set(s) in which track overflow records are present but not indicated by the Format 1 DSCB. The TRACKOVERFLOW keyword is not needed for data sets indicated as having overflow records.

Note: When you perform a logical DUMP, the use of TRACKOVERFLOW marks the data set on the dump tape. This allows logical RESTORE to correctly track-pack the data set if required.⁷ The target's Format 1 DSCB will be set to indicate overflow records are present when the data set is restored. If OPTIMIZE(2), (3), or (4) is specified, specifying TRACKOVERFLOW can result in fewer EXCPs on the source DASD device.

UNCATALOG



applies to physical and logical data set DUMP.

For a *physical* data set DUMP, UNCATALOG instructs DFDSS to uncatalog any single-volume, non-VSAM, cataloged data sets successfully dumped from the current volume.

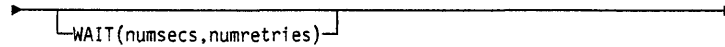
Note: A data set with a high-level qualifier of SYS1 cannot be uncataloged.

For a *logical* data set DUMP, UNCATALOG instructs DFDSS to uncatalog any successfully dumped single- or multivolume non-VSAM data sets that are currently cataloged. (For VSAM data sets, use the DELETE keyword.)

Note: UNCATALOG is ignored for logical and SMS-managed data sets.

⁷ Track-packing is a technique used by DFDSS to build target tracks for any DASD device by using input physical record information. For example, when copying a data set from a DASD with smaller track capacity to a DASD with larger track capacity, DFDSS reads full tracks from the source and copies the physical records from one or more input tracks into a single output track of the target data set.

WAIT



For a *volume*, *tracks*, or *physical* data set DUMP, WAIT specifies to DFDSS the length of a wait and the number of retries to obtain control of a volume or data set.

numsecs

is a number from 1 to 255 that specifies the interval, in seconds, between retries.

numretries

is a 1- or 2-digit number that specifies the number of times DFDSS must retry to gain control of a volume or data set.

For a *logical data set* DUMP, WAIT specifies to DFDSS the length of wait and the number of passes to be made through the list of selected data sets to obtain control of a data set.

numsecs

is a number from 1 to 255 that specifies the interval, in seconds, to wait before attempting another pass through the list of selected data sets.

numretries

is a 1- or 2-digit number that specifies the number of passes to make through the list of selected data sets in an attempt to obtain control of a data set.

The default for *numsecs,numretries* is (2,2), which specifies two retries at 2-second intervals. If you do not want to wait for a resource, specify 0 for either *numsecs* or *numretries*.

For a data set COPY or logical data set DUMP, the WAIT keyword has a different meaning when: (1) data sets are being serialized, (2) multiple data sets are being processed, and (3) WAIT(0,0) is not specified. In this case, DFDSS makes multiple passes through the list of selected data sets. On each pass, DFDSS processes the data sets that (1) can be serialized without waiting for the resource and (2) were not processed before. At the end of a pass, if none of the data sets could be processed without waiting for a resource, then, in the next pass, at the first occurrence of a data set that was not processed, a WAIT will be issued. That data set and the remainder of the list will be processed if possible. The above procedure will be repeated until all data sets are processed or the WAIT limits are reached. For example, if WAIT(3,10) is specified and 5 data sets are left to be processed, up to 10 passes are made. On each pass, an unprocessed data set is waited upon for 3 seconds. Thus, only a 30-second maximum will ever be WAITed, not 150 (5 times 3 times 10).

Data Integrity Considerations for Full or Tracks DUMP

For a full or tracks DUMP, DFDSS serializes the VTOC to preclude DADSM functions such as ALLOCATE, EXTEND, RENAME, and SCRATCH from changing the contents of the VTOC on the volume during the DUMP. Data sets are *not* serialized on these full or tracks operations. Therefore, some data sets might be opened by other jobs during the DUMP. The net result might be that partially updated data sets are dumped. For mountable devices, you can minimize this possibility by doing the following:

1. Specify PRIVATE and DEFER in the DD statement.
2. Specify unit affinity on a second DD statement that defines a mountable device.

3. Specify a volume count in the VOLUME keyword of a DD statement that is greater than the number of mountable devices to be allocated.
4. Take the dump when there is low system activity.

Full data integrity can always be guaranteed by performing DUMP operations by data set only.

Product Implementation Information

Format of the Output Data Set

For a physical DUMP, each logical volume of a dump data set contains data in the following sequence:

1. Volume header record: identifies and contains data pertinent to the whole volume and identifies the type of operation that created the dump.
2. Map record or records: maps the tracks that were dumped.
3. Track 0: dump of track 0 of cylinder 0.
4. VVDS track records if VSAM data sets exist on the volume and were dumped.
5. VTOC track records.
6. Data track records, including VVDS if it was part of the dump. Item 6 is repeated for each data set.
7. Two volume trailer records: identifies the end of the data for the DASD volume.

For a logical DUMP, the format of the output data set is:

1. Tape header record.
2. List of potential data sets record.
3. Data set header record.
4. Sphere information record (if the SPHERE keyword was specified and this data set is part of a sphere).
5. Volume header record.
6. Data track record(s) (one for each track of the data set on this volume). Items 5 and 6 are repeated for each volume of the data set.
7. Two data set trailer records. Items 3 through 7 are repeated for each data set.

See Appendix E, "Data Area Descriptions" on page 220 for a detailed mapping of the ADRBMB mapping macro and the ADRTAPB mapping macro.

End of Product Implementation Information

Examples of Full and Tracks DUMP Operations

In the following example, DASD volume 111111 is to be dumped to the first data set of standard label tape volumes TAPE01 and TAPE02.

The command input to be substituted for a full and tracks dump are shown below in Example 1A and Example 1B, respectively. To restore the same volume, refer to Examples 1. 1A, and 1B of the RESTORE command.

Example 1:

```
//JOB1 JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=A
//DASD DD UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//TAPE DD UNIT=(3480,2)VOL=SER=(TAPE01,TAPE02),
// LABEL=(1,SL),DISP=(NEW,CATLG),DSNAME=USER2.BACKUP
//SYSIN DD *
command input (See Examples 1A and 1B below)
/*
```

Example 1A: A Full DUMP Operation

```
DUMP INDDNAME(DASD) OUTDDNAME(TAPE)
```

Example 1B: A Tracks DUMP Operation

```
DUMP TRACKS(1,0,1,5) INDDNAME(DASD) -
OUTDDNAME(TAPE)
```

Examples of Physical Data Set DUMP Operations

Example 2 shows specified data sets on DASD volumes 111111 and 222222 being dumped to the first data set of standard label tape volume TAPE02.

Examples 2A through 2G below complement Examples 2A through 2D in the RESTORE section, in any combination; for example, the dump tape produced in Example 2C can be used as the input tape for Example 2A under the RESTORE command.

Example 2:

```
//JOB1 JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=A
//DASD1 DD UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//DASD2 DD UNIT=3380,VOL=(PRIVATE,SER=222222),DISP=OLD
//TAPE DD UNIT=3480,VOL=SER=TAPE02,
// LABEL=(1,SL),DISP=(NEW,CATLG),DSNAME=USER2.BACKUP
//SYSIN DD *
command input (see Examples 2A, 2B, ... 2G that follow)
/*
```

Example 2A: Using the INCLUDE Subkeyword

```
DUMP INDDNAME(DASD1,DASD2) OUTDDNAME(TAPE) -
DATASET(INCLUDE(USER2.**,USER3.*))
```

Example 2B: Using the INCLUDE and EXCLUDE Subkeywords

```
DUMP INDDNAME(DASD1,DASD2) OUTDDNAME(TAPE) -
DATASET(INCLUDE(USER2.**,USER3.*) -
EXCLUDE(USER2.**.REP))
```

Example 2C: Using the INCLUDE, EXCLUDE, and BY Subkeywords

```
DUMP INDDNAME(DASD1,DASD2) OUTDDNAME(TAPE) -
DATASET(INCLUDE(USER2.**,USER3.*) -
EXCLUDE(USER2.**.REP) -
BY((DSCHA,EQ,1)))
```

Example 2D: With Filtering Data in a Data Set

```
DUMP INDDNAME(DASD1,DASD2) OUTDDNAME(TAPE) -  
      DATASET(FILTERDD(A1))
```

Note: The following DD statement must be added to the JCL shown above:

```
//A1 DD DSNAME=USER2.FILTER,DISP=SHR
```

This cataloged data set (USER2.FILTER) contains three card-image records. The information shown is positioned in columns 2 through 72 of each record:

```
INCLUDE(USER2.** ,USER3.*) -  
EXCLUDE(USER2.**.REP) -  
BY((DSCHA,EQ,1))
```


Example 2E: With Passwords in the Input Stream

```
DUMP INDDNAME(DASD1,DASD2) OUTDDNAME(TAPE) -
      DATASET(INCLUDE(USER2.**,USER3.*)) -
      PASSWORD(USER2.ABC.DEF/PSWD1,USER2.XYZ/PSWD2)
```

Example 2F: With Passwords in a Data Set

```
DUMP INDDNAME(DASD1,DASD2) OUTDDNAME(TAPE) -
      DATASET(INCLUDE(USER2.**,USER3.*)) -
      PASSWORD(PDD)
```

statement must be added to the JCL shown above:

```
//PDD DD DSNAME=USER2.PASSWORD,DISP=SHR
```

This cataloged data set (USER2.PASSWORD) contains a single card-image record. The information shown is positioned in columns 2 through 72:

```
USER2.ABC.DEF/PSWD1,USER2.XYZ/PSWD2
```

Example 2G: Wait for Data Sets if They or Other Data Sets with the Same Name Are In Use by Other Jobs

```
DUMP INDDNAME(DASD1) OUTDDNAME(TAPE) -
      DATASET(INCLUDE(**)) -
      WAIT(1,99)
```

If a data set is in use, DFDSS waits for one second, then tries to gain access to the resource again. This is done as many as 99 times for each data set.

Examples of Logical Data Set DUMP Operations**Example 1: Dumping data sets that are constantly in use**

```
//JOB2 JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=A
//DASD1 DD UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//DASD2 DD UNIT=3380,VOL=(PRIVATE,SER=222222),DISP=OLD
//TAPE DD UNIT=3480,VOL=SER=TAPE02,
// LABEL=(1,SL),DISP=(NEW,CATLG),DSNAME=USER2.BACKUP
//SYSIN DD *
DUMP INDDNAME(DASD1) OUTDDNAME(TAPE) -
      DATASET(INCLUDE(**)) TOL(ENQF) WAIT(0,0)
/*
```

DFDSS does not wait (WAIT(0,0)) if a data set is in use. Instead, it processes the data set without serialization or enqueueing (TOL(ENQF)).

Note: In order to dump an open VSAM data set, you must specify the fully-qualified data set name in addition to TOL(ENQF).

Example 2: Dumping a User Catalog and Its Aliases: To dump a user catalog, you perform a logical data set dump with the fully-qualified user catalog name as the data set name. No filtering is allowed. If the user catalog has any aliases, the aliases are automatically dumped.

```
//JOB2    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADDRSSU
//SYSPRINT DD   SYSOUT=A
//DASD1   DD     UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//TAPE    DD     UNIT=3480,VOL=SER=TAPE02,
// LABEL=(1,SL),DISP=(NEW,CATLG),DSNAME=USER2.BACKUP
//SYSIN   DD     *
DUMP OUTDDNAME(TAPE) -
      DS(INCLUDE(MY.USER.CAT))
/*
```

Example 3: A Logical Data Set DUMP with Catalog Filtering

```
//JOB3    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADDRSSU
//TAPE    DD     UNIT=3480,VOL=SER=TAPE04,
// LABEL=(1,SL),DISP=(NEW,CATLG),DSN=USER3.BACKUP
//SYSPRINT DD   SYSOUT=A
//SYSIN   DD     *
DUMP OUTDD(TAPE) -
      DS(INCL(USER1.**))
/*
```

All data sets cataloged in the standard search order whose first-level qualifier is USER1 are to be dumped. Because some of these data sets are multivolume, source DASD volumes are not specified, resulting in data set selection by catalog.

Example 3 can be modified as follows to dump only data sets changed since the last backup. In addition, data sets that end with a qualifier of LISTING are not to be dumped (EXCL(**.LISTING)).

```
//SYSIN   DD     *
DUMP OUTDD(TAPE) -
      DS(INCL(USER1.**)) -
      EXCL(**.LISTING) -
      BY((DSCHA EQ 1)))
/*
```

Example 4: A Logical Data Set DUMP with VTOC Filtering

```
//JOB5    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADDRSSU
//SYSPRINT DD   SYSOUT=A
//DASD1   DD     VOL=SER=338001,UNIT=3380,DISP=OLD
//TAPE    DD     UNIT=3480,VOL=SER=TAPE04,
// LABEL=(1,SL),DISP=(NEW,CATLG),DSN=USER3.BACKUP
//SYSIN   DD     *
DUMP DATASET(INCLUDE(USER3.**)) -
      LOGINDDNAME(DASD1) -
      OUTDDNAME(TAPE) -
      DELETE PURGE
/*
```

All data sets on volume 338001 whose first qualifier is USER3 are included in a logical data set DUMP. DFDSS filters using the VTOC of volume 338001. Catalogs are also used, as needed, for multivolume and VSAM data sets.

Example 4 can be modified as follows to dynamically allocate volume 338001 and to perform ALLMULTI processing. All single volume data sets on volume 338001 are included in a logical data set DUMP. ALLMULTI specifies that a cataloged data set residing on volumes 338001, 338003, and 338005 will be dumped even though 338003 and 338005 are missing from the LOGINDYNAM volume list.

```
//JOB6 JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRSSU
//SYSPRINT DD SYSOUT=A
//TAPE DD UNIT=3480,VOL=SER=TAPE04,
// LABEL=(1,SL),DISP=(NEW,CATLG),DSN=USER3.BACKUP
//SYSIN DD *
DUMP DATASET(INCLUDE(**)) -
ALLMULTI -
LOGINDYNAM(338001) -
OUTDDNAME(TAPE) -
DELETE PURGE
/*
```

Example 5: A Logical Data Set DUMP for Storage Management Subsystem (SMS)

```
//MYJOB JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRSSU
//SYSPRINT DD SYSOUT=A
//TAPE DD UNIT=3480,VOL=SER=TAPE04,
// LABEL=(1,SL),DISP=(NEW,CATLG),DSN=USER3.BACKUP
//SYSIN DD *
DUMP LOGINDYNAM(338001) -
ALLMULTI -
DATASET(INCLUDE(**)) -
OUTDDNAME(TAPE) -
DELETE
/*
```

A volume is backed up for converting to or from SMS management by using a logical data set DUMP.

PRINT Command

With the PRINT command, you can print:

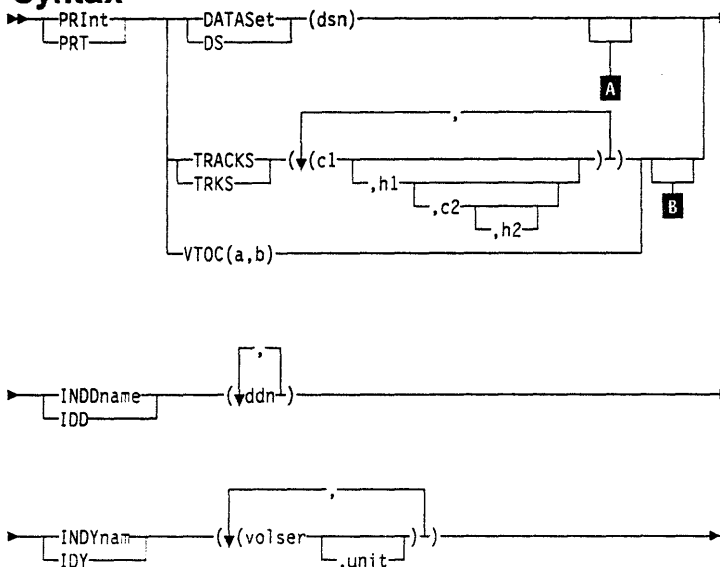
- A single-volume non-VSAM data set, as specified by a fully-qualified name. You must specify the volume where the data set resides, but you do not need to specify the range of tracks it occupies.
- A single-volume VSAM data set component (not cluster). The component name specified must be the name in the Format 1 DSCB in the VTOC, not the name in the catalog.
- Ranges of tracks.
- All or part of the VTOC. The VTOC location need not be known.

Note: In order to print a multivolume data set, multiple PRINT commands with the appropriate INDD/INDY keywords must be used.

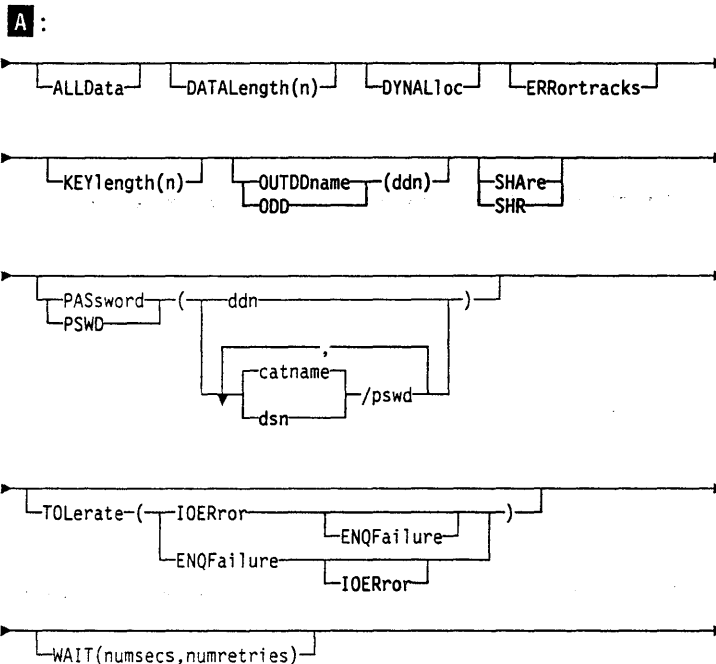
Unless ALLDATA is specified, only the used space is printed for SAM or PAM data sets or data sets with a null DSORG field.

If an error occurs in reading a record, DFDSS attempts to print the record in error. You can print all requested tracks or just a subset of the tracks that have data checks. For more information about authorization checking, see Appendix B, "Serialization and Authorization Checking" on page 165.

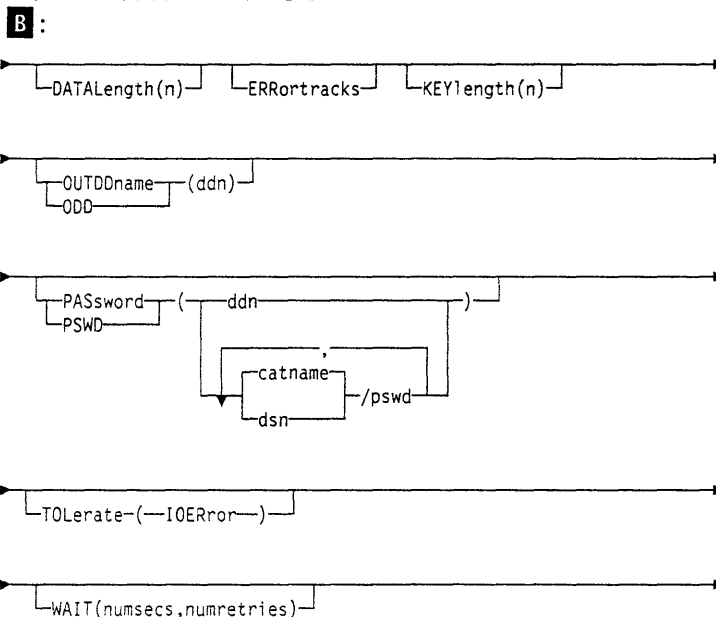
PRINT Command Syntax



Optional Keywords with DATASET

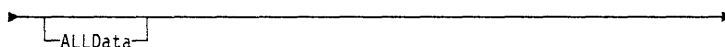


Optional Keywords with TRACKS or VTOC



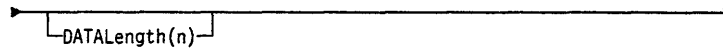
Explanation of PRINT Command Keywords

ALLData



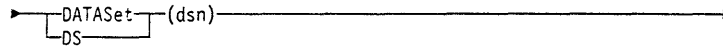
specifies, when the DATASET keyword is also specified, that *all* allocated space in the data set is to be printed.

DATALength



n
specifies the logical length, in decimal format, of the data portion of a record. It is used only if the count field of a record on any track has a data check.

DATASet

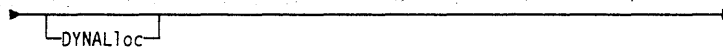


dsn
specifies the fully-qualified name of the data set to be printed. The data set is printed in logical sequence.

Note: Data set filtering is not allowed with the PRINT command.

VSAM catalogs are not supported for data set processing. If VSAM catalogs are processed, the results can be unpredictable.

DYNALloc

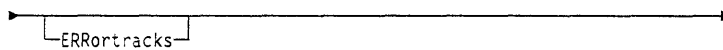


specifies dynamic allocation, instead of enqueue, to serialize the use of data sets. This allows cross-system serialization in a JES3/MVS environment.

Consider:

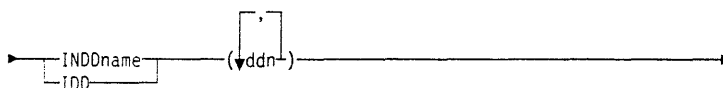
- The serialization is of value only when the dynamic allocation/ JES3 interface is not disabled.
- SVC 99 is used to serialize data sets.
- Run time increases when you use DYNALLOC to serialize data sets (as opposed to enqueue) because overhead is involved in dynamic allocation and serialization across multiple processors.
- Using INDYNAM instead of DD statements to allocate DASD volumes does not appreciably increase run time and permits easier coding of JCL and command input.

ERRortracks



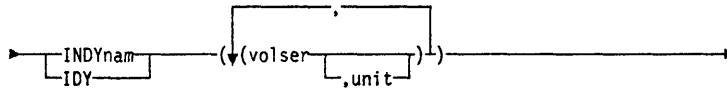
specifies that only tracks on which data checks occur are to be printed.

INDDname



ddn
specifies the name of the DD statement that identifies a volume that contains the data set, range of tracks, or the VTOC to be printed. If you want to print a multi-volume data set, you must print one volume at a time.

INDYnam



specifies that the volume that contains the data set, range of tracks, or the VTOC to be printed is to be dynamically allocated. If real (not virtual), the volume must be **mounted and online**.

volser

specifies the volume serial number of a DASD volume to be printed.

unit

specifies the device type of a DASD volume to be printed. This keyword is optional.

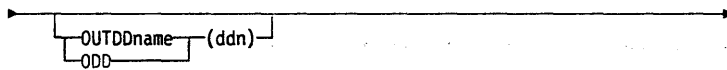
KEYlength



n

specifies the key length, in decimal format, of a record. It is used only if the count field of a record on any track has a data check.

OUTDDname



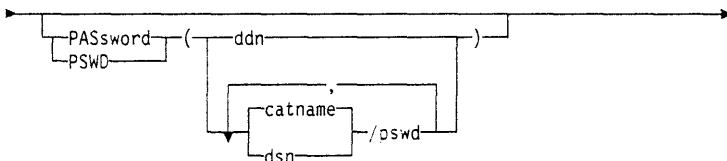
ddn

specifies the name of the DD statement that identifies the (output) print data set. Each of the DD statements corresponding to a DDNAME (*ddn*) must identify only one volume serial number. If this keyword is not specified, the default is SYSPRINT.

Notes:

1. If the DCB keyword LRECL is specified on the DD statement, it must be in the range of 84 to 137 inclusive. If BLKSIZE is specified, it must be at least four greater than the LRECL.
2. If an LRECL less than 84 is chosen, the return code is 8 and an error message is issued.
3. If the specified LRECL is greater than 137, LRECL and BLKSIZE are set to 137 and 141, respectively.

PASsword



specifies the passwords DFDSS is to use for password-protected data sets. (Password checking is bypassed for RACF-protected data sets.) This is required only if:

- You do not have the required volume-level RACF DASDVOL or RACF DATASET access.
- The installation authorization exit does not bypass the checks.
- You do not want to be prompted for the password.

For VSAM data sets, passwords are checked at the cluster level only.

Note: The **PASSWORD** keyword is not valid for a **PRINT VTOC** command.

ddn

specifies the name of the DD statement that identifies the sequential data set, or member of a partitioned data set, that contains data set names and their passwords. **This data set must contain card-image records in DFDSS command syntax format.**

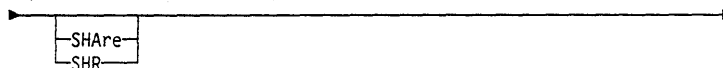
dsn/pswd

dsn is a fully-qualified data set name. *pswd* is its password. If no password follows the slash (/), *dsn* is treated as though it were *ddn*.

Printing of actual data set passwords specified in your input command stream is suppressed in the **SYSPRINT** output.

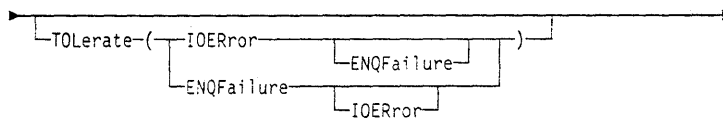
Passwords for password-protected VSAM data sets must be supplied by the operator. They cannot be supplied with the input stream. They are required on tracks **PRINT** operations only.

SHAre



specifies that **DFDSS** is to share, for read access with other programs, the data set that is to be printed.

TOLerate



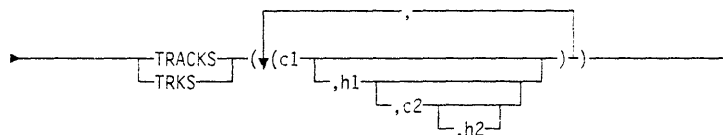
ENQFailure

specifies that data sets are to be processed even though shared or exclusive access fails. For more information on **TOL(ENQF)**, see Appendix B, "Serialization and Authorization Checking" on page 165.

IOERror

specifies that **DFDSS** is to continue processing even though I/O errors occur, but is to end after 100 errors.

TRACKS



specifies ranges of tracks to be printed.

c1,h1

specifies the cylinder and head number of the beginning of the range. Specify hexadecimal numbers as **X'c1'** or **X'h1'**.

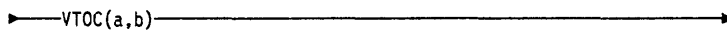
c2,h2

specifies the cylinder and head number of the end of the range. Specify hexadecimal numbers as **X'c2'** or **X'h2'**. The **c2** must be greater than or equal to **c1**. If **c2** equals **c1**, **h2** must be greater than or equal to **h1**.

DFDSS verifies that the range is within the limits of the device. If you do not specify all four values for a range, DFDSS provides the missing values unless the omitted value causes a syntax error. No intervening values can be omitted. For example:

Specified	Results
<i>None</i>	Syntax error
<i>c1</i>	<i>c1,0,c1,maximum head number</i>
<i>c1,h1</i>	<i>c1,h1,c1,maximum head number</i>
<i>c1,h1,c2</i>	<i>c1,h1,c2,maximum head number</i>
<i>c1,,c2</i>	Syntax error
<i>,h1</i>	Syntax error

VTOC



specifies that all or part of the VTOC is to be printed. Omitting the VTOC, DATASET, and TRACKS keywords causes the entire VTOC to be printed. Part of the VTOC can be printed by specifying:

VTOC(*a,b*)

where *a* and *b* are the relative track numbers (1 is the first track) of the first and last tracks to be printed. The value of *b* must be equal to or greater than the value of *a* and equal to or less than 65535 (X'FFFF'). Either of these numbers can be specified in decimal or hexadecimal. To specify a hexadecimal number, code X'nn'.

If only the first value (*a*) is specified, only that track is printed.

If the second value (*b*) is greater than the relative track number of the last physical track of the VTOC, DFDSS prints up to and including the last track of the VTOC. Therefore, another way of printing the entire VTOC would be to specify:

VTOC(1,X'FFFF')

Note: The PASSWORD keyword is not valid for a PRINT VTOC command.

WAIT



specifies to DFDSS the length of a wait in seconds and the number of retries to obtain control of a volume or a data set.

numsecs

is a number from 1 to 255 that specifies the interval, in seconds, between retries.

numretries

is a 1- or 2-digit number that specifies the number of times an attempt to gain control of a volume or data set is to be retried.

The default for *numsecs,numretries* is (2,2), which specifies two retries at 2-second intervals. If you do not want to wait for a resource, specify 0 for either numsecs or numretries.

Examples of PRINT Operations

Example 1: Print a Data Set

```
//JOB1    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADRDSSU
//SYSPRINT DD   SYSOUT=A
//DASD    DD     UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//SYSIN   DD     *
PRINT DATASET(USER2.COMPILE.LIST) INDDNAME(DASD) -
SHARE PASSWORD(USER2.COMPILE.LIST/PSWD1)
/*
```

Print a hard copy of data set USER2.COMPILE.LIST residing on volume 111111.

Example 2: Print a Range of Tracks

```
//JOB2    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADRDSSU
//SYSPRINT DD   SYSOUT=A
//DASD    DD     UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//SYSIN   DD     *
PRINT TRACKS(1,0,1,5) INDDNAME(DASD)
/*
```

Print a hard copy of tracks 0 through 5 from cylinder 1 on volume 111111.

Example 3: Print a Component of a VSAM Cluster

```
//JOB3    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADRDSSU
//SYSPRINT DD   SYSOUT=A
//SYSIN   DD     *
PRINT INDYNAM(338000) /* ALLOC VOL 338000 DYNAMICALLY */ -
DS(PARTS.VSAM1.INDEX) /* DATA SET THAT HAS BAD TRACK */ -
WAIT(0,0) /* DO NOT WAIT IF ENQ FAILS */ -
TOL(ENQF) /* IGNORE ENQ FAILURES */ -
PSWD(PARTS.VSAM1/USERPSWD) /* PASSWORD FOR CLUSTER */
/*
```

A component of a VSAM data set is to be printed. The printing is to proceed even if the component cannot be serialized (enqueued).

Example 4: Print a Data Set

```
//JOB4    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADRDSSU
//SYSPRINT DD   SYSOUT=*
//DD1     DD     VOL=SER=D8E040,UNIT=3380,DISP=OLD
//SYSIN   DD     *
PRINT INDD(DD1) DS(PRINT.SAMPLE)
/*
```

The output shown in Figure 6 on page 91 was produced from the control statement:
PRINT INDD(DD1) DS(PRINT.SAMPLE)

Note: If the last record on the track is a track overflow record, the count field of the record is followed by (+).


```

PAGE 0001 DFSS 5665-327 DATA FACILITY DATA SET SERVICES V2R5.0 89.067 16:08
PRINT INDD(DD1) DS(PRINT.SAMPLE) 150000
ADR101I RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'PRINT '
ADR109I RI01 (01), 89067 16:08:40 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR006I (001)-SETUP(01), 89067 16:08:40 PROCESSING BEGINS
*** TRACK(CCHH) 00010000 RO DATA 0000000000000000
COUNT 0001000001080100
0000 FFFFFFFF FFFFFFFF 0026D4C5 D4F0F140 40400000 0400D4C5 D4F0F240 40400000 *.....MEM01.....MEM02.....*
0020 0600FFFF FFFFFFFF FFFF0000 00000000 00000000 00000000 00000000 00000000 *.....*
0040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0060 TO 00FF SAME AS ABOVE
0100 00000000 00000000 *.....*
COUNT 0001000002080100 ALL KEY/DATA 00
COUNT 0001000003000000
COUNT 00010000040000A0
0000 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9F0F1 F2F3F4F5 *ABCDEFGHJKLMNPQRSTUVWXYZ012345*
0020 F6F7F8F9 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9F0F1 *6789ABCDEFGHIJKLMNPQRSTUVWXYZ01*
0040 F2F3F4F5 F6F7F8F9 C1C2C3C4 C5C6C7C8 C2C3C4C5 C6C7C8C9 D1D2D3D4 D5D6D7D8 *23456789ABCDEFGHIBCDEFGHIJKLMNPQ*
0060 D9E2E3E4 E5E6E7E8 E9F0F1F2 F3F4F5F6 F7F8F9C1 C2C3C4C5 C6C7C8C9 D1D2D3D4 *RSTUVWXYZ0123456789ABCDEFGHIJKLM*
0080 D0D6D7D8 D9E2E3E4 E5E6E7E8 E9F0F1F2 F3F4F5F6 F7F8F9C1 C2C3C4C5 C6C7C840 *NOPQRSTUVWXYZ0123456789ABCDEFGHI.*
COUNT 0001000005000000
COUNT 00010000060000F0
0000 C3C4C5C6 C7C8C9D1 D2D3D4D5 D6D7D8D9 E2E3E4E5 E6E7E8E9 F0F1F2F3 F4F5F6F7 *CDEFGHJKLMNPQRSTUVWXYZ01234567*
0020 F8F9C1C2 C3C4C5C6 C7C8C9D1 D2D3D4D5 D6D7D8D9 E2E3E4E5 E6E7E8E9 F0F1F2F3 *89ABCDEFGHIJKLMNPQRSTUVWXYZ0123*
0040 F4F5F6F7 F8F9C1C2 C3C4C5C6 C7C84040 C4C5C6C7 C8C9D1D2 D3D4D5D6 D7D8D9E2 *456789ABCDEFGHI..DEFGHIJKLMNQRS*
0060 E3E4E5E6 E7E8E9F0 F1F2F3F4 F5F6F7F8 F9C1C2C3 C4C5C6C7 C8C9D1D2 D3D4D5D6 *TUVWXYZ0123456789ABCDEFGHIJKLMNO*
0080 D7D8D9E2 E3E4E5E6 E7E8E9F0 F1F2F3F4 F5F6F7F8 F9C1C2C3 C4C5C6C7 C8404040 *PQRSTUVWXYZ0123456789ABCDEFGHI..*
00A0 C5C6C7C8 C9D1D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9F0F1 F2F3F4F5 F6F7F8F9 *EFGHIJKLMNPQRSTUVWXYZ0123456789*
00C0 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9F0F1 F2F3F4F5 *ABCDEFGHJKLMNPQRSTUVWXYZ012345*
00E0 F6F7F8F9 C1C2C3C4 C5C6C7C8 40404040 *6789ABCDEFGHI....*
COUNT 0001000007000000
ADR013I (001)-CLTSK(01), 89067 16:08:40 TASK COMPLETED WITH RETURN CODE 0000
ADR012I DSSU (01), 89067 16:08:41 DFSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000

```

Figure 6. Output Resulting from a PRINT Command

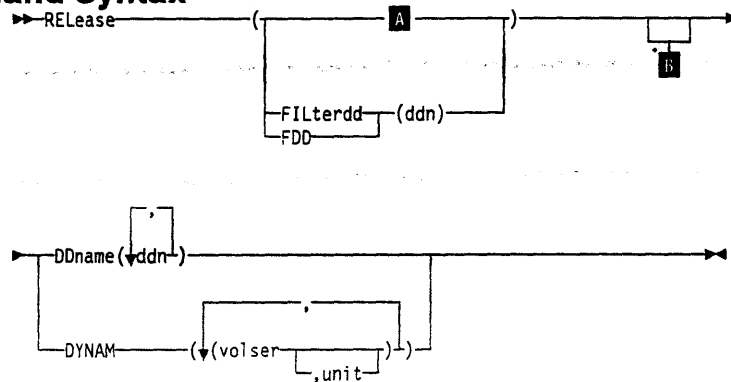
RELEASE Command

The RELEASE command releases, on a specified volume(s), allocated but unused space from all sequential and partitioned data sets that you select with INCLUDE, EXCLUDE, or BY criteria. For an explanation of these criteria, see Chapter 2, "Choosing the Data Sets You Want Processed—Filtering" on page 9. DFSS only selects data sets that have releasable space.

Note: The RELEASE function requires MVS/XA DFP Version 2 Release 1.0 or a later release.

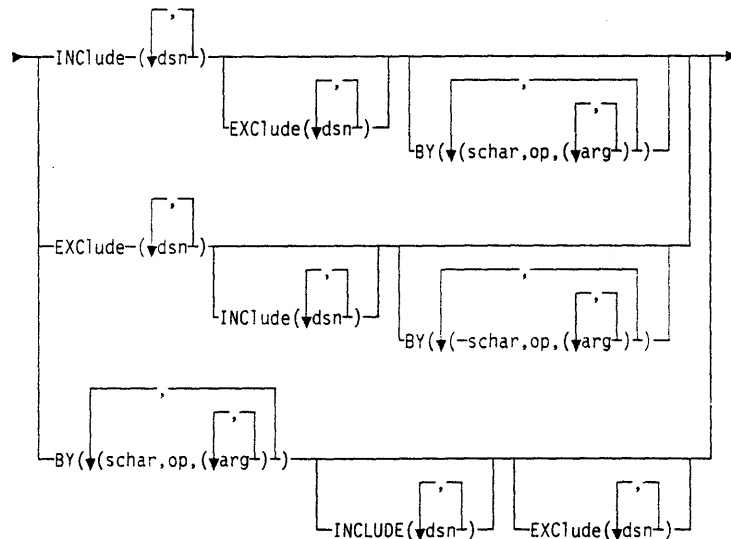
You must exclude, by using the EXCLUDE keyword, data sets whose DS1STAR (last block pointer in the Format 1 DSCB) field is not maintained in the VTOC. This can occur if you use an access method other than BSAM, QSAM, or BPAM. DFSS does not release space for data sets whose DS1STAR field is 0.

RELEASE Command Syntax

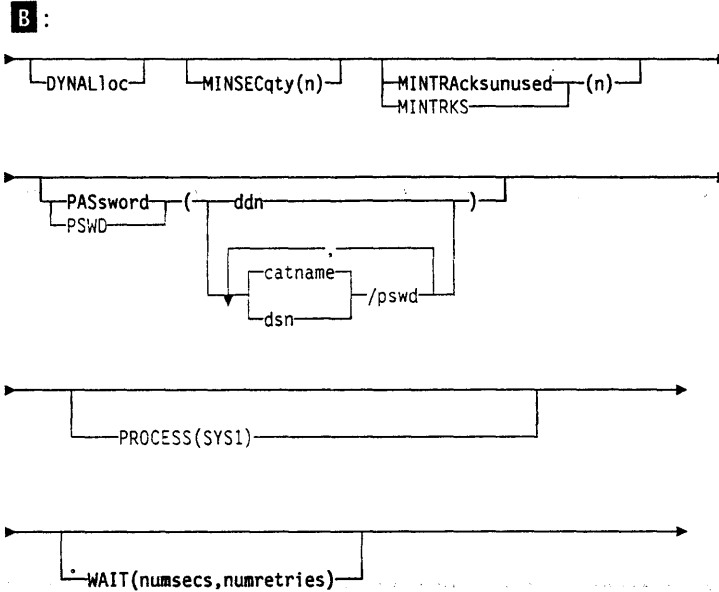


Additional Keywords Used with RELEASE

A :



Optional Keywords

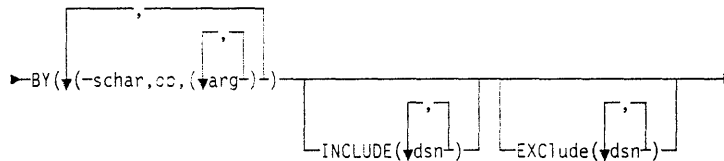


Explanation of RELEASE Command Keywords

The following applies to the RELEASE command:

- A data set with 16 allocated extents will not have any space released.
- Free tracks in cylinder-allocated extents are not released. Only free cylinders are released.
- DFDSS excludes system data sets beginning with SYS1, unless the PROCESS keyword is used.

BY



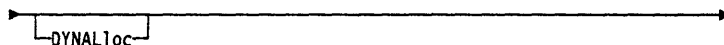
specifies that the data sets selected up to this point, by the processing of the INCLUDE and EXCLUDE keywords, are to be further filtered. To select the data set, *all* BY criteria must be met. See the separate discussions of INCLUDE and EXCLUDE for information on how these keywords are specified.

DDName



specifies the name of the DD statement that identifies a volume whose sequential and partitioned data sets, if selected, are to have their unused space released. To assure correct processing, each of the DD statements corresponding to a DDNAME (*ddn*) must identify only one volume serial number.

DYNALloc

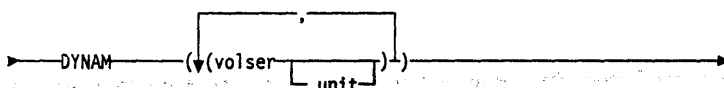


specifies dynamic allocation, instead of the ENQ macro, to serialize the use of data sets. This allows cross-system serialization in a JES3/MVS environment.

Consider:

- This serialization is of value only when the dynamic allocation/JES3 interface is not disabled.
- SVC 99 is used to serialize data sets.
- Run time increases when you use DYNALLOC to serialize data sets (as opposed to enqueue) because overhead is involved in dynamic allocation and serialization across multiple processors.

DYNAM



specifies that the volume to be processed is to be dynamically allocated. If the volume is real (not virtual), it must be mounted and online. Using DYNAM instead of DD statements to allocated DASD volumes will not appreciably increase run time and permits easier coding of JCL and command input.

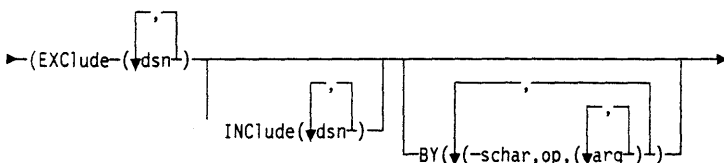
volser

specifies the volume serial number of a DASD volume to be processed.

unit

specifies the device type of a DASD volume to be processed. This keyword is optional.

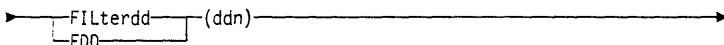
EXClude



dsn

specifies the name of a data set to be excluded from the data sets selected by INCLUDE. Either a fully- or a partially-qualified data set name can be used. See the separate discussions of INCLUDE and BY for information on how these keywords are specified.

FILTerdd

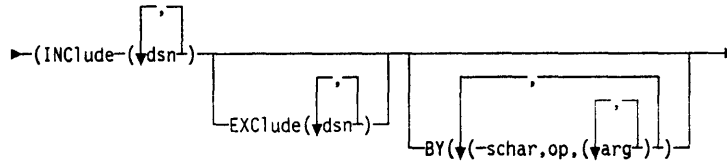


ddn

specifies the name of the DD statement that identifies the sequential data set or member of a partitioned data set that contains the filtering criteria to use. This is in the form of card-image records, in DFDSS command syntax, that contain the INCLUDE, EXCLUDE, and BY keywords that complete the RELEASE syntax.

Note: You must use FILTERDD when you have more than 255 entries in the INCLUDE, EXCLUDE, or BY list of subkeywords.

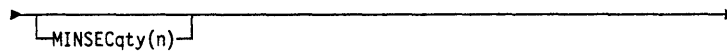
INClude



dsn

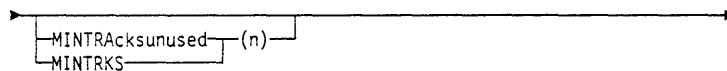
specifies the name of a data set whose unused space is eligible to be released. Either a fully- or a partially-qualified data set name can be used. See "Filtering by Data Set Names" on page 10. If INCLUDE is omitted (but EXCLUDE or BY is specified) or if INCLUDE(**) is specified, *all* data sets are eligible to be selected for releasing. See the separate discussions of EXCLUDE and BY for information on how these keywords are specified.

MINSECqty



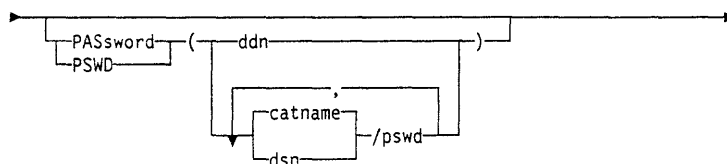
Because all the allocated but unused space will be released when you do not specify a secondary allocation quantity, you will not be able to add records to the data set after the RELEASE operation. MINSECQTY solves this problem. The RELEASE operation is not performed unless the secondary allocation quantity in the VTOC is equal to or greater than *n* tracks and the data set has not reached 16 extents. *n* is a 1- to 5-digit decimal number with a range from 0 to 99999. If MINSECQTY (*n*) is not specified, a default of 1 track is used.

MINTRAcksunused



specifies that the RELEASE operation is to be performed only if the number of unused tracks for a selected data set is equal to or greater than *n*, where *n* is a 1- to 5-digit decimal number. When MINTRKS is not specified, a default value of 1 is used. If a RELEASE is done, *all* unused tracks will be released.

PASsword



specifies the passwords DFDSS is to use for selected password-protected sequential and partitioned data sets. (Password checking is bypassed for RACF-protected data sets.) This is required only if:

- You do not have the required volume-level RACF DASDVOL or RACF DATASET access.
- The installation authorization exit does not bypass the checks.
- You do not want to be prompted for the password.

ddn

specifies the name of the DD statement that identifies the sequential data set or member of a partitioned data set that contains data set names and their passwords. This data set must contain card-image records in DFDSS command syntax format.

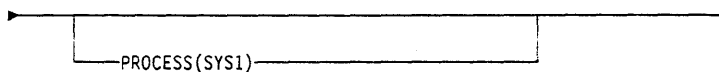
RELEASE Command

dsn/pswd

dsn is a fully-qualified data set name. *pswd* is its password. If no password follows the slash (/), *dsn* is treated as though it were *ddn*.

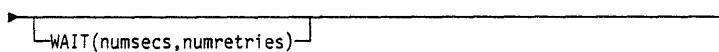
Printing of actual data set passwords specified in your input command stream is suppressed in the SYSPRINT output.

PROCESS



specifies that the RELEASE operation is to be performed for data sets with a high-level qualifier of SYS1. To specify PROCESS, you must have the proper RACF Facility Class authorization. Refer to *Resource Access Control Facility (RACF) Security Administrator's Guide* for additional information.

WAIT



specifies to DFDSS the length of a wait in seconds and the number of retries to obtain control of a volume or serialize a data set.

numsecs

is a number from 1 to 255 that specifies the interval, in seconds, between retries.

numretries

is a 1- or 2-digit number that specifies the number of times an attempt to gain control of a volume or data set is to be retried.

The default for *numsecs,numretries* is (2,2), which specifies two retries at 2-second intervals. If you do not want to wait for a resource, specify 0 for either *numsecs* or *numretries*.

Example of a RELEASE Operation

A RELEASE Operation on Selected Sequential and Partitioned Data Sets

```
//JOB1    JOB  accounting information,REGION=nnnnK
//STEP1   EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=A
//SYSIN   DD   *
RELEASE INCLUDE(**) -
  DYNAM(338000) /* DYNAM ALLOC VOL 338000 */ -
  MINTRKS(10) /* THERE ARE 10 OR MORE UNUSED TRKS */ -
                /* MINSEC NOT SPEC. IT DEFAULTS TO 1 */
/*
```

Unused tracks of sequential and partitioned data sets on volume 338000 are to be released if both:

- The number of unused tracks in the data set is greater than or equal to 10.
- The data set can be extended later if required (MINSEC(1)). This need not be specified, because it is the default.

The above example can be modified as follows to release unused tracks of all sequential and partitioned data sets, other than system data sets, that have any unused tracks:

```
//SYSIN DD *  
RELEASE INCLUDE(**) -  
DYNAM(338000) /* DYNAM ALLOC VOL 338000 */  
/*
```

RESTORE Command

With the RESTORE command, you can restore data to a DASD volume(s) from a DFSS-produced dump volume(s). You can restore data sets, an entire volume, or ranges of tracks. You can restore to unlike devices from a logical dump tape.

Note: The FULL keyword for the RESTORE command specifies that an entire DASD volume is to be restored. The TRACKS keyword for RESTORE specifies ranges of tracks to be restored.

DFSS offers two ways to process RESTORE commands:

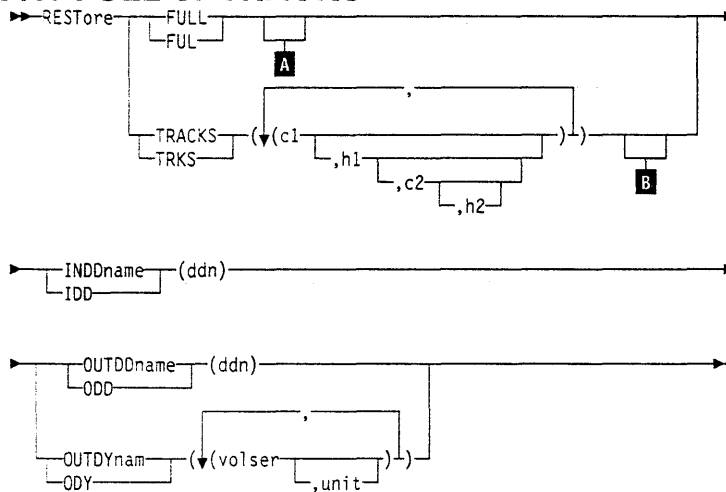
- *Logical processing* is data-set oriented, which means it operates against data sets and volumes independently of physical device format.
- *Physical processing* can operate against data sets, volumes, and tracks, but is oriented toward moving data at the track-image level.

The processing method is determined by what type of dump tape is used as input and by the keywords specified on the command.

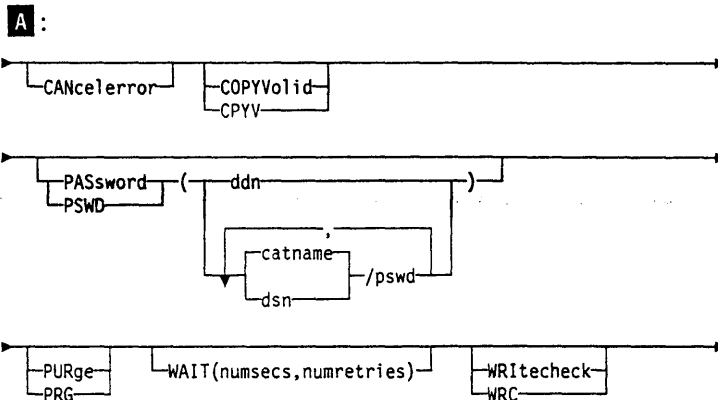
DFSS logical RESTORE processing cannot be used to process PAM data sets containing location-dependent information that does not reside in note lists or in the directory. For example, PDS load libraries that contain overlay modules are not supported during logical RESTORE processing.

For more information on using the RESTORE command, see *Data Facility Data Set Services: User's Guide*.

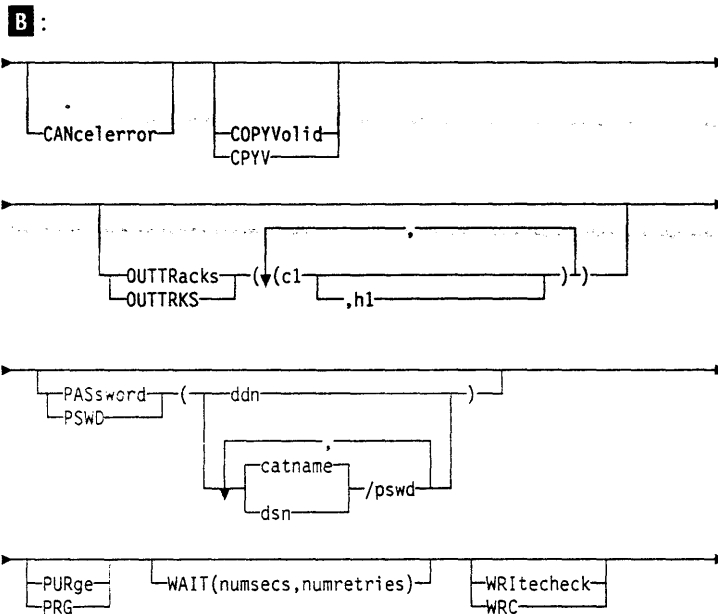
RESTORE Syntax for FULL or TRACKS



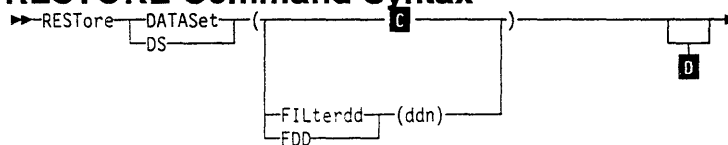
Optional Keywords Used with FULL



Optional Keywords Used with TRACKS



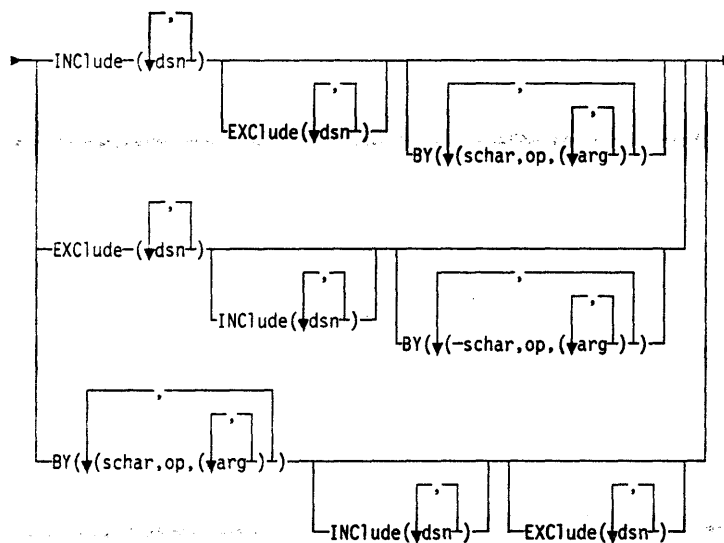
Logical Data Set RESTORE Command Syntax



RESTORE Command

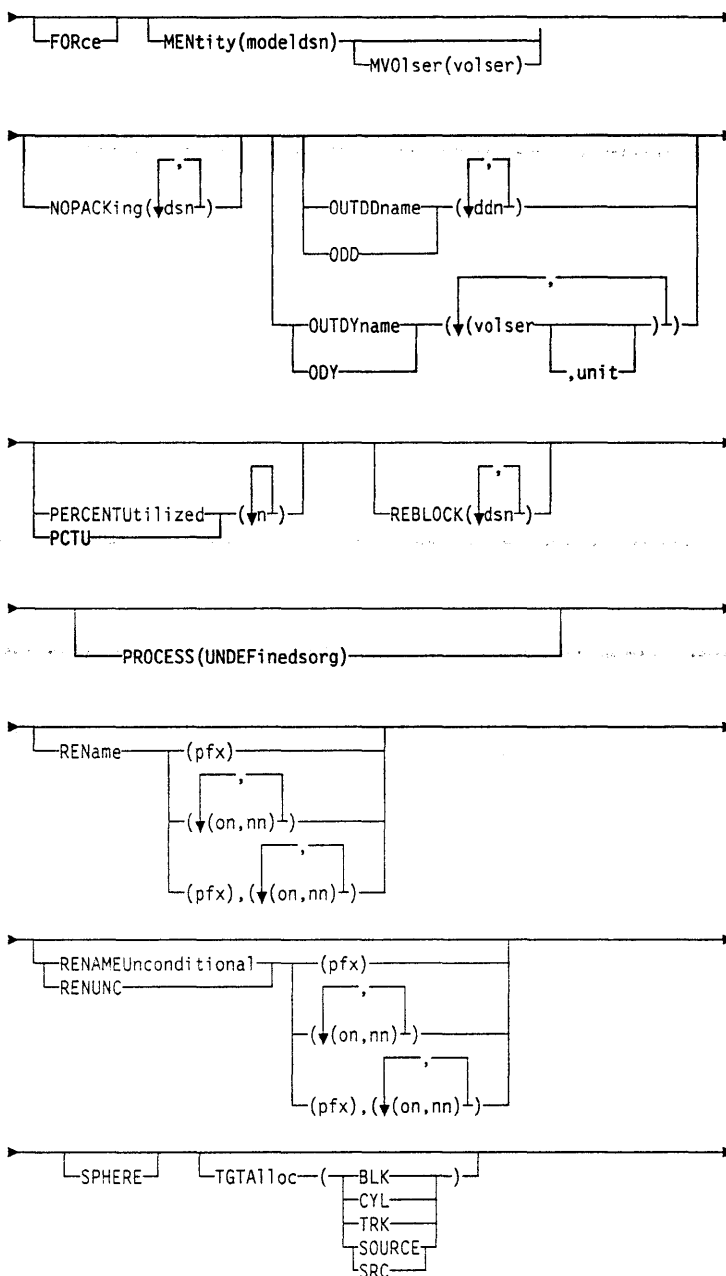
Additional Keywords Used with DATASet

G :



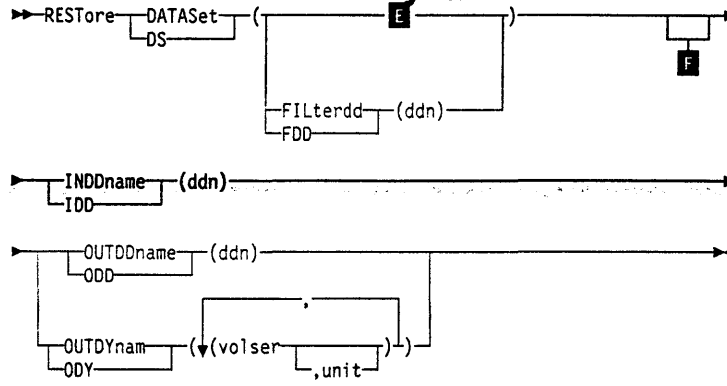
Optional Logical Data Set RESTORE Keywords

D:



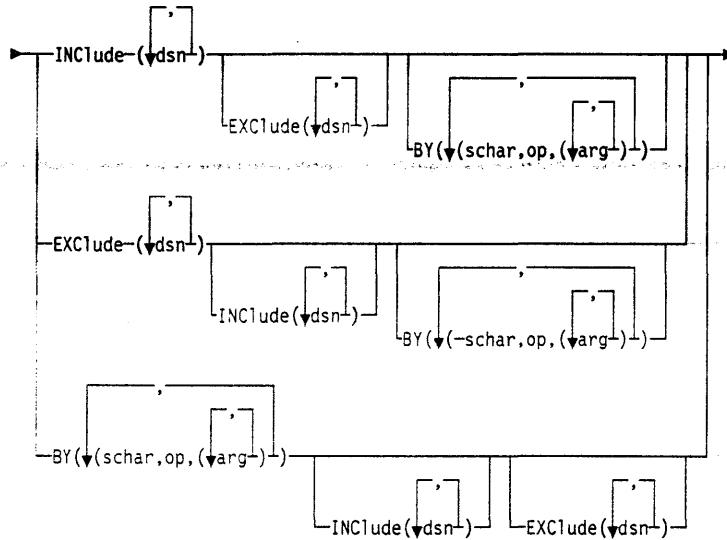
RESTORE Command

Physical Data Set RESTORE Command Syntax



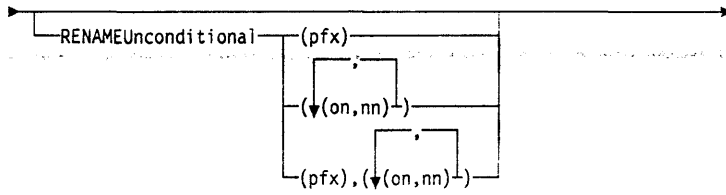
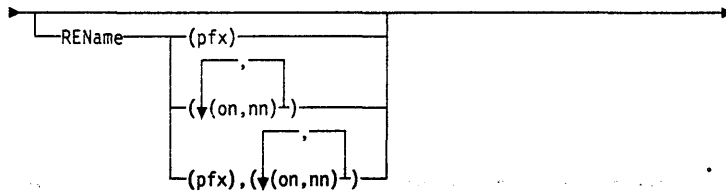
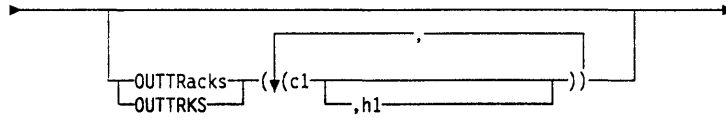
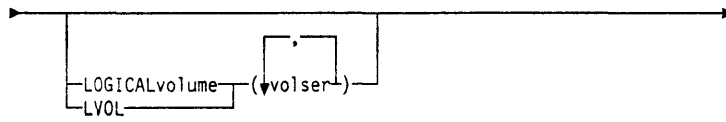
Additional Keywords Used with DATASET

E :

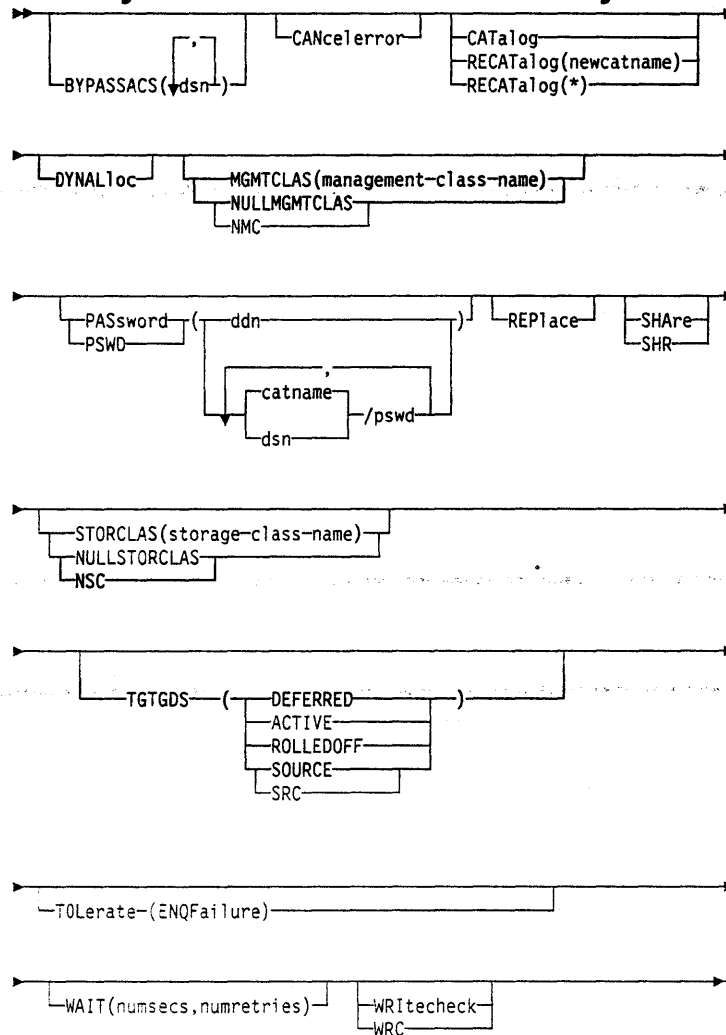


Optional Physical Data Set RESTORE Keywords

F:

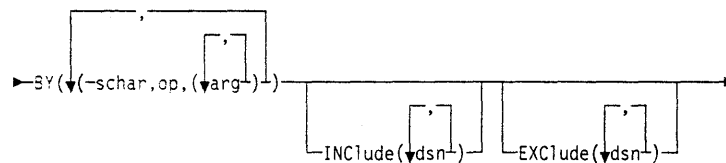


Optional Logical and Physical Data Set RESTORE Keywords

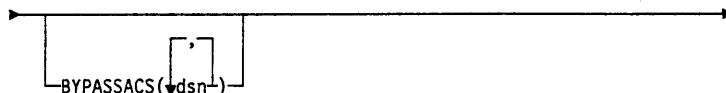


Explanation of RESTORE Command Keywords

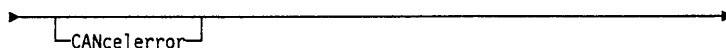
BY



specifies that the data sets selected up to this point, by the processing of the INCLUDE and EXCLUDE keywords, are to be further filtered. To select the data set, *all* BY criteria must be met. See "Filtering by Data Set Characteristics" on page 12 for further information. Also, separate discussions of INCLUDE and EXCLUDE provide information on how these keywords are specified.

BYPASSACS

specifies that Automatic Class Selection (ACS) routines are not to be invoked to determine the target data set's storage class or management class names. In order to bypass ACS filtering, the user must have RACF Facility Class authorization. See *Data Facility Data Set Services: User's Guide* for profile names.

CANcelerror

specifies that the RESTORE is to be stopped if a permanent read error occurs. If CANCELERROR is not specified, DFDSS attempts to recover from input errors, but the results can be unpredictable because of the difficulty in repositioning and assembling the data.

CATalog or RECATalog

For a *logical* RESTORE, CATALOG instructs DFDSS to catalog data sets that it allocates. For a *physical* RESTORE, CATALOG is used for non-VSAM single volume data sets; RECATALOG is ignored.

CATALOG

catalogs the target data set in a catalog as determined by the standard catalog search order. This is the default for VSAM data sets, multivolume data sets, and SMS-managed data sets (during logical data set restore).

RECATALOG(newcatname)

catalogs the target data set in the *newcatname* catalog.

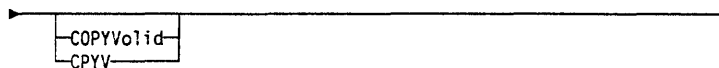
RECATALOG(*)

catalogs the target data set in the same catalog that points to the source data set. If the source data set was not cataloged, the new data set is not cataloged either.

Notes:

1. CATALOG or RECATALOG fails if the target data set is already cataloged in the same catalog and either RENAME is not specified.
2. CATALOG and RECATALOG are ignored when the target data set is preallocated.
3. RECATALOG is ignored for SMS-managed data sets.
4. Be careful when using RECATALOG(*newcatname*) because the target data set may already be cataloged outside of the standard order of search.
5. CATALOG must be specified to catalog non-VSAM SMS-managed data sets during physical data set restore.

COPYVolid



specifies that the VOLID from the dumped DASD volume is to be copied to the output DASD volume. This applies to a full RESTORE operation; it applies to a tracks RESTORE only if track 0 is to be restored.

When the volume serial number on a DASD volume is changed, the volume serial field in the UCB of the receiving volume is cleared and the operator is notified. The operating system then initiates either:

- A demount if there is another volume with the same serial number
- A mount to get the volume with the new serial number mounted.

This might change the mount attributes of the volume. You should exercise operating precautions if there are two or more processors sharing the same DASD volume.

When the volume serial number is changed by using a COPYVOLID keyword or when both the dumped volume and the volume to be restored have different serial numbers, profiles are not built for the RACF-protected data sets on the restored volume or for the RACF DASDVOL for RACF-protected DASD volumes.

Notes:

1. If you are doing a full RESTORE and the input has VSAM data sets, the VOLID must be copied.
2. COPYVolid is required for a full volume RESTORE of SMS-managed source volumes.

DATASet

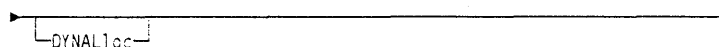


specifies a data set RESTORE operation, using filtering. See Chapter 2, "Choosing the Data Sets You Want Processed—Filtering" on page 9 for an explanation of the filtering process used.

Notes:

1. Either the FILTERDD or the INCLUDE keyword must be substituted when DATASET is selected.
2. VSAM catalogs are not supported for data set processing. If VSAM catalogs are processed, the results can be unpredictable.

DYNALloc



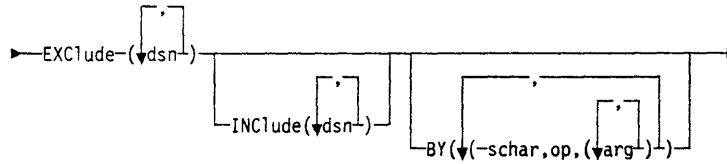
specifies dynamic allocation, instead of enqueue, to serialize the use of data sets. This allows cross-system serialization in a JES3/MVS environment.

Consider:

- The serialization is of value only when the dynamic allocation/JES3 interface is not disabled.
- SVC 99 is used to serialize data sets.

- Run time increases when you use DYNALLOC to serialize data sets (as opposed to enqueue) because overhead is involved in dynamic allocation and serialization across multiple processors.
- Using INDYNAM instead of DD statements to allocate DASD volumes does not appreciably increase run time and permits easier coding of JCL and command input.

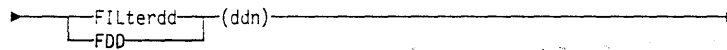
EXclude



dsn

specifies the name of a data set to be excluded from the data sets selected by INCLUDE. Either a fully- or a partially-qualified data set name can be used. See the separate discussion of INCLUDE and BY for information on how these keywords are specified.

FILterdd

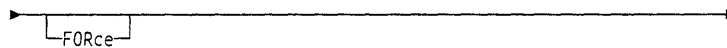


ddn

specifies the name of the DD statement that identifies the sequential data set or member of a partitioned data set that contains the filtering criteria to use. This is in the form of card-image records, in DFDSS command syntax, containing the keywords INCLUDE, EXCLUDE, and BY that complete the RESTORE syntax below.

Note: You must use FILTERDD when you have more than 255 entries in the INCLUDE, EXCLUDE, or BY list of subkeywords.

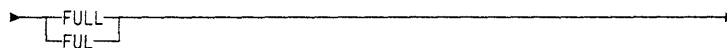
FORce



For a **logical** data set RESTORE, FORCE specifies that DFDSS will allow an unmovable data set or a data set allocated by absolute track allocation to be moved.

FORCE does not apply to ISAM data sets.

FULL



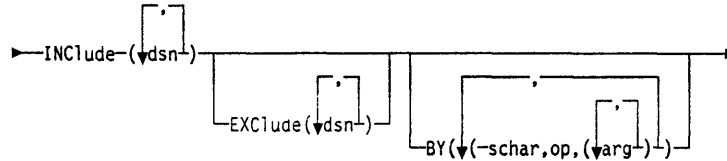
specifies that an entire DASD volume is to be restored. This is the default.

Notes:

1. You cannot specify SHARE or TOL(ENQF) for FULL operations.
2. VSAM catalogs are not supported for full volume processing. If VSAM catalogs are processed, the results can be unpredictable.

RESTORE Command

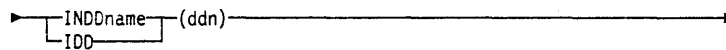
INClude



dsn

specifies the name of a data set eligible to be restored. Either a fully- or a partially-qualified data set name can be used. See "Filtering by Data Set Names" on page 10. If INCLUDE is omitted (but EXCLUDE or BY is specified) or INCLUDE(**) is specified, *all* data sets are eligible to be selected for restoring. See the separate discussions of EXCLUDE and BY for information about how these keywords are specified.

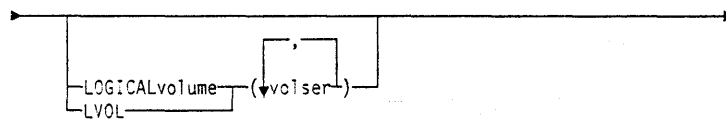
INDDname



ddn

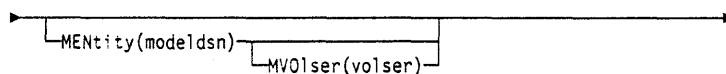
specifies the name of the DD statement that identifies the (input) dump data set. This data set can be on a tape or DASD volume.

LOGicalvolume



specifies, for a *physical* data set RESTORE operation, the volume serial number(s) of the source DASD volume(s) that are to be processed. For example, if you have taken a data set dump from volumes 111111, 222222, and so forth, but you want to restore only some data sets from source volume 222222, specify LOGICALVOLUME (222222). LOGICALVOLUME is useful for restoring multi-volume data sets.

MENTity

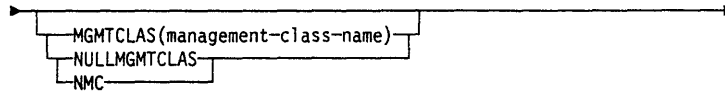


specifies a model entity and, optionally, the serial number of the volume containing that entity (*volser*) to be used when DFDSS defines discrete profiles. These keywords are used to define the data sets to RACF. Specification of MVOLSER is optional when the model entity (MENTITY) is either (1) cataloged in an integrated catalog facility catalog or (2) a non-VSAM data set cataloged in the standard catalog search order. When MVOLSER is specified for a VSAM model entity, the *volser* specified must be the volume serial number of the catalog in which the model entity is cataloged. If these keywords are not specified, DFDSS defines the data set to RACF without using a model.

During a RESTORE operation, DFDSS defines a data set to RACF only if:

- The data set was RACF-protected with a discrete profile during DUMP
- Either the data set is not preallocated or it is preallocated but not RACF-protected with a discrete profile.

For a description of RACF data sets, refer to *Resource Access Control Facility (RACF) General Information Manual*.

MGMTCLAS

specifies the user-desired management class that is to replace the source management class as input to the ACS routines. You must have the proper RACF authority for the management class specified. The keyword itself does not require RACF authorization.

NULLMGMTCLAS specifies that the input to the ACS routines is to be a null management class rather than the source data set's management class.

MGMTCLAS and NULLMGMTCLAS are mutually exclusive.

Note: All SMS-managed data sets specified in the BYPASSACS keyword will be assigned the specified management class because the ACS routines will not be invoked. Non-SMS-managed data sets do not have a management class.

NOPACKing

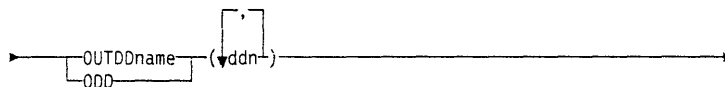
specifies that DFDSS is to allocate the target partitioned data set only to devices that are the same or like device type as the source, and that DFDSS is to use track level I/O to perform data movement. This results in an exact track-for-track image of the source data set on the target volume.

dsn

specifies the fully- or partially-qualified name(s) of a PDS to be processed.

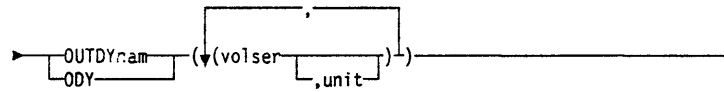
NOPACKing is only valid with a PDS. If REBLOCK is specified, REBLOCK is ignored for the data set.

A PDS restored using NOPACKing is not compressed during data movement. NOPACKing can be used for a damaged PDS that is currently usable by an application but would be made unusable by compression or other rearrangement of the physical layout of the data.

OUTDDname***ddn***

specifies the name of the DD statement that identifies a volume to be restored. To assure correct processing, each of the DD statements corresponding to a DDNAME (*ddn*) must identify only one volser. For a logical data set RESTORE or when you specify spill files, you can specify multiple names, separated by commas.

OUTDynam



specifies that the volume to be restored is to be dynamically allocated. If real (not virtual), the volume must be mounted and online. For a logical data set RESTORE or when you specify spill files, one or more volumes are allowed.

volser

specifies the volume serial number of a DASD volume to be restored.

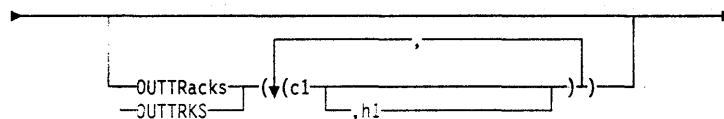
unit

specifies the device type of a DASD volume to be restored. This keyword is optional.

Note: OUTDDNAME or OUTDYNAM is required for restoring from a physical dump and optional for restoring from a logical dump. They are optional for a logical RESTORE, except:

- For multivolume data sets that are preallocated on volumes that are different from the original source volumes; or
- When the original source volume is not available.

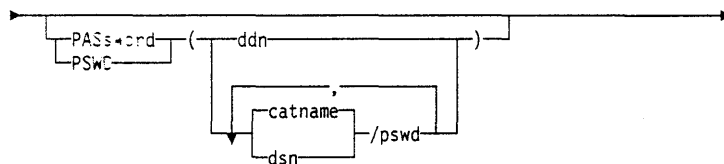
OUTTRacks



specifies, for a track RESTORE, the beginning location of the cylinder and head number of the target volume to which the track is to be restored.

If OUTTRKS is not specified, the track is restored to its original cylinder and head number.

PASsword



specifies the passwords DFDSS is to use for password-protected data sets for all RESTORE operations. (Password checking is bypassed for RACF-protected data sets.) This is required only if:

- You do not have the required volume-level RACF DASDVOL or RACF DATASET access.
- The installation authorization exit does not bypass the checks.
- You do not want to be prompted for the password.

ddn

specifies the name of the DD statement that identifies the sequential data set, or member of a partitioned data set, that contains data set names and their passwords in the format *dsn/pswd*[...]. This data set must contain card-image records in DFDSS command syntax format.

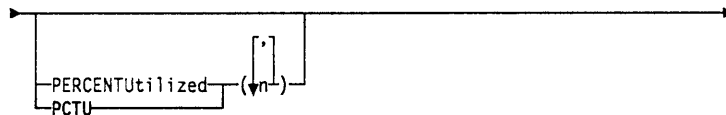
dsn/pswd

dsn is a fully- or partially-qualified data set name. *pswd* is its password. If no password follows the slash (/), *dsn* is treated as though it were *ddn*.

catname/pswd

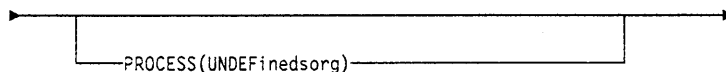
catname is a fully-qualified catalog data set name. *pswd* is its password. This is supported for VSAM data sets only.

Note: Printing of actual data set passwords specified in your input command stream is suppressed in the SYSPRINT output.

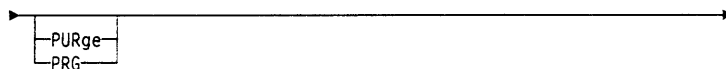
PERCENTUtilized

specifies that DFDSS must stop allocating data sets to the target volume(s) when the allocated space reaches *n* percent of the total space on the target volume. The default is 100. Specify more than one *n* if you have more than one target volume (for instance, a volume for overflow). PERCENTUTILIZED is ignored when the target data set is preallocated.

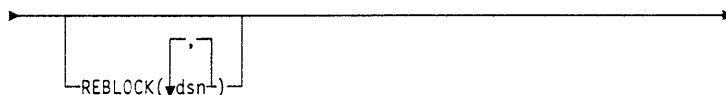
Note: PERCENTUTILIZED is not supported in an SMS environment. This keyword is valid only for logical data set restore operations.

PROCESS

specifies that the logical data set RESTORE operation is to be allowed for undefined DSORG data sets going to an unlike target device of a larger capacity and that DFDSS is to use track level I/O to perform data movement. This results in an exact track-for-track image of the source data set on target volume.

PURge

specifies that a full or tracks RESTORE is not to be stopped if there are unexpired data sets on the target volume. See the REPLACE keyword for a data set RESTORE.

REBLock

specifies that DFDSS is to reblock one or more of the selected SAM or PAM data sets unless otherwise overridden by the user reblock exit routine.

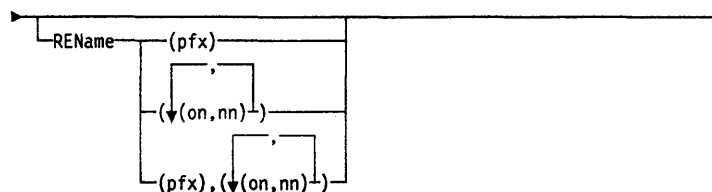
dsn

specifies the fully- or partially-qualified name(s) of a SAM or PAM data set to be copied and reblocked.

The following data sets are not eligible for reblocking:

- PAM data sets with NOTE lists and load modules
- SAM and PAM data sets with RECFM = U, RECFM = variable spanned.

REName



specifies that, if a data set with the old name exists on the output DASD volume, DFDSS is to allocate a new data set with the new name and restore the data set. If the data set with the old name does not exist on the volume, the data set is restored with the old name. For a VSAM data set that already exists on another DASD volume and is cataloged, the VSAM data set is restored with the new name unless the new name also exists and is cataloged.

For a *physical* data set RESTORE, RENAME is not supported for VSAM data sets. This keyword only applies to *movable* data sets, therefore, unmovable or ISAM data sets will not be renamed. See “Special Considerations for Restore” on page 117 for details. RENAME and RENAMEUNCONDITIONAL are mutually exclusive; you cannot specify these keywords together.

pfx

Specifies the prefix to be used to replace the first-level qualifier of the data set name. It is optional, but if specified, must be the first parameter in the list of subkeywords. The prefix is used only if the *(on,nn)* parameters are not specified or the old name filters do not match the data set name.

on

Specifies the old name to be used as filtering criteria for matching data set names on the target volume.

nn

Specifies the new name to be used to derive the new data set name if the data set name selected by the corresponding old name filtering criteria matches the name of a data set that already exists on the target volume.

If none of the old name filters match the data set name and the prefix is specified, the prefix is used to derive the new name. If old name filters do not match and the prefix is not specified, the data set is not restored. If the old name filter matches and there is an error in the new name filter, the data set is not restored.

The syntax for the **prefix** is as follows:

- Single-level, fully-qualified, unquoted DSNAME
- 8 characters or less
- The first character should be an alphabetic character
- The rest of the characters can be alphanumeric.

The syntax for the **old name** filter is exactly like that of the INCLUDE filter, and their rules match. For more information, see the INCLUDE keyword for the RESTORE command, or see “Filtering by Data Set Names” on page 10.

Examples of valid syntax for the **new name** filter are:

- ******—Restore the data set with the old name. This provides a powerful tool whereby some data sets can be restored with the old name and others can be restored with the new name.
- *****—If DSNAME has one level, then restore with old name.

- A.**—First level of DSNAME replaced by A.
- A.B.**—First two levels of DSNAME replaced by “A.B”.
- *.A.**—Second level of DSNAME replaced by A.
- **.BCD—Last level of DSNAME replaced by BCD.
- DATE.**.LIST—First and last levels are replaced by DATE and LIST.
- Q.*—If DSNAME has two levels, replace the first by Q.
- Q.*.B—If DSNAME has three levels, replace the first and last by Q and B.
- *.*.SYSLIST—If DSNAME has three levels, replace the last by SYSLIST.
- ABC.DEF—No asterisk in substring; replace the entire name with “ABC.DEF”.

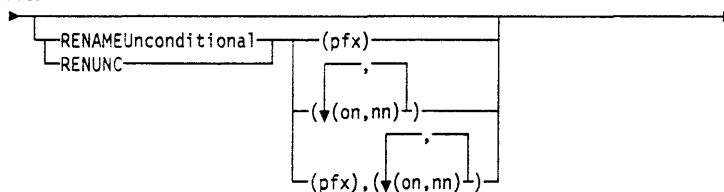
Examples of invalid syntax for the **new name filter** are:

- **.DATA.**—Invalid (level to be replaced is ambiguous).
- *SYS*—Invalid (a qualifier is not completely replaced).
- SYS*—Invalid (a qualifier is not completely replaced).
- *SYS—Invalid (a qualifier is not completely replaced).
- SYS*TEM—Invalid (a qualifier is not completely replaced).

If the new name filter has errors, the data set is not restored. The new name that is derived is truncated to fit 44 characters. If it ends with a period, that period is also truncated.

If the new name is not fully qualified, then it must contain the same number of qualifiers as the old name. For example, given the old name filter DATE.** and the new name filter DATE.*.*.LIST, DATE.MARCH.TODAY.OLDLIST would be renamed, but DATE.MARCH.OLDLIST would not.

RENAMEUnconditional



specifies that the data set should be restored with the new name, whether or not the data set exists on DASD with the old name. If the data set exists on the volume with the new name, it is not restored. This keyword only applies to *movable* data sets, therefore, unmovable or ISAM data sets will not be renamed. If the old name filter matches and there is an error in the new name filter, the data set is not restored. See “Special Considerations for Restore” on page 117 for details.

Notes:

1. RENAME and RENAMEUNCONDITIONAL are mutually exclusive; you cannot specify these keywords together. RENAMEUNCONDITIONAL is not supported for physical restore of VSAM data sets.
2. RENAMEUNCONDITIONAL specifies that the data set must be restored with the new name, regardless of whether the data set exists on DASD with the old name. If the data set exists on the target volume with the new name, an allo-

ation error occurs and the data set is not restored, regardless of whether **REPLACE** is specified.

px

specifies the prefix used to replace the first-level qualifier of the data set name. It is optional but, if specified, must be the first parameter in the list of subkeywords. The prefix is used only if the (*on,nn*) parameters are not specified or the old name filters do not match the data set name.

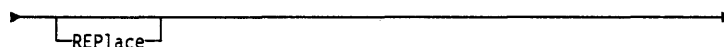
on

Specifies the old name to be used as a filtering criterion to check if it matches the data set name.

nn

Specifies the new name to be used to derive the new data set name if the data set name matches the corresponding old name filtering criterion.

For syntax rules, see the discussion of *px*, *on*, and *nn* under "REName" on page 112.

REPlace

specifies that if the data set exists on DASD, it can be replaced with the data set from the source volume. DFDSS searches the output volumes for preallocated data sets. If no output volumes are specified, DFDSS searches the volumes on which the data set resided at dump time. If no preallocated target is found, DFDSS attempts to allocate a data set. If allocation is successful, the data set is restored. **PURGE** is accepted and is treated the same as **REPLACE**.

For non-VSAM data sets, if the target data set is smaller than the source data set, DFDSS scratches the target data set and reallocates it with the size of the source data set.

Notes:

1. **REPLACE** must be specified to restore preallocated unmovable data sets.
2. **REPLACE** is required for movable data sets if a preallocated data set is to be replaced.
3. If **REPLACE** is specified with **RESTORE**, the classes already associated with the preallocated data set remain the same. See "Special Considerations for Restore" on page 117 for more information.

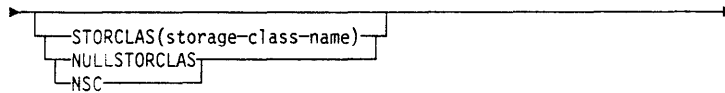
SHAre

specifies that DFDSS is to share, for read access with other programs, the data sets that are to be restored. The resetting function is bypassed if **SHARE** is specified on a data set **RESTORE**.

SHARE and **FULL** are mutually exclusive; you cannot specify these keywords together.

SPHERE

specifies that for any VSAM cluster dumped with the SPHERE keyword, DFDSS must also restore all associated AIX clusters and paths. Individual sphere component names need not be specified; only the base cluster name is required.

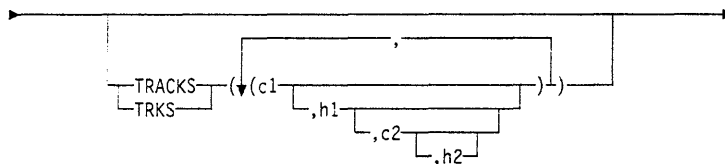
STORCLAS

specifies the user-desired storage class that is to replace the source storage class as input to the ACS routines. The user must have the proper RACF authorization for the storage class specified. The keyword itself does not require authorization.

NULLSTORCLAS specifies that the input to the ACS routines is to be a null storage class rather than the source data set's storage class.

STORCLAS and NULLSTORCLAS are mutually exclusive.

Note: If BYPASSACS(dsn) is specified, all data sets that pass the BYPASSACS selection criteria are guaranteed the specified storage class. The combination of NULLSTORCLAS and BYPASSACS(dsn) forces the selected data sets to be non-SMS managed.

TRACKS

specifies ranges of tracks to be restored (that is, a tracks RESTORE). If any of the requested tracks are not in the input file, the RESTORE operation is stopped.

c1,h1

specifies the cylinder and head number of the beginning of the range. Specify hexadecimal numbers as X'c1' or X'h1'.

c2,h2

specifies the cylinder and head number of the end of the range. Specify hexadecimal numbers as X'c2' or X'h2'.

Notes:

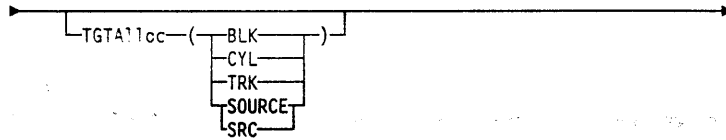
1. The c2 must be greater than or equal to c1.
2. If c2 equals c1, h2 must be greater than or equal to h1.

DFDSS verifies that the range is within the limits of the device. If you do not specify all four values for a range, DFDSS provides the missing values unless the omitted value causes a syntax error. No intervening values can be omitted. For example:

Specified	Results
<i>None</i>	Syntax error
<i>c1</i>	c1,0,c1,maximum head number
<i>c1,h1</i>	c1,h1,c1,maximum head number
<i>c1,h1,c2</i>	c1,h1,c2,maximum head number

c1,c2 Syntax error
,h1 Syntax error

TGTALloc



specifies, during a logical data set RESTORE, how DFDSS will allocate the target data set.

BLK
specifies to allocate by blocks.

CYL
specifies to allocate by cylinders.

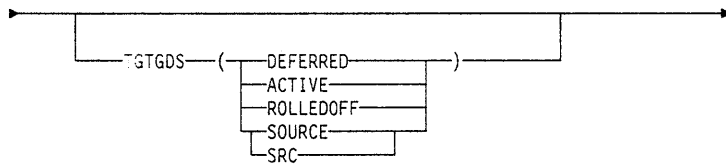
TRK
specifies to allocate by tracks.

SOURCE/SRC
specifies to allocate with the same space allocation type as that of the source data set.

Notes:

1. If the TGTALLOC keyword is omitted, the target allocation defaults to source.
2. If BLK is specified for VSAM data sets, TRK is used instead. The final VSAM allocation can be different from the requested one because of VSAM allocation rules.

TGTGDS



specifies in what status, during a data set operation, that DFDSS is to place non-preallocated SMS-managed GDG data sets:

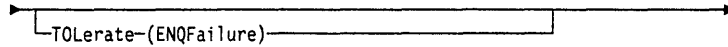
DEFERRED
specifies that the target data set is to be assigned the DEFERRED status.

ACTIVE
specifies that the target data set is to be assigned the ACTIVE status, for example, rolled into the GDG base.

ROLLEDOFF
specifies that the target data set is to be assigned the rolled-off status.

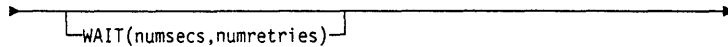
SOURCE/SRC
specifies that the target data set is to be assigned the same status as that of the source data set.

Note: The requested target status of generation data sets must not violate generation data group rules.

TOLerate**ENQFailure**

specifies that data sets are to be processed even though shared or exclusive access fails. TOL(ENQF) and FULL are mutually exclusive; you cannot specify these keywords together.

For more information on TOL(ENQF), see Appendix B, “Serialization and Authorization Checking” on page 165.

WAIT

specifies to DFDSS the length of a wait in seconds and the number of retries to obtain control of a volume or data set.

numsecs

is a number from 1 to 255 that specifies the time interval, in seconds, between retries.

numretries

is a 1- or 2-digit number that specifies the number of times an attempt to gain control of a volume or data set is to be retried.

The default for *numsec,numretries* is (2,2), which specifies two retries at 2-second intervals. If you do not want to wait for a resource, specify 0 for either numsecs or numretries.

WRltecheck

specifies that the data restored is to be verified for successful completion. This keyword increases the overall elapsed time. The default is no WRITECHECK.

Special Considerations for Restore

The actions DFDSS performs for RENAME, RENAMEUNCONDITIONAL, or REPLACE depend on the keywords you specify and the configurations of data sets on volumes. This section contains figures that describe specific environments for DFDSS RESTORE operations.

Figure 7 on page 118 describes, in decision table format, DFDSS RESTORE actions for physical processing of SMS-managed data sets. The specified RESTORE keywords and the existence of the data set are shown in the upper half. The actions taken are shown in the lower half.

Figure 8 on page 119 describes *general* DFDSS actions for both physical and logical RESTORE on non-VSAM data sets. It is *not* a program flowchart. Use it to clarify the RESTORE actions on non-VSAM data sets under varying conditions.

	Non-VSAM Physical Restore	VSAM Phy Restore
RENAME	N N Y Y Y Y - - - - N N	- - - - -
RENUNC	N N - - - - - Y Y Y Y N N	- - - - -
REPLACE	Y Y N N Y Y N N Y Y N N	Y Y N N
OLD DATA SET EXISTS	Y N Y N Y N Y N Y N Y N	Y N Y N
NEW NAME ON USERVOL	- - N X N - - - - - - - -	- - - - -
NEW NAME ON VDSSVOL	- - - - X - X N N N N - -	- - - - -
Overlay old data set on USERVOL	T - - - - - - - - - - - -	T - - - -
Overlay old data set on VDSSVOL	- - - - - - - - - - - -	- - - - -
ALLOC with old name on USERVOL & restore	- - - - - - - - - - - -	- - - - -
ALLOC with new name on USERVOL & restore	- - T - T - - - - - - - -	- - - - -
ALLOC with old name on VDSSVOL & restore	- T - T - T - - - - - T	- T - T
ALLOC with new name on VDSSVOL	- - - - - - T T T T - -	- - - - -
Do not restore	- - - - - - - - - - T -	- - T -
<p>Legend:</p> <p>Y Yes N No T Action taken X Don't Care - Not Applicable</p> <p>USERVOL Output volume specified in Restore command VDSSVOL Output volume selected by VTOC/Data Set Services</p> <p>Note: Once a target volume is selected based on the matrix above, all the selected data sets on the same input logical volume must be restored to the same target volume except that when certain errors occur, a message is issued and the data set is not restored.</p>		

Figure 7. Physical Data Set Restore Actions on SMS-Managed Data Sets

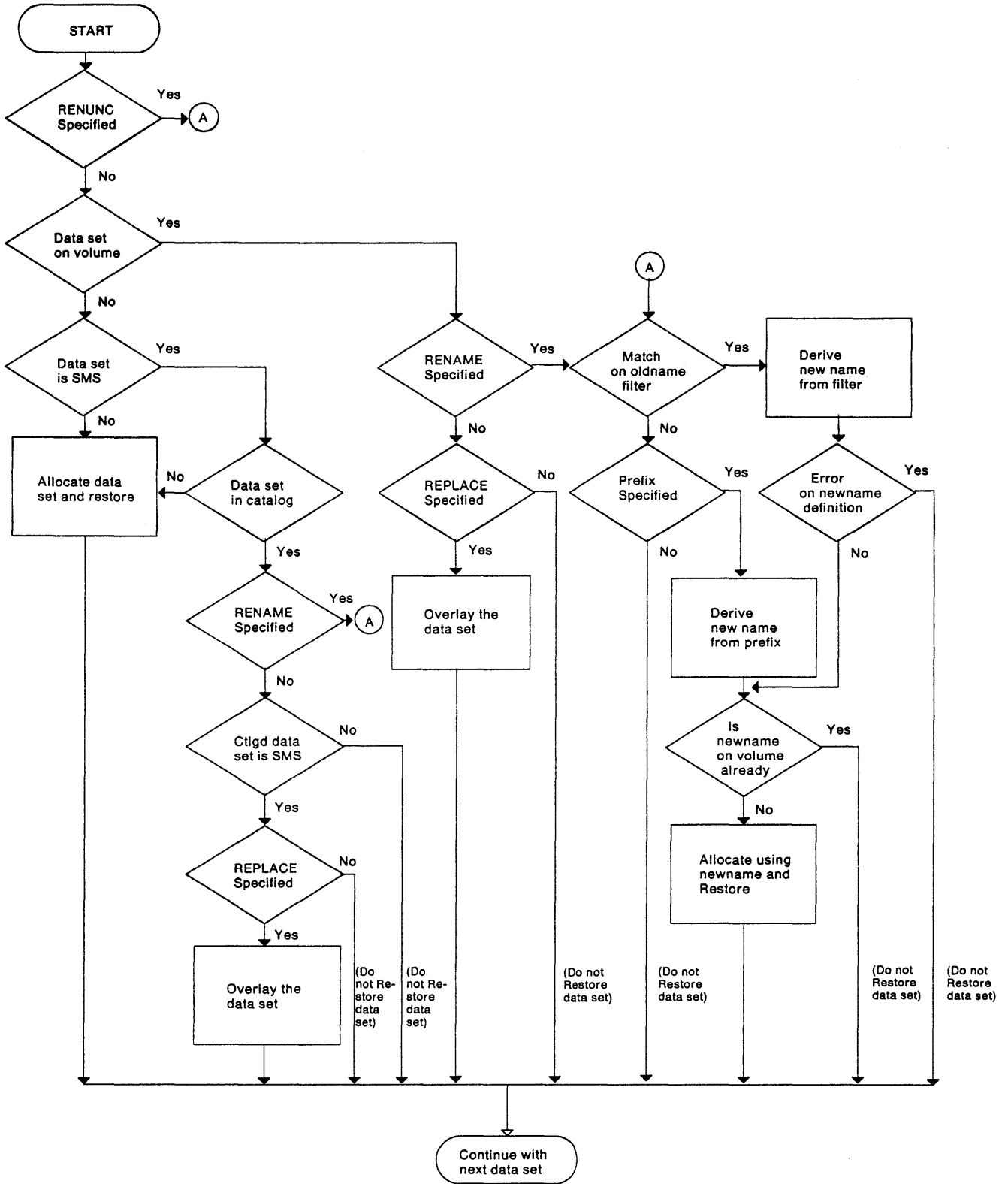


Figure 8. RESTORE Actions on Non-VSAM Data Sets. These actions apply to a data set RESTORE with RENAME, RENAMEUNCONDITIONAL, and REPLACE keywords.

Assignment of Class Names by Using RESTORE and COPY

In an SMS environment, you can use `STORCLAS`, `MGMTCLAS`, `NULLSTORCLAS`, `NULLMGMTCLAS`, and `BYPASSACS` keywords with `RESTORE` and `COPY` to influence the class names assigned to a data set. Figure 9 shows how these keywords can influence the storage and management class names of the target data set in a `RESTORE` or `COPY` operation.

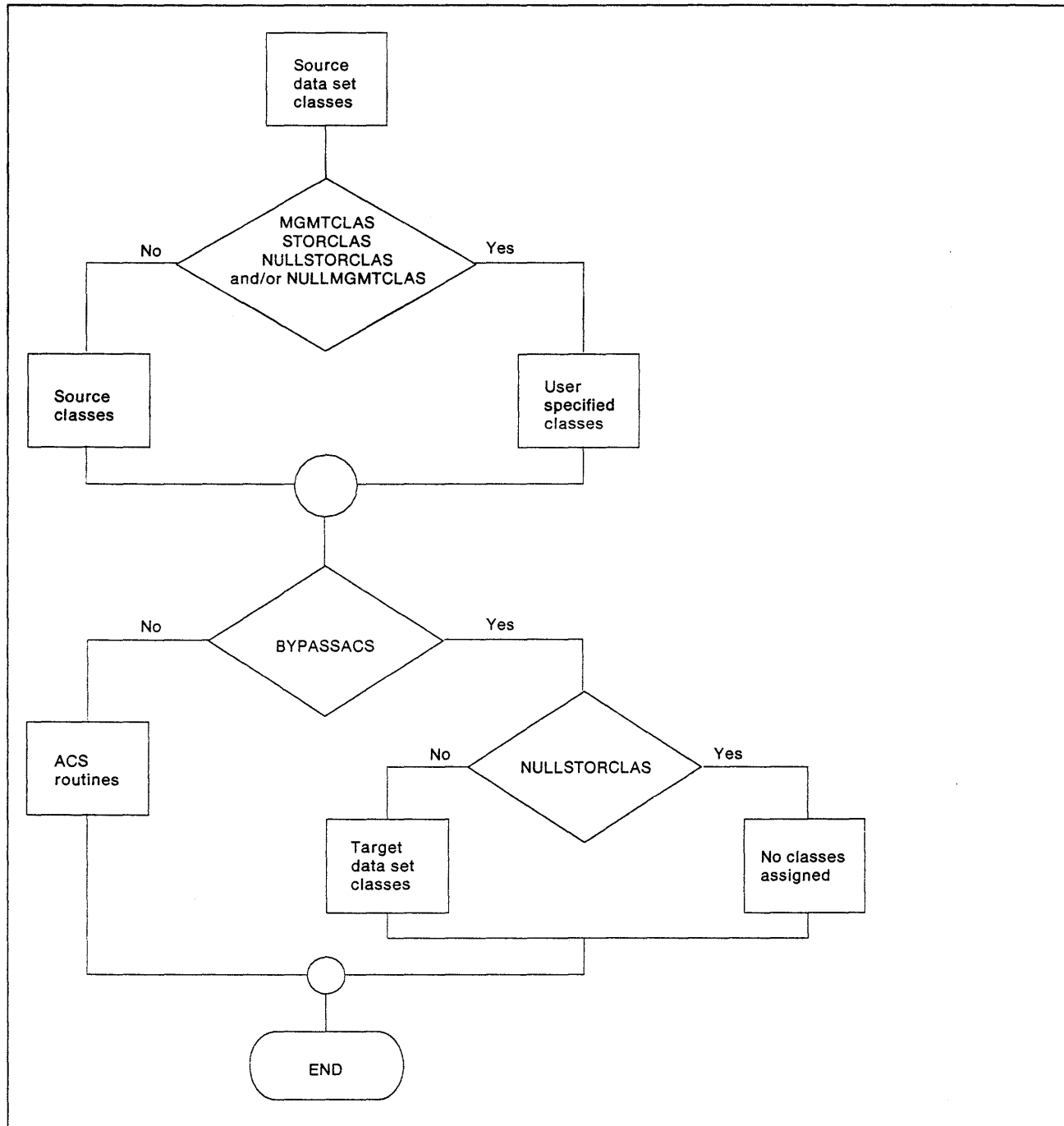


Figure 9. DFDSS Target Class Selection

Notes to Figure 9 on page 120:

- If you specify **BYPASSACS**, the source data set's class names or the class names specified with **STORCLAS** and **MGMTCLAS** are assigned to the target data set. If you do not specify **BYPASSACS**, ACS uses the source data set's class names, or the class names specified with **STORCLAS** and **MGMTCLAS** as input to assign the target data set's class names.
- If you specify **NULLSTORCLAS**, DFDSS passes a null storage class to ACS, which selects a storage class for the data set. If you specify **NULLSTORCLAS** and **BYPASSACS** together, the data set becomes non-SMS managed.
- **NULLMGMTCLAS** can only be used with SMS-managed data sets. Specifying **NULLMGMTCLAS** and **BYPASSACS** together causes the removal of the original management class of the data set.

Data Integrity Considerations for Full or Tracks RESTORE

For a full or tracks RESTORE, DFDSS serializes the VTOC to preclude DADSM functions such as **ALLOCATE**, **EXTEND**, **RENAME**, and **SCRATCH** from changing the contents of the VTOC on the volume during the RESTORE. Data sets are *not* serialized on these full or tracks operations. Therefore, some data sets might be opened by other jobs during the RESTORE. The result might be that partially updated data sets are restored. For mountable devices, you can minimize this possibility by doing the following:

```
VOL=(PRIVATE,SER=xxxxxx)
```

Full data integrity can always be guaranteed by performing RESTORE operations by data set only.

Examples of Full and Tracks RESTORE Operations

Example 1 shows that DASD volume 111111 is to be restored from the first data set of standard label tape volumes TAPE01 and TAPE02.

The command input to be substituted for a full and tracks RESTORE are shown below in Example 1A and 1B respectively. To dump the same volume, refer to Examples 1, 1A, and 1B of the DUMP command.

Example 1:

```
//JOB1 JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRSSU
//SYSPRINT DD SYSOUT=A
//TAPE DD UNIT=3480,VOL=SER=(TAPE01,TAPE02),
// LABEL=(1,SL),DISP=(OLD,KEEP),DSNAME=USER2.BACKUP
//DASD DD UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//SYSIN DD *
command input (see Examples 1A and 1B below)
/*
```

Example 1A: A Full RESTORE

```
RESTORE INDDNAME(TAPE) OUTDDNAME(DASD) PURGE
```

Example 1B: A Tracks RESTORE

```
RESTORE TRACKS(1,0,1,5) INDDNAME(TAPE) -
OUTDDNAME(DASD) PURGE
```

Example 1C: A Tracks RESTORE—Restore to Different Tracks

```
//JOBTRKS JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRDSSU
//TAPE DD UNIT=3480,VOL=SER=TAPE04,
// LABEL=(1,SL),DISP=(OLD,KEEP),DSN=USER2.BACKUP
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
RESTORE INDD(TAPE) OUTDYNAM(338000) -
TRKS(200,10,200,10) /* INPUT TRK 200,10 ON SOURCE VOL*/ -
OUTTRKS(100,0) /*RESTORE TO 100,0 ON TARGET VOL */ -
PURGE /* OK TO OVERLAY THE TRK */ -
/* EVEN IF UNEXPIRED DATA SET AT THE LOCATION */ -
PSWD(ABC/WRITPSWD) /* PASSWORD FOR THE DATA SET */
/*
```

A track of dump data is restored to a cylinder and head number other than that from which it was dumped. The dump tape (which might have resulted from a full, tracks, or data set DUMP) contains a track dumped from cylinder 200 head 10 that is restored to cylinder 100 head 0.

Examples of Physical Data Set RESTORE Operations

Example 2 specifies data sets on standard label tape volume TAPE02 are to be restored to DASD volume 111111.

Examples 2A through 2D below complement Examples 2A through 2G in “Examples of Physical Data Set DUMP Operations” on page 80 in any combination. For example, the dump tape produced in Example 2C in “DUMP Command” can be used as the input tape for Example 2A below.

Example 2:

```
//JOB2 JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=A
//TAPE DD UNIT=3480,VOL=SER=TAPE02,
// LABEL=(1,SL),DISP=(OLD,KEEP),DSNAME=USER2.BACKUP
//DASD1 DD UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//SYSIN DD *
command input (see Examples 2A, 2B, ... below)
/*
```

Example 2A: Using the INCLUDE Subkeyword to Restore All Data Sets on a Dump Tape

```
RESTORE INDDNAME(TAPE) OUTDDNAME(DASD1) -
DATASET(INCLUDE(**))
```

Example 2B: Using the INCLUDE and EXCLUDE Subkeywords

```
RESTORE INDDNAME(TAPE) OUTDDNAME(DASD1) -
DATASET(INCLUDE(**) -
EXCLUDE(**.LIST))
```

All data sets with a final qualifier of LIST are restored.

Example 2C: Using the INCLUDE, EXCLUDE, and BY Subkeywords

```
RESTORE INDDNAME(TAPE) OUTDDNAME(DASD1) -
DATASET(INCLUDE(**) -
EXCLUDE(**.LIST) -
BY((EXPDT,LE,80045)))
```


All data sets that satisfy the BY subkeyword except those specified in the EXCLUDE subkeyword are restored.

Example 2D: With Filtering Data in a Data Set

```
RESTORE INDDNAME(TAPE) OUTDDNAME(DASD1) -
        DATASET(FILTERDD(A1))
```

Note to Example 2D: The following DD statement must be added to the JCL shown in Example 2:

```
//A1 DD DSNAME=USER2.FILTER,DISP=SHR
```

This cataloged data set (USER2.FILTER) contains three card-image records. The information shown below is positioned in columns 2 through 72 of each record:

```
INCLUDE(**) -
EXCLUDE(**.LIST) -
BY((DSCHA,EQ,1))
```

Example 2E: Using the LOGICALVOLUME and REPLACE Keywords

```
RESTORE INDDNAME(TAPE) OUTDDNAME(DASD1) -
        DATASET(INCLUDE(**)) LOGICALVOLUME(111111) -
        REPLACE
```

Although the dump tape contains data sets from source volumes 111111 and 222222, only the data sets from source volume 111111 are restored. If movable or unmovable data sets exist on the volume, you must replace the data sets. If no movable data sets exist on the volume, you must allocate the data set and restore.

Example 2F: Using the REPLACE and RENAME Keywords

```
RESTORE INDDNAME(TAPE) OUTDDNAME(DASD1) -
        DATASET(INCLUDE(**)) LOGICALVOLUME(111111) -
        REPLACE -
        RENAME((USER2),(USER4.**,USER3.**))
```

In the above example, renaming takes place only for data sets that exist on DASD with the old name. Data sets with a first-level qualifier of USER4 are renamed to a first-level qualifier of USER3. The first-level qualifiers of all other data sets are replaced by USER2. Unmovable data sets are not renamed.

Example 2G: Using the REPLACE and RENAMEUNCONDITIONAL Keywords

```
RESTORE INDDNAME(TAPE) OUTDDNAME(DASD1) -
        DATASET(INCLUDE(**)) LOGICALVOLUME(111111) -
        REPLACE -
        RENAMEUNCONDITIONAL((USER2),(*.PEAR.**,*PLUM.**), -
                            (MY.SPECIFIC.DS,YOUR.ANY))
```

RENAMEUNCONDITIONAL is used for movable data sets; REPLACE is used for unmovable data sets. With the RENAMEUNCONDITIONAL keyword, movable data sets are renamed whether or not they exist on DASD with the old name. In the example, data sets with a second-level qualifier of PEAR are renamed by using a second-level qualifier of PLUM. MY.SPECIFIC.DS is renamed as YOUR.ANY. The first-level qualifier of all other movable data sets is changed to USER2. Unmovable data sets are not renamed.

Example 2H: RESTORE for All Data Sets

```
RESTORE INDDNAME(TAPE) OUTDDNAME(DASD1) -
        WAIT (1,99) DATASET(INCLUDE(**))
```

Example 2H shows you what to do if the data sets are in use for a short time interval during RESTORE. DFDSS waits for a second at a time and retries as many as 99 times if the data set is in use by another job.

Example 2I: Restore Data Sets

```
RESTORE INDDNAME(TAPE) OUTDDNAME(DASD1) -
        DATASET(INCLUDE(**)) TOL(ENQF) WAIT(0,0)
```

In Example 2I, DFDSS tries to serialize (ENQ) each data set. If the ENQ fails, DFDSS does not wait (WAIT(0,0)), and the data set is processed without serialization or enqueueing (TOL(ENQF)).

Example 3: Using the RENAME Keyword to Restore a VSAM Data Set

```
//JOB3      JOB  accounting information,REGION=nnnnK
//STEP1     EXEC PGM=ADRDSSU
//TAPE      DD  UNIT=3480,VOL=SER=TAPE04,
//          LABEL=(1,SL),DISP=(OLD,KEEP),DSN=USER3.BACKUP
//SYSPRINT DD  SYSOUT=A
//SYSIN     DD  *
RESTORE INDD(TAPE) OUTDYNAM(338000) -
        DS(INCL(PARTS.VSAM1)) -
        RENAME(*.VSAM1,*.VSAM2) -
        CATALOG -
        PSWD(PARTS.VSAM1/MYPSWD)
/*
```

A VSAM key-sequenced data set, PARTS.VSAM1, is restored from a *logical* dump tape in this example. It is renamed as PARTS.VSAM2 and cataloged in the same catalog as PARTS.VSAM1. The cluster's components, PARTS.VSAM1.DATA and PARTS.VSAM1.INDEX, are also renamed.

Example 4: Using the RECATALOG Keyword

```
//JOB4      JOB  accounting information,REGION=nnnnK
//STEP1     EXEC PGM=ADRDSSU
//TAPE      DD  UNIT=3480,VOL=SER=TAPE04,
//          LABEL=(1,SL),DISP=(OLD,KEEP),DSN=USER3.BACKUP
//SYSPRINT DD  SYSOUT=A
//SYSIN     DD  *
RESTORE INDD(TAPE) OUTDYNAM((338001),(338002)) -
        DS(INC(PARTS.**)) /* OR DS(INC(**)) */ -
        PCTU(80) -
        RECATALOG(USERCAT2) -
        PSWD(USERCAT/MASTPSWD) /* PASSWORD OF THE CATALOG */ -
        TGTALLOC(SOURCE)
/*
```

In this example, data sets with a first-level qualifier of *PARTS* were dumped logically. All are restored to volume 338001 and cataloged in catalog USERCAT2. These data sets were on volume 338000 at the time of dump. Because the data sets may not fit on volume 338001, a spill volume, 338002, is specified. To ensure that the data sets on volume 338001 can be extended, 20% of the total space on volume 338001 is to be left as free space (PCTU(80)). The password for the catalog is specified instead of the password for each data set. TGTALLOC(SOURCE) specifies

that the data sets are to be restored with the same allocation type they had when they were dumped.

Examples of Data Set RESTORE Logical Operations

Example 1: A Logical Data Set RESTORE—Output Volumes not Specified

```
//JOB5      JOB  accounting information,REGION=nnnnK
//STEP1     EXEC PGM=ADRSSU
//TAPE      DD  UNIT=3480,VOL=SER=TAPE04,
//           LABEL=(1,SL),DISP=(OLD,KEEP),DSN=USER3.BACKUP
//SYSPRINT  DD  SYSOUT=A
//SYSIN     DD  *
RESTORE INDD(TAPE) -
          DS(INCL(USER1.MULTVOL)) -
          REPLACE
/*
```

In Example 1, data set USER1.MULTVOL is restored. The location to which it is to be restored is not given. This RESTORE statement also applies when the data set has been scratched inadvertently after the DUMP. A multivolume data set is restored to the volumes from which it was dumped, provided the data set is preallocated on the output volumes. If it is not preallocated, the data set is restored to the first volume from which it was dumped that has adequate space to restore the data set as a single-volume data set.

The RESTORE statement can be modified as follows to support multiple restores of both single and multivolume data sets from dump tapes created by one of the above examples:

```
//SYSIN     DD  *
RESTORE INDD(TAPE) /* RESTORE                               */ -
          DS(INCL(USER1.CNTL,**)) /* USER'S CONTROL DATA SETS */ -
          REPLACE          /* OVERLAY DATA SETS IF THEY EXIST */
/*
```

Example 2: A Logical RESTORE of an Unmovable Data Set

```
//JOB6      JOB  accounting information,REGION=nnnnK
//STEP1     EXEC PGM=ADRSSU
//TAPE      DD  UNIT=3480,VOL=SER=TAPE04,
//           LABEL=(1,SL),DISP=(OLD,KEEP),DSN=USER4.BACKUP
//SYSPRINT  DD  SYSOUT=A
//SYSIN     DD  *
RESTORE INDD(TAPE) OUTDYNAM(338000) -
          DS(INCL(HIGH.PERF)) -
          FORCE          /* TO FORCE RESTORE OF UNMOVABLE DATA SET */
/*
```

The unmovable data set, HIGH.PERF, does not exist on volume 338000. This data set existed on volume 338000 at the time of dump. The location, relative to the beginning of the volume, where this data set is to be restored is occupied by other data sets. The data set is marked as unmovable on volume 338000 because either:

- You do not want DFHSM to move it to an unlike device type.
- You do not want the data set to be relocated by DEFRAG for performance reasons.
- It was allocated as an ABSTR data set for performance reasons.

Example 3: A Logical Data Set DUMP, Followed by a RESTORE to an Unlike Device

```
//JOB1      JOB  accounting information,REGION=nnnnK
//STEP1     EXEC PGM=ADRDSSU
//TAPE      DD  UNIT=3480,VOL=TAPE01,
// LABEL=(1,SL),DISP=(NEW,CATLG),DSNAME=USER2.BACKUP
//SYSPRINT  DD  SYSOUT=A
//SYSIN     DD  *
      DUMP DATASET(INCL(USER2.OLDDS)) -
      OUTDD(TAPE)
/*

//JOB2      JOB  accounting information,REGION=nnnnK
//STEP1     EXEC PGM=ADRDSSU
//SYSPRINT  DD  SYSOUT=A
//TAPE      DD  UNIT=3480,VOL=TAPE01,
// LABEL=(1,SL),DISP=(OLD,KEEP),DSNAME=USER2.BACKUP
//DASD     DD  UNIT=3380,VOL=(PRIVATE,SER=222222),DISP=OLD
//SYSIN     DD  *
      RESTORE DATASET( -
      INCLUDE(USER2.OLDDS) ) -
      INDDNAME(TAPE) -
      OUTDDNAME(DASD) -
      RENAME(*.OLDDS,*.NEWDS)
/*
```

In the first part of example 3, DFDSS dumps a cataloged data set (USER2.OLDDS) from a 3350 source volume to an IBM standard label dump tape (TAPE01). Next, DFDSS restores USER2.OLDDS from TAPE01 to a 3380 target volume (DASD volume 222222). The RENAME keyword is used to change the name of the data set to USER2.NEWDS.

Example 4: DUMP/RESTORE for Converting to SMS

```
//SYSPRINT  DD  SYSOUT=*
//DTAPE01   DD  DISP=(,CATLG),DSN=V338001.USER3.BACKUP,
// LABEL=(1,SL),UNIT=3400-6,VOL=SER=TAPE01
//SYSIN     DD  *
      DUMP -
      DS( INC(**) ) -
      LOGINDYNAM ( -
      (338001) -
      ) -
      DELETE COMPRESS -
      OUTDDNAME (DTAPE01)
/*
```

This part of the example dumps all of the data sets on the non-SMS volume 338001 to TAPE01 with the DELETE option. DELETE is required to avoid duplications in the RESTORE.

```
//SYSPRINT  DD  SYSOUT=*
//DTAPE01   DD  DISP=(OLD,KEEP),DSN=V338001.USER3.BACKUP,
// LABEL=(1,SL),UNIT=3400-6,VOL=SER=TAPE01
//SYSIN     DD  *
      RESTORE DS( INC(**) ) -
      STORCLAS(SC01MJA1) -
      INDDNAME (DTAPE01)
/*
```

This second part of the example restores all of the data sets that were dumped in the first half of this example. Because no output volume is specified, most of the data

sets will be allocated on SMS volumes throughout the system. The **STORCLAS** keyword indicates that the storage administrator wants the data sets to have a storage class of **SC01MJA1**. The ACS routines might or might not assign the target data sets the storage class specified. All data sets not converted to SMS (ACS **STORCLAS** routine returns a null storage class) will be restored to the original volume.

Chapter 4. Syntax—Auxiliary Commands

This chapter describes the following **auxiliary** commands you can use to further refine DFDSS processing:

Write-to-Operator (WTO)
SERIAL
PARALLEL
SET.

Write-to-Operator Command

The WTO command lets you write a message to the system console. It can be used, for example, to allow the mounting of a required volume or to stop other jobs in the system before continuing processing of your DFDSS job. The message is limited to 70 characters. The command syntax is:

→ WTO message →

The message that is issued as a result of using the WTO command will not be prefixed with a message ID by DFDSS. If you want a message ID, you must supply it as part of the message text.

One of the following routing codes will be assigned to the message:

- 2 Master console information
- 11 Programmer information

The following descriptor code is assigned to the message:

- 3 Event action required

When a WTO command is encountered, DFDSS waits until the last-requested function command completes before issuing the WTO command.

Scheduling Tasks

The following commands are useful for task scheduling activities:

SERIAL
PARALLEL

The keywords PARALLEL and SERIAL can be truncated to PAR and SER, respectively. No other auxiliary commands can be truncated. There are no alternative abbreviations for auxiliary commands.

SERIAL Command

The SERIAL command allows you to re-initiate serial task scheduling (only one task at a time) after you have used PARALLEL task scheduling. Tasks are processed in the order in which they appear in the input stream.

If neither SERIAL nor PARALLEL has been issued, SERIAL is the default. The command syntax is:

►SERIAL◄

PARALLEL Command

The PARALLEL command lets you initiate parallel task scheduling; that is, two or more tasks are processed concurrently, provided the required system resources (such as virtual storage, DASD, or tape volumes) are available. The commands may not necessarily be processed in the order in which they appear in the input stream. The command syntax is:

►PARALLEL◄

If a switch occurs from SERIAL to PARALLEL mode or vice versa, DFDSS waits for all previously scheduled tasks to be completed before continuing. If PARALLEL is specified but required system resources are not available, some tasks may be delayed in processing until the resources are available (that is, when other tasks have ended). If DFDSS is operating in PARALLEL mode, an IF statement ensures that all prior commands are processed.

Controlling Task Processing

With the *IF-THEN-ELSE-DO-END-SET-EOJ* group of commands, you can direct DFDSS through a logical path in your command sequence, based on the condition (return) codes of previously completed operations.

Condition Codes

The condition codes that are tested in the IF-THEN-ELSE command sequence are:

- 0 Indicates that the function was processed as directed and expected. Informational messages may have been issued.
- 4 Indicates that some problem was encountered in processing the complete function, but it was possible to continue. The continuation may not provide you with exactly what you wanted, but no permanent harm will have been done by such continuation. A warning message was issued.
- 8 Indicates that a requested function did not process, began processing but ended prematurely, or the job ran without processing all requested functions. An error (E) message is issued. If an abend occurs in any of the DFDSS subtasks, the return code is set to 8.
- 12 Indicates that the job did not process. No functions were processed.
- 16 Indicates that a function processed and left at least one volume or data set in an unusable condition. For example, a full volume DUMP ended prematurely, leaving the output tape in an unusable condition.

The condition codes that are tested in the IF-THEN-ELSE command sequence and that can be set by the SET command cannot be passed from one job step to the next. However, the final maximum condition code is passed to the MVS system when DFDSS returns control to the system at the completion of step processing.

Setting Condition Codes

With the SET command, you can set the LASTCC and MAXCC variables to any value from 0 to 16, inclusive. By doing this, you can influence the logical path DFDSS takes in the command sequence following the SET command.

Caution: SET LASTCC should not be used before the first function command.

The condition code of the last-requested operation (LASTCC) and the maximum code of all completed operations (MAXCC) can be determined with an IF command. When an IF MAXCC or SET MAXCC command is encountered, DFDSS waits for *all previously requested function commands* to complete before the highest return code is determined. Also, when an IF LASTCC or SET LASTCC command is encountered, DFDSS waits for the *last-requested function command* to complete before the return code is determined. Following this, if the condition code is tested and satisfied, DFDSS ends the commands in the DO-END group or the single command that comes after the THEN keyword. If the tested condition is *not* satisfied, DFDSS bypasses the DO-END group or single command that comes after the THEN keyword.

SET Command

The syntax of the SET command is:

```

▶▶SET-----▶
   |
   | MAXCC=number
   |-----|
   | LASTCC=number
  
```

SET

specifies that MAXCC or LASTCC is to be set to a new condition-code value. A SET command that occurs within an unprocessed THEN or ELSE clause is not processed.

MAXCC

specifies that the variable MAXCC is to be set to a new condition-code value. Setting MAXCC does not affect the value of LASTCC.

LASTCC

specifies that the variable LASTCC is to be set to a new condition-code value. If the value assigned to LASTCC is higher than the value of MAXCC, MAXCC is also set to the higher value.

number

specifies the value to be assigned to MAXCC or LASTCC. The maximum value that can be assigned is 16; a higher value is reduced to 16.

Examples of Controlling Tasks for the SET Command: The examples that follow show the use of the SET command.

Example 1: If you want to set the last condition code established to 12, specify:

```
SET LASTCC=12
```

Example 2: If you want to replace the highest condition code established in processing so far with 8, specify:

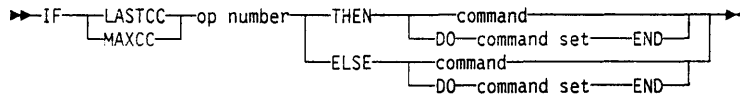
```
SET MAXCC=8
```


Using Condition Codes

Condition codes are used to set up the IF-THEN-ELSE statements. LASTCC specifies a comparison to the last condition code and MAXCC specifies the maximum condition code for comparison.

Using the IF-THEN-ELSE Command Sequence

The syntax of the IF-THEN-ELSE command sequence is:



where:

IF

specifies that a comparison is to be made, the outcome of which determines which logical path (in your command sequence) DFDSS is to take.

LASTCC

specifies that LASTCC (condition code of the last-requested operation) is to be compared to a specified number. LASTCC is initialized to zero upon entry to DFDSS.

MAXCC

specifies that MAXCC (maximum condition code of all completed operations) is to be compared to a specified number. MAXCC is initialized to zero upon entry to DFDSS.

op

specifies an operator that describes the comparison to be made between the condition code specified by LASTCC or MAXCC and the number that follows the operator. This may be any one of six possible comparisons:

- EQ or = Equal to
- LE or <= Less than or equal to
- LT or < Less than
- GT or > Greater than
- GE or >= Greater than or equal to
- NE or ≠ Not equal to.

number

specifies the decimal integer that is to be compared with MAXCC or LASTCC. Values greater than 16 are reduced to 16.

THEN

specifies that a single command or a group of commands (enclosed by DO and END) is to be processed if the tested condition is satisfied. THEN can be followed by another IF command. The ELSE clause can neither be on the same line as the THEN clause, nor can it be on the same line as the continuation of the THEN clause.

ELSE

specifies that a single command or a group of commands (enclosed by DO and END) is to be processed if the tested condition is not satisfied. ELSE can be followed by another IF command.

Nesting IF Commands: When an IF command appears in a THEN or an ELSE clause, it is called a nested IF command. The maximum level of allowed nesting is 10, starting with the first time you specify IF.

Within a nest of IF commands, the innermost ELSE clause is associated with the innermost THEN clause, the next innermost ELSE clause with the next innermost THEN clause, and so on. If there is an IF command that does not require an ELSE clause, use a null ELSE clause (see "NULL Commands," below), unless the nesting structure does not require one. If a nesting structure does not require a null ELSE clause, you will be told so by the DFDSS job stream.

DO

specifies that the group of commands that follow is to be treated as a single unit, that is, to be processed as a result of a single IF command. The set of commands is ended by END. A command following a DO must begin on a new line.

END

specifies the end of a set of commands initiated by the nearest unended DO. END must be on a line by itself.

Examples of Controlling Task Processing for the IF-THEN-ELSE Command: The examples that follow show the use of the IF-THEN-ELSE command sequence.

Example 1: In this example, nested IF commands are used to determine whether a COPYDUMP, EOJ, or PRINT command is to be processed.

```
IF LASTCC > 4 -  
  THEN IF MAXCC < 12 -  
    THEN COPYDUMP ...  
    ELSE EOJ  
  ELSE IF LASTCC = 4 -  
    THEN  
    ELSE PRINT...
```

This example specifies that, if the value of LASTCC is greater than 4, the value of MAXCC is to be tested. If the value of MAXCC is less than 12, the COPYDUMP command is processed, but if the value of MAXCC is 12 or greater, the EOJ command is processed instead. Again, if the value of LASTCC is 4 or less, LASTCC is tested for being exactly 4: No action is to be taken in this case. If, however, LASTCC is less than 4, the PRINT command is to be processed.

Example 2: In this example, nested IF commands are used to determine whether a COPYDUMP or a PRINT command is to be processed.

```
IF LASTCC > 4 -  
  THEN IF MAXCC < 12 -  
    THEN COPYDUMP ...  
    ELSE  
  ELSE IF LASTCC = 4 -  
    THEN PRINT ...
```

Should the first IF command determine that LASTCC is greater than 4 and the second IF command determine that MAXCC is 12 or greater, no function command in the example is processed. The null ELSE command is used here to specify that the next ELSE is to correspond to the first THEN.

Example 3: In this example, if the maximum condition code is zero, a COPYDUMP is to be processed and a data set is to be printed.

```
IF MAXCC=0 -
  THEN DO
    COPYDUMP...
    PRINT ...
  END
```

Creating a Null Statement

If THEN or ELSE is not followed by a continuation character or by a command on the same line, the THEN or ELSE results in a null command. The null command is used to specify that no action is to be taken if the IF clause is satisfied (a null THEN command) or if the IF clause is not satisfied (a null ELSE command). A semicolon after the THEN or ELSE keyword also results in a null command.

If you want to specify a null THEN command, specify:

```
IF ... THEN           or           IF ... THEN;
  ELSE ...             ELSE ...
```

If you want to specify a null ELSE command, specify:

```
IF ... THEN ...       or           IF ... THEN ...
  ELSE                 ELSE;
```

Common Continuation Errors

The continuation rules, described in Chapter 1, "Specifying DFDSS Commands" on page 1, must be used carefully when auxiliary commands, comments, or blank records appear in your input. You must also be careful not to inadvertently specify a null clause when continuing auxiliary commands.

The following examples show common continuation errors.

Error Example 1:

```
IF LASTCC = 0 -
  THEN
  PRINT ...
```

A continuation character (hyphen) is missing after *THEN*; consequently, a null THEN clause is assumed. The PRINT command is unconditionally processed.

Error Example 2:

```
IF LASTCC = 0 -
  THEN -
    COPYDUMP ...
    /* ALTERNATE PATH */
  ELSE -
  PRINT ...
```

Because no continuation character (hyphen) follows the comment, a null ELSE clause is assumed. ELSE is not matched with THEN, and an error message is issued. The PRINT command is ignored. Notice the correct use of the continuation character on the other lines.

Error Example 3:

```
PRINT INDD( - /*COMMENT*/  
DDN1)
```

The DDN1 on the second line is ignored and nothing is printed because characters other than blanks appear after the continuation character (hyphen).

Ending Your DFDSS Job

With the end-of-job (EOJ) command, you can end your DFDSS job after the currently processing operation(s) or scheduled task(s) is completed. The command syntax is:

```
▶▶EOJ◀◀
```

Chapter 5. Stand-alone DFDSS Restore Program

This chapter describes how to use the stand-alone DFDSS restore program and provides examples of job streams to run stand-alone DFDSS restore.

You can use stand-alone DFDSS restore (ADRDMPRS) to perform either a full or a tracks restore from the first data set of DFDSS-produced dump tape(s) without the use of a system environment. Alternatively, you can run the stand-alone DFDSS restore program in a VM/370 virtual machine. ADRDMPRS operates in either ESA/370 mode, System/370 XA mode or System/370 mode.

Compatibility with System DFDSS

The dump format used by both the stand-alone DFDSS restore program and system DFDSS minimizes the time required for DUMP and RESTORE operations and facilitates the restoring of multiple data sets. Stand-alone DFDSS RESTORE can restore any physical dump made by DFDSS as long as the physical dump is the first data set on the tape volume. The dump tapes may contain standard user labels.

Input and Output

DFDSS uses as input:

- Control statements
- DFDSS-produced dump tape(s).

DFDSS produces as output:

- A restored direct-access volume
- A message data set.

Control Statements

The stand-alone DFDSS restore program is controlled by using control statements. Because it is an independent program, operating-system job control statements are not used.

DFDSS control statements consist of:

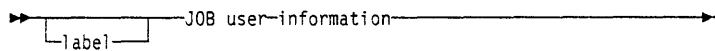
- A JOB statement to begin your job.
- A MSG statement to define an output device for operator messages.
- A RESTORE statement to identify both the tape source volume (containing dumped data to be restored) and the DASD receiving volume.
- An optional VDRL statement to specify the range of tracks to be restored. This is required for a tracks RESTORE only.
- An END statement to end your job.

The label on each statement is optional and, if specified, must start in column 1. If a label is not specified for any of the above statements, each such statement must be preceded by a blank character.

Control statements are coded as card images, and are contained in columns 1 through 71. A statement that exceeds 71 characters must be continued on one or more additional card images. A nonblank character must be placed in column 72 to indicate continuation. A control statement can be interrupted either in column 71 or after any comma. The continued portion of the control statement *must* begin in column 16 of the following card image.

JOB Statement

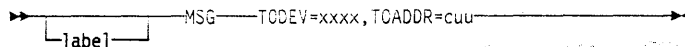
The JOB statement indicates the beginning of a job. The syntax of the JOB statement is:



JOB must be preceded and followed by at least one blank. If *label* is specified, it must start in column 1.

MSG Statement

The MSG statement defines the message data device. It follows the JOB statement and precedes the RESTORE statement. The syntax of the MSG statement is:



where:

MSG must be preceded and followed by at least one blank. If *label* is specified, it must start in column 1.

TCDEV = xxxxx

specifies the type of output device to receive messages. This can be specified generically or by a specific device type. Device types that can be specified are:

- For console: **CNSL** (for console), 1052, 3210, 3215, 3270, or 3277
Specify 1052 for the 3138, 3148, or 3158 console. Specify 3270 for the 3278 Model 2A or 3036 console. Specify 3270 or CNSL only if it is the operator's console. If you have a console other than those listed, specifying CNSL may work. (See "Running the Stand-alone DFDSS Restore Program" on page 139.)
- For printer: **PRNT** (for printer), 1403, 1443, 3211, 3262, or 3800
Specify 3210 for the 3203 printer. Specify PRNT for the 4245 or 4248 printer.
- For tape: **TAPE** (for tape), 2400, 2420, 3400, or 3480.

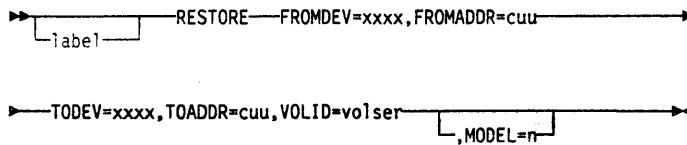
TOADDR = cuu

specifies the channel number, *c*, and unit number, *uu*, of the message output device.

RESTORE Statement

With a RESTORE operation, you can restore a subset of the tracks or all the tracks from a full volume dump tape(s) or from a tracks (one or more track ranges) dump tape(s). (For information on restoring a track or tracks from a physical data set dump tape, see "VDRL Statement" on page 138.)

With the **RESTORE** statement, you specify both the tape source volume (containing dumped data to be restored) and the DASD receiving volume. The syntax of the **RESTORE** statement is:



where:

RESTORE must be preceded and followed by at least one blank. If *label* is specified, it must start in column 1.

FROMDEV = xxxxx

specifies the type of the source device; for example, 2400. Eligible device names are 2400, 2420, 3400, 3424, 3430, 3480, and TAPE. (To use a 3430, specify either 3430 or TAPE. To use a 3480, specify 3480 or TAPE.)

Note: The IBM 3424 Magnetic Tape Device is used for Brazil only. It is a rack-mounted tape unit that uses 0.5-inch tape on 10.5-inch reels; and writes data in densities of either 1600 or 6250 bits per inch.

FROMADDR = cuu

specifies the channel number, *c*, and unit number, *uu*, of the source device.

TODEV = xxxxx

specifies the type of the receiving device; for example, 3380. This device type must be the same as the device type of the volume originally dumped. Eligible devices are: 3330, 3330-1, 2305 (or 2305-2), 3340, 3350, 3375, and 3380 (3344 is specified as 3340).

TOADDR = cuu

specifies the channel number, *c*, and unit number, *uu*, of the receiving device.

VOLID = volser

specifies the volume serial number of the receiving volume. If *volser* does not match the volume serial number found on the receiving volume, the operator is notified and the job is ended.

MODEL = n

specifies a decimal model number (2) for a 2305. If **MODEL** is omitted and **TODEV = 2305** is specified, 2305-2 is assumed.

Restore time can be minimized by selecting devices assigned to different channels. For example:

Col. 16	Col. 72
RESTORE FROMDEV=2400, FROMADDR=282, TODEV=3380,	C
TOADDR=190, VOLID=111111	

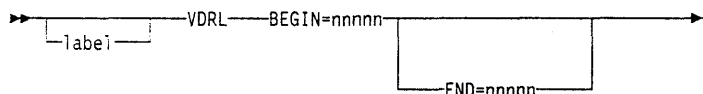
Note: The continuation symbol must appear in column 72 and the next line of text must start in column 16.

VDRL Statement

The optional volume restore limits (VDRL) statement is used to specify the upper and lower limits of a tracks RESTORE. When the VDRL statement is used, it must be preceded by a RESTORE statement and must be followed by an END statement. The upper and lower limits you specify can encompass a subset of the tracks or all of the tracks from the following:

- A full volume dump. If you omit VDRL, you restore the full volume.
- A tracks (one or more track ranges) dump. You *must* specify VDRL in this case.
- The first logical volume of a physical data set dump. You *must* specify VDRL in this case.

The syntax of the VDRL statement is:



where:

VDRL must be preceded and followed by at least one blank. If *label* is specified, it must start in column 1.

BEGIN = nnnnn

specifies a 1 to 5 decimal-digit relative track address (cylinder 0, track 0 = 0) of the first track to be restored.

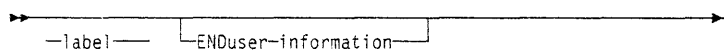
END = nnnnn

specifies a 1 to 5 decimal-digit relative track address of the last track to be restored. If only one track is to be restored, this address is the same as that of the beginning address. If **END** is omitted, the last track of the volume, excluding those tracks reserved as alternates, is assumed to be the upper limit.

Note: The BEGIN and END values can encompass two or more track ranges that have been dumped to tape, even though these values span tracks that were not actually dumped.

END Statement

The END statement marks the end of a job. It appears after the VDRL statement (or RESTORE statement, if there is no VDRL statement). The syntax of the END statement is:



END must be preceded and followed by at least one blank. If *label* is specified, it must start in column 1.

Generic Device Names

Generic device names can be used instead of non-DASD device names when one of the supported devices of that class is intended. For example, if an IBM 2520, 2540, 2501, or 3505 Card Reader is to be used, the generic name CARD may be substituted as follows in response to the ADR505A DEFINE INPUT DEVICE message:

```
INPUT=CARD,00C
```

Similarly, TAPE can be used for tape devices, PRNT for printer devices, and CNSL for consoles.

Copying to Tape

After DFDSS is installed and accepted by System Modification Program (SMP) or System Modification Program Extended (SMP/E) (as described in the Program Directory), the stand-alone DFDSS restore program resides in SYS1.SAMPLIB. The following sample job stream will copy the stand-alone DFDSS restore program to an unlabeled tape:

```
//COPYSA JOB accounting information
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD DSN=SYS1.SAMPLIB(ADRMPRS),DISP=SHR
//SYSUT2 DD DSN=SAMPLIB,UNIT=3480,LABEL=(,NL),
// DISP=(,KEEP),VOL=SER=nnnnnn,
// DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
```

Note: BLKSIZE=80 is required on the tape data set.

Running the Stand-alone DFDSS Restore Program

The stand-alone DFDSS restore program runs independently of the MVS system environment. It is not supported, however, by the IBM 3066 Console, which is only used with the IBM System/370 Model 168.

The stand-alone DFDSS restore program must operate in EC mode. Therefore, before running the stand-alone DFDSS restore program under VM/370, you should enter:

```
CP SET ECMODE ON
```

The stand-alone DFDSS restore program is loaded as card images on an unlabeled tape. Control statements can follow the last card image of the program or can be entered on a separate input device. The data to be restored must be on a tape mounted on a different tape drive. To run the stand-alone DFDSS restore program:

1. Place the stand-alone DFDSS restore program deck in the reader or mount the tape reel that contains the program. Make the device ready.
2. If the DFDSS control statements are to be read from a separate card reader or tape drive, make that device ready.
3. Mount the input tape containing the data to be restored (later specified by the keyword FROMADDR=) and make it ready.
4. Load the program from the reader or tape drive. (Follow the initial program load (IPL) procedure for the processor that will process the DFDSS program.) When the program is successfully loaded, the processor enters the wait state.

The rightmost 2 bytes of the program status word (PSW) of the IPLed processor contain the hexadecimal value FFFF.

5. Define the control statement input device in one of the following ways:
 - a. If a supported, channel-attached, operating console is available:
 - 1) Press the **REQUEST** or **ENTER** key of the console.
 - 2) In response to the message **ADR505A DEFINE INPUT DEVICE**, enter **INPUT = xxxx, cuu** where *xxxx* is device type, *c* is the channel address, and *uu* is the unit address. The device type can be 2400, 2420, 2501, 2520, 2540, 3400, 3424, 3480, 3505, 3270, 1052, **CNSL**, **CARD**, or **TAPE**.
 - b. If a supported, channel-attached, operating console is **not** available:
 - 1) Follow the procedure to alter real storage for the processor and enter at storage location 0110 (hexadecimal): *0cuu* (for example, X'000C0000') for 2540 or 3505 card readers; *1cuu* for 2501 or 2520 card readers; or *2cuu* for 2400 series or 3400 series 9-track tapes or 3480 Tape Subsystems.
 - 2) Cause an **EXTERNAL INTERRUPT**. Follow the procedure required by the processor to generate the external interrupt. Normally the interrupt is generated by pressing a keyboard function key on the processor's system console or pressing a button on the processor's operator control panel.
6. The control statements are now read from the specified device and the restore operation is started. Control statements are printed on the message output device.
7. At the end of the job, **END OF JOB** is printed on the message output device and the processor enters a wait state. The processor's PSW contains the hexadecimal value **EEEE**.
8. When the dump file is on multiple tapes, the following message is issued at the end of each reel:

```
ADR561A END OF TAPE. MOUNT TAPE ON cuu. DEPRESS INTERRUPT KEY
```

To continue the restore at this point, unload the current reel, mount the next reel, ready the device, and cause an external interrupt.

WAIT State Codes

Explanations of the programming status word (PSW) WAIT state codes follow.

Code	Explanation
00E2	WAIT state if a machine check occurs.
1111	Temporary WAIT for completion of input/output.
BCBC	An attempt was made to IPL a processor that does not support EC mode.
6666	WAIT for operator intervention.
EEEE	End WAIT state when the stand-alone DFDSS restore program has completed the job.
FFFF	WAIT for operator input command.
FFFFFF	Error WAIT in IPL loader because one of the following bits is on in the CSW unit status: attention, status modifier, unit check, or unit exception. Register 3 contains the unit status/channel status of the CSW.

Examples of Stand-alone DFDSS Restore Program Jobs

You can use either IBM standard label or nonlabeled tapes as the source volumes for the stand-alone DFDSS restore program operation.

Example 1: A Full RESTORE Operation

A full IBM 3380 DASD volume (serial number 111111) at address 150 is to be restored from IBM 3400 tapes at address 280.

Col.	Col.	Col.
2	16	72
JOB	RESTORE 3380 FROM TAPE	
MSG	TODEV=3210, TOADDR=009	
RESTORE	FROMDEV=3400, FROMADDR=280, TODEV=3380,	C
	TOADDR=150, VOLID=111111	
END		

Example 2: A Tracks RESTORE Operation

An IBM 3380 DASD volume (serial number 111111) at address 150 is to be partially restored from IBM 3480 tapes at address 280.

Col.	Col.	Col.
2	16	72
JOB	TRACKS RESTORE 3380 FROM TAPE	
MSG	TODEV=3210, TOADDR=009	
RESTORE	FROMDEV=3480, FROMADDR=280, TODEV=3380,	C
	TOADDR=150, VOLID=111111	
VDRL	BEGIN=20, END=219	
END		

Example 3: A Tracks RESTORE Operation Using Generic Device Names

Col.	Col.	Col.
2	16	72
JOB	TRACKS RESTORE 3380 FROM TAPE	
MSG	TODEV=CNSL, TOADDR=009	
RESTORE	FROMDEV=TAPE, FROMADDR=280, TODEV=3380,	C
	TOADDR=150, VOLID=111111	
VDRL	BEGIN=20, END=219	
END		

Storage Requirements

The storage requirement, in bytes, to run the stand-alone DFDSS restore program is 220K.

Chapter 6. Customizing DFDSS and Its Interfaces

This chapter explains how you can customize DFDSS by coding exit routines and tailoring Interactive Storage Management Facility (ISMF) panels. Product-sensitive programming interfaces are contained in this chapter.

The installation exits described in this chapter are interfaces which are provided to allow the customer installation to perform tasks such as product tailoring, monitoring, modification, or diagnosis. They are dependent on the detailed design or implementation of DFDSS. Such interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new releases or versions, or as a result of service.

Coding Installation Exit Routines

Installation exit routines supplied with DFDSS are:

Installation Authorization Exit Routine

forces authorization checking of protected data sets.

Installation Enqueue Exit Routine

forces enqueueing of the VTOC for the duration of the full or tracks DUMP or COPY, or tracks PRINT, or, if TOLERATE(ENQFAILURE) is specified, for the duration of a data set DUMP.

Installation Options Exit Routine

does not override any default or user-specified command options in the input stream.

Installation Reblock Exit Routine

during a data set COPY or data set RESTORE, allows DFDSS to use the block size it selects for the target data set.

For MVS/XA installations, all DFDSS exit routines must be link-edited with the ADRDSSU load module with AMODE=31 specified in the link-edit PARMs.

Installation Authorization Exit Routine (ADRUPSWD)

DFDSS supports authorization checking of password- and RACF-protected data sets and volumes. DFDSS provides an exit routine to enable you to control or override the authorization checks (RACF, password, or operator) done by DFDSS.

If the user is APF-authorized (the JSCBPASS bit is set) or has the proper level of RACF DASDVOL access authority, the ADRUPSWD exit is not called. For DEFrag, if a RACF DASDVOL profile exists for the volume being processed, the ADRUPSWD exit is not called. If the DASDVOL profile exists at all, either it allows the volume to be accessed (in which case there is no need to call the exit) or it causes the DEFrag task to fail with a 913 abend (in which case DFDSS cannot call the exit). Otherwise, the exit routine supplied with DFDSS causes DFDSS to:

1. Not relocate protected data sets for which you do not have read access during DEFrag. That is, you must supply a password for each protected data set.
2. Check authorization at the data set level for all functions other than COPYDUMP.

If a data set is protected by using both RACF and a password, only RACF checking is done. Authorization for password-protected non-VSAM data sets is checked by comparing the user-supplied password to that in the PASSWORD data set. Authorization for VSAM data sets is checked by using the VSAM catalog management services. Authorization for VSAM data sets not cataloged in the integrated catalog facility catalog is checked by opening (OPENJ) the data set for read or write. (This OPENJ will fail if the data set has more than 16 extents.)

Some Suggestions for Using the Installation Authorization Exit

One use of the installation authorization exit is to improve DFDSS performance. If you use this exit to bypass authorization checking, you reduce the amount of processing DFDSS does and improve performance. However, you should carefully consider the trade-off between security and performance before you use this exit to improve performance.

Installation-Supplied Authorization Exit Routine

Your installation can supply an exit routine to bypass authorization checking for both non-VSAM and VSAM data sets. The name of the routine must be **ADRUPSWD** and be link-edited to the DFDSS load module **ADRDSSU**. It must be written in reentrant code. (This routine is not given control for a **COPYDUMP** operation.)

The routine is given control at two levels: the volume level and the data set level (except for **DEFRAG**, which is given control at the volume level only). Control is always given at the volume level first. A description of the two levels follows.

Volume Level: At this level, on **DUMP**, **COPY**, **PRINT**, and **RESTORE**, you can determine if the user is to be allowed to proceed with the specified function. During **DEFRAG**, you can determine if protected data sets are to be relocated.

Your installation authorization exit routine can do one of the following:

1. End the operation.
2. Request processing without authorization checking at the data set level.
3. Request that the installation authorization exit be entered at the data set level.

Data Set Level: The routine is given control at this level during **DUMP**, **COPY**, **PRINT**, **RESTORE**, **COMPRESS**, and **RELEASE** operations only if the authorization exit routine instructs DFDSS to check authorization at the data set level. However, the routine is not given control at this level during **DEFRAG**. At the data set level, you can do one of the following:

1. End processing of the volume.
2. End processing of the current data set.
3. Bypass authorization checking of the current data set.
4. Have DFDSS check authorization for the current data set.
5. Also, for 2, 3, and 4 above, you can specify whether the exit routine is to be entered for subsequent data sets.

ADRSSU loads the following registers as input to ADRUPSWD:

Register	Content
1	Pointer to a 4-word parameter list.
13	Address of a 72-byte standard register save area.
14	Return address.
15	Address of ADRUPSWD. This register should be used to pass back a return code.

Standard register save and restore conventions must be observed.

The four word parameter list contains the following. You can use the ADRUPB mapping macro to map this parameter list. See Figure 11 on page 148.

- First word Pointer to the Format 1 DSCB for the protected non-VSAM data set or the cluster name for a VSAM data set to 0 for volume level entry.
- Second word Pointer to the catalog name. Set to 0 for volume level entry.
- Third word Pointer to the volume serial number.
- Fourth word Flags that are set as follows:

Byte 0

Bit	Meaning
0	End of the parameter list. This is always set to 1.
1	Set to 1 if read access is required on the volume. Set to 0 if write access is required.
2	Set to 1 for a DUMP operation.
3	Set to 1 for a RESTORE operation.
4	Set to 1 for a COPY operation.
5	Set to 1 for a PRINT operation.
6	Set to 1 for a DEFrag operation.
7	Set to 1 for a COMPRESS operation.

Byte 1

Bit	Meaning
0	Set to 1 for DEFrag, PRINT, DUMP, or RESTORE for a data set operation.
1	Set to 1 for a full volume DUMP, RESTORE, or COPY operation.
2	Set to 1 for a tracks DUMP, RESTORE, COPY, or PRINT operation.
3	Set to 1 for a RELEASE operation.
4 to 7	Reserved.

Byte 2

Bit	Meaning
0	Set to 1 for volume level entry. Set to 0 for data set level entry.
1	If it is 1, it is entered for a VSAM data set cataloged under the integrated catalog facility. Cluster and catalog names are passed. If it is 0, and byte 2, bit 0 is also 0, it is entered at the data set level for all data sets except VSAM data sets.

Byte 3 Reserved.

Your ADRUPSWD routine should issue a return code in register 15 to indicate the course of action for further processing by DFDSS. The return codes, in decimal, and their meanings are shown in Figure 10 on page 147.

Note: The installation authorization exit routine supplied with DFDSS passes a return code of 12.

The first time ADRUPSWD is entered, the DSNAME address is not passed, but the VOLSER address is passed. If you elect to check the passwords and have ADRDSSU return to you for each data set, then, on the next entry, the DSNAME address is also passed.

An example of how to link-edit your ADRUPSWD routine with DFDSS module ADRDSSU follows. This example assumes that ADRUPSWD is in USERLIB (a cataloged data set).

```
//NAME      JOB (user accounting information)
//          EXEC      PGM=HEWL,
//          PARM='LIST,LET,NCAL,XREF,RENT,AC=1,AMODE=31'
//SYSPRINT  DD        SYSOUT=A
//SYSUT1   DD        UNIT=SYSDA,SPACE=(CYL,(1))
//SYSLMOD  DD        DSN=SYS1.LINKLIB,DISP=OLD
//USER     DD        DSN=USERLIB,DISP=SHR
//SYSLIN   DD        *
INCLUDE    USER(ADRUPSWD)
INCLUDE    SYSLMOD(ADRDSSU)
ENTRY     ADRDSSU
NAME      ADRDSSU(R)
/*
```


1. Volume level entry: All functions except COPYDUMP

Return Code	Meaning		
	End processing for volume?	Data set level checks required?	Should ADRUPSWD be entered at data set level?
0	No	No	No
8	No	Yes	Yes
12	No	Yes	No
20	Yes	N/A	N/A

If a nonzero return code less than or equal to 12 is returned for DEFRAg, all data sets on the volume are to be checked for authorization; any nonzero return code greater than 12 ends the function; if any data sets are protected and you do not have DASDVOL authority a return code of 20 will always end the function.

For all other functions, the function is ended with message ADR402E-15 if a return code other than those listed above is returned by ADRUPSWD.

2. Data set level entry: All functions except DEFRAg and COPYDUMP

Return Code	Meaning			
	End processing for volume?	End processing of data set?	Checking required?	Should ADRUPSWD be entered again?
0	No	No	No	No
4	No	No	No	Yes
8	No	No	Yes	Yes
12	No	No	Yes	No
16	No	Yes	N/A	Yes
20	Yes	N/A	N/A	N/A

Note: N/A means not applicable.

Figure 10. Return Codes From the Installation Authorization Exit Routine

Customizing DFDSS

Offsets	Type	Length	Name	Description	
0	(0)	CHARACTER	16	ADRUPB	
0	(0)	ADDRESS	4	UPDSCBAD	0 IF UPIND IS A 1
F1 DSCB ADDR IF BOTH UPIND AND UPVSMDAT ARE ZERO ADDR OF 44-BYTE CLUSTER NAME IF UPIND IS ZERO AND UPVSMDAT IS A 1					
4	(4)	ADDRESS	4	UPCATP	ADDR OF 44 BYTE CATALOG
NAME IF UPVSMDAT IS A 1					
8	(8)	ADDRESS	4	UPUCBVSA	UCB VOLSER ADDR
12	(C)	BITSTRING	1	UPFLG1	FLAG BYTE 1
		1... ..		UPEOL	END OF LIST INDICATOR, ALWAYS 1
		.1.		UPFRTO	1 = SOURCE VOL, 0 = DEST. VOL
		.1.		UPDUMP	1 = DUMP FUNCTION
		...1		UPREST	1 = RESTORE FUNCTION
	 1...		UPCOPY	1 = COPY FUNCTION
	1..		UPPRINT	1 = PRINT FUNCTION
	1.		UPDEFRAG	1 = DEFRAG FUNCTION
	1		UPCOMPR	1 = COMPRESS FUNCTION
13	(D)	BITSTRING	1	UPFLG2	FLAG BYTE 2
		1... ..		UPDSN	1 = DATASET OPERATION
		.1.		UPFULL	1 = FULL VOLUME OPERATION
		.1.		UPPART	1 = PARTIAL VOLUME OPERATION
		...1		UPRLSE	1 = RELEASE FUNCTION
	 1111		*	RESERVED
14	(E)	BITSTRING	1	UPFLG3	FLAG BYTE 3
		1... ..		UPIND	1 = VOLUME LEVEL CHECK
0 = DATASET LEVEL CHECK					
		.1.		UPVSMDAT	1 = CLUSTER AND CATALOG
NAMES PASSED					
15	(F)	BITSTRING	1	UPFLG4	FLAG BYTE 4 RESERVED
16	(10)	CHARACTER	*	*	FORCE WORD ALIGNMENT

Figure 11. ADRUPB Mapping Macro

Name	Hex Offset	Struct Level
ADRUPB	0	1
UPCATP	4	2
UPCOMPR	C	3
UPCOPY	C	3
UPDEFRAG	C	3
UPDSCBAD	0	2
UPDSN	D	3
UPDUMP	C	3
UPEOL	C	3
UPFLG1	C	2
UPFLG2	D	2
UPFLG3	E	2
UPFLG4	F	2
UPFRTO	C	3
UPFULL	D	3
UPIND	E	3
UPPART	D	3
UPPRINT	C	3
UPREST	C	3
UPRLSE	D	3
UPUCBVA	8	2
UPVSM DAT	E	3

Installation Enqueue Exit Routine

The installation enqueue exit routine supplied with DFDSS works as follows:

- For a full or tracks COPY or DUMP, or tracks PRINT—enqueues the VTOC during the entire operation
- For a data set DUMP (where TOL(ENQF) is not specified—enqueues the VTOC only until the VTOC is processed
- For a data set DUMP (where TOL(ENQF) is specified—enqueues the VTOC during the entire DUMP.

Some Suggestions for Using the Installation Enqueue Exit

You can use this exit to prevent DFDSS from doing enqueue processing. By not doing enqueue processing, you can both improve performance and decrease the chance of deadlock. However, you must carefully consider the trade-offs between data integrity and performance and reduced chance of deadlock.

Installation-Supplied Enqueue Exit Routine

You may want to access a volume while it is being dumped, either by another job under the control of a second initiator or by another processor in a shared DASD environment. If so, your installation can supply an exit routine for the above operations that will cause the VTOC to be enqueued only until it is processed.

The name of the routine must be ADRUENQ and be link-edited with the DFDSS load module ADRDSSU. The routine must be written in reentrant code.

Note: The installation enqueue exit routine will have no effect on the DEFRAG, COPYDUMP, RESTORE, or data set PRINT commands.

DFDSS loads the following registers as input to ADRUENQ:

Register	Content
1	Pointer to a 2-word parameter list.
13	Pointer to a standard 18-word save area.
14	Return address.
15	Address of ADRUENQ.

The two word parameter list contains the following. You can use the ADRUNQB mapping macro, see Figure 12 on page 151, to map this parameter list.

The first word of the parameter list is a pointer to the volume serial field for the volume being dumped. The second word of the parameter list contains flags that are set as follows:

Byte 0

Bit	Meaning
0	End of the parameter list. This is always set to 1.
1	Reserved.
2	Set to 1 for a DUMP operation
3	Reserved.
4	Set to 1 for a COPY operation.
5	Set to 1 for a PRINT operation.
6	Reserved.
7	Reserved.

Byte 1

Bit	Meaning
0	Set to 1 for a data set operation.
1	Set to 1 for a full volume DUMP or COPY operation.
2	Set to 1 for a tracks DUMP, COPY, or PRINT operation.
3 to 7	Reserved.

Byte 2

Bit	Meaning
0	Set to 1 if TOLERATE(ENQFAILURE) is used.
1-7	Reserved.

Byte 3 Reserved.

Your ADRUENQ routine should issue a return code in register 15 to indicate the course of action for further processing by DFDSS. Standard register save and restore conventions must be observed.

The return codes and their meanings are:

Return Code	Meaning
0	Enqueue on the volume being dumped or copied for the duration of the operation.
4	Enqueue on the volume being dumped or copied only for the duration of VTOC access.

An example of how to link-edit your ADRUENQ routine with DFDSS follows. This example assumes that module ADRUENQ is in USERLIB (a cataloged data set).

```
//NAME      JOB (user accounting information)
//          EXEC      PGM=HEWL,
//          PARM='LIST,LET,NCAL,XREF,RENT,AC=1,AMODE=31'
//SYSPRINT  DD        SYSOUT=A
//SYSUT1    DD        UNIT=SYSDA,SPACE=(CYL,(1))
//SYSLMOD   DD        DSN=SYS1.LINKLIB,DISP=OLD
//USER      DD        DSN=USERLIB,DISP=SHR
//SYSLIN    DD        *
            INCLUDE   USER(ADRUENQ)
            INCLUDE   SYSLMOD(ADRSSU)
            ENTRY     ADRSSU
            NAME      ADRSSU(R)
/*
```

Offsets	Type	Length	Name	Description	
0	(0)	CHARACTER	8	ADRUNQB	
0	(0)	ADDRESS	4	UNUCBVSA	UCB VOLSER ADDR
4	(4)	BITSTRING	1	UNFLG1	FLAG BYTE 1
		1... ..		UNEOL	END OF LIST INDICATOR, ALWAYS 1
		.1.. ..	*		RESERVED
		..1. ..	UNDUMP		1 = DUMP FUNCTION
		...1 ..	*		RESERVED
	 1...	UNCOPY		1 = COPY FUNCTION
	1..	UNPRINT		1 = PRINT FUNCTION
	1.	*		RESERVED
5	(5)	BITSTRING	1	UNFLG2	FLAG BYTE 2
		1... ..	UNDSN		1 = DATASET OPERATION
		.1.. ..	UNFULL		1 = FULL VOLUME OPERATION
		..1.	UNPART		1 = PARTIAL VOLUME OPERATION
		...1 1111	*		RESERVED
6	(6)	BITSTRING	1	UNFLG3	FLAG BYTE 2
		1... ..	UNTOLNQF		1 = DATASET OPERATION
7	(7)	CHARACTER	1	*	RESERVED

Figure 12. ADRUNQB Mapping Macro

Name	Hex Offset	Struct Level
ADRUNQB	0	1
UNCOPY	4	3
UNDSN	5	3
UNDUMP	4	3
UNEOL	4	3
UNFLG1	4	2
UNFLG2	5	2
UNFLG3	6	2
UNFULL	5	3
UNPART	5	3
UNPRINT	4	3
UNTOLNQF	6	3
UNUCBVSA	0	2

Installation Options Exit Routine

The installation options exit routine can control the processing of tasks or change some of the defaults or specified options. It is given control initially when DFDSS is invoked and also before processing of each task. During initial entry, it can specify where the I/O and application interface buffers are to reside and it can also change the default SERIAL mode to PARALLEL. When entered for each task, the routine can change the options, defaults, or values (as applicable) listed in Figure 13.

When you are doing a DEFRAG, an extent of a protected data set is relocated and the old extent is cleared (written over with 0's) for security. This adds time to the DEFRAG process. You can prevent the clearing of the old extent with the installation options exit.

The exit routine can force the DUMP function to end if a write error occurs on any of the output copies. This is done by setting bit UFOIACPY on. The default is to continue to produce as many error-free output copies as possible.

The exit routine supplied with DFDSS does not change any defaults, options, or values.

Item to be Changed	Change
ALLDATA(*)/no ALLDATA	to no ALLDATA/ALLDATA(*)
ALLEXCP/no ALLEXCP	to no ALLEXCP/ALLEXCP
ALLMULTI/no ALLMULTI	to no ALLMULTI/ALLMULTI
CATALOG/no CATALOG	to no CATALOG/CATALOG
COMPRESS/no COMPRESS	to no COMPRESS/COMPRESS
COPYVOLID/no COPYVOLID	to no COPYVOLID/COPYVOLID
DELETE/no DELETE	to no DELETE/DELETE
DYNALLOC/no DYNALLOC	to no DYNALLOC/DYNALLOC
FORCE/no FORCE	to no FORCE/FORCE
NULLMGMTCLAS/no NULLMGMTCLAS	to no NULLMGMTCLAS/NULLMGMTCLAS
NULLSTORCLAS/no NULLSTORCLAS	to no NULLSTORCLAS/NULLSTORCLAS
PROCESS(SYS1)/no PROCESS(SYS1)	to no PROCESS(SYS1)/PROCESS(SYS1)
PROCESS(UNDEF)/no PROCESS(UNDEF)	to no PROCESS(UNDEF)/PROCESS(UNDEF)
PURGE/no PURGE	to no PURGE/PURGE
REPLACE/no REPLACE	to no REPLACE/REPLACE
RACFLOG/no RACFLOG	to no RACFLOG/RACFLOG
REBLOCK no REBLOCK	to no REBLOCK/REBLOCK
RECATALOG no RECATALOG	to no RECATALOG/RECATALOG
RESET/no RESET	to no RESET/RESET
SERIAL/PARALLEL	to PARALLEL/SERIAL
SPHERE/no SPHERE	to no SPHERE/SPHERE
TOLERATE(ENQFAILURE)/no TOLERATE(ENQFAILURE)	to no TOLERATE(ENQFAILURE)/TOLERATE(ENQFAILURE)
TGTALLOC/no TGTALLOC	to no TGTALLOC/TGTALLOC
UNCATALOG/no UNCATALOG	to no UNCATALOG/UNCATALOG
WRITECHECK no WRITECHECK	to no WRITECHECK/WRITECHECK
OPTIMIZE, FRAGMENTATIONINDEX, SHARE, and WAIT	Count values

Figure 13. Task Alterations Made with Installation Options Exit

Some Suggestions for Using the Installation Options Exit

You can use this exit to inhibit DFDSS functions. You can then force the use of your own options and controls.

Installation-Supplied Installation Options Exit Routine

Your installation can supply an exit routine to override certain defaults and user-specified parameters in DFDSS commands. The name of the routine must be **ADRUIXIT** and it must be link-edited with the DFDSS load module **ADRDSSU**. It must be written in reentrant code.

DFDSS loads the following registers as input to **ADRUIXIT**:

Register	Content
1	Pointer to a parameter list, illustrated in the ADRUF0 data area below.
13	Pointer to a standard 18-word save area.
14	Return address.
15	Address of ADRUIXIT . This register is also used by you to pass back a return code.

Registers 2 through 13 must be returned unchanged.

The return codes, in hexadecimal, and their meanings are:

Return Code	Meaning
0	No changes were made by this exit routine.
4	The parameter list has been modified.
8	Do not schedule this function (valid only for function command entry).

This exit is entered for the following reasons only, during processing of DFDSS:

During program initialization. This is called the *PARM change* entry. (The **UFFUNCT** field contains all zeros.)

Just before scheduling a function command. This is called the *function* entry. (The function type is indicated by the **UFFUNCT** field.)

You can use the **ADRUF0** mapping macro, see Figure 14 on page 155, to map the parameter list. The parameter list consists of the following sections:

UFOHDR

contains information describing the type of entry and, for a function entry, details of the function to be scheduled. It also contains an offset to **UFFUNCT** or **UFOPARM** and the addresses of **UFOVOL** for input and **UFOVOL** for output.

UFOFUNCT

contains information about the function to be scheduled and which can be altered by this exit routine. The bits are described in the parameter list. It is created for the function entry.

UFOPARM

contains bits that can be set to change defaults or override commands. The bits are described in the parameter list. It is created for the **PARM change** entry.

UFOVOL

is an array of entries describing the volumes used for input and output for the specific function commands.

Figure 14 (Page 1 of 3). ADRUFO Mapping Macro

Offsets	Type	Length	Name	Description	
0	(0)	CHARACTER	20	ADRUF0B	
0	(0)	CHARACTER	20	UFOHDR	HEADER OF UFO PARM LIST
0	(0)	CHARACTER	4	UFID	IDENTIFIER EBCDIC "UFO "
4	(4)	FIXED	2	UFLEN	LENGTH OF PARM LIST
6	(6)	FIXED	2	UFB DYOFF	OFFSET TO UFOFUNCT OR UFOPARM
8	(8)	ADDRESS	4	UFVOLI@	ADDR OF INPUT VOL LIST 0
12	(C)	ADDRESS	4	UFVOLO@	ADDR OF OUTPUT VOL LIST 0
16	(10)	BITSTRING	2	UFFUNCT	FUNCTION BEING PERFORMED '0000'X = PARM CHANGE ENTRY (THIS FIELD MUST BE KEPT IN SYNCHORNIZATION WITH FBMAJOR IN ADRFUNCB)
16	(10)	BITSTRING	1	UFFUNCT1	FUNCTION BYTE ONE
		1... ..		UFFUDEF	1 = DEFRAG OPERATION
		.1.. ..		UFFUCOPY	1 = COPY OPERATION
		..1.		UFFUREST	1 = RESTORE OPERATION
		...1		UFFUDUMP	1 = DUMP OPERATION
	 1...		UFFUPRT	1 = PRINT OPERATION
	1..		UFFUCPYD	1 = COPYDUMP OPERATION
	1.		*	RESERVED
	1		UFFUCOMP	1 = COMPRESS OPERATION
17	(11)	BITSTRING	1	UFFUNCT2	FUNCTION BYTE TWO
		1... ..		UFFURLSE	1 = RELEASE OPERATION
		.1..		UFFUCONV	1 = CONVERTV OPERATION
		..11 1111		*	RESERVED
18	(12)	CHARACTER	1	UFFIND	FUNCTIONAL INDICATORS
		1... ..		UFFIFULL	1 = FULL VOLUME REQUEST (DUMP, RESTORE, COPY & DEFRAG)
		.1..		UFFIPART	1 = PARTIAL REQUEST (DUMP, RESTORE, COPY, DEFRAG & PRINT)
		..1.		UFFIFILT	1 = REQUEST BY FILTER/DSNAME (DUMP, RESTORE & PRINT)
		...1		UFFIPRTV	1 = PRINT VTOC
	 1...		UFFLOGCL	1 = LOGICAL DUMP EITHER NO INPUT VOLUMES SPECIFIED (CATALOG FILTERING) OR LOGINDDNAME, LOGINDYNAM SPECIFIED
	111		*	RESERVED
19	(13)	CHARACTER	1	UFAIFLGS	APPLICATION INTERFACE FLGS
		1... ..		UFAINV	1=INVOKED BY APPL. INTERF.
		.1..		UFUIMAL	1=ADRUIM NOT TO BE GIVEN CONTROL
		..1.		UFUIMCH	1=DO NOT ALLOW ADRUIM TO MODIFY OPTIONS. VALUES
		...1		UFSTOP	1=DO NOT SCHEDULE TASK (SAME AS RETURN CODE 8 FROM ADRUEXIT)
	 1...		UFSYSIN	1=SYSIN OR ALTERNATE NOT PRESENT ALLOWED IF UFPARAM=XX1000XX
	1..		UFSYSPR	1=SYSPRINT/ALTERNATE NOT PRESENT ALLOWED IF UFPARAM=XX1000XX
	1.		UFNOIN	1=NO INPUT TAPE ONLY FOR RESTORE
	1		UFNOOUT	1=NO OUTPUT TAPE ONLY FOR DUMP
20	(14)	CHARACTER	*	*	
0	(0)	CHARACTER	28	UFOFUNCT	LIST OF OPTIONS FOR FUNCTION. POINTED TO BY THE ADDRESS OF UFOHDR + UFB DYOFF. PRESENT IF ANY BITS ARE ON IN UFFUNCT
0	(0)	BITSTRING	1	UFO1FLGS	1ST SET OF OPTION FLAGS
		1... ..		UFO1COMP	1 = COMPRESS (DUMP)
		.1..		UFO1CVOL	1 = COPYVOLID (RESTORE & COPY)
		..1.		UFO1PURG	1 = PURGE (DUMP, COPY & RESTORE)
		...1		UFO1RESE	1 = RESET CHANGE BIT (DUMP)
	 1...		UFO1WRCK	1 = WRITECHECK
	1..		UFO1ALD	1 = ALLDATA
	1.		UFO1ALDL	1 = ALLDATA(LIST), 0 = ALLDATA(.), VALID ONLY IF UFO1ALD = 1. BIT MAY NOT BE SET ON BY THE EXIT BUT MAY BE RESET TO CHANGE ALLDATA(LIST) TO ALLDATA(.)
	1		UFO1ALLE	1 = ALLEXCP
1	(1)	BITSTRING	1	UFO2FLGS	2ND SET OF OPTION FLAGS
		1... ..		UFO2DYNQ	0 = USE ENQ TO HOLD DATASET, 1 = USE DYNALOC TO HOLD DS (DUMP, DEFRAG, PRINT & RESTORE)
		.1..		UFO2ENQE	1 = ENQ EXCLUSIVE (DUMP, RESTORE AND PRINT)

Figure 14 (Page 2 of 3). ADRUFO Mapping Macro

Offsets	Type	Length	Name	Description
	..1.		UFO2ENQS	1 = ENQ SHARED IF EXCL FAILS DUMP, RESTORE & PRINT
	...1		UFO2ENQN	1 = DO NOT ENQ IF EXCL & SHR FAIL DUMP, RESTORE & PRINT
 1..		UFO2DEL	1 = DELETE AFTER DS DUMP
1..		UFO2CTLG	1 = CATALOG DATA SETS DURING A DATA SET RESTORE, COPY
1..		UFO2RECT	1 = RECATALOG DURING A DATA SET RESTORE, COPY
1.		UFO2UNC	1 = UNCATALOG DATA SETS AFTER A DATA SET DUMP, COPY
1		*	RESERVED
2	(2) FIXED	1	UFDOPTM	OPTIMIZE VALUE (DUMP/COPY) (1, 2, 3, OR 4)
3	(3) BITSTRING	1	UFOINSOP	INSTALLATION OPTIONS
	1...		UFOERASE	1 = ERASE DASD TRACKS
	.1..		UFOIACPY	1 = DUMP MUST PRODUCE ALL OUTPUT COPIES OR NONE AT ALL
	..1.		UFOBLDIX	1 = INVOKE ICKDSF TO REBUILD VTOC INDEX
	...1		UFORACLG	RACFLAG = YES SPECIFIED OR FORCE RACF LOGGING
 1..		UFOBK32K	TAPE BLK SIZE 32K
111		*	RESERVED
4	(4) FIXED	4	UFOFRAGI	FRAGMENTATION INDEX(DEFRAG) 9,90,900=900, 09=90, 009=9
8	(8) FIXED	4	UFOMNGAS	MINIMUM WASTED SPACE ON A PDS COMPRESSPDS
12	(C) FIXED	4	UFOMNSQT	MINIMUM SECONDARY ALLOCATION QUANTITY
16	(10) FIXED	4	UFOMNTUS	MINIMUM UNUSED TRACKS RELEASEUNUSED, CHANGEALLOC
20	(14) BITSTRING	1	UFO3FLGS	3RD OPTION FLAG BYTE
	1...		UFO3FORC	1 = FORCE UNMOVABLES ON COPY, RESTORE
	.1..		UFO3REPL	1 = REPLACE(DATASET COPY, RESTORE)
	..1.		UFOFRBLK	1 = FORCE REBLOCKING OF DATA SETS (COPY, RESTORE)
	...1		UFODRBLK	1 = DISABLE REBLOCKING OF DATA SETS (COPY, RESTORE)
 1..		UFOALLMU	1=SEARCH ALL VOLUMES FOR COPY OR DUMP
1..		UFOSPHER	1=PERFORM SPHERE PROCESSING
1..		UFONOSMS	1=NULLSTORCLAS SPECIFIED
1..		UFONMGMT	1=NULLMGMTCLAS SPECIFIED
21	(15) FIXED	1	UFOWAITS	WAIT TIME IN SECONDS BETWEEN RESERVE & ENQ
22	(16) FIXED	1	UFOWAITR	NUMBER OF RETRIES ON RESERVE OR ENQ
23	(17) BITSTRING	1	UFOTGTAL	TGTALLOC FLAGS
	1...		UFOTGTCY	1 = CYLINDER
	.1..		UFOTGTTR	1 = TRACK
	..1.		UFOTGTBL	1 = BLOCK
	...1		UFOTGTSR	1 = SOURCE
 1111		*	RESERVED
24	(18) BITSTRING	1	UFOPROCK	PROCESS OPTIONS (THIS FIELD MUST BE KEPT IN SYNC WITH FBPROCKW)
	1...		UFOPRUND	1 = PROCESS UNDEFDSORG
	.1..		UFOPRSYS	1 = PROCESS SYS1
	..11 1111		*	UNDEFINED
28	(1C) CHARACTER			
0	(0) CHARACTER	16	UFOPARM	EXECUTE CARD PARAMETER OPTION LIST. POINTED TO BY THE ADDRESS OF UFOHDR + UFBDYOFF. PRESENT IF ALL BITS ARE OFF IN UFFUNCT
0	(0) CHARACTER	1	UFSEPAR	SERIAL / PARALLEL
	1...		UFFORSER	1 = FORCE TO SERIAL
	.1..		UFDEFPAR	1 = DEFAULT TO PARALLEL. IGNORED IF UFFORSER IS 1.
	..11 1111		*	RESERVED
1	(1) CHARACTER	1	UFXAFLAG	CNTL FLAGS FOR XA MODE
	1...		UFABUFF	1=I/O BUF ABOVE 16M REQ
	.1..		UFAI31B	1=A1 BUF ABOVE 16M REQ
	..11 1111		*	RESERVED
2	(2) CHARACTER	8	UFWKUNIT	WORKUNIT PARAMETER
10	(A) CHARACTER	6	UFWKVOL	WORKVOL PARAMETER
16	(10) CHARACTER		*	

Figure 14 (Page 3 of 3). ADRUFO Mapping Macro

Offsets	Type	Length	Name	Description	
0	(0)	CHARACTER	*	UFOVOL	VOLUME LIST HEADER. POINTED TO BY UFVOLI AND UFVOLO
0	(0)	CHARACTER	4	UFOVHDR	HEADER
0	(0)	FIXED	2	UFOVCNT	COUNT OF VOLUME LIST ENTRIES
2	(2)	CHARACTER	2	*	RESERVED
4	(4)	CHARACTER	20	UFOVENT (*)	VOLUME LIST ENTRY
4	(4)	ADDRESS	4	UFOVUCB@	ADDR OF UCB 0
8	(8)	CHARACTER	8	UFODDNAM	DDNAME BLANK
16	(10)	CHARACTER	6	UFOVOLID	VOLUME SERIAL BLANK
22	(16)	CHARACTER	2	*	RESERVED

Name	Hex Offset	Hex Value	Struct Level	Name	Hex Offset	Hex Value	Struct Level
ADRUF0B	0		1	UF0TGTCY	17		3
UF0IFLGS	13		3	UF0TGTSR	17		3
UF0IINV	13		4	UF0TGTR	17		3
UF0I3IB	1		3	UFOVCNT	0		3
UFBDYOFF	6		3	UFOVENT	4		2
UFDEFPAR	0		3	UFOVHDR	0		2
UFDUOPTM	2		2	UFOVOL	0		1
UFFIFILT	12		4	UFOVOLID	10		3
UFFIFULL	12		4	UFOVUCB@	4		3
UFFIND	12		3	UFOWAITR	16		2
UFFIPART	12		4	UFOWAITS	15		2
UFFIPRTV	12		4	UFO1ALD	0		3
UFFLOGCL	12		4	UFO1ALDL	0		3
UFFORSER	0		3	UFO1ALLE	0		3
UFFUCOMP	10		5	UFO1COMP	0		3
UFFUCONV	11		5	UFO1CVOL	0		3
UFFUCOPY	10		5	UFO1FLGS	0		2
UFFUCPYD	10		5	UFO1PURG	0		3
UFFUDEF	0		5	UFO1RESE	0		3
UFFUDLMP	10		5	UFO1WRCK	0		3
UFFUNCT	10		3	UFO2CTLG	1		3
UFFUNCT1	10		4	UFO2DEL	1		3
UFFUNCT2	11		4	UFO2DYNQ	1		3
UFFUPRT	10		5	UFO2ENQE	1		3
UFFUREST	10		5	UFO2ENQN	1		3
UFFURLSE	11		5	UFO2ENQS	1		3
UFID	0		3	UFO2FLGS	1		2
UFLEN	4		3	UFO2RECT	1		4
UFNOIN	13		4	UFO2UNC	1		3
UFNOOUT	13		4	UFO3FLGS	14		2
UFOALLMU	14		3	UFO3FORC	14		3
UFOBK2K	3		3	UFO3REPL	14		3
UFOBLDIX	3		3	UFSERPAR	0		2
UFODDNAM	8		3	UFSTOP	13		4
UFODRBLK	14		3	UFSYSIN	13		4
UFOERASE	3		3	UFSYSR	13		4
UFOFRAGI	4		2	UFUIMAL	13		4
UFOFRBLK	14		3	UFUIMCH	13		4
UFOFUNCT	0		1	UFVOLI@	8		3
UFOHDR	0		2	UFVOLO@	C		3
UFOIACPY	3		3	UFWKUNIT	2		2
UFOINSOP	3		2	UFWKVOL	A		2
UFOMNGAS	8		2	UFXABUFF	1		3
UFOMNSQT	C		2	UFXAFLAG	1		2
UFOMNTUS	10		2				
UFONMGMT	14		3				
UFONOSMS	14		3				
UFOPARM	0		1				
UFOPROCK	18		2				
UFOPRSYS	18		3				
UFOPRUND	18		3				
UFORACLG	3		3				
UFOSPHER	14		3				
UFOTGTAL	17		2				
UFOTGTBL	17		3				

Installation Reblock Exit Routine

The installation reblock exit routine is given control when a SAM or PAM data set is being processed during a data set COPY or RESTORE if either:

- You specify REBLOCK(dsn [...]) in the COPY or RESTORE statement, the installation options exit routine (ADRUIXIT) does not override it, and the data set being copied meets the filtering criteria passed in the REBLOCK sublist (fully- or partially-qualified data set names).
- The installation options exit routine forces all data set COPY or RESTORE functions to use the reblock exit routine.

Note: The installation reblock exit is not given control for RECFM=U data sets. However, PDS load module data sets are reblocked, if requested, by IEBCOPY COPYMOD.

The installation options exit routine cannot change the REBLOCK sublist (fully or partially qualified data set names), except:

- It can prevent DFDSS from reblocking if you specified REBLOCK(dsn [...]), and
- It can force DFDSS to reblock all SAM/PAM data sets being copied (equivalent to REBLOCK(**)).
- If the reblockable indicator is on in the data set's Format 1 DSCB (DS1REBLK), DFDSS automatically reblocks the data set to a system determined blocksize. The reblock exit routine is not called.

The DFDSS reblock exit routine passes a return code of zero, which allows DFDSS to use an optimal system generated block size.

Some Suggestions for Using the Installation Reblock Exit

You can use this exit to verify or change blocksize. For DFP Version 3 Release 1, the reblock exit can set a new reblockable indicator that will cause the data set to be automatically reblocked when the data set is copied by DFDSS or another program.

Installation-Supplied Reblock Exit Routine

Your installation can replace the DFDSS exit routine to override the DFDSS-selected block size. If the exit routine passes an invalid block size, the data set is not processed and message ADR453E is issued.

The name of the reblock exit routine must be ADRREBLK, it must be link-edited with the DFDSS load module ADRDSSU, and it must be written in reentrant code.

DFDSS loads the following registers as input to ADRREBLK:

- Register 1: Pointer to a 6-word parameter list.
- Register 13: Pointer to a standard 18-word save area.
- Register 14: Return address.
- Register 15: Address of ADRREBLK.

The six word parameter list contains the following information. You can use the ADRRBLKB mapping macro, see Figure 15 on page 159, to map this parameter list.

First word—

Pointer to a 140-byte work area that has the Format 1 DSCB. For multivolume data sets, the extent fields are for the first volume only.

Second word—

Pointer to the 6-byte volume serial of the input volume. For a multivolume data set, this volume serial is for the first input volume.

Third word—

Pointer to the 4-byte device type field of the input volume. This is in the format of the UCBTYP field in the UCB.

Fourth word—

Pointer to the 6-byte volume serial of the output volume. For a data set that spans multiple output volumes, the serial is for the first volume.

Fifth word—

Pointer to the 4-byte device type field of the output volume. Its format is similar to that for the input volume.

Sixth word—

High-order bit (bit 0) is on to indicate the end of the parameter list. The next bit (bit 1) indicates that the MVS/XA DFP Version 2 DASDCALC function was used to calculate the optimum block size for the target data set.

The 3rd and 4th bytes of the sixth word contain the block size selected by DFDSS for the data set on the output volume. This is in the same format as the DS1BLKL field in the Format 1 DSCB (fixed binary).

Your ADRREBLK routine should issue a return code in register 15 to indicate the course of action for further processing by DFDSS. Standard register save and restore conventions must be observed.

The return codes and their meanings are:

Return Code Meaning

0	Block size not changed. Use the DFDSS-selected block size.
4	Block size has been changed by the exit; the new block size is indicated in last 2 bytes of word 6 of the parameter list.
8	Use the input block size (do not reblock).
12	System determined block size will be used. In addition, the reblockable indicator in the Format 1 DSCB (DS1REBLK) will be turned on.

The example below shows you how to link-edit your ADRREBLK routine. This example assumes that module ADRREBLK is in USERLIB (a cataloged APF authorized library).

```
//NAME          JOB (user accounting information)
//              EXEC      PGM=HEWL,
//              PARM='LIST,LET,NCAL,RENT,AC=1,XREF,AMODE=31'
//SYSPRINT      DD          SYSOUT=A
//SYSUT1        DD          UNIT=SYSDA,SPACE=(CYL,(1))
//SYSLMOD       DD          DSN=SYS1.LINKLIB,DISP=OLD
//USER          DD          DSN=USERLIB,DISP=SHR
//SYSLIN        DD          *
  INCLUDE       USER(ADRREBLK)
  INCLUDE       SYSLMOD(ADRSSU)
  ENTRY        ADRSSU
  NAME         ADRSSU(R)
/*
```

Offsets	Type	Length	Name	Description
0	(0)	24	ADRRBLKB	
0	(0)	4	RBDSCB@	PTR TO FORMAT 1 DSCB
4	(4)	4	RBIVOL@	PTR TO INPUT VOLSER
8	(8)	4	RBIDEV@	PTR TO INPUT DEVICE TYPE
12	(C)	4	RBOVOL@	PTR TO OUTPUT VOLSER
16	(10)	4	RBODEV@	PTR TO OUTPUT DEVICE TYPE
20	(14)	1	RBEND	LAST WORD IN PARM LIST
	1... ..		RBENDLST	END OF PARAMETER LIST
	.1.. ..		RBDASDCS	1=DASD CALCULATION SERVICE
	..1.		RBRSTCPY	1= CALLED FOR RESTORE 0= CALLED FOR COPY
	...1 1111		*	UNUSED
21	(15)	1	*	UNUSED
22	(16)	2	RBLKSIZE	SELECTED BLOCKSIZE

Figure 15. ADRRBLKB Mapping Macro

Name	Hex Offset	Struct Level
ADRRBLKB	-0	1
RBDASDCS	14	3
RBDSCB@	0	2
RBEND	14	2
RBENDLST	14	3
RBIDEV@	8	2
RBIVOL@	4	2
RBLKSIZE	16	2
RBODEV@	10	2
RBOVOL@	C	2
RBRSTCPY	14	3

Customizing ISMF

There may be times when you wish to customize ISMF. Following are the libraries you can customize and some restrictions when you customize. *MVS/DFP Version 3 Release 2: Customization* discusses ISMF restrictions and customization with DFDSS in detail.

ISMF Libraries That Can Be Customized

You can customize the following libraries of ISMF:

The Panel Library

ISMF allows you to make the following changes to the panel library:

- Change the initial priming values that ISMF ships
- Change the default values for data entry panels
- Provide additional restrictions to values entered for certain fields on panels
- Remove fields from functional panels
- Change highlighting and color
- Change the format of the panel
- Modify existing functional panel text and help text
- Add new fields to panels
- Add new panels.

The Message Library

In the message library you can modify existing messages and add new messages.

The Skeleton Library

In the skeleton library you can modify the job skeletons for ISMF commands and line operators.

The Table Library

In the table library you can modify the ISPF command tables.

The Load Library

In the load library you can modify the ISMF command and line operator tables. The tables are contained in nonexecutable CSECTS in the load library.

The CLIST Library

In the CLIST library you can modify the options on the CLIST CONTROL statement.

Restrictions to Customizing ISMF

Following are some general restrictions to customizing ISMF:

1. Before you change anything, you should make a backup copy of ISMF. Keep this unmodified version of the product for diagnostic purposes. IBM support and maintenance is provided for **only the unmodified version of ISMF**.
2. Do not delete or rename any of the parts of ISMF. Deleting or renaming a part could severely impact processing or cause ISMF to fail.
3. ISMF is copyrighted. Under the IBM licensing agreement you may modify ISMF for your own use. You may not modify it for commercial resale.

Appendix A. Compatibility Considerations for Prior Releases

Access Authority

The type of access authority required by DFDSS Version 2 for processing **RACF-protected data sets and password-protected data sets** is the same as in DFDSS Version 1 Releases 2.0 and 2.1. In addition, DFDSS Version 2 and DFDSS Version 1 Release 2.1 support data sets protected by RACF generic profiles. If you do not have DATASET access authority to a multivolume data set, you must have DASDVOL access authority to all volumes on which the data set resides.

Deleting Data Sets

In Version 2, if DELETE is specified on a data set COPY, a successfully processed data set is scratched from all the volumes on which it resides and is uncataloged. (On a data set DUMP, this is true only for a logical dump.) Otherwise, it processes as in Version 1 Release 2.1, where only single volume non-VSAM data sets are scratched and uncataloged. In Version 1 Release 1 through Release 2.0, non-VSAM data sets are scratched but not uncataloged from the source dump volume.

Data Set DUMP

In Version 1, not specifying the input volume(s) resulted in an error message. In Version 2 Releases 1 and 2, not specifying the input volumes resulted in a logical data set DUMP. The data sets and the volumes to be processed were determined from the catalog(s). In Version 2 Releases 3 and 4, a logical data set dump can be done on a volume if you use LOGINDD or LOGINDDY.

Data Set RESTORE

In Version 1, not specifying the output volume(s) resulted in an error message. In Version 2, if output volume(s) are not specified, the target volume(s) for the RESTORE are determined from either:

- The information kept on the dump tape if the data set does **not** exist in the catalog

- The catalog (standard catalog search order) if the data set **does** exist. In this case, REPLACE must be specified.

In DFDSS Version 2, you can do a data set RESTORE from full volume or data set dump tapes created by Version 1. The following limitations apply:

- Because multivolume data sets were not supported on data set dumps in Version 1, these data sets cannot be restored in their entirety. Also, multivolume data sets will not be cataloged during a RESTORE.

- DFDSS will not be able to redefine the VSAM data sets. You may have to define them with the RECATALOG option.

LOGICALVOLUME

In Version 2, the LOGICALVOLUME keyword is ignored when you do a COPYDUMP on data from a Version 2 logical dump. The COPYDUMP function cannot isolate data by volume from these tapes.

WORKSIZE

In Version 2, the DEFrag WORKSIZE keyword is ignored.

VSAM key range data sets

In Version 2, data set COPY, logical data set DUMP, logical data set RESTORE, and PRINT can process VSAM key range data sets. The cluster names of these data sets should be specified in the EXCLUDE keyword if they are to be excluded from processing.

PRINT

In Version 2, a data set PRINT results in printing tracks in the logical sequence of the data set on the volume, not by cylinder-head sequence. Also, you can print VSAM components.

Restoring by Using a Different DFDSS Release

Tapes dumped by using an older version or release of DFDSS can be restored by using a newer version or release. IBM does not recommend *and does not guarantee* that tapes dumped by using a newer version or release of DFDSS can be restored by using an older release. However, if the dump tape does not contain any functional changes introduced in a newer release, a restore using an older release may be successful.

Note: VSAM data sets residing on DFDSS data set Dump tapes created in an MVS/XA DFP Version 2 Release 2 (or later release) system cannot be restored in an MVS/XA DFP Version 2 Release 1 or an MVS/370 DFP Version 1 system without certain VSAM N*512 toleration PTFs.

Figure 16 on page 163 shows whether a tape dumped using one DFDSS release can be restored using a different release. Figure 17 on page 163 shows similar information for stand-alone DFDSS restore.

Dump From	RESTORE From V1 R2.1			RESTORE From V2		
	Full	Tracks	Data Set	Full	Tracks	Data Set
V1 R2.1 Full	Yes	Yes	Yes	Yes	Yes	Yes
V2 Full	Yes	Yes	Yes	Yes	Yes	Yes
V1 R2.1 Tracks	No	Yes	No	No	Yes	No
V2 Tracks	No	Yes	No	No	Yes	No
V1 R2.1 Physical data set	No	Yes	Yes	No	Yes	Yes
V2 Physical data set	No	Yes	Yes	No	Yes	Yes
V2 Logical data set	No	No	No	No	No	Yes ⁸

Figure 16. System DFDSS RESTORE Compatibility among Releases

Dump From	RESTORE With V1 R2.1		RESTORE With V2	
	Full	Tracks	Full	Tracks
V1 R2.1 Full	Yes	Yes	Yes	Yes
V2 Full	Yes	Yes	Yes	Yes
V1 R2.1 Tracks	Yes	Yes	Yes	Yes
V2 Tracks	Yes	Yes	Yes	Yes
V1 R2.1 Physical data set	No	Yes	No	Yes
V2 Physical data set	No	Yes	No	Yes
V2 Logical data set	No	No	No	No

Figure 17. Stand-alone DFDSS RESTORE Compatibility Among Releases

Legend for Figure 16 on page 163 and Figure 17:

- V1 R2.1 DFDSS Version 1 Release 2.1
- V2 DFDSS Version 2 (all releases)

⁸ Tapes dumped by using an older version or release of DFDSS can be restored by using a newer version or release. IBM does not recommend **and does not guarantee** that tapes dumped by using a newer version or release of DFDSS can be restored by using an older release. However, if the dump tape does not contain any functional changes introduced in a newer release, a restore using an older release may be successful. In particular, restoring VSAM data sets using a previous DFDSS release is not recommended.

Appendix B. Serialization and Authorization Checking

Data Integrity—Serialization

DFDSS uses **volume serialization** and **data set serialization** functions to ensure that data sets are not modified during the processing of DFDSS commands. Volume serialization is accomplished by using the RESERVE macro. Data set serialization is accomplished by using the ENQ macro and the DFDSS DYNALLOC function. In case of shared DASD, volume serialization ensures that data sets are not being added, deleted, renamed, or extended from either the same processor or other processors.

You can override data set serialization with the user enqueue exit routine. (See Chapter 6, “Customizing DFDSS and Its Interfaces” on page 143 for additional information regarding installation enqueue exit routines.) This allows other programs to access volumes or data sets while DFDSS is running. It does not affect the integrity of any data set other than the one currently being processed. Volume serialization, however, cannot be overridden.

Volume Serialization

For volume serialization, DFDSS issues a RESERVE on the SYSVTOC to prevent direct access device storage management (DADSM) functions from changing the VTOC entries during DFDSS processing. The RESERVE is issued before processing is begun on a particular volume and released when processing is completed. However, DFDSS does not issue a RESERVE for a COPYDUMP, data set RESTORE, or data set PRINT operation. You will notice that volume serialization by itself does not ensure integrity at the data set level. For example, in the case of a dump other than a data set DUMP, data sets are not serialized, and they could be changed while a DUMP is being taken.

DFDSS reserves a VTOC when it is accessed during COMPRESS, RELEASE, and COPY (with input volume specified). It is also reserved when updating the RACF-defined flag.

DFDSS operates at the volume and data set levels. To prevent activity on the VTOC at the volume level (such as addition, deletion, extension, or renaming of data sets), the VTOC is locked during COPY, DEFrag, DUMP, PRINT (except data set prints), and RESTORE (tracks and full). Of these, the COPY, DEFrag, DUMP, PRINT (tracks), and RESTORE functions may need to access catalogs for VSAM data sets, to check authorization to access the data sets on the volume. If another job on the system (for example, an Access Method Services DEFINE command) has already locked the catalog and needs the VTOC for a DADSM function (such as the ones described above), a deadlock can result between DFDSS and the other job. To avoid this, you should not run any jobs (for example, VSAM/Access Method Services jobs) that need to gain control of both the catalog and the VTOC while DFDSS is executing.

Avoiding Lockout

When an application program attaches a DFDSS dump or restore task and a task that invokes dynamic allocation to allocate a data set (for example, logical RESTORE) or invokes DADSM to scratch a data set, there is a possibility that the two tasks will lock each other out. To avoid this lockout, DFDSS uses the following ENQ scheme:

- Before DFDSS invokes DADSM SCRATCH, an exclusive ENQ is requested for the resource with a major name of ADRLOCK and a minor name of the volser (padded with blanks to 8 bytes).
- Before DFDSS invokes dynamic allocation to allocate a new data set on a specific volume, a shared ENQ is requested for the resource with a major name of ADRLOCK and a minor name of the volser (padded with blanks 8 bytes). If this is a non-specific dynamic allocation request, an exclusive ENQ with the resource with a major name of ADRLOCK and a minor name of NONSPEC is requested.
- If an application program invokes DFDSS FULL DUMP and plans on attaching another DFDSS logical RESTORE task, the application should request the first resource in exclusive mode (ADRLOCK, volser). The application should request the second resource in shared mode (ADRLOCK, NONSPEC) before attaching the FULL DUMP task. Multiple DUMPs can be active in the same address space and a logical RESTORE request will not lock out that task.

If an application program issues a DADSM request, the application must enqueue on ADRLOCK/volser in a shared ENQ or ADRLOCK/NONSPEC in an exclusive ENQ.

Notes:

1. When DFDSS is invoked with JCL, lockout does not occur because DFDSS allocates a volume before invoking DADSM. If a FULL DUMP has the volume enqueued and a restore task has been applied against the same volume, the restore task is held until the volume is available.
2. Only non-SMS, non-VSAM DADSM requests require serialization.

Data Set Serialization

Enqueuing—ENQ

The following material describes using enqueue options for data set serialization. Enqueues are done on the data set name for data set COPY, data set DUMP, data set RESTORE, DEFRAG, PRINT, COMPRESS, and RELEASE to prevent multiple, simultaneous updates to the same data set.

The SHARE option applies to non-VSAM data sets only (RESTORE, DUMP, and COPY).

If SHARE is not specified by the user with a DFDSS command, DFDSS tries to provide the highest level of data integrity by defaulting to exclusive enqueueing. If the enqueue fails, the command is put in a wait state for X seconds, where X is specified in WAIT(numsecs,numretries), and the enqueue is retried. If the wait-enqueue sequence fails after Y retries, where Y is specified in WAIT(numsecs,numretries), processing ends for the data set.

If you specify the SHARE keyword, DFDSS tries an enqueue for share. If it fails, it goes through the same logic as if SHARE had not been specified. If the retries all fail, processing ends for the data set.

You can specify TOLERATE(ENQFAILURE) in addition to the default, ENQ, or the SHARE option. If TOLERATE(ENQFAILURE) is specified, DFDSS attempts to get the specified level of enqueue, exclusive or share. If the enqueue fails after the specified or default number of retries, DFDSS processes the data set without an enqueue. You should specify TOLERATE(ENQFAILURE) if you are willing to tolerate the exposure of not having data integrity in order to force the successful completion of that particular data set operation. DFDSS respects data set integrity by not processing partially-qualified VSAM data sets which are opened for update. In order to force TOLERATE(ENQFAILURE) on data set that is in open status, a fully-qualified name is required. This is particularly useful when an installation has duplicate data sets (different data sets with the same name but on different volumes) and you want to run DFDSS on a data set on one volume while the data set with the same name is being used by the system or another job on a different volume.

Note: Because of ENQ contention, SYSPRINT data sets should not be allocated on volumes being processed. TOLERATE(ENQFAILURE) cannot apply to data movements that involve the use of utilities.

Figure 18 on page 167 shows the data set enqueue options.

TOLERATE(ENQFAILURE) can be used on VSAM and non-VSAM data sets.

Options	None	SHARE	TOLERATE(ENQFAILURE)	SHARE and TOLERATE(ENQFAILURE)
Type of enqueue	Exclusive	Share	Exclusive	Share
If enqueue is successful	Process	Process	Process	Process
If enqueue is not successful	Do not process	Do not process	Process	Process

Figure 18. Data Set Enqueue Options for Non-VSAM Data Sets Specified on DFDSS Commands

Dynamic Allocation—DYNALLOC

DYNALLOC is an option for data set DUMP, data set COPY, data set RESTORE, data set PRINT, DEFRAG, COMPRESS, and RELEASE. It allows serialization of data sets across processors with shared DASD, with JES3, and with the interface between dynamic allocation and JES3 enabled. SVC 99 causes DYNALLOC to serialize data sets. Processing time increases because overhead is involved in dynamic allocation and serialization across multiple processors.

If you use DYNALLOC to serialize data sets (as opposed to ENQ) the job run time will increase. If you use INDYNAM instead of DD statements to allocate DASD volumes you do not appreciably increase run time and coding of JCL and command input is easier.

Enqueuing versus Dynamic Allocation of Data Sets

For data set operations, the default method of serializing usage of data sets is to enqueue on the data set name. If DYNALLOC is coded in the control cards, the time taken to serialize is much greater than that taken by using the ENQ macro. So the DYNALLOC keyword should be used judiciously. It should be used only on a JES3 system if the interface between the allocation function and JES3 is not disabled. If the interface is disabled, then the end result of serialization using ENQ is the same as using SVC 99; ENQ takes much less time to serialize data sets.

When you use DYNALLOC, the selected data sets are allocated to DFDSS. If some data sets are allocated to the system, they can only be processed by DFDSS with DYNALLOC if SHARE is also specified. Figure 20 on page 170 provides additional information on data set serialization.

Read/Write Serialization Scheme

Figure 19 on page 169 shows the serialization scheme used by all operations, except COPYDUMP, for read/write access.

	ENQ Names Major Name	ENQ Names Minor Name	ENQ Control, Exclusive (E) or Share (S)	ENQ Scope
<i>Non-VSAM Data Sets</i>	SYSDSN	Data set name	E (SHARE not specified) S (SHARE specified)	SYSTEM
<i>VSAM Data Sets</i>				
On each component	SYSVSAM	See Note	E	SYSTEMS
On the cluster	SYSDSN	Cluster name	E (SHARE not specified) S (SHARE specified)	SYSTEM
<i>Integrated catalog facility catalog</i>				
In addition to above	SYSIGGV2	Catalog name	E	SYSTEMS
<i>On the Volume</i>	SYSVTOC	Volume serial number	E	SYSTEMS
<i>VVDS Data Set</i>	SYSZVVDS	SYS1.VVDS. Vvolser		SYSTEMS
Input access			S	
Output access			E	

If DYNALLOC is used:

For non-VSAM data sets, instead of ENQ, the data set is allocated automatically via SVC99, with a disposition of OLD if SHARE is not used; otherwise, SHARE is the disposition. For VSAM data sets, in addition to the ENQ, the cluster is allocated dynamically just as for non-VSAM data sets.

Note: The minor name consists of:

component name	catalog name	L1	L2	L3	A
----------------	--------------	----	----	----	---

L1: Total length of minor name
L2: Component name length
L3: Catalog name length

On a data set DUMP, an ENQ is done once with the character A=B and again with A=O.
On a data set RESTORE and DEFRAG, an ENQ is done once with A=B and again with A=I.
On a data set COPY, an ENQ is done once with A=O.

Figure 19. Read Write Access Serialization Scheme

WAIT Option

This option allows you to specify how long, in seconds, DFDSS is to wait for a resource and the number of times DFDSS is to retry, should an ENQ or RESERVE fail. The default is WAIT(2,2). This means DFDSS is to retry twice at 2-second intervals.

Serialization and Authorization Checking

For a data set COPY or logical data set DUMP, the WAIT option has a different meaning for serializing data sets when multiple data sets are to be processed and WAIT(0,0) is not specified. Multiple passes are made through the list of data sets that are selected. On each pass, those data sets that can be serialized without waiting for the resource and that were not processed before are processed. At the end of a pass, if none of the data sets could be processed without waiting for a resource, then, in the next pass, at the first occurrence of a data set that was not processed a WAIT will be issued. That data set and the remainder of the list will be processed if possible. The above procedure is repeated until all data sets are processed or the WAIT limits are reached. For example, if WAIT(3,10) is specified and 5 data sets are left to be processed, up to 10 passes are made. On each pass, the first unprocessed data set is waited upon for 3 seconds. Thus, only a 30-second maximum will ever be WAITed, not 150 (5 times 3 times 10).

Function	Data Set Type	Volume Level Serialization		Data Set Level Serialization (ENQ or DYNALLOC)
		VTOC	VVDS	
COMPRESS	Non-VSAM	Yes	N/A	DSName
CONVERTV	Non-VSAM	Yes	N/A	DSName
Data Set COPY	Non-VSAM	Yes	N/A	DSName
	VSAM	Yes	Yes	Component and Cluster Names
DEFRAG	Non-VSAM	Yes	N/A	DSName
	VSAM	Yes	Yes	Component and Cluster Names
Full DUMP	Non-VSAM	Yes	N/A	N/A
	VSAM	Yes	Yes	N/A
Data Set DUMP	Non-VSAM	Yes	N/A	DSName
	VSAM	Yes	Yes	Component and Cluster Names
Data Set PRINT	Non-VSAM	Yes	N/A	DSName
	VSAM	Yes	Yes	Component Name
Tracks PRINT		Yes	N/A	N/A
RELEASE	Non-VSAM	Yes	N/A	DSName
Full RESTORE	Non-VSAM	Yes	N/A	N/A
	VSAM	Yes	Yes	N/A
Data Set RESTORE	Non-VSAM	N/A	N/A	DSName
	VSAM	N/A	Yes	Component and Cluster Names

Figure 20. Resource Serialization

Notes to Figure 20:

- VSAM data sets must be cataloged in an integrated catalog facility catalog.
- N/A means not applicable.

An Example of RESERVE-ENQUEUE Processing

If you were to enter the following:

```
DUMP INDD(IN1) OUTDD(OUT1) -
DATASET(INCLUDE(MY.DUMP.DATASET1,MY.DUMP.DATASET2)) -
WAIT(5,2) -
SHARE
```

The following would result:

1. DFDSS issues a RESERVE on SYSVTOC (this is the default)

If RESERVE fails, DFDSS retries two times at 5-second intervals.

If RESERVE is successful, DFDSS continues.

If either of the data set was a VSAM or SMS-managed data set, the VVDS would also be serialized by DFDSS. If VVDS serialization fails, none of the data sets are dumped.

2. DFDSS issues a shared enqueue on data set name. If the shared enqueue fails, DFDSS retries two times at 5-second intervals.

If the shared enqueue is successful:

DFDSS adds the data set name to the list to be dumped.

DFDSS loops back to the beginning of this step until an enqueue is tried for all data sets.

If the shared enqueue fails:

DFDSS issues message ADR412E.

DFDSS sets condition code = 8.

DFDSS loops back to the beginning of this step until an enqueue is tried for all data sets.

3. After an enqueue is tried for all specified data sets:

DFDSS dumps the VVDS (if an integrated catalog facility data set was selected) and the VTOC and dequeues the VTOC.

If VVDS was enqueued, it is dequeued by DFDSS.

DFDSS dumps each data set that was successfully enqueued.

DFDSS dequeues the enqueued data sets.

DFDSS issues message ADR3781.

Keep in mind that, if you specify TOLERATE(ENQFAILURE) with the DUMP command and the user enqueue exit routine does not override it, the VTOC is not dequeued until all the data tracks for all the data sets are dumped.

DFDSS Authorization Checking

DFDSS checks authorization on DASD volumes and data sets to ensure that you have sufficient authority to access them. The required access authority depends on the function being performed and the protection in effect for the entity. The protection mechanism can be either Resource Access Control Facility (RACF) or password. RACF checking takes precedence over password checking. The system operator is also queried for authorization in certain instances. Some system data sets and VSAM data sets are defined to accept authorization from the operator when RACF or password checking cannot be used (for example, in accessing the VTOC,

Serialization and Authorization Checking

VTOC index, or VVDS). System Authorization Facility (SAF) is also supported (SAF is not supported for MVS/370).

Note: If you enter data set passwords in the command data set and your job abnormally ends, these passwords may appear in the dump. To prevent this, you should put the passwords in a data set and refer to this data set with a DD statement. If you enter data set passwords in the command data set, they are suppressed and do not appear in the SYSPRINT data set listing.

DFDSS checks Resource Access Control Facility (RACF) authorization at the volume and data set levels. If RACF Version 1 Release 5 or later release is installed, generic profiles are also checked. If RACF Version 1 Release 7 (or later release) and MVS/XA Data Facility Product Version 2 Release 1 (or later release) are installed, the erase-on-scratch RACF data set profile attribute is supported during data set DUMP with DELETE, data set COPY with DELETE, DEFrag, and RELEASE.

Access authority to the volume is checked first. If you have the required RACF authority at the volume level (for example, alter authority for a full volume RESTORE), checking at the data set level is not performed. If you do not have the required volume level authority or if the volume is not protected or the protection status of the volume cannot be determined, only then is checking performed at the data set level.

For a full volume operation, you must have required access authority to the volume itself or to all the data sets on the volume. For a tracks or data set operation, you must have the required access authority to the volume itself or to the data sets that map the required extents. Full and tracks operations are ended if you do not have sufficient authority. For data set level operations, if you do not have volume level access, only the subset of those required data sets that you have access to is processed.

Authorization checks are not done for COPYDUMP.

Authorization checking at the data set level is bypassed in the following cases, where it is considered to be equivalent to your having volume level access:

The JSCBPASS (bypass authorization check) indicator is on in the JSCB (job step control block), or

Your authorization exit routine so directs.

An installation-supplied installation authorization exit routine can allow special privileged users to bypass all or some of the authorization checking. Security processing does increase the time required to perform various DFDSS functions. If RACF, password checking, or other security features are used for a large number of data sets, the elapsed time of an operation can be increased. Note that when you invoke this exit you cannot bypass authorization checking initiated by utilities (such as IEBCOPY) that are invoked by DFDSS. See "Installation Authorization Exit Routine (ADRUPSWD)" on page 143 for details on the installation authorization exit. See Figure 22 on page 176 for information on when DFDSS invokes utilities.

Volume Authorization Checking

RACF DASDVOL protection is the only protection checked by DFDSS at the volume level. If other means of protection are normally used in a given installation, an installation-supplied installation authorization exit routine must check your access authority. See "Installation Authorization Exit Routine (ADRUPSWD)" on page 143 for details. The installation authorization exit routine supplied with DFDSS does not provide this function. DFDSS does, however, check for DASDVOL access authority if RACF DASDVOL class is active on the system.

If RACF DASDVOL class is active and a profile exists for the specified volume, your access authority is determined. If RACF DASDVOL class is active but a profile does not exist for the volume, the volume is not protected. If RACF DASDVOL class itself is inactive, the protection status of the volume cannot be determined. If RACF DASDVOL is inactive or no profile exists, the user is given authority.

Figure 21 on page 175 lists the RACF DASDVOL access authority checked by DFDSS, by function and type of access.

Data Set Authorization Checking

Data set authorization is checked if:

- System utilities are invoked for data movement.
- You do not have the required RACF DASDVOL access.
- Data set authorization checking is not bypassed either with the JSCBPASS bit indicator or your authorization exit routine.

Data sets can be protected by the RACF DATASET class mechanism, the password mechanism, or both. If a data set is protected by both mechanisms, only the RACF DATASET access authority is checked. If a data set is not protected by either mechanism, DFDSS assumes that all users have required access to it unless the data set is any of the following:

- A checkpoint/restart data set
- A VTOC index data set
- A VVDS data set
- Marked as RACF protected but RACF is inactive
- A VSAM data set cataloged in an integrated catalog facility catalog but the VVDS is not on the volume.

In these cases, DFDSS prompts the system operator to determine whether you can be allowed the required access. Keep in mind that the operator is always prompted for the VVDS data set even if it is protected by other means. Also, the operator message, ADR369D, contains only the generic phrase *RACF PROTECTED SYSTEM DATASET* but not individual data set names. The operator is only prompted for the first data set encountered in this class. The reply given by the operator is used for subsequent data sets of this type on the volume.

The checking mechanism used is based on the type of data set: VSAM, non-VSAM, and so on. The access authority required depends on the function and type of operation being performed:

- Non-VSAM RACF-protected data sets. If RACF Version 1 Release 5 or System Authorization Facility (SAF) is installed on your system, all data sets are checked as though they were RACF protected. This is because of the **generic profile** function supported by RACF Version 1 Release 5 and System Authori-

zation Facility (SAF). If RACF Version 1 Release 4 is installed on your system but SAF is not, the data sets are checked for RACF protection only if they are specified as RACF protected. Authorization is checked by calling the RACF facility. The access authority required is given in Figure 21 on page 175.

- VSAM data sets. Authorization for accessing VSAM data sets (except VVDS) is checked by using the catalog management services. Authorization for overlaying the VVDS is obtained by prompting the operator. The master password is required to read from or write to a catalog for:

A DUMP, COPY, or RESTORE
A tracks PRINT.

The master password is also required to write to the output volume for:

A full or tracks RESTORE
A full or tracks COPY.

Authorization is checked by calling the RACF facility. The access authority required is given in Figure 21 on page 175.

- VSAM data sets not cataloged under the integrated catalog facility catalog. Authorization is checked by opening the data space for input (for input operations) or output (for output operations). The data set must be cataloged in the standard order of search.
- Non-VSAM password-protected data sets. If the DASD volume is the target volume for a RESTORE or COPY operation or if it is a DUMP command, the write password must be supplied. In all other cases, a read password suffices. Passwords must be supplied by using the PASSWORD keyword of the applicable command.
- When RACF Version 1 Release 5 or an earlier release is installed, if you have DASDVOL access at the proper authorization level but not the proper data set authorization and you do not have the RACF special and operations attributes:
 - When DFDSS attaches utilities in processing non-VSAM data sets for COPY and COMPRESS operations, the utilities abend with a 913 code.
 - When restoring KSDS and key range data sets to unlike devices, message ADR437E (VSAM open error) is issued and the data set is not restored.
- When RACF Version 1 Release 6 or a later release is installed, RACF DASDVOL authorization checking is honored. However, DASDVOL authority is not sufficient if you want to use to move (COPY with DELETE) a RACF protected VSAM data set. To copy a data set, DFDSS uses ALTER NEWNAME to change the name of the data set being copied. ALTER renames both the data set and the data set's RACF profile. If the user doing the copy is not a member of the group containing the profile, ALTER will fail, causing DFDSS to fail the copy.

To copy a RACF protected data set, at least one of the following conditions must be met:

- You are a member of the group containing the data set's profile.
- You have RACF OPERATIONS authority.
- The data set's profile takes the form 'HLQ*'; a more specific profile name prevents DFDSS from copying the data set.

If the data set is protected by a discrete profile, you must be a member of the group because OPERATIONS authority is not sufficient to rename the discrete profile.

These same conditions apply for copying non-VSAM data sets into SMS-managed storage.

To restore a data set that was RACF protected at dump time, you must have update access to the target data set. If the data set is to be renamed during RESTORE, you must have read access to the source data set as well as update access to the target data set.

For non-VSAM data sets that were protected under a discrete profile that no longer exists, you must predefine a new discrete profile or preallocate that data set with a discrete profile.

These conditions do not apply if data set authorization checking is bypassed.

Function	Type of Access on Volume or Data Set	Type of Access		
		Full	Tracks	Data Set
COMPRESS	Input	N/A	N/A	Update
COPY with DELETE	Input Output	N/A N/A	N/A N/A	Alter Update
COPY without DELETE	Input Output	Read Alter	Read Alter	Read Update
COPYDUMP	N/A	N/A	N/A	N/A
DEFRAG ⁹	Input	Read	Read	Read
DEFRAG ¹⁰	Input	Update	Update	Update
DUMP with DELETE	Input	N/A	N/A	Alter
DUMP without DELETE	Input	Update	Update	Update ¹¹
PRINT	Input	N/A	Read	Read
RELEASE	Input	N/A	N/A	Update
RESTORE	Input Output	N/A Alter	N/A Alter	Read Update

Figure 21. Access Authority Requirement—RACF-Protected Data Sets and DASDVOL

⁹ RACF-protected data set access requirement

¹⁰ DASDVOL access requirement

¹¹ ALTER is required for Integrated Catalog Facility VSAM catalogs.

Utilities Used for Moving Data Sets

In some cases, DFDSS invokes a utility to move a data set. Figure 22 shows when DFDSS invokes a utility for a data set COPY.

When you move a data set and a utility is used, the data set must be cataloged in the standard search order.

When DFDSS invokes IEBCOPY to copy a LOADMOD, message IEC507D is issued requesting operator authorization to overwrite an unexpired area when the source data set has an incorrect RLD count and an unexpired date.

When DFDSS invokes IEHMOVE to copy data sets, IEHMOVE has DD statement requirements that DFDSS cannot always satisfy. To avoid potential abnormal ends, do one or both of the following:

- Specify the source and target volumes as PRIVATE.
- Ensure that the source and target volumes are not in the list of default volumes for dynamic allocation.

Data Set Type	Like Devices	Unlike Devices
SAM	DFDSS	DFDSS
PAM	DFDSS ¹²	DFDSS ¹²
PAM load modules	DFDSS	IEBCOPY ¹³
DAM nonrelative block address mode	DFDSS	IEHMOVE ¹⁴
DAM relative block address mode ¹⁵	DFDSS	DFDSS
ISAM to same track. MVS/XA DFP Version 2 Release 1 or later installed. Target volume uses VTOC index. Target space available.	DFDSS	N/A
ISAM (all other cases)	IEBISAM	IEBISAM
ESDS	DFDSS	DFDSS

¹² All PAM data sets that are not load modules are compressed during a copy to a like or unlike device. DFDSS also compresses track overflow data sets.

¹³ If copying PAM load modules with REBLOCK, then IEBCOPY is used instead of DFDSS to copy the data set.

¹⁴ Not copied if the target data set is preallocated or if the target device has a smaller track capacity than the source.

¹⁵ DFDSS RELBLOCKADDRESS parameter must be specified.

¹⁶ IDCAMS is used if the target's CISIZE, CASIZE, physical record size, physical block size, or imbed or span attributes are different from the source's.

¹⁷ IDCAMS is used if the source and target CASIZE, physical record size, or physical block size are different; if the components span multiple volumes: for a KSDS with IMBED and either the source HURBA = HARBA or it has extended indexes.

Data Set Type	Like Devices	Unlike Devices
RRDS	DFDSS	DFDSS
KSDS	DFDSS ¹⁶	IDCAMS (REPRO)
Key range data set	DFDSS ¹⁷	IDCAMS (REPRO)
CVOL	IEHMOVE	IEHMOVE
Integrated catalog facility user catalogs	IDCAMS (EXPORT/IMPORT)	IDCAMS (EXPORT/IMPORT)
Undefined DSORG	DFDSS	DFDSS

Figure 22. Data Mover Selection Matrix for Data Set COPY.

RACF Logging during Authorization Checking

If you specify the RACFLOG = YES keyword in your EXEC statement or if the installation options exit requests RACF logging, DFDSS allows RACF logging as follows:

- During RACF DASDVOL checking, DFDSS allows logging of successful authorizations only. This allows tracking of successful authorizations with DASDVOL. Unsuccessful attempts are not logged because the user may still gain authorization during data set level checking.
- During data set level checking of VSAM data sets, DFDSS has no control of RACF logging. Logging of these data sets normally occurs as specified in the applicable RACF profile.
- During data set level checking of non-VSAM data sets, DFDSS allows RACF logging as specified in the applicable RACF profile.

Figure 21 on page 175 provides additional information regarding RACF-protected data sets and DASDVOL.

Example of Security Processing

Dump data sets by filter criteria; assume RACF and password-protected data sets are on the volume.

```
DUMP DATASET(INCL(MY.**))
```

The security processing that occurs is as follows:

Authorization checking is not done if any of the following are true:

1. The IEFSDPPT program property table indicates that DFDSS does not require authorization processing.
2. An installation exit indicates to bypass authorization checking.
3. The user has RACF DASDVOL update access.

The following occurs if authorization checking is performed:

1. Each data set that is RACF-protected is checked for access authority. If you do not have authority or if the RACF profile is missing and the data set is RACF indicated, the data set is dumped and message ADR402 is issued. Processing continues for other data sets.
2. Password checking is done if the data set is not RACF-protected but is password protected. Password checking is also done if the data set is protected with both the password and RACF protection mechanisms and RACF is

Serialization and Authorization Checking

inactive or not supported on the system. If authorization is not obtained, message ADR402 is issued, the data set is not dumped, and processing continues for other data sets. If this is a system data set (for example, checkpoint/restart), operator authorization is obtained.

Appendix C. User Interaction Module Exit Options

This appendix provides information that you may need to refer to when you use the User Interaction Modules. General-use programming interfaces are contained in this appendix.

Calling Block Structure

The parameter structure outlined below can be viewed in block form as shown in Figure 23 on page 181.

ADDRSSU

is the name of the DFDSS module main entry point.

OPTPTR

is the name of the pointer to the option list and is similar to the OPTIONADDR description used in *MVS/DFP Version 3 Release 2: Utilities*. This provides a way to specify processing options. This parameter must be specified even if the list is a null list. This means that if you do not want to specify any parameters to DFDSS, this pointer must be specified and must point to a halfword of binary zeros. The same values that can be specified in the EXEC PARM field in the JCL for DFDSS can be specified in the option list. The first field of this list is a halfword field (not on a fullword boundary) giving the length of the remainder of the list. (This field is hereafter referred to as the option list length field.) If you do not want to specify any options, you must set the option list length field to binary zeros. If you specify any options in the option list, you must set the total length of the option list (excluding the length field itself) in the option list length as a binary value. The options must comply with the parameter syntax of the DFDSS EXEC parameter values.

If you do not want to specify subsequent parameters, you can omit them from the list.

DDPTR

is the name of the pointer to the DDNAME list and is similar to the DDNAMEADDR description used in *MVS/DFP Version 3 Release 2: Utilities*. The DDNAME list provides a way for the user to specify alternate names for the SYSIN and/or SYSPRINT data sets. The DDNAME list is a variable length field made up of a halfword field followed by unseparated 8-character left-justified (right padded with blanks) fields. Each 8-character field is reserved for a specific DDNAME. DFDSS uses only two of these fields, that of SYSIN (the fifth field) and SYSPRINT (the sixth field). The other fields are provided as a standard implementation consistent with existing system utility invocation procedures. All fields other than the SYSIN and SYSPRINT fields must be filled with binary zeros and are ignored by DFDSS.

The first field of this list is a halfword (not on a fullword boundary) field giving the length of the remainder of the list and contains the binary value representing the number 48. (This field is hereafter referred to as the DDNAME list length field.) If you specify this parameter but do not want to specify any alternate DDNAMES for the SYSIN or SYSPRINT data sets, you must set the DDNAME list length field to the correct length and all of the 8-character fields to binary zeros.

If you want to specify an alternate DDNAME for the SYSIN data set, you must specify this parameter and must fill the fifth 8-character field with the alternate DDNAME; otherwise, this field must be binary zeros.

If you want to specify an alternate DDNAME for the SYSPRINT data set, you must specify this parameter and must fill the sixth 8-character field with the alternate DDNAME; otherwise, this field must be binary zeros.

If you do not want to specify subsequent parameters, you can omit them from the list.

PAGEPTR

is the name of the pointer to the page number list and is similar to the HDINGADDR description used in *MVS/DFP Version 3 Release 2: Utilities*. This provides a way to specify the starting page number for system output on the SYSPRINT data set. The page number list is a fixed-length 6-byte field made up of a halfword field followed by a 4-byte EBCDIC page count, which specifies the starting page number for DFDSS to use for the SYSPRINT data set. If this value is specified here and in the OPTPTR list, this value is used, and the corresponding OPTPTR value is ignored.

The first field of this list is a halfword field giving the length of the remainder of the list and normally contains the binary value representing the number 4. This field is hereafter referred to as the page number list length field.

If you do not want to specify a starting Page Number, you must set the page number list length field to binary zeros. If you want to specify a starting Page Number, you must specify this parameter and specify a 4-character page value. If the Page Number is specified (page number list length field is not binary zeros), DFDSS resets this field to the current page number upon completion of the present invocation. If the Page Number was not specified (page number list length field is binary zeros), this field is not changed by DFDSS.

If you do not want to specify subsequent parameters, you can omit them from the list.

PARAM

if you had JCL EXEC PARM = values specified in the calling program, DFDSS ignores the parameters in the lists pointed to by register 1 upon entry to DFDSS.

UIMPTR

is the name of the pointer to the user interaction module (UIM) list. There is no comparable parameter described in *MVS/DFP Version 3 Release 2: Utilities*. This parameter provides a way to specify the name or address of a vector-implemented exit module that is to interact with DFDSS for the various I/O and data set operations. The UIM list is of variable length and consists of a halfword field. The halfword field is followed by a 4-byte address or 8-character left-justified (padded on the right with blanks) string field that specifies the address of the UIM entry point or the load module name (locatable through the normal LINKLIB structure) of the UIM. If you do not want to specify a UIM, you should not specify this parameter.

The first field of this list is a halfword field (not on a full word boundary) giving the length of the remainder of the list. It contains the binary value representing the number 4 if the address of the UIM is being passed, or the binary value representing the number 8 when the name of the UIM is being passed. (This field is hereafter referred to as the UIM list length field.)

If you want to specify a UIM, you must specify the address of the UIM or the name of the UIM in the field following the UIM list length field. Fill that length field with the proper binary value of 4 or 8 respectively. See "System Programming Information" on page 182 for more details on the use of this module.

When the address of the UIM is specified, DFDSS uses bit 0 of the address to determine the addressing mode of the UIM in MVS/XA. If bit 0 is 1, then DFDSS assumes that the UIM is to be called in 31-bit addressing mode (which is set at link-edit time by specifying AMODE = 31). If bit 0 is 0, the UIM is called in 24-bit addressing mode.

UAPTR

is the name of the pointer to the user area list. There is no comparable parameter in DFDSS Version 2 Release 1 or prior releases. This parameter provides a way to specify the address of an area that you want to be passed to the UIM at each DFDSS exit point. The user area list is of fixed length, consisting of a halfword length field followed by a single 4-byte address that locates the user area starting address.

If you use the user area, the address of the user area must be specified in the field after the user area list length field, and the length must be set to 4.

VL

indicates that the list is variable length. Both the ATTACH and LINK macros require specifying VL = 1.

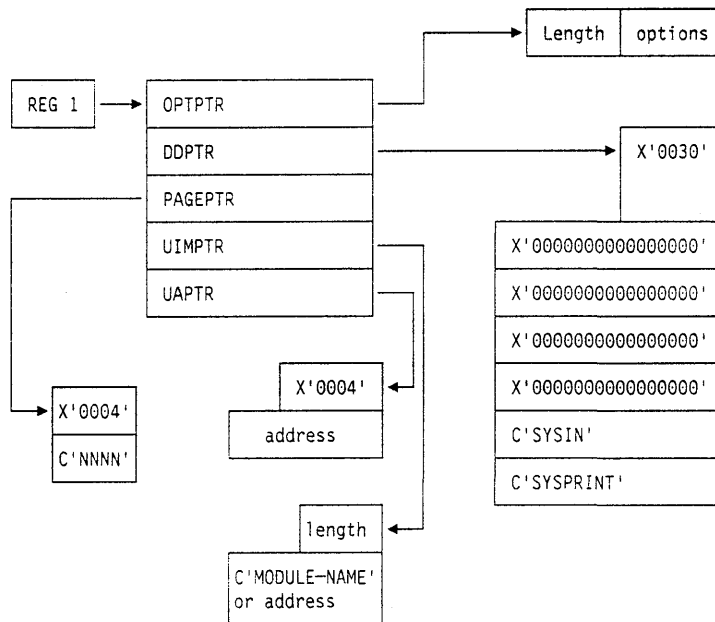


Figure 23. DFDSS Application Interface Structure

User Interactions

User interactions can only take place when DFDSS has been invoked by using the Application Interface and the name of the user interaction module (UIM) exit routine is supplied to DFDSS. DFDSS can be invoked by any of the system macros shown in Figure 24.

```
ATTACH EP=ADRSSU,PARAM=(OPTPTR,DDPTR,PAGEPTR,UIMPTR,UAPTR),VL=1
or
LINK EP=ADRSSU,PARAM=(OPTPTR,DDPTR,PAGEPTR,UIMPTR,UAPTR),VL=1
or
CALL (15),(OPTPTR,DDPTR,PAGEPTR,UIMPTR,UAPTR),VL
```

Figure 24. Invoking DFDSS with Its Application Interface

As shown further in Figure 23, a pointer to the UIM exit is passed in the parameter list (UIMPTR). Generally, user interactions can be used for:

- Record management
- Controlling processing of data sets
- Gathering statistics on DFDSS operations.

See Appendix D, “Examples of the Application Program with the User Interaction Module (UIM)” on page 204 for an explanation of the various UIM exits.

When a UIM exit routine is specified, DFDSS processes normally, then at each point in the process (hereafter referred to as DFDSS exit points), the UIM exit routine is called conditionally to allow some types of user operations.

If a DFDSS subtask abnormally ends for any reason, it will be unable to make any further calls to the UIM, including the *function ending* call.

Note: DFDSS runs as an authorized problem program. Any program invoking DFDSS must also be authorized and in a non-supervisor state.

System Programming Information

There are some bit definitions in the installation options control block (ADRUF0) that permit DFDSS to communicate with the installation options exit routine. When the installation options exit does not want a function to be scheduled, it returns a code of 8; for the UIM to do this requires a bit definition of UFSTOP to be set in the ADRUF0. If there is not to be a SYSIN and/or SYSPRINT data set allocated or all SYSIN/SYSPRINT is to be handled **in storage**, UFSYSIN and UFSYSPR specify this to DFDSS. Three additional bits are referred to—UFAIINV, UFUIMAL and UFUIMCH¹⁸—which determine allowances within the UIM interaction.

Two additional bits are defined to indicate the existence or absence of an input restore/copydump data set or an output dump copydump data set. These are UFNOIN and UFNOOUT respectively.

¹⁸ See *Data Facility Data Set Services: User's Guide* for a definition of these and other acronyms.

When DFDSS has been invoked by using the Application Interface and the UIM has been specified and is to be called, the following information is passed to the UIM on every exit call:

1. Register 1 points to the interface parameter list pointer.
2. The interface parameter list pointer is the address of the DFDSS exit identification block. See "ADREID0 Data Area" on page 198 for a detailed description of this block. This list basically consists of:
 - A halfword field specifying the length of the remainder of the list.
 - The remainder of the list that is mapped by the macro ADREID0. See Figure 26 on page 183 for a block format of this area.

Upon return from the UIM, the user return code field is examined to determine the disposition of the current I/O record or data set (within the limits allowed to the exit).

Application Interface Blocks: The parameter structure described in Figure 23 on page 181 can be viewed in block form as shown in Figure 25.

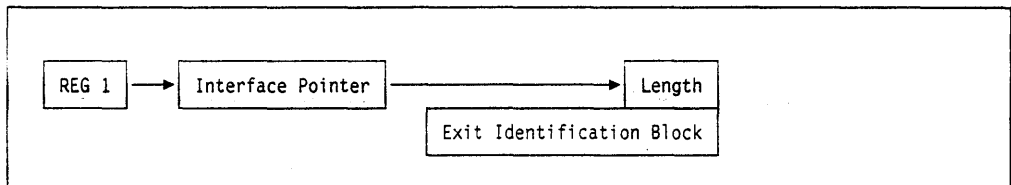


Figure 25. DFDSS Exit Interface Structure

Exit Identification Block: The exit identification block is shown in block form in Figure 26.

Control Block Eye-catcher	(supplied by DFDSS)
TASK-ID	(supplied by DFDSS)
Installation Exit Allowance	(supplied by DFDSS)
DFDSS Processing Option	(supplied by DFDSS)
User Return Code	(supplied by UIM)
Record Area Length	(supplied by DFDSS)
Original Record Length	(supplied by DFDSS)
Original Record Pointer	(supplied by DFDSS)
User Area Pointer	(supplied by DFDSS)
DDNAME/VOLID Pointer	(supplied by DFDSS)

Figure 26. DFDSS Exit Identification Block

The exit identification block is passed to the user interaction module every time DFDSS gives control to it. Each field is described below, but see “ADREID0 Data Area” on page 198 for the formal declarations.

- **Control Block Eye-catcher**—This field is a 4-character string field that is supplied by DFDSS. It contains the character string *EIDB* and can be of use in locating the control block when you are viewing a storage dump during **DEBUG processing**.
- **TASK-ID**—This field is a fullword binary field that is supplied by DFDSS. The number contained in this field is assigned by DFDSS to each function command statement submitted in *SYSIN*, whether obtained from the data set or from the UIM. This number is binary zero when DFDSS is calling the UIM for a function that is not related to a user command statement. Each command is numbered in sequence. When a task is scheduled to process this command, all messages associated with this task and all calls to the user interaction module for this task are accompanied by this unique number. In this way, the UIM can identify which task is being processed and what function is associated with that task.
- **Installation Exit Allowance**—this field is a fullword binary field supplied by DFDSS. The 32 bits defined in this field can be used as flags to determine what actions the UIM can perform with respect to the record presented. Some actions conditionally allowed are:
 - View and conditionally override the installation options.
 - Insert data prior to current record.
 - Replace the current data record with an exit-record supplied.
 - Delete the current data record.
 - Modify the current data record.
 - Disconnect the exit from further interaction.
 - Recognize when a disallowed option has been attempted.
 - End processing of the data set.
 - End processing of the task.
- **DFDSS Processing Option**—This field is a halfword binary field that is supplied by DFDSS. The number contained in this field can be used by the user interaction module (UIM) to vector branch to the appropriate processing routine for the record or data set being presented to the exit.

At appropriate locations in the DFDSS processing modules, the user interaction module is considered for receiving control. These locations are referred to as DFDSS Exit Points.

- **User Return Code**—This field is a halfword binary field that is filled in by the UIM. The number contained in this field identifies the action expected by the exit on the record or data set being presented to the exit. This field is examined by DFDSS when control returns from the UIM. Although not all the following return codes are allowed at any given exit call, the list does include all acceptable return codes. If a code that is not allowed is returned from the UIM, DFDSS passes the *EIDB* back to the UIM with the *EIXERR* flag set. This allows the UIM one chance to correct the option. If a code that is not allowed is then returned again, it is ignored by DFDSS and the record is processed as though the exit had returned the code zero (0). This sequence is followed for any subsequent incorrect return codes. The meaning of each return code for those exits presenting records to the UIM (*Eioptions 1 to 20*) is:
 - 0 The record is processed as would normally have happened if there had not been a UIM. The original record is not changed in any way by the exit.

- 4 The record was replaced by the exit. The new record must be placed in the area pointed to by the original record pointer field and its length stored in the original record length field.
- 8 The record is to be inserted. The address of the new record must be stored in the original record pointer field or the new record must be stored in the area pointed to by the original record pointer field and its length must be stored in the original record length field. When this exit is next called, the original record is presented again.
- 12 The record is to be deleted. The record presented to the UIM is ignored in the processing, thus deleting it.
- 16 The record was modified by the exit. This return code is the only one allowing the original record to be altered and then to be processed by DFDSS. Any changes made to the record must be logically correct because DFDSS cannot assure the validity of these changes.

Note: If the record being processed is the installation options record (ADRUF0) and any values have been changed, return code 16 must be returned or the changes are ignored.
- 20 The record is to be processed as though a return code zero (0) had been given, but this particular DFDSS exit point will no longer be called from the current functional task, although it might be called from others. You must be cautious in using this code because multiple record types use the same DFDSS exit point. It would be better for the UIM to inspect each record type and only give a return code zero (0) when a record is not of interest to the exit.
- 24 The record is to be processed but at all future visits of DFDSS at this exit point, only user statistical records are to be presented to the UIM.
- 28 The notification-of-response return code. The WTOR has been handled by the UIM and the UIM has supplied the proper response from the WTOR in the area pointed to by the original record pointer. The interface allows you to handle all WTOR processing in the UIM exit routine and to supply the required response to DFDSS in lieu of DFDSS's issuing the WTOR itself. Note that the UIM must set the original record length to the proper value.
- 32 DFDSS ends the current functional task and issues message ADR356E. This return code is supported only for full volume DUMP and RESTORE.

The meaning of each return code for those exits controlling data set processing (Eioptions 21, 22, and 23) is:

- 0 The data set is processed as would normally have happened if there had not been a UIM. Data set processing is not altered in any way.
- 16 This return code is valid from exit 22 only. It indicates to DFDSS to examine the bypass processing flags and to modify processing of the data set according to which flags are on. These bypass flags are only examined by DFDSS if a return code of 16 is returned. Refer to "Bypass Verification Exit (Eioption 22)" on page 194 for more details about these flags.
- 20 The record is to be processed as though a return code zero (0) had been given, but this particular DFDSS exit point will no longer be called from the current functional task, although it might be called from others. You must be cautious in using this code because multiple record types use the same DFDSS exit point. It would be better for the UIM to inspect each record

type and only give a return code zero (0) when a record is not of interest to the exit.

32 DFDSS ends the current functional task and issues message ADR356E. This return code is supported only for logical data set COPY, logical data set DUMP, and logical data set RESTORE.

36 DFDSS ends processing of the data set named in the parameter passed to the exit. Processing continues with the next data set (if any). If Exit 23 sets this return code after the data set has already been processed, then DFDSS does not undo the processing, but deletes the data set from the successfully-processed message list (if applicable) and includes it in the unsuccessfully-processed message list even though the data set may have been successfully processed up to that point

- **Record Area Length**—This field is a fullword binary field that is supplied by DFDSS. The number contained in this field represents the total length of the area in which the original record is stored. The number can be used by the UIM to verify that replacement and inserted records will fit in the provided buffer area pointed to by the original record pointer field. The value of this field is zero (0) when the UIM is called by exits 21, 22, or 23.
- **Original Record Length**—This field is a fullword binary field that is supplied by DFDSS but can be changed by the UIM. The number contained in this field represents the total length of the record pointed to by the original record pointer. The length includes just the length of the record pointed to by the original record pointer field and does not include the length of this field itself. The value of this field is zero when the UIM is called to supply in-storage SYSIN data. The value of this field is zero (0) when the UIM is called by exits 21, 22, or 23.
- **Original Record Pointer**—This field is a fullword address field that is supplied by DFDSS but can be changed by the UIM. This field contains the address of the record being passed to the UIM on this call. This address will normally not be changed by the exit unless an insertion is being used. Refer to the proper return code descriptions for the effect on this field. The value of this field is zero when the UIM is called to supply in-storage SYSIN data or called by Exits 21, 22, or 23. The location of the original record, above or below the 16-megabyte (Mb) virtual storage line, is controlled by using the installation options exit. If this record is to be above 16 Mb, the UIM has to run in 31-bit addressing mode in order to address the record.
- **User Area Pointer**—This field is a fullword address field that is supplied by the application program and is maintained by DFDSS. This field contains the address of the user data/work area that was supplied by the application program as a communications area for UIM internal process controls. DFDSS saves this pointer and supplies it in the EIUSEPTR field of the exit identification block on each call to the UIM. If any UIM exit changes the user area pointer, DFDSS presents the updated pointer on subsequent calls to the UIM.

Each DFDSS functional task keeps its own copy of the user area pointer. If the pointer is changed by the UIM for one task, it does not be changed in any other task. At the beginning of each task, the user area pointer is the one passed to DFDSS by the application program.

- **DDNAME/VOLID Pointer**—This field is a fullword address field supplied by DFDSS. It contains the address of an area containing the DDNAME of the output dump data set, left-justified in an 8-byte area, followed by a 6-byte area containing the volume serial number of the volume containing the dump data set, also left-justified. This pointer is valid only for the full volume DUMP exit, EIOP06.

Application Interface Descriptions

Following are the descriptions for all of the exit points available with DFDSS.

Function Startup (Eioption 0)

This exit point is called when DFDSS starts a function such as DUMP or RESTORE.

The valid return codes for function startup are:

- 0 (continue normal processing)
- 20 (inhibit all UIM calls)
- 32 (end processing -- Only valid for a full DUMP and a full RESTORE).

Reading SYSIN Record (Eioption 1)

This exit point is called after DFDSS reads a SYSIN record. You can replace, insert, delete, or modify a SYSIN record at this exit point. The EIRECPTR points to the SYSIN record.

The valid return codes for reading SYSIN records are:

- 0 (continue normal processing)
- 4 (record replaced)
- 8 (insert record)
- 12 (delete record)
- 16 (record modified)
- 20 (disconnect exit).

Printing SYSPRINT Record (Eioption 2)

This exit point is called when DFDSS is ready to print a SYSPRINT record. You can replace, insert, delete, or modify a SYSPRINT record at this exit point. The EIRECPTR points to the SYSPRINT record.

The valid return codes for printing SYSPRINT records are:

- 0 (continue normal processing)
- 4 (record replaced)
- 8 (insert record)
- 12 (delete record)
- 16 (record modified)
- 20 (disconnect exit).

Note: When DFDSS is printing a page header record, you can only modify or delete the record.

Reading Physical Tape Record (Eioption 3)

This exit point is called when DFDSS has read a record from a data set that has been dumped (on tape or DASD) by using the DUMP command. The EIRECPTR points to the tape record. The valid return codes for reading physical tape records are:

- 0 (continue normal processing)
- 8 (insert record—Only valid when UFNOIN is set in the user installation options exit).
- 20 (disconnect exit)
- 32 (end function—Only valid for a full DUMP and a full RESTORE).

Reading Logical Tape Record (Eioption 4)

This exit point is called when DFDSS has read a record from a data set which was created with a logical DUMP command and dumped on tape or DASD. The EIRECPTR points to the tape record.

The valid return codes for reading logical tape records are:

- 0 (continue normal processing)
- 8 (insert record)
- 20 (disconnect exit)
- 24 (select user statistics records)
- 32 (end function—Only valid for a full DUMP and a full RESTORE).

Note: If code 24 is returned, DFDSS will only call this exit point when DFDSS processes user statistics records.

Writing Logical Tape Record (Eioption 5)

This exit point is called when DFDSS writes a logical record to a dumped dataset, on tape or DASD, while performing a logical DUMP. If you insert a record at this exit point, DFDSS marks it as a statistics record, not a data record. The EIRECPTR points to the tape record.

The valid return codes for writing logical tape records are:

- 0 (continue normal processing)
- 8 (insert record)
- 20 (disconnect exit).

Writing Physical Tape Record (Eioption 6)

This exit point is called when DFDSS writes a physical record to a dumped data set on tape or DASD. The EIRECPTR points to the tape record and the EIDDID points to DDINFO.

The valid return codes for writing physical tape records are:

- 0 (continue normal processing)
- 12 (delete record—Only valid when UFNOOUT is set in the user installation options Exit).
- 20 (disconnect exit)
- 32 (end function—Only valid for a full DUMP).

Reading Disk Track (Eioption 7)

This exit point is called when DFDSS has read a track from DASD. The EIRECPTR points to the track buffer.

The valid return codes for reading disk tracks are:

- 0 (continue normal processing)
- 20 (disconnect exit)
- 32 (end function—Only valid for a full DUMP).

Writing Disk Track (Eioption 8)

This exit point is called when DFDSS is ready to write a track to DASD. The EIRECPTR points to the track buffer.

The valid return codes for writing disk tracks are:

- 0 (continue normal processing)
- 20 (disconnect exit)
- 32 (end function—Only valid for a full RESTORE).

Reading Utility SYSPRINT (Eioption 9)

This exit point is called when DFDSS is reading output from an attached utility. You can replace, insert, delete, or modify a utility SYSPRINT record at this exit point. The EIRECPTR points to the Utility SYSPRINT record.

The valid return codes for reading utility SYSPRINT are:

- 0 (continue normal processing)
- 4 (record replaced)
- 8 (insert record)
- 12 (delete record)
- 16 (record modified)
- 20 (disconnect exit).

Reserved (Eioption 10)

This exit point is not currently used by DFDSS.

Writing WTO Message (Eioption 11)

This exit point is called when DFDSS is ready to write a WTO message which was specified in the SYSIN. You can replace, insert, delete, or modify the WTO message at this exit point. The EIRECPTR points to the WTO message.

The valid return codes for writing WTO messages are:

- 0 (continue normal processing)
- 4 (record replaced)
- 8 (insert record)
- 12 (delete record)
- 16 (record modified)
- 20 (disconnect exit).

- For each component or data set:
 - The number of volumes it resides on
 - The volser of each volume
 - The RBA of the control interval in the VVDS of the VVDS entry of each volume.

EIRECPTR points to EIREC20. EI20DA@ and EI20IX@ point to EI20DSI.

Note: Because SMS-managed multivolume non-VSAM data sets have a VVDS entry only on the first volume, RBA information is passed for just one volume. The RBA in all other volume entries will be zero. Likewise, the RBA information for non-SMS-managed non-VSAM data sets will always be zero.

The valid return codes for volume notification are:

- 0 (continue normal processing)
- 20 (disconnect exit)
- 32 (end function—Only valid for full or physical data set DUMP and full or physical data set RESTORE).

Data Set Verification (Eioption 21)

This exit allows the user interaction module (UIM) to end logical COPY, DUMP, and RESTORE processing for individual data sets. This exit is given control through the UIM at the start of logical COPY, logical DUMP, and logical RESTORE processing for each data set. DFDSS provides the UIM with the data set name through the EIREC21 structure within the exit identification block, ADREID0. EIREC21 is shown in Figure 27 on page 197. The name provided is the original data set name prior to rename processing, if any, for COPY and RESTORE. The UIM returns to DFDSS with a return code in the exit identification block.

The valid return codes for data set verification are:

- 0 (continue normal processing)
- 20 (disconnect exit)
- 32 (end function)
- 36 (end data set)

DFDSS will continue processing with the next data set, if any. If the user specifies the SPHERE keyword and processing for a base cluster is ended through this exit, processing for all AIXs related to the base cluster is also ended. If the user specifies the SPHERE keyword and the UIM attempts to end processing for an AIX that is related to a base cluster that was selected for processing, DFDSS invokes the UIM again. If the UIM attempts to end processing for the AIX again, DFDSS issues warning message ADR770W and ignores the request. If the user does not specify the SPHERE keyword, AIXs can be ended even if they are related to base clusters selected for processing. Ending a base cluster without the SPHERE keyword does not cause any related AIXs to be ended.

Note: See the Note for Eioptions 21, 22, 23 on page 194.

Bypass Verification Exit (Eioption 22)

This exit allows a user to force DFDSS to bypass serialization and security verification during logical COPY, DUMP, and RESTORE processing for individual data sets. It also allows a user to turn on a tolerate-migrated-volser indicator that forces DFDSS to restore a data set with a migrated volser. This exit is given control through the UIM at the start of logical COPY, logical DUMP, and logical RESTORE processing for each data set. DFDSS provides the UIM with the data set name through the EIREC22 structure within the exit identification block, ADREID0. EIREC22 is shown in Figure 27 on page 197. The data set name provided is the original name prior to rename processing, if any, for COPY or RESTORE. The UIM returns to DFDSS with a return code in the exit identification block.

The valid return codes for data set verification are:

- 0 (continue normal processing)
- 16 (bypass one or more of the following:)
- Serialization via ENQ or SVC 99 is bypassed if the UIM exit turns on the bypass serialization indicator, EI22BSER. Serialization is bypassed for the source data set for COPY and DUMP. Serialization is also bypassed for the target data set for COPY and RESTORE. DFDSS assumes that all necessary serialization has already been performed by the invoker of DFDSS. However, DFDSS does no checking to ensure that this is true. DFDSS performs normal serialization if the UIM exit does not turn on the bypass serialization indicator.

DFDSS does not delete or uncatalog data sets that were bypassed for serialization even if the DELETE or UNCAT keywords are specified. If a pre-allocated data set is not large enough during a data set RESTORE and the bypass serialization indicator is on, DFDSS does not scratch and reallocate that target data set and the RESTORE fails.

- RACF verification and all other data set security checking is bypassed if the UIM exit turns on the bypass RACF indicator, EI22BSEC. If EI22BSEC is set on, DFDSS checks to ensure that the application program is authorized to bypass RACF and security processing. The application program is authorized to bypass RACF and security processing if the JSCBPASS flag is on in the job step control block. If the EI22BSEC and JSCBPASS indicators are on, RACF verification and all other data set security processing including password checking is bypassed. DFDSS does not do any RACF authorization checks. DFDSS does not create any RACF profiles for the copied or restored target data set. DFDSS assumes that all necessary RACF authorization and security checking has already been performed by the invoker of DFDSS. For a COPY or RESTORE operation, RACF profiles are not created for the target data set.

If the user turns on the bypass RACF indicator and the JSCBPASS indicator is off, DFDSS sets the error flag, EIXERR, on in the exit identification block and invokes the UIM again. If the user sets the bypass RACF indicator again, DFDSS issues error message ADR772W and processing continues as normal. DFDSS does still allow serialization to be bypassed and tolerate migrated volser processing if those indicators are set.

DFDSS performs normal RACF and security processing if the UIM exit does not turn on the bypass RACF indicator.

- If the tolerate-migrated-volser indicator is set on by the UIM, DFDSS takes special action to support the restoration of a non-VSAM data set with a

volser of MIGRAT for a logical data set RESTORE operation in an SMS environment or when CATALOG has been specified. DFDSS ignores the EI22BMIG indicator for COPY and DUMP operations and VSAM data sets.

When DFDSS is restoring a non-VSAM data set and the tolerate-migrated-volser indicator is set on, a catalog LOCATE is issued to determine the status of the data set being restored. Based on the result of that LOCATE, different actions will be taken.

- No catalog entry is found: In this case, the data set being restored is not considered to be migrated. A normal DFDSS logical restore is performed. The tolerate-migrated-volser indicator is ignored.
- A catalog entry is found, but the VOLSER is not MIGRAT: Again, the data set is not considered to be migrated. As above, normal processing continues as though the indicator had not been set.
- A catalog entry is found and the VOLSER is MIGRAT: The data set is indeed migrated and requires special processing. Instead of cataloging the data set after allocating it, DFDSS alters the existing MIGRAT volser entry to the actual volume the data set was restored to. Once the catalog entry has been changed from MIGRAT to the new data set's volsers, the restore continues as normal.

Note: Because the migrated data set is not recalled during this restore, the user of this interface is responsible for deleting the migrated copy of the data set and updating the necessary control files.

DFDSS performs normal non-VSAM data set cataloging if the UIM exit does not turn on the tolerate-migrated-volser indicator.

If the UIM exit sets the return code to EIRC16 but fails to turn on any of the bypass verification indicators, DFDSS ignores all bypass options and performs normally. If the UIM exit sets the return code to EIRC16 for a VSAM data set and turns on the tolerate-migrated-volser indicator, DFDSS does not treat it as an error condition and ignores the tolerate-migrated-volser indicator.

- 20 (disconnect exit)
- 32 (end function)
- 36 (end data set).

Refer to EIRC36 in "Data Set Verification (Eioption 21)" on page 192 for details on ending spheres.

Note: See the Note for Eioptions 21, 22, 23 on page 194.

Data Set Processed Notification Exit (Eioption 23)

This exit indicates to the UIM whether or not the logical COPY, DUMP, or RESTORE processing for individual data sets was successful. This exit is given control through the UIM at the conclusion of logical COPY, logical DUMP, and logical RESTORE processing for each data set. DFDSS provides the UIM with information through the EIREC23 structure within the exit identification block, ADREID0. EIREC23 is shown in Figure 27 on page 197..

DFDSS provides the UIM with the following information at the conclusion of processing for logical DUMP of each data set:

- The data set name that was dumped.
- A return code for the data set that indicates whether processing was successful.

- 0 (Data set completely successful (informational))
- 4 (Data set partially successful (warning))
- 8 (Data set unsuccessful (error))
- 12 (Ending error)
- 16 (Ending error)
- The source SMS flag (DSISMSFG) from the Format 1 DSCB.
- The source data set organization (DSIDSORG) from the Format 1 DSCB for a non-VSAM data set.
- Flags to indicate data set type for a VSAM data set.

DFDSS provides the UIM with the following information at the conclusion of processing for logical COPY and logical RESTORE of each data set:

- The original data set name that was copied or restored. This is the data set name prior to rename processing, if any.
- The new data set name being copied or restored if rename processing was performed.
- A return code for the data set that indicates whether processing succeeded:
 - 0 Data set completely successful (informational)
 - 4 Data set partially successful (warning)
 - 8 Data set unsuccessful (error)
 - 12 Ending error
 - 16 Ending error
- The source and target SMS flags (DSISMSFG) from the Format 1 DSCB.
- The source and target data set organization (DSIDSORG) from the Format 1 DSCB for a non-VSAM data set.
- Flags to indicate data set type for a VSAM data set.
- A count of the volumes that the data set was copied or restored to.
- A list of the volume serial numbers (VOLSERs) that the data set was copied or restored to.

The UIM returns to DFDSS with a return code in the exit identification block. The valid return codes for data set verification are:

- 0 (continue normal processing)
- 20 (disconnect exit)
- 32 (end function)
- 36 (end data set)

It does not undo any successful processing, but deletes the data set from the successfully-processed message list (if applicable) and includes it in the unsuccessfully-processed message list. It is the responsibility of the user of this interface to delete the copied, dumped, or restored data set from the target volumes. DFDSS continues processing with the next data set, if any. Refer to EIRC36 in "Data Set Verification (Eioption 21)" on page 192 for details on ending spheres.

Note for Eioptions 21, 22, 23: If Eioptions 21, 22, and 23 end the function (the User Interaction Module (UIM) passes back a return code 32 to DFDSS) during data set COPY, DFDSS can be processing more than one data set at the time. This can happen if utilities are needed to process some of the data sets. DFDSS does the following steps before ending:

- No unprocessed data sets are scheduled for processing.
- All data sets in utility processing are allowed to complete normally.
- No new calls are made to exits 21, 22, or 23. This includes data sets in utility processing that are allowed to complete.

- Spheres that are being processed because of the SPHERE keyword and that are not complete at the end of the function will have the target parts deleted to preserve sphere integrity. Preallocated spheres are left partially copied.
- Spheres being processed without the SPHERE keyword or individual sphere components being copied are left partially completed.

Avoiding Lockout

Refer to "Avoiding Lockout" on page 166 for a description of the ENQ scheme to prevent lockout.

Application Interface Summary

Record Processing: If the UIM makes any changes to the presented record (assuming it can validly do so), return code 16 must be returned or the changes are ignored. If the record is to be replaced in total, return code 4 must be returned or the original record is used. If a record is to be inserted before the current record, return code 8 must be returned or the record is ignored. If the current record is to be deleted, return code 12 must be returned or the current record is processed. If the exit is no longer interested in processing any records appearing at the current DFDSS exit point, return code 20 must be returned or the exit is called at the next visitation of DFDSS to the exit point. The current record will still be processed in either event. If you want to receive only user statistical records at any specific exit, return code 24 must be returned. If you want to process a WTOR within the UIM, the response must be returned to DFDSS with a return code 28.

If a record is being returned that is longer than the original record, the exit must either replace the record in the area pointed to by the original record pointer or supply another area and store the address to the record in the original record pointer field. In either case, the length must be stored in the original record length field and a return code of 4 must be used. If a record is being returned that is shorter than the original record, the exit can supply an area and store the address to the record in the original record pointer field or, if allowed, can merely replace the original record with the shorter one. In either case, the length of the new record must be placed in the original record length field. If the new area option was used, return code 4 must be used. If the new record overlaid the original, return code 16 must be used. Figure 27 on page 197 describes the data area corresponding to the DFDSS exit identification block.

Data Set Processing: Eioptions 21, 22, and 23 are three exits that give you added control over data set processing during a logical data set COPY, DUMP, and RESTORE. These exits are called immediately prior to and immediately after each data set's being processed, on a data set by data set basis, and allow you to make a number of processing changes.

Eioptions 21 and 22 are called, one after the other, at the start of processing of each data set. DFDSS passes each exit the name of the data set being processed. Based on that name, each exit allows you to end processing of just that data set (return code 36); end processing of the whole functional task (return code 32); disconnect the exit so that it is no longer called prior to subsequent data sets being processed by that task (return code 20); or do nothing (return code 0). Eioption 22 also allows you to modify the way in which the data set is to be processed (return code 16). Refer to "Bypass Verification Exit (Eioption 22)" on page 193 for more details.

Eioption 23 is called immediately after a data set has been processed and also allows you to end processing of the data set, end processing of the task, disconnect the exit, or do nothing.

ADREID0 Data Area

The following figures describe the data area corresponding to the DFDSS exit identification block.

Figure 27 (Page 1 of 3). ADREID0 Mapping Macro

Offsets	Type	Length	Name	Description	
0	(0)	CHARACTER	42	ADREIB	
0	(0)	FIXED	2	EIDLLEN	LENGTH OF ADREIB 2
2	(2)	CHARACTER	4	EIID	BLOCK IDENTIFIER EBCDIC "EIDB"
6	(6)	FIXED	4	EITSKID	TASK ID NUMBER
10	(A)	BITSTRING	4	EIXALLOW	USER EXIT ALLOWANCE OPTIONS
10	(A)	BITSTRING	1	EIXALLOW0	ALLOWANCE OPTION BYTE 1
		1... ..		EIXREP	ALLOW REPLACE OF RECORD
		.1.. ..		EIXINS	ALLOW INSERTION OF RECORD
		..1.		EIXDEL	ALLOW DELETION OF RECORD
		...1		EIXMOD	ALLOW MODIFICATION OF RECORD
	 1...		EIXDIS	ALLOW DISCONNECT OF EXIT
	1..		EIXWTOR	ALLOW WTOR RESPONSE
	1.		EIXSTAT	ALLOW SELECTION OF USER STATS
	1		EIXTERM	ALLOW FUNCTION ENDING
11	(B)	BITSTRING	1	EIXALLOW1	ALLOWANCE OPTION BYTE 2
		1... ..		EIXTDSET	ALLOW DATA SET ENDING
		.111 1111		*	UNUSED
12	(C)	BITSTRING	1	EIXALLOW2	ALLOWANCE OPTION BYTE 3
12	(C)	BITSTRING	1	*	RESERVED
13	(D)	BITSTRING	1	EIXALLOW3	ALLOWANCE OPTION BYTE 4
		1... ..		EIXERR	DISALLOWED OPTION ATTEMPTED
		.111 1111		*	RESERVED
14	(E)	FIXED	2	EIOPTION	PROCESSING OPTION
16	(10)	FIXED	2	EIRETCOD	EXIT RETURN CODE
18	(12)	FIXED	4	EIRECALN	RECORD AREA LENGTH
22	(16)	FIXED	4	EIRECLEN	ORIGINAL RECORD LENGTH
26	(1A)	ADDRESS	4	EIRECPTR	ORIGINAL RECORD ADDRESS
30	(1E)	ADDRESS	4	EIUSEPTR	USER DATA AREA ADDRESS
34	(22)	ADDRESS	4	EIIDID	EIOP06 DDNAME VALID PTR
38	(26)	BITSTRING	4	EIXFLAGS	OTHER FLAGS
38	(26)	BITSTRING	1	EIXFLAG0	FLAG BYTE 0
		1... ..		EIXABEND	FOR EIOP02 ONLY. 1=MESSAGE IS ADR013 INDICATING AN ABEND CONDITION
		..11 1111		EIXNTERR	FOR EIOP02 ONLY. 1=MESSAGE IS TYPE 'E' AND IS NOT 324 OR 347
				*	RESERVED
39	(27)	BITSTRING	1	EIXFLAG1	FLAG BYTE 1
39	(27)	BITSTRING	1	*	RESERVED
40	(28)	BITSTRING	1	EIXFLAG2	FLAG BYTE 2
40	(28)	BITSTRING	1	*	RESERVED
41	(29)	BITSTRING	1	EIXFLAG3	FLAG BYTE 3
41	(29)	BITSTRING	1	*	RESERVED
42	(2A)	CHARACTER		*	FORCE SIZE OF CONTROL BLOCK TO WORD BOUNDARY
0	(0)	CHARACTER	16	EIDINFO	UIM 06 INFO
0	(0)	CHARACTER	3	EIDDDNAME	O P DEVICE DDNAME
8	(8)	CHARACTER	6	EIVOLID	O P VOLSER
14	(E)	FIXED	1	EIRETC	RESERVED FOR RETCODE
15	(F)	FIXED	1	*	RESERVED FOR FLAGS
0	(0)	CHARACTER	56	EIREC20	
0	(0)	CHARACTER	44	EI20DSN	DATA SET NAME CLUSTER NAME
44	(2C)	BITSTRING	1	EI20FLGS	SOME FLAGS
		1... ..		EI20VSAM	1=DATA SET IS VSAM 0=DATA SET IS NONVSAM
		..1.		EI20RACF	1=DATA SET IS PROTECTED BY A DISCRETE RACF PROFILE

Figure 27 (Page 2 of 3). ADREID0 Mapping Macro

Offsets	Type	Length	Name	Description
	..11 1111		*	RESERVED
45	(2D) CHARACTER	1	*	RESERVED
46	(2E) FIXED	1	EI20DA#	NUMBER OF DATA COMPONENTS
47	(2F) FIXED	1	EI20IX#	NUMBER OF INDEX COMPONENTS (0 IF NONVSAM)
48	(30) ADDRESS	4	EI20DA@	POINTER TO DATA COMPONENT INFO FOR VSAM/DATA SET INFO FOR NONVSAM
52	(34) ADDRESS	4	EI20IX@	POINTER TO INDEX COMPONENT INFO FOR VSAM/0 FOR NONVSAM
0	(0) CHARACTER	60	EI20DSI	DATA SET INFO
0	(0) CHARACTER	44	EI20CON	COMPONENT NAME (BLANKS FOR NONVSAM)
44	(2C) FIXED	2	EI20NVOL	# OF VOLUMES IN DS
46	(2E) CHARACTER	2	*	RESERVED
48	(30) CHARACTER	12	EI20VLI	VOLUME INFORMATION
48	(30) CHARACTER	6	EI20VOL	VOLSER
54	(36) CHARACTER	2	*	RESERVED
56	(38) FIXED	4	EI20RBA	RBA OF VVDS ENTRY/0 IF NONE
0	(0) CHARACTER	44	EIREC21	
0	(0) CHARACTER	44	EI21DSN	DATA SET/CLUSTER NAME
0	(0) CHARACTER	50	EIREC22	
0	(0) CHARACTER	44	EI22DSN	DATA SET/CLUSTER NAME
44	(2C) BITSTRING	1	EI22FLGS	EXIT 22 FLAGS
	1... ..		EI22BSER	1 = BYPASS ENQ/SVC 99 SERIALIZATION
	.1.. ..		EI22BSEC	1 = BYPASS SECURITY CHECKING RACF PROFILE CREATION
	..1.		EI22BMIG	1 = TOLERATE MIGRATED VOLSER FOR NONVSAM DSETS (RESTORE ONLY)
	...1		EI22BFBU	RESERVED
 1111		*	RESERVED
45	(2D) CHARACTER	5	EI22SFLG	SOURCE DATA SET FLAGS
45	(2D) CHARACTER	1	E22SSMSF	SOURCE SMS FLAGS (DS1SMSFG) FROM FORMAT 1 DSCB
46	(2E) CHARACTER	2	E22SDSRG	SOURCE DATA SET ORG(DS1DSORG) FROM FORMAT 1 DSCB
48	(30) BITSTRING	2	E22SVFLG	SOURCE VSAM DSET FLAGS
	1... ..		E22SESDS	1 = ESDS DATA SET
	.1.. ..		E22SKSDS	1 = KSDS DATA SET
	..1.		E22SKRDS	1 = KRDS DATA SET
	...1		E22SLDS	1 = LINEAR DATA SET
 1...		E22SRRDS	1 = RRDS DATA SET
1..		E22SPSSI	1 = PAGE/SWAP STGINDEX, ETC. UNSUPPORTED DATA SETS
1.		E22SVVDS	1 = VVDS DATA SET
1		E22SBCS	1 = BCS DATA SET
	1... ..		E22SAIX	1 = AIX DATA SET
	.111 1111		*	RESERVED
0	(0) CHARACTER	*	EIREC23	
0	(0) CHARACTER	44	EI23DSN	DATA SET CLUSTER NAME
44	(2C) CHARACTER	44	EI23NEWN	NEW DSET CLUSTER NAME
88	(58) FIXED	2	EI23DSRC	RETURN CODE FOR DATA SET PROCESSING
90	(5A) CHARACTER	5	EI23SFLG	SOURCE DATA SET FLAGS
90	(5A) CHARACTER	1	E23SSMSF	SOURCE SMS FLAGS (DS1SMSFG) FROM FORMAT 1 DSCB
91	(5B) CHARACTER	2	E23SDSRG	SOURCE DATA SET ORG(DS1DSORG) FROM FORMAT 1 DSCB
93	(5D) BITSTRING	2	E23SVFLG	SOURCE VSAM DSET FLAGS
	1... ..		E23SESDS	1 = ESDS DATA SET
	.1.. ..		E23KSDS	1 = KSDS DATA SET
	..1.		E23SKRDS	1 = KRDS DATA SET
	...1		E23SLDS	1 = LINEAR DATA SET
 1...		E23SRRDS	1 = RRDS DATA SET
1..		E23SPSSI	1 = PAGE/SWAP STGINDEX, ETC. UNSUPPORTED DATA SETS
1.		E23SVVDS	1 = VVDS DATA SET
1		E23SBCS	1 = BCS DATA SET
	1... ..		E23SAIX	1 = AIX DATA SET
	.111 1111		*	RESERVED
95	(5F) CHARACTER	5	EI23TFLG	SOURCE DATA SET FLAGS
95	(5F) CHARACTER	1	E23TSMF	SOURCE SMS FLAGS (DS1SMSFG) FROM FORMAT 1 DSCB

Figure 27 (Page 3 of 3). ADREID0 Mapping Macro

Offsets	Type	Length	Name	Description
96	(60) CHARACTER	2	E23TDSRG	SOURCE DATA SET ORG(DS1DSORG) FROM FORMAT 1 DSCB
98	(62) BITSTRING	2	E23TVFLG	TARGET VSAM DSET FLAGS
	1... ..		E23TESDS	1 = ESDS DATA SET
	.1..		E23TKSDS	1 = KSDS DATA SET
	..1.		E23TKRDS	1 = KRDS DATA SET
	...1		E23TLDS	1 = LINEAR DATA SET
 1..		E23TRRDS	1 = RRDS DATA SET
1..		E23TPSSI	1 = PAGE/SWAP/STGINDEX. ETC. UNSUPPORTED DATA SETS
1.		E23TVVDS	1 = VVDS DATA SET
1		E23TBCS	1 = BCS DATA SET
	1...		E23TAIX	1 = AIX DATA SET
	.111 1111		*	RESERVED
100	(64) FIXED	1	EI23VOL#	NUMBER OF VOLUMES
101	(65) CHARACTER	6	EI23VSR (*)	VOLSER ARRAY

Figure 28. Constants

Length	Type	Value	Name	Description
4	CHARACTER	EIDB	ADREIBID	BLOCK IDENTIFIER
2	DECIMAL	0	EIRC00	CONTINUE NORMAL PROCESS
2	DECIMAL	4	EIRC04	RECORD REPLACED
2	DECIMAL	8	EIRC08	INSERT RECORD
2	DECIMAL	12	EIRC12	DELETE RECORD
2	DECIMAL	16	EIRC16	RECORD MODIFIED
2	DECIMAL	20	EIRC20	DISCONNECT EXIT
2	DECIMAL	24	EIRC24	SELECT USER STATS RECS
2	DECIMAL	28	EIRC28	WTOR RESPONSE
2	DECIMAL	32	EIRC32	END FUNCTION
2	DECIMAL	36	EIRC36	END DSET ONLY
2	DECIMAL	0	EIOP00	FUNCTION STARTUP ENTRY
2	DECIMAL	1	EIOP01	READING SYSIN RECORD
2	DECIMAL	2	EIOP02	PRINTING SYSPRINT RECORD
2	DECIMAL	3	EIOP03	READING PHYSICAL TAPE
2	DECIMAL	4	EIOP04	READING LOGICAL TAPE
2	DECIMAL	5	EIOP05	WRITING LOGICAL TAPE
2	DECIMAL	6	EIOP06	WRITING PHYSICAL TAPE
2	DECIMAL	7	EIOP07	READING DISK TRACK
2	DECIMAL	8	EIOP08	WRITING DISK TRACK
2	DECIMAL	9	EIOP09	READING UTILITY SYSPRINT
2	DECIMAL	10	EIOP10	WRITING UTILITY SYSPRINT
2	DECIMAL	11	EIOP11	WRITING WTO MESSAGE
2	DECIMAL	12	EIOP12	WRITING WTO MESSAGE
2	DECIMAL	13	EIOP13	PRESENTING ADRUFO REC
2	DECIMAL	14	EIOP14	FUNCTION ENDING
2	DECIMAL	15	EIOP15	PRESENTING WTO RESPONSE
2	DECIMAL	16	EIOP16	TAPE VOL SECURITY
2	DECIMAL	17	EIOP17	TAPE MOUNT(NON SPEC)
2	DECIMAL	18	EIOP18	INSERT LOGICAL REC
2	DECIMAL	19	EIOP19	TAPE OUTPUT ERROR
2	DECIMAL	20	EIOP20	VOLUME NOTIFICATION
2	DECIMAL	21	EIOP21	DSET VERIFICATION
2	DECIMAL	22	EIOP22	BYPASS VERIFICATION
2	DECIMAL	23	EIOP23	DS PROC NOTIFICATION
4	DECIMAL	42	EIDBLEN	LENGTH OF BLOCK

Application Interface

Name	Hex Offset	Struct Level	Name	Hex Offset	Struct Level
ADREIB	0	1	E22SPSSI	30	4
EIDDID	22	2	E22SRRDS	30	4
EIDDINFO	0	1	E22SSMSF	2D	3
EIDDNAME	0	2	E22SVFLG	30	3
EIDLLEN	0	2	E22SVVDS	30	4
EIID	2	2	E23SAIX	5E	4
EIOPTION	E	2	E23SBCS	5D	4
EIRECALN	12	2	E23SDSRG	5B	3
EIRECLEN	16	2	E23SESDS	5D	4
EIRECPTR	1A	2	E23SKRDS	5D	4
EIREC20	0	1	E23SKSDS	5D	4
EIREC21	0	1	E23SLDS	5D	4
EIREC22	0	1	E23SPSSI	5D	4
EIREC23	0	1	E23SRRDS	5D	4
EIRETC	E	2	E23SSMSF	5A	3
EIRETCOD	10	2	E23SVFLG	5D	3
EITSKID	6	2	E23SVVDS	5D	4
EIUSEPTR	1E	2	E23TAIX	63	4
EIVOLID	8	2	E23TBCS	62	4
EIXABEND	26	4	E23TDSRG	60	3
EIXALLOW	A	2	E23TESDS	62	4
EIXALLOW0	A	3	E23TKRDS	62	4
EIXALLOW1	B	3	E23TKSDS	62	4
EIXALLOW2	C	3	E23TLDS	62	4
EIXALLOW3	D	3	E23TPSSI	62	4
EIXDEL	A	4	E23TRRDS	62	4
EIXDIS	A	4	E23TSMSF	5F	3
EIXERR	D	4	E23TVFLG	62	3
EIXFLAGS	26	2	E23TVVDS	62	4
EIXFLAG0	26	3			
EIXFLAG1	27	3			
EIXFLAG2	28	3			
EIXFLAG3	29	3			
EIXINS	A	4			
EIXMOD	A	4			
EIXNTERR	26	4			
EIXREP	A	4			
EIXSTAT	A	4			
EIXTDSET	B	4			
EIXTERM	A	4			
EIXWTOR	A	4			
EI20CON	0	2			
EI20DA#	2E	2			
EI20DA@	30	2			
EI20DSI	0	1			
EI20DSN	0	2			
EI20FLGS	2C	2			
EI20IX#	2F	2			
EI20IX@	34	2			
EI20NVOL	2C	2			
EI20RACF	2C	3			
EI20RBA	38	3			
EI20VLI	30	2			
EI20VOL	30	3			
EI20VSAM	2C	3			
EI21DSN	0	2			
EI22BFBU	2C	3			
EI22BMIG	2C	3			
EI22BSEC	2C	3			
EI22BSER	2C	3			
EI22DSN	0	2			
EI22FLGS	2C	2			
EI22SFLG	2D	2			
EI23DSN	0	2			
EI23DSRC	58	2			
EI23NEWN	2C	2			
EI23SFLG	5A	2			
EI23TFLG	5F	2			
EI23VOL#	64	2			
EI23VSER	65	2			
E22SAIX	31	4			
E22SBCS	30	4			
E22SDSRG	2E	3			
E22SESDS	30	4			
E22SKRDS	30	4			
E22SKSDS	30	4			
E22SLDS	30	4			

An Example of Invoking DFDSS by Using an Application Program

The following example shows that a full DASD volume is to be dumped to a tape volume(s). DFDSS is LINKed somewhere in MYJOB to perform the DUMP, and, conditionally, the DEFRAg functions.

```
//JOB1    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=MYJOB
//STEPLIB DD    DSN=MY.LINKLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=A
//DASD    DD    UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//TAPE    DD    UNIT=3480,VOL=SER=(TAPE01,TAPE02),
//        LABEL=(1,NL),DISP=(NEW,KEEP)
//SYSIN   DD    *
          DUMP INDD(DASD) OUTDD(TAPE)
          IF LASTCC = 0 -
            THEN DEFRA DDN(DASD)
/*
```

The preceding example does not show how the invocation of DFDSS was brought about, but does show that the user program, MYJOB, was run. Somewhere within MYJOB it is desired to run DFDSS to perform the functions specified in the SYSIN data set. At this point, the next example shows the necessary code used to LINK to DFDSS. Because no EXEC PARMs were specified and the standard SYSIN and SYSPRINT data set names are to be used, there is no need to pass special parameters.

```

.
.
.
LINK EP=ADRSSU,PARAM=(OPTPTR),VL=1
.
.
.
          CNOP 2,4
OPTPTR  DC  H(0)
```

Note: For additional information on the Application Interface, see Appendix D, “Examples of the Application Program with the User Interaction Module (UIM)” on page 203

How to Determine DFDSS Version, Release, and Modification Level

Subsystems that invoke DFDSS dynamically need to know if DFDSS is installed on the system, and if it is, its version, release, and modification level. A DFDSS-provided macro attempts to load the DFDSS load module. If the load is successful, it determines the DFDSS version, release, and modification level and passes the requested information in a register. The macro also causes the load module to be deleted from main storage.

ADRMCLVL (in SYS1.SAMPLIB) is an in-line executable assembler-language macro that can be invoked by a caller. The caller can be in problem program state and have a user key. The caller must save registers 0, 1, 14, and 15 before invoking the macro. No other registers are disturbed.

On return, register 1 contains information as follows:

- If the release level of ADRDSSU cannot be determined, register 1 contains X'04000000'.
- Otherwise, register 1, byte 0 contains X'00', and the three low-order bytes contain:
 - Byte 1: Version number (in binary)
 - 1 = Version 1
 - 2 = Version 2
 - Byte 2: Release number (in binary)
 - 1 = Release 1
 - 2 = Release 2
 - 3 = Release 3
 - 4 = Release 4
 - 5 = Release 5
 - Byte 3: Modification level (in binary)
 - 0 = Modification level 0
 - 1 = Modification level 1
 - 2 = Modification level 2

Figure 29 shows the expansion of the macro ADRMCLVL.

```

MACRO
ADRMCLVL
LOAD EP=ADRRELVL,ERRET=ADR1&SYSNDX
LR 1,0
- 1,0(1)
B ADR2&SYSNDX
ADR1&SYSNDX LA 1,4(0,0)
SLL 1,24(0)
B ADR3&SYSNDX
ADR2&SYSNDX DELETE EP=ADRRELVL
ADR3&SYSNDX DS OH
MEND
* register 1 contents:
* byte 0: 0 if no errors, 4 otherwise
* byte 1: version number (0 if errors or indeend)
* byte 2: release number (0 if errors or indeend)
* byte 3: modification number (0 if errors or indeend)
    
```

Figure 29. Expansion of Macro ADRMCLVL

Appendix D. Examples of the Application Program with the User Interaction Module (UIM)

Note: The example shown is not written in reentrant code. If you are planning to share a UIM between tasks, you should code the module in reentrant code.

The example has not been submitted to any formal test and is distributed as it is, without any warranty either expressed or implied. The use of this example or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. Although each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk. See Figure 31 on page 204 and Figure 32 on page 206 for a complete sample program listing showing how a user might use all of the UIM exit functions to receive control from DFDSS. Figure 33 on page 218 shows what an output listing might look like as the result of the sample program in Figure 31 on page 204 and Figure 32 on page 206.

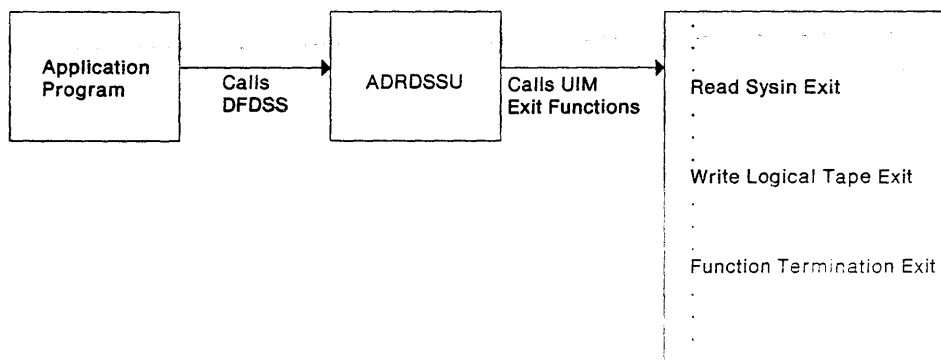


Figure 30. The Application Program Process

```

*****
*
* Module Name      = USRAIPGM
*
* Descriptive Name = DFDSS Application Interface Program Example
*                   for the Reference Guide.
*
* Function         = Invoke DFDSS supplying alternate UIM: USRUIM
*
* Operation       = Place SYSIN records in the Userarea. The UIM
*                   will pass these records to DFDSS.
*
*****
USRAIPGM CSECT
USRAIPGM AMODE 31
USRAIPGM RMODE ANY
*
      STM 14,12,12(13)
      BALR 12,0
      USING *,12
      LA 3,SAVEAREA
      ST 3,8(13)
      ST 13,4(3)
      LR 13,3
*
      LOAD EP=ADRDUSSU
      LR 15,0
      LA 3,USERAREA
      ST 3,UAG
      CALL (15), (OPTPTR,DDNPTR,PAGEPTR,UIMPTR,UAPTR),VL
      L 13,4(13)
      RETURN (14,12),,RC=(15)
*
      CNOP 2,4
OPTPTR DC AL2(OPTLEN)
OPTIONS DC C'SIZE=4096K'
OPTLEN EQU *-OPTIONS
      CNOP 2,4
DDNPTR DC AL2(DDNLEN)
DDNAMES DC XL8'00'
      DC XL8'00'
      DC XL8'00'
      DC XL8'00'
      DC CL8'MYDATA'
      DC CL8'SYSRINT'
DDNLEN EQU *-DDNAMES
      CNOP 2,4
PAGEPTR DC AL2(PAGELEN)
PAGENO DC CL4'0100'
PAGELEN EQU *-PAGEPTR
      CNOP 2,4
UIMPTR DC AL2(UIMLEN)
UIM DC CL8'USRUIM'
UIMLEN EQU *-UIM
*

```

Figure 31 (Part 1 of 2). Application Interface Program Example


```

      CNOP 2,4
UAPTR DC AL2(UALEN)
UA@   DS   A
UALEN EQU *-UA@
*
USERAREA DC AL1(COM1)           Length of first command
        DC   C' WTO ''USRAIPGM INVOKING DFDSS''
COM1    EQU *-USERAREA-1
SECNDCOM DC AL1(COM2)           Length of first command
        DC   C' DUMP DS(INC(**)) LOGINDD(DASD) OUTDD(TAPE)'
COM2    EQU *-SECNDCOM-1
THIRDCOM DC AL1(COM3)           Length of third command
        DC   C' IF LASTCC=0 -'
COM3    EQU *-THIRDCOM-1
FOURCOM DC AL1(COM4)           Length of fourth command
        DC   C' THEN DEFRAG DDN(DASD)'
COM4    EQU *-FOURCOM-1
ENDCOMM DC X'00'               End of command flag
*
SAVEAREA DS 18F
*
      END

```

Figure 31 (Part 2 of 2). Application Interface Program Example

```

*****
*
* Module Name      = USRUIM
*
* Descriptive Name = DFDSS User Interaction Module Example for the
*                   Reference Guide.
*
* Function         = Perform various functions at different exit
*                   points.
*
* Exit Point      Operation
* -----
* 0               Initialize counters for task
* 1               Insert SYSIN records from Userarea
* 2               Insert table of accounting data
* 3               Count Tape Blocks Read
* 4               Count Logical Tape Blocks Read
* 5               Count Logical Tape Blocks Written
* 6               Count Tape Blocks Written
* 7               Count DASD Tracks Read
* 8               Count DASD Tracks Written
* 9               Nothing
* 10              (Reserved)
* 11              Insert a WTO message
* 12              Never allow ADR369D
* 13              Force Reblocking
* 14              Convert counts to printable characters
* 15              Nothing
* 16              Always bypass password protection for tape
* 17              Supply TAPE01 as a tape volser
* 18              Nothing
* 19              Save Tape DDNAME and volser which had error
* 20              Save Data Set name
* 21              Do not process temporary data sets
* 22              Bypass serialization of SYS1 data sets
* 23              End function if processing errors
*****
*
USRUIM CSECT
USRUIM AMODE 31
USRUIM RMODE ANY
*****
* Registers with special uses
*****
*
EIDBASE EQU 2           Base reg for ADREIB.
OPTION  EQU 3           Reg to test option
WORKREG EQU 4           Work register
LENGTH  EQU 5           Reg to get length of records
TEMPBASE EQU 6          Temporary base reg for exits
RC      EQU 7           Register for return code
SYSPBASE EQU 8          Register for sysprint recs
MSGOFF  EQU 9           Register for offsets to table

```

Figure 32 (Part 1 of 12). User Interaction Module Example

```

*****
* Save Registers and Establish Addressability. *
*****
    USING *,15           Initial Addressability
    STM 14,12,12(13)     Save regs
    BALR 12,0           New base addressability
    USING *,12
    DROP 15
    B BEGIN
    DC CL16'CSECT - USRUIM'
*****
* Establish Addressability to EIDB *
*****
BEGIN LR 15,13          Save caller SA pointer
      LA 13,SAVEAREA     Set my SA pointer
      ST 15,SAVEAREA+4   Backward Chain
      ST 13,SAVEAREA+8   Forward Chain
      USING ADREIB,EIDBASE Addressability to EIDB
      L EIDBASE,0(,1)    Address of EIDB
      LH OPTION,EIOPTION Get DFSS processing option
      SLL OPTION,2       Multiply times four
      LA 15,VECTABLE     Point to vector table start
      L 15,0(OPTION,15)  Point to processor routine
      BALR 14,15         Go to processor routine
      L 13,SAVEAREA+4   Restore SA pointer
      RETURN (14,12)    Return to DFSS
*****
* Processing Routines *
*****
* AIOPT00: Function Startup *
* Initialize counters for tape and dasd accounting. *
* Insert message identifying task into table *
*****
AIOPT00 XC RTBCNT,RTBCNT Initialize Tape Blocks Read
        XC WTBCNT,WTBCNT Initialize Tape Blocks Written
        XC RLTCNT,RLTCNT Init Logical Tape Blocks Read
        XC WLTBCNT,WLTBCNT Init Logical Tape Blocks Written
        XC RDTCNT,RDTCNT Initialize Disk Tracks Read
        XC WDTCNT,WDTCNT Initialize Disk Tracks Written
        ICM WORKREG,15,EITSKID Get current task identifier
        CVD WORKREG,CVDWORK Convert to decimal
        UNPK OPT00TSK(2),CVDWORK+6(2) Unpack into message insert
        OI OPT00TSK+1,X'F0' Make last character printable
        L MSGOFF,MSGCOUNT Get current MSGCOUNT
        MH MSGOFF,MSGLEN Multiply by MSGLEN to get
        LA WORKREG,MSGTABLE offset into message table
        LA TEMPBASE,0(MSGOFF,WORKREG)
        USING TABLEMAP,TEMPBASE
        MVC TABENTRY(133),OPT00MSG Move into msgtable
        DROP TEMPBASE
        L WORKREG,MSGCOUNT Increment MSGCOUNT
        LA WORKREG,1(WORKREG)
        ST WORKREG,MSGCOUNT Save new count
        LH RC,EIRC00 Set normal process retcode

```

Figure 32 (Part 2 of 12). User Interaction Module Example

	STH	RC,EIRETCOD	Store retcode in EIDB
	BR	14	Return to intercept processor

	* AIOPT01: Read SYSIN Record		*
	* Get SYSIN records from the UserArea and give them to DFDSS		*
	* using the Insert Record return code.		*

AIOPT01	TM	SYSINFL,SYSIFRST	Q. First entry to SYSIN exit?
	BO	NXTSYS	A. No, get next input
	OI	SYSINFL,SYSIFRST	Indicate first time taken
	ICM	TEMPBASE,15,EIUSEPTR	Get address of UA
	ST	TEMPBASE,CURCOMM	Save pointer to first record
NXTSYS	L	TEMPBASE,CURCOMM	Get current command pointer
	LA	WORKREG,1(,TEMPBASE)	Get past length
	STCM	WORKREG,15,EIRECPTR	Initialize the pointer
	SR	LENGTH,LENGTH	Clear workreg
	IC	LENGTH,0(,TEMPBASE)	Get length of record
	LTR	LENGTH,LENGTH	Q. End of SYSIN data?
	BZ	SYSDCL	A. Yes, bypass SYSIN DS
	STCM	LENGTH,15,EIRECLEN	Set record length
	LA	TEMPBASE,1(LENGTH,TEMPBASE)	Point to new record
	ST	TEMPBASE,CURCOMM	Save new pointer
	LH	RC,EIRC08	Set insert record retcode
	B	SYSEXIT	Continue setup
SYSDCL	ICM	LENGTH,15,EIRECLEN	Get current record length
	LTR	LENGTH,LENGTH	Q. EOF condition ? (reclen=0)
	BZ	SYSRC00	A. Yes, process it.
	LH	RC,EIRC12	Set delete record retcode
	B	SYSEXIT	Continue Setup
SYSRC00	LH	RC,EIRC00	
SYSEXIT	STH	RC,EIRETCOD	Store retcode in EIDB
	BR	14	Return to intercept processor

	* AIOPT02: Write SYSPRINT		*
	* Insert SYSPRINT records before message ADR012I		*

AIOPT02	ICM	SYSBASE,15,EIRECPTR	
	USING	SYSMPMAP,SYSBASE	
	CLC	MSGID(8),ADR012I	Q. Last SYSPRINT rcd?
	BNE	NOTLAST	A. No, don't insert records
	TM	SYSRFL,INSDONE	Q. Are we done inserting?
	BO	NOTLAST	A. Yes, don't insert records
	L	WORKREG,INSERTCT	
	LR	MSGOFF,WORKREG	Save INSERTCT for offset
	LA	WORKREG,1(WORKREG)	Increment INSERTCT and
	C	WORKREG,MSGCOUNT	Q. Have we printed all rcds?
	BNH	INSERT	A. No, insert another record
	OI	SYSRFL,INSDONE	Indicate we are done inserting
	LA	WORKREG,TRAILER	Point to trailer record
	STCM	WORKREG,15,EIRECPTR	Store into EIRECPTR
	LH	WORKREG,MSGLEN	Update EIRECLEN
	STCM	WORKREG,15,EIRECLEN	
	LH	RC,EIRC08	Set insert record retcode
	B	SYSEXIT	

Figure 32 (Part 3 of 12). User Interaction Module Example

```

INSERT  MH  MSGOFF,MSGLEN      Multiply by MSGLEN to get
        LA  WORKREG,MSGTABLE  offset into message table
        LA  WORKREG,0(MSGOFF,WORKREG)
        STCM WORKREG,15,EIRECPTR  Store address in EIRECPTR
        LH  WORKREG,MSGLEN
        STCM WORKREG,15,EIRECLEN  Store length in EIRECLEN
        L   WORKREG,INSERTCT     Increment MSGCOUNT
        LA  WORKREG,1(WORKREG)
        ST  WORKREG,INSERTCT     Save new count
        LH  RC,EIRC08           Set insert record retcode
        B   SYSPEXIT
NOTLAST LH  RC,EIRC00           Set normal process retcode
SYSPEXIT STH RC,EIRETCOD       Store retcode in EIDB
        BR  14                 Return to intercept processor
*****
* AIOPT03: Read Tape Block Exit *
*   Count Tape Block records read *
*****
AIOPT03 LH  WORKREG,RTBCNT      Increment Read Tape Block
        LA  WORKREG,1(WORKREG)  count
        STH WORKREG,RTBCNT
        LH  RC,EIRC00           Set normal process retcode
        STH RC,EIRETCOD       Store retcode in EIDB
        BR  14                 Return to intercept processor
*****
* AIOPT04: Read Logical Tape Block Exit *
*   Count Logical Tape Block records read *
*****
AIOPT04 LH  WORKREG,RLTBCNT     Increment Read logical Tape
        LA  WORKREG,1(WORKREG)  Block count
        STH WORKREG,RLTBCNT
        LH  RC,EIRC00           Set normal process retcode
        STH RC,EIRETCOD       Store retcode in EIDB
        BR  14                 Return to intercept processor
*****
* AIOPT05: Write Logical Tape Block Exit *
*   Count Logical Tape Block records written *
*****
AIOPT05 LH  WORKREG,WLTBCNT     Increment Write logical Tape
        LA  WORKREG,1(WORKREG)  Block count
        STH WORKREG,WLTBCNT
        LH  RC,EIRC00           Set normal process retcode
        STH RC,EIRETCOD       Store retcode in EIDB
        BR  14                 Return to intercept processor
*****
* AIOPT06: Write Tape Block Exit *
*   Count Tape Block records written *
*****
AIOPT06 LH  WORKREG,WTBCNT      Increment Write Logical
        LA  WORKREG,1(WORKREG)  Tape Block count
        STH WORKREG,WTBCNT
        LH  RC,EIRC00           Set normal process retcode
        STH RC,EIRETCOD       Store retcode in EIDB
        BR  14                 Return to intercept processor
*

```

Figure 32 (Part 4 of 12). User Interaction Module Example

```

*****
* AIOPT07: Read DASD Track Exit *
* Count DASD Tracks read *
*****
AIOPT07 LH WORKREG,RDTCNT Increment Read DASD Track
        LA WORKREG,1(WORKREG) count
        STH WORKREG,RDTCNT
        LH RC,EIRC00 Set normal process retcode
        STH RC,EIRETCOD Store retcode in EIDB
        BR 14 Return to intercept processor
*****
* AIOPT08: Write DASD Track Exit *
* Count DASD Tracks written *
*****
AIOPT08 LH WORKREG,WDTCNT Increment Write DASD Track
        LA WORKREG,1(WORKREG) count
        STH WORKREG,WDTCNT
        LH RC,EIRC00 Set normal process retcode
        STH RC,EIRETCOD Store retcode in EIDB
        BR 14 Return to intercept processor
*****
* AIOPT09: Read Utility SYSPRINT *
*****
AIOPT09 LH RC,EIRC00 Set normal process retcode
        STH RC,EIRETCOD Store retcode in EIDB
        BR 14 Return to intercept processor
*****
* AIOPT10: Reserved *
*****
AIOPT10 LH RC,EIRC00 Set normal process retcode
        STH RC,EIRETCOD Store retcode in EIDB
        BR 14 Return to intercept processor
*****
* AIOPT11: Write WTO Message *
* Delete the WTO and insert my own *
*****
AIOPT11 LH WORKREG,WTOCT Check WTO count
        LTR WORKREG,WORKREG Q. Is WTOCT 0?
        BZ OPT11INS A. Yes, insert one
OPT11DEL LH RC,EIRC12 Set delete record retcode
        B OPT11XIT (delete all other WTOs)
OPT11INS LA WORKREG,1(WORKREG) Increment WTOCT
        STH WORKREG,WTOCT
        LA WORKREG,UIMWTO Get address of WTO
        STCM WORKREG,15,EIRECPTR Store it in EIRECPTR
        LA WORKREG,WTOLEN Get length of WTO
        STCM WORKREG,15,EIRECLEN Store length it in EIRECLEN
        LH RC,EIRC08 Set insert record retcode
        B OPT11XIT
OPT11XIT STH RC,EIRETCOD Store retcode in EIDB
        BR 14 Return to intercept processor

```

Figure 32 (Part 5 of 12). User Interaction Module Example

```

*****
* AIOPT12: Write WTOR Message *
* Check for ADR369D, never allow it to write over VTOC, etc *
*****
AIOPT12 ICM WORKREG,15,EIRECPTR
        CLC 12(8,WORKREG),ADR369D Q. Authorize request?
        BNE OPT12RC0 A. No, let DFDSS issue WTOR
        LA WORKREG,UIMRESP Get address of response
        STCM WORKREG,15,EIRECPTR Store in EIRECPTR
        LA WORKREG,RESPLEN Get length of response
        STCM WORKREG,15,EIRECLEN Store in EIRECLEN
        LH RC,EIRC28 Set WTOR Response retcode
        B OPT12XIT
OPT12RC0 LH RC,EIRC00 Set normal process retcode
OPT12XIT STH RC,EIRETCOD Store retcode in EIDB
        BR 14 Return to intercept processor
*****
* AIOPT13: Present ADRUFO Record *
* If parm call, set IO and AI buffers above 16M *
* If function call and Copy, force reblocking. *
*****
AIOPT13 ICM TEMPBASE,15,EIRECPTR
        USING ADRUFOB,TEMPBASE Establish addressability to UFO
        CLC UFFUNCT,PARM Q. Is this a PARM call?
        BE PARMCALL A. Yes, go to parm call
FUNCTION TM UFFUNCT1,UFFUCOPY Is this a COPY?
        BNO OPT13RC0 No, don't change options
        SR WORKREG,WORKREG
        ICM WORKREG,3,UFBDOFF Get offset for PARM list
        AR WORKREG,TEMPBASE Add to address of UFO
        USING UFOFUNCT,WORKREG Establish addressability
        OI UFOINSOP,UFOFRBLK Force reblocking
        LH RC,EIRC16 Set modify record retcode
        B OPT13XIT
PARMCALL SR WORKREG,WORKREG This is a parm call.
        ICM WORKREG,3,UFBDOFF Get offset for PARM list
        AR WORKREG,TEMPBASE Add to address of UFO
        USING UFOFUNCT,WORKREG Establish addressability
        OI UFXAFLAG,UFXABUFF Make sure IO buffers are >16M
        OI UFXAFLAG,UFAI31B Make sure AI buffers are >16M
        LH RC,EIRC16 Set modify record retcode
        LA WORKREG,2 Initialize message table:
        ST WORKREG,MSGCOUNT Initialize msgcount
        LA TEMPBASE,MSGTABLE Get address of table
        USING TABLEMAP,TEMPBASE
        MVC TABENTRY(133),HEADER Move header into msgtable
        LA TEMPBASE,133(TEMPBASE) Move past first message
        MVC TABENTRY(133),BLANKS Move blank line into table
        B OPT13XIT Exit.
OPT13RC0 LH RC,EIRC00 Set normal process retcode
OPT13XIT DROP TEMPBASE
        STH RC,EIRETCOD Store retcode in EIDB
        BR 14 Return to intercept processor

```

Figure 32 (Part 6 of 12). User Interaction Module Example

```

*****
* AIOPT14: Function Ending *
*****
AIOPT14  LH   WORKREG,RTBCNT      Get Tape Blocks Read
          CVD  WORKREG,CVDWORK   Convert to decimal
          UNPK RTBINS,CVDWORK    Unpack into message insert
          OI   RTBINS+7,X'F0'    Make last character printable
          LH   WORKREG,WTBCNT    Get Tape Blocks Written
          CVD  WORKREG,CVDWORK   Convert to decimal
          UNPK WTBINS,CVDWORK    Unpack into message insert
          OI   WTBINS+7,X'F0'    Make last character printable
          LH   WORKREG,RLTBCNT   Get Tape Logical Blocks Read
          CVD  WORKREG,CVDWORK   Convert to decimal
          UNPK RLTBINS,CVDWORK   Unpack into message insert
          OI   RLTBINS+7,X'F0'  Make last character printable
          LH   WORKREG,WLTBCNT   Get Tape Logical Blocks Written
          CVD  WORKREG,CVDWORK   Convert to decimal
          UNPK WLTBINS,CVDWORK   Unpack into message insert
          OI   WLTBINS+7,X'F0'  Make last character printable
          LH   WORKREG,RDTCNT    Get Disk Tracks Read
          CVD  WORKREG,CVDWORK   Convert to decimal
          UNPK RDTINS,CVDWORK    Unpack into message insert
          OI   RDTINS+7,X'F0'   Make last character printable
          LH   WORKREG,WDTCNT    Get Disk Tracks Written
          CVD  WORKREG,CVDWORK   Convert to decimal
          UNPK WDTINS,CVDWORK    Unpack into message insert
          OI   WDTINS+7,X'F0'   Make last character printable
          L    MSGOFF,MSGCOUNT  Get current MSGCOUNT
          MH   MSGOFF,MSGLEN     Multiply by MSGLEN to get
          LA   WORKREG,MSGTABLE  offset into message table
          LA   TEMPBASE,0(MSGOFF,WORKREG)
          USING TABLEMAP,TEMPBASE
          MVC  TABENTRY(133),READMSG  Move into msgtable
          LA   TEMPBASE,133(TEMPBASE) Increment pointer for 2nd msg
          MVC  TABENTRY(133),WRITEMSG Move into msgtable
          DROP TEMPBASE
          L    WORKREG,MSGCOUNT    Increment MSGCOUNT
          LA   WORKREG,2(WORKREG)
          ST   WORKREG,MSGCOUNT    Save new count
          LH   RC,EIRCOO           Set normal process retcode
          STH  RC,EIRETCOD        Store retcode in EIDB
          BR   14                  Return to intercept processor
*****
* AIOPT15: Present WTOR Response *
*****
AIOPT15  LH   RC,EIRCOO           Set normal process retcode
          STH  RC,EIRETCOD        Store retcode in EIDB
          BR   14                  Return to intercept processor

```

Figure 32 (Part 7 of 12). User Interaction Module Example


```

*****
* AIOPT16: OPEN/EOF Tape Security and Verification *
* Always bypass password protection *
*****
AIOPT16 ICM TEMPBASE,15,EIRECPTR Get address of record
        LH RC,EIRC16 Bypass password protection
        ST RC,0(TEMPBASE) Store in EIRECPTR
        LH RC,EIRC16 Set modify record retcode
        STH RC,EIRETCOD Store retcode in EIDB
        BR 14 Return to intercept processor
*****
* AIOPT17: OPEN/EOF Non-specific Tape Mount *
* Supply DFSS with TAPE01 *
*****
AIOPT17 ICM TEMPBASE,15,EIRECPTR Get address of record
        MVC 0(6,TEMPBASE),TAPE01 Supply TAPE01 as specific volser
        LH RC,EIRC16 Set modify record retcode
        STH RC,EIRETCOD Store retcode in EIDB
        BR 14 Return to intercept processor
*****
* AIOPT18: Insert logical VSAM Record during Logical Restore *
*****
AIOPT18 LH RC,EIRC00 Set normal process retcode
        STH RC,EIRETCOD Store retcode in EIDB
        BR 14 Return to intercept processor
*****
* AIOPT19: Output Tape I/O Error *
* Save DDNAME, Volser, and Return code, print record. *
*****
AIOPT19 ICM TEMPBASE,15,EIRECPTR Get address of EIRECORD
        USING EIDINFO,TEMPBASE Est addressability to DDINFO
        MVC OPT19DD,EIDDDNAME Move DDNAME to message
        MVC OPT19VS,EIVOLID Move VOLSER to message
        SR WORKREG,WORKREG Clear out work register
        IC WORKREG,EIRETC Get return code
        CVD WORKREG,CVDWORK Convert to decimal
        UNPK OPT19RC(2),CVDWORK+6(2) Unpack into message insert
        OI OPT19RC+1,X'F0' Make last character printable
        DROP TEMPBASE
        L MSGOFF,MSGCOUNT Get current MSGCOUNT
        MH MSGOFF,MSGLEN Multiply by MSGLEN to get
        LA WORKREG,MSGTABLE offset into message table
        LA TEMPBASE,0(MSGOFF,WORKREG)
        USING TABLEMAP,TEMPBASE
        MVC TABENTRY(133),OPT19MSG Move into msgtable
        DROP TEMPBASE
        L WORKREG,MSGCOUNT Increment MSGCOUNT
        LA WORKREG,1(WORKREG)
        ST WORKREG,MSGCOUNT Save new count
        LH RC,EIRC00 Set normal process retcode
        STH RC,EIRETCOD Store retcode in EIDB
        BR 14 Return to intercept processor

```

Figure 32 (Part 8 of 12). User Interaction Module Example

```

*****
* AIOPT20: Volume Information Exit *
* Save Data Set name *
*****
AIOPT20 ICM TEMPBASE,15,EIRECPTR Establish addressability to
        USING EIREC20,TEMPBASE EIREC20
        MVC OPT20DSN(44),EI20DSN Move data set name into message
        DROP TEMPBASE
        L MSGOFF,MSGCOUNT Get current MSGCOUNT
        MH MSGOFF,MSGLEN Multiply by MSGLEN to get
        LA WORKREG,MSGTABLE offset into message table
        LA TEMPBASE,0(MSGOFF,WORKREG)
        USING TABLEMAP,TEMPBASE
        MVC TABENTRY(133),OPT20MSG Move into msgtable
        DROP TEMPBASE
        L WORKREG,MSGCOUNT Increment MSGCOUNT
        LA WORKREG,1(WORKREG)
        ST WORKREG,MSGCOUNT Save new count
        LH RC,EIRC00 Set normal process retcode
        STH RC,EIRETCOD Store retcode in EIDB
        BR 14 Return to intercept processor
*****
* AIOPT21: Data Set Verification Exit
* Do Not Process TEMP Data Sets
*****
AIOPT21 CLC EI21DSN(4),TMPL If first 4 chars of dsname='TEMP'
        BNE LABEL1 Then
        LH RC,EIRC36 Do not process data set
        B LABEL2 Else
        LABEL1 LH RC,EIRC00 Set normal return code
        LABEL2 STH RC,EIRETCOD Store retcode in EIDB
        BR 14 Return to intercept processor
*****
* AIOPT22: Bypass Verification Exit
* Bypass Serialization of SYS1 Data Sets
*****
AIOPT22 CLC EI21DSN(4),SYSL If first 4 chars of dsname='SYS1'
        BNE LAB1 Then
        LH RC,EIRC16 Bypass serialization of data set
        NI EI22FLGS,B'01111111' Set EI22BSER bit on
        B LAB2 Else
        LAB1 LH RC,EIRC00 Set normal return code
        LAB2 STH RC,EIRETCOD Store retcode in EIDB
        BR 14 Return to intercept processor

```

Figure 32 (Part 9 of 12). User Interaction Module Example

```

*****
* AIOPT23: Data Set Processed Notification Exit
* End Function If Errors in Processing
*****
AIOPT23 LH WORKREG,EI23DSRC If data set processing RC not zero
        LTR WORKREG,WORKREG Then
        BZ LABL1
        LH RC,EIRC36 End function
        B LABL2 Else
LABL1 LH RC,EIRC00 Set normal return code
LABL2 STH RC,EIRETCOD Store retcode in EIDB
        BR 14 Return to intercept processor
*
CURCOMM DC F'0' Address of current command
SYSINFL DC X'00' SYSIN Flag
SYSIFRST EQU X'80' First entry SYSIN Performed
SYSPRFL DC X'00' SYSPRINT Flag
INSDONE EQU X'80' Done inserting SYSPRINT recs
TMPL DC C'TEMP' TEMP high-level qualifier
SYSL DC C'SYS1' SYS1 high-level qualifier
*
SAVEAREA DC 18F'0' My save area
*
VECTABLE DC A(AIOPT00) Function Startup Exit
VEC01 DC A(AIOPT01) Read SYSIN Exit
VEC02 DC A(AIOPT02) Write SYSPRINT Exit
VEC03 DC A(AIOPT03) Read Tape Block Exit
VEC04 DC A(AIOPT04) Read Logical Tape Exit
VEC05 DC A(AIOPT05) Write Logical Tape Exit
VEC06 DC A(AIOPT06) Write Tape Block Exit
VEC07 DC A(AIOPT07) Read DASD Track Exit
VEC08 DC A(AIOPT08) Write DASD Track Exit
VEC09 DC A(AIOPT09) Read Utility Sysprint Exit
VEC10 DC A(AIOPT10) Reserved
VEC11 DC A(AIOPT11) Write WTO Message Exit
VEC12 DC A(AIOPT12) Write WTOR Message Exit
VEC13 DC A(AIOPT13) Present ADRUFO Record Exit
VEC14 DC A(AIOPT14) Function Ending Exit
VEC15 DC A(AIOPT15) Present WTOR Response
VEC16 DC A(AIOPT16) OPEN/EOF Tape Sec/Ver Exit
VEC17 DC A(AIOPT17) OPEN/EOF Non-Spec Tape Mount
VEC18 DC A(AIOPT18) Insert logical VSAM Rcd -Restore
VEC19 DC A(AIOPT19) Output Tape I/O Error Exit
VEC20 DC A(AIOPT20) Volume Information Exit
VEC21 DC A(AIOPT21) Data Set Verification Exit
VEC22 DC A(AIOPT22) Bypass Verification Exit
VEC23 DC A(AIOPT23) Data Set Processed Notification Exit
*

```

Figure 32 (Part 10 of 12). User Interaction Module Example

```

UIMWTO DC C'>>> USRUIM is active <<<'
WTOLEN EQU *-UIMWTO Show that we are active
*
UIMRESP DC C'T' Reply T to ADR369
RESPLEN EQU *-UIMRESP
*
ADR369D DC CL8'ADR369D' DFDSS Authorization WTOR
ADR012I DC CL8'ADR012I' Last DFDSS message id
*
MSGLEN DC H'133' Length of messages
INSERTCT DC F'0' Count of messages given to DFDSS
MSGCOUNT DC F'0' Count of messages in table
*
RTBCNT DC H'0' Count of Tape Blocks Read
WTBCNT DC H'0' Count of Tape Blocks Written
*
RLTBCNT DC H'0' Count Logical Tape Blocks Read
WLTBCNT DC H'0' Count Logical Tape Blocks Writn
*
RDTCNT DC H'0' Count Disk Tracks Read
WDTCNT DC H'0' Count Disk Tracks Written
*
WTOCT DC H'0' Count of WTOs
*
PARAM DC X'0000' Parm call test for Opt 13
*
TAPE01 DC CL6'TAPE01' Tape volser for Opt 17
*
*
HEADER DC C' ***** Begin USRUIM Messages '
DC CL(133-(*-HEADER))'*****C
*****'
*
TRAILER DC C' ***** End USRUIM Messages '
DC CL(133-(*-TRAILER))'*****C
*****'
*
BLANKS DC CL133' ' Blank message
*
READMSG DC C' Disk Tracks Read: '
RDTINS DC C'XXXXXXX'
DC C' Tape Blocks Read: '
RTBINS DC C'XXXXXXX'
DC C' Logical Tape Blocks Read: '
RLTBINS DC C'XXXXXXX'
DC CL(133-(*-READMSG))' '
*

```

Figure 32 (Part 11 of 12). User Interaction Module Example

```

WRITEMSG DC C' Disk Tracks Written: '
WDTINS DC C'XXXXXXXX'
DC C' Tape Blocks Written: '
WTBINS DC C'XXXXXXXX'
DC C' Logical Tape Blocks Written: '
WLTBINS DC C'XXXXXXXX'
DC CL(133-(*-WRITEMSG))' '
*
OPT00MSG DC C' Messages follow for task: '
OPT00TSK DC C'XX'
DC CL(133-(*-OPT00MSG))' '
*
OPT19MSG DC C' Permanent Tape Error for DDNAME: '
OPT19DD DC C'XXXXXXXX'
DC C' VOLSER: '
OPT19VS DC C'XXXXXX'
DC C' Return Code: '
OPT19RC DC C'XX'
DC CL(133-(*-OPT19MSG))' '
*
OPT20MSG DC C' EI20DSN = '''
OPT20DSN DC C'XXXXXXXXX.XXXXXXXXXX.XXXXXXXXXX.XXXXXXXXXX.XXXXXXXXXX'
DC C''''
DC CL(133-(*-OPT20MSG))' '
*
CVDWORK DS D Workarea for decimal conversions
*
ADREID0 Macro to map EIDB
*
ADRUF0 Macro to map UFO
*
USRUIM CSECT
MSGTABLE DS CL13300 Room in table for 100 msgs
*
TABLEMAP DSECT DSECT to insert msgs into table
TABENTRY DS CL133
*
SYSPMAP DSECT DSECT to find MSGID in DFSS
DS CL1 messages
MSGID DS CL7
*
END

```

Figure 32 (Part 12 of 12). User Interaction Module Example

```

IPAGE 0001 DFDSS 5665-327 DATA FACILITY DATA SET SERVICES V2R5.0 88.008 16:28
-ADR010I PRIME(01), SIZE VALUE OF 4096K WILL BE USED FOR GETMAIN
WTO 'USRAIPGM INVOKING DFDSS'
ADR101I RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'WTO '
DUMP DS(INC(**)) LOGINDD(DASD) OUTDD(TAPE)
ADR101I RI01 (01), TASKID 002 HAS BEEN ASSIGNED TO COMMAND 'DUMP '
IF LASTCC=0 -
ADR101I RI01 (01), TASKID 003 HAS BEEN ASSIGNED TO COMMAND 'IF '
THEN DEFrag DDN(DASD)
ADR101I RI01 (01), TASKID 004 HAS BEEN ASSIGNED TO COMMAND 'DEFrag '
ADR109I RI01 (01), 88008 16:28:31 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
0ADR006I (002)-SETUP(01), 88008 16:28:31 PROCESSING BEGINS
0ADR317I (002)-OPNCL(01), THE DEVICE SIZE FIELD IN THE VTOC IS INCORRECT IF VOLUME SYSTEM IS NOT A MINIVOLUME. DEVICE SIZE OF 0100
CYLINDERS WILL BE USED
0ADR454I (002)-DTDSC(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
0
0 SYS1.LINKLIB
0 SYS1.CMDLIB
0 SYS1.TELCLIB
0 SYS1.VTAMLIB
0 SYS1.HELP
0 SYS2.LINKLIB
0 SYS1.DGTCLIB
0 SYS1.DGTSLIB
0 SYS1.DGTPLIB
0 SYS1.DGTMLIB
0 SYS1.DGTTLIB
0 SYS1.DGTLIB
0 USERA.LINKLIB
0 USERC.LINKLIB
0ADR013I (002)-CLTSK(01), 88008 16:30:04 TASK COMPLETED WITH RETURN CODE 0000
0ADR006I (004)-SETUP(01), 88008 16:30:06 PROCESSING BEGINS
0ADR317I (004)-OPNCL(01), THE DEVICE SIZE FIELD IN THE VTOC IS INCORRECT IF VOLUME SYSTEM IS NOT A MINIVOLUME. DEVICE SIZE OF 0100
CYLINDERS WILL BE USED
0ADR317I (004)-OPNCL(01), THE DEVICE SIZE FIELD IN THE VTOC IS INCORRECT IF VOLUME SYSTEM IS NOT A MINIVOLUME. DEVICE SIZE OF 0100
CYLINDERS WILL BE USED
0ADR208I (004)-DANAL(01), BEGINNING STATISTICS ON SYSTEM:
IPAGE 0002 DFDSS 5665-327 DATA FACILITY DATA SET SERVICES V2R5.0 88.008 16:28
-
0 FREE CYLINDERS 0009
0 FREE TRACKS 0013
0 FREE EXTENTS 0003
0 LARGEST FREE EXTENT (CYL,TRK) 0005,0000
0 FRAGMENTATION INDEX .258
0 PERCENT FREE SPACE 09
0ADR220I (004)-DANAL(01), INTERVAL BEGINS AT CC:HH 0091:0000 A
ND ENDS AT CC:HH 0098:0000
0ADR209I (004)-DEFrag(01), MOVED EXTENT 001 FROM 0096:0000-0097:0014 TO 0084:0000-0085:0014 FOR USERC.LINKLIB
0ADR213I (004)-DANAL(01), ENDING STATISTICS ON SYSTEM:
0 DATA SET EXTENTS RELOCATED 0001
0 TRACKS RELOCATED 0030
0 FREE CYLINDERS 0009
0 FREE TRACKS 0013
0 FREE EXTENTS 0002
0 LARGEST FREE EXTENT (CYL,TRK) 0009,0000
0 FRAGMENTATION INDEX .074
IPAGE 0003 DFDSS 5665-327 DATA FACILITY DATA SET SERVICES V2R5.0 88.008 16:28
-ADR212I (004)-DANAL(01), EXTENT DISTRIBUTION MAP FOR SYSTEM:

```

Figure 33 (Part 1 of 2). Output Resulting from Use of the UIM Exits

0	EXTENT	**INPUT FREE**		**OUTPUT FREE**		** OCCUPIED **	
	IN SIZE TRACKS	NO. EXTS	CUM. PCT/100	NO. EXTS	CUM. PCT/100	NO. EXTS	CUM. PCT/100
	1		0.000		0.000	3	0.002
	2		0.000		0.000	7	0.012
	5		0.000		0.000	2	0.019
	10		0.000		0.000	2	0.034
	12		0.000		0.000	1	0.043
	13		0.000	1	0.087		0.043
	14		0.000		0.087	1	0.053
	20		0.000		0.087	2	0.083
	30	1	0.202		0.087	2	0.127
	36		0.202		0.087	1	0.154
	43	1	0.493		0.087	1	0.186
	75	1	1.000		0.087	1	0.241
	100		1.000		0.087	2	0.389
	135		1.000	1	1.000		0.389
	>499		1.000		1.000	1	1.000

ADRO13I (004)-CLTSK(01), 88008 16:30:20 TASK COMPLETED WITH RETURN CODE 0000
 ***** Begin USRUIM Messages *****

Messages follow for task: 02
 Disk Tracks Read: 00001159 Tape Blocks Read: 00000000 Logical Tape Blocks Read: 00000000
 Disk Tracks Written: 00000000 Tape Blocks Written: 00001990 Logical Tape Blocks Written: 00001158

Messages follow for task: 04
 Disk Tracks Read: 00000030 Tape Blocks Read: 00000000 Logical Tape Blocks Read: 00000000
 Disk Tracks Written: 00000060 Tape Blocks Written: 00000000 Logical Tape Blocks Written: 00000000

***** End USRUIM Messages *****

0ADR012I DSSU (01), 88008 16:30:20 DFSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
 1

Figure 33 (Part 2 of 2). Output Resulting from Use of the UIM Exits

Appendix E. Data Area Descriptions

This appendix contains two data area descriptions corresponding to control blocks ADRBMB and ADRTAPB.

For a description of ADRUFO, see Figure 14 on page 155.

For a description of ADREID0, see Figure 27 on page 197.

Product implementation information is contained in this chapter.

ADRBMB Data Area

Product Implementation Information

Figure 34. ADRBMB Mapping Macro

Offsets	Type	Length	Name	Description
0	(0) CHARACTER	*	ADRBMB	
0	(0) CHARACTER	12	BMHDR	BITMAP HEADER
0	(0) CHARACTER	4	BMID	BLK IDENTIFIER EBCDIC "BMBB"
4	(4) ADDRESS	4	BMNPTR	ADDR OF NEXT BITMAP SEGMENT/0
8	(8) FIXED	2	BMNTRK	NUMBER OF TRACKS MAPPED BY SEGMENT
10	(A) CHARACTER	2	*	RESERVED
12	(C) BITSTRING	0	BMBITMAP	BITMAP LAYOUT, CONTIGUOUS BITS FOR EACH TRACK. 1 = OPERATE ON CORRESPONDING TRACK. 0 = SKIP TRACK

Name	Hex Offset	Struct Level
ADRBMB	0	(0)
BMBITMAP	12	(C)
BMHDR	0	(0)
BMID	0	(0)
BMNPTR	4	(4)
BMNTRK	8	(8)

ADRTAPB Data Area

Figure 35 (Page 1 of 3). ADRTAPB Mapping Macro

Offsets	Type	Length	Name	Description
0	(0) CHARACTER	16	ADRTAPB	
0	(0) CHARACTER	6	DTPSCHK	
0	(0) FIXED	4	DTPSEQNO	SEGMENT SEQUENCE NUMBER
4	(4) FIXED	1	DTPNOSEG	NUM OF SEGMENTS PER RECORD
5	(5) FIXED	1	DTPSEGNO	SEGMENT NUM OF RECORD
6	(6) FIXED	2	DTPSEGLN	SEGMENT LENGTH INCL PREFIX
8	(8) FIXED	1	DTPPFXLN	LENGTH OF PREFIX(CONSTANT 16)
9	(9) CHARACTER	1	DTPDMPID	TYPE OF DUMP ID
	.1.		DTPFULD	80 FULL DUMP
	.1.		DTPPARTD	40 PARTIAL DUMP
	.1.		DTPDASDD	20 DATASET DUMP
	.1.		DTPLOGCL	10 LOGICAL DUMP THIS BIT IS CALLED DTPCATDD IN V2 R1. BIT IN STRUCTURE TO MAINTAIN COMPATIBILITY BETWEEN V2 R1, V2 R2 AND V2 R3
 1111		*	RESERVED DO NOT USE LAST 4 BITS IN DTPDMPID TO ENSURE DOWNWARD COMPATIBILITY
10	(A) BITSTRING	1	DTPRCID1	RECORD IDENTIFIER 1
	.1.		DTPVHDR	80 VOLUME HEADER
	.1.		DTPHADR	TAPE HEADER
	.1.		DTPBTM	40 MAP OF DUMPED TRKS
	.1.		DTPDSNL	DS NAME/CATALOG LST
	.1.		DTPTRK0	20 TRACK 0
	.1.		DTPDSHDR	DATA SET HEADER
	.1.		DTPVTOC	10 VTOC TRACK
	.1.		DTPVOLD	VOLUME DEFINITION
 1..		DTPDATA	08 DATA TRACK
 1..		DTPVTRLR	04 VOLUME TRAILER
 1..		DTPDTRLR	DS TRAILER
 1..		DTPVVDS	02 VVDS TRACK
 1..		DTPSPHDR	01 SPHERE REC HEADER
11	(B) BITSTRING	1	DTPRCFLI	FLAG BYTE
	.1.		DTPDDISP	IF ON, DATA EQUAL TO LENGTH OF ADRTAPB HAS BEEN DISPLACED FROM THIS SEGMENT TO THE LAST SEGMENT OF THE TRACK
		DTPDDDSP	IF ON, LAST SEGMENT DISPLACED DATA MUST BE REPLACED FIRST (SEE DTPDDISP)
	.1.		DTPENDCD	1 = END OF CDF DATA SET
	.1. 1111		*	RESERVED
12	(C) CHARACTER	4	*	RESERVED
16	(0) CHARACTER		DTPBODY	START OF REMAINDER OF RECORD
DTVOL				
0	(0) CHARACTER	46	DTVOL	
0	(0) CHARACTER	4	DTVTOCB	VTOC BEGINNING CCHH
4	(4) CHARACTER	4	DTVTOCE	VTOC ENDING CCHH
8	(8) CHARACTER	4	DTVOLSZ	VOLUME SIZE(DS+DEVSZ)
8	(8) FIXED	2	DTVLOGCY	# OF CYLINDERS IN VOLUME
10	(A) FIXED	2	DTVTRKCY	# OF TRACKS PER CYLINDER
12	(C) FIXED	2	DTVBLKSZ	MAXIMUM BLOCKSIZE
14	(E) FIXED	2	DTVMAXCB	MAXIMUM COMPRESS BUFFER IN WORDS
16	(10) CHARACTER	6	DTVSERL	VOLUME SERIAL OF SOURCE VOL
22	(16) CHARACTER	1	DTVVTOCI	VTOC INDICATORS
23	(17) CHARACTER	8	DTVTIMD	DATE & TIME OF DAY OF DUMP
23	(17) CHARACTER	4	DTVDAY	DATE 00YYDDDC
27	(1B) CHARACTER	4	DTVTIME	TIME OF DAY IN DECIMAL
31	(1F) CHARACTER	4	DTVDEVTY	DEVNTP OF VOLUME(UCB)BT#)
35	(23) FIXED	1	DTVMODNO	MODEL NUMBER
36	(24) BITSTRING	1	DTVINDI	VOLUME TYPE INDICATORS
	.1.		DTVVIRT	80 VIRTUAL VOLUME
	.1.		DTVMINI	40 MINI DISK
	.1.		DTVCVAF	20 VOLUME HAS INDEXED VTOC
	.1.		*	10 NOT USED
	.1.		DTVFCMP	08 FILE COMPRESSED
	.1.		DTVUNLCD	04 UNALLOCATED SPACE DUMPED
	.1.		DTVLVF	02 OTHER LOGICALVOLUMES MAY FOLLOW THIS LOGICALVOLUME
 1..		DTVVVDS	01 VVDS DATASET DUMPED BEFORE VTOC
37	(25) FIXED	2	DTVBMSZ	BITMAP SIZE IN WORDS
39	(27) BITSTRING	1	DTVIND2	VOLUME TYPE INDICATORS
	.1.		DTVLNVI	80 NONVSAM DATA SETS NOT ON VOLUME
	.1.		DTVLVI	40 VSAM DATA SETS NOT ON VOLUME
	.1.		DTVGVI	20 NONVSAM DATA SETS NOT ON ANY VOLUME

Figure 35 (Page 2 of 3). ADRTAPB Mapping Macro

Offsets	Type	Length	Name	Description
	...1		DTVGV1	10 VSAM DATA SETS NOT ON ANY VOLUME
 1..		DTVSMS	SOURCE VOLUME IS SMS MANAGED
 1..		DTVSMSI	SOURCE VOL IS SMS INITIAL STAGE UNUSED
DTVLEN				
DTVLEN				
40	(28) FIXED	2	DTVLEN	LEN OF HEADER = SEGMENT LENGTH PREFIX
42	(2A) FIXED	1	DTVVERNO	DFDSS VERSION NUMBER
43	(2B) FIXED	1	DTVLVLNO	DFDSS MODIFIC. NUMBER
44	(2C) FIXED	1	DTVVXTNO	NUMBER OF VVDS EXTENTS
45	(2D) FIXED	1	DTVVXTOF	OFFSET TO VVDS EXTENT
DTHDR				
0	(0) CHARACTER	18	DTHDR	
0	(0) CHARACTER	8	DTHTIMD	DATE & TIME DAY OF DUMP
0	(0) CHARACTER	4	DTHDAY	DAY
4	(4) CHARACTER	4	DTHTIME	TIME
8	(8) CHARACTER	1	DTHIND2	DATA SET TYPE INDICATORS
	1..		DTHGNVI	NO NON VSAM DATA SETS
	..1.		DTHGVI	NO VSAM DATA SETS
	..11 1111		*	RESERVED
9	(9) FIXED	2	DTHLEN	HEADER LENGTH
11	(B) FIXED	1	DTHVERNO	DFDSS VERSION NUMBER
12	(C) FIXED	1	DTHLVLNO	DFDSS MODIFIC. NUMBER
13	(D) FIXED	2	DTHBLKSZ	MAX BLKSIZE
15	(F) FIXED	2	DTHNDS	NO. POTENTIAL DS
17	(11) CHARACTER	1	DTHIND1	INDICATORS
	1..		DTHFCMP	FILE COMPRESSED
	..1.		DTHUNLCD	UNALLOCATED SPACE DUMPED
	..1.		DTHSFER	SPHERE OPTION
	...1 1111		*	RESERVED
DTLDSN				
0	(0) CHARACTER	45	DTLDSN	
0	(0) FIXED	1	DTLLEN	LENGTH OF DATA SET NAME
1	(1) CHARACTER	44	DTLCAT	CATALOG NAME
DTDSHDR				
0	(0) CHARACTER	106	DTDSHDR	
0	(0) FIXED	1	DTDLEN	LENGTH OF DATA SET NAME
1	(1) FIXED	1	DTDCATLN	LENGTH OF CATALOG NAME
2	(2) CHARACTER	2	DTDDSORG	DATA SET ORGA (FROM F1)
4	(4) CHARACTER	1	DTDOPTCD	DS OPTION CODE (FROM F1)
5	(5) CHARACTER	1	DTDNVOL	NUMBER VOLS FOR DATA SET
6	(6) BITSTRING	1	DTDIND	DATA SET INDICATOR
	1..		DTDIPWD	PASSWORD SUPPLIED 1 = YES
	..1.		DTDTPWD	TYPE PASSWORD 0 = D.S.
	..1.		DTDIRACF	CATALOG RACE PROFILE
	...1 1..		DTDRACFP	RACE PROFILE FLAGS
	...1 1..		DTDRACFD	1 = RACE DISCRETE PROFILE
	...1 1..		DTDRACFG	1 = RACE GENERIC PROFILE
	...1 1..		DTDALIAS	1 = USER CATALOG ALIAS
	...1 1..		DTDSPER	1 = SPHERE RECORD FOLLOWS
	...1 1..		DTDSMS	1 = SMS MANAGED DATA SET
7	(7) CHARACTER	8	DTDIPWD	PASSWORD
15	(F) CHARACTER	44	DTDCAT	CATALOG NAME
59	(3B) CHARACTER	44	DTDSSN	DATA SET NAME
103	(67) CHARACTER	2	DTDVOLCT	VOLCOUNT (FROM BCS)
103	(67) FIXED	1	DTDVCTD	VOLCOUNT FOR DATA COMPONENT OR NONVSAM DATA SET
104	(68) FIXED	1	DTDVCTI	VOLCOUNT FOR INDEX COMPONENT OR ZERO FOR NO INDEX
105	(69) CHARACTER	1	DTDIND2	DATA SET INDICATOR 2
	1..		DTDAIXSP	AIXR PART OF A SPHERE
	..1.		DTDCDF	1 = COMMON DATA FORMAT DSET
	..1.		DTDPDSE	1 = PDSE DATA SET
	...1 1111		*	UNUSED
DTSPPHERE				
0	(0) CHARACTER	4	DTSPPHERE	
0	(0) FIXED	4	DTSLEN	LENGTH OF SPHERE RECORD
DTSINFO				
0	(0) CHARACTER	102	DTSINFO	SPHERE INFORMATION
0	(0) CHARACTER	44	DTSAXNM	AIX NAME
44	(2C) CHARACTER	44	DTSPATHN	PATH NAME
88	(58) CHARACTER	1	DTSPATHA	PATH ATTRIBUTE
89	(59) BITSTRING	1	DTSPATHF	PATH INFO FLAG
	1..		DTSPATHI	IF SET, PATH OWNERID IS CONTAINED IN THIS BLOCK (SEE DTSPHON FOR DESCRIPTION)
	..1.		DTSPATHE	PATH EXPIRATION DATE IS CONTAINED IN THIS BLOCK (SEE DTSPHER FOR DESCRIPTION). IF NOT SET, IGNORE THE EXPIRE FIELDS.

Figure 35 (Page 3 of 3). ADRTAPB Mapping Macro

Offsets	Type	Length	Name	Description
	.1.		DTSPATHP	PATH HAS PASSWORD THAT IS CONTAINED IN THIS BLOCK (SEE DTSEXPIR FOR DESCRIPTION). IF EXISTS, BEGINS AT DTSPATHD.
	... 1...		DTSPRACF	PATH RACF FLAGS
	... 1....		DTSRACFD	1 = DISCRETE PROFILE
	... 1...		DTSRACFG	1 = GENERIC PROFILE
111		*	NOT USED
90	(5A) CHARACTER	8	DTSPTHON	FORMAT OF OWNERID
98	(62) CHARACTER	4	DTSPTHEP	EXPIRATION DATE FORMAT:
98	(62) CHARACTER	3	DTSEXPIR	YYDDDF FORMAT
98	(62) FIXED	1	DTSEYEAR	YY 8 BIT IN DECIM
99	(63) FIXED	2	DTSERDAY	12 BITS DAY OF IN DECIMAL & 4 BIT SIGN UNPACKED
101	(65) FIXED	1	DTSEXCNY	CENTURY BYTE IN DECIMAL
DTSPASSW				
0	(0) CHARACTER	52	DTSPASSW	SECURITY INFO TABLE
0	(0) CHARACTER	8	DTSMSTRP	MASTER PASSWORD. MUST HAVE IN ORDER TO HAVE ANY OTHER PASSWORDS.
8	(8) CHARACTER	8	DTSCNTLP	CONTROL INTERVAL PASSWORD
16	(10) CHARACTER	8	DTSUPDAT	UPDATE PASSWORD
24	(13) CHARACTER	8	DTSREADP	READ PASSWORD
32	(20) CHARACTER	8	DTSCODEN	CODE NAME
40	(28) FIXED	2	DTSNUMAT	NUMBER OF ATTEMPTS
42	(2A) CHARACTER	8	D TSAUTNM	ADDR OF AUTHORIZATION MOD
50	(32) FIXED	2	D TSAUTHR	AUTHORIZATION REC LENGTH
DTMVOL				
0	(0) CHARACTER	28	DTMVOL	VOLUME SERIAL ID
0	(0) CHARACTER	6	DTMVSERL	DEVTYPE (UCBTBYT4)
6	(6) CHARACTER	4	DTMDEVTY	SLACK FILLER
10	(A) CHARACTER	2	*	VOLUME SIZE (DS4DEVSZ)
12	(C) CHARACTER	8	DTMVOLSZ	#BYTES/TRK (TRACK CAPACITY WITH OVERHEAD).
12	(C) FIXED	4	DTMTRKCP	# CYLINDERS PER VOLUME
16	(10) FIXED	2	DTMLOGCY	# TRACKS PER VOLUME
18	(12) FIXED	2	DTMTRKCY	# TRACKS PER VOLUME
20	(14) FIXED	2	DTMMAXCB	MAX COMPRESS BUF IN WORDS
22	(16) CHARACTER	2	DTMIND	VOLUME INDICAT
	1...		DTMVIRT	VIRTUAL VOLUME
	.. 1.		DTMMINI	MINI VOLUME
	.. 1.		DTMCVAF	VOLUME HAS INDEXED VTOC
24	(18) FIXED	1	DTM#VVRS	# OF VVRS NVRS DUMPED
25	(19) FIXED	1	DTM#DSCB	# OF DSCBS DUMPED
26	(1A) FIXED	1	DTM#EXT	# OF EXTENTS DUMPED
27	(1B) CHARACTER	1	DTMMODNO	MODEL NUMBER
DTBTMR				
0	(0) CHARACTER	12	DTBTMR	RESERVED
0	(0) CHARACTER	12	DTBHDR	RESERVED
0	(0) CHARACTER	4	DTBBMID	ID OF THE BLOCK = 3MBB
4	(4) FIXED	4	DTBADDR	OF NEXT 3MBB SEGMENT COPIED FROM ADR3MB BLOCK
8	(8) FIXED	2	DTBTRK#	# OF TRKS MAPPED IN 3MBB SEG
10	(A) CHARACTER	2	*	RESERVED
DTTRK				
0	(0) CHARACTER	24	DTTRK	MAPS THE TRACK
0	(0) CHARACTER	16	DTTHDR	HEADER FOR EACH TRACK SEGMENT
0	(0) FIXED	2	DTTRKLN	LENGTH OF DATA ON TRK
2	(2) CHARACTER	1	DTTRKID	TRACK INDICATORS
	1...		DTTIOER	I/O ERROR ON TRACK
	.. 1.		DTTROVVF	LAST REC ON TRK IS OVFLD REC
	.. 1.		DTTCCMP	IF ON, TRACK COMPRESSED
	... 1...		DTTVFRST	FIRST VVDS RECORD
	... 1...		DTTINVT	INVALID TRACK FORMAT
	... 1...		DTTSTAT	USER STATISTICAL RECORD
3	(3) CHARACTER	4	DTCCCHH	CCHH OF TRACK
3	(3) FIXED	2	DTTCC	CYLINDER NUMBER
5	(5) FIXED	2	DTTHH	TRACK NUMBER
7	(7) CHARACTER	6	DTTVSERL	VOLUME SERIAL
16	(10) CHARACTER	8	DTTR0DAT	RCRD 0 DATA, ONLY ON 1ST SEG
24	(18) CHARACTER		DTT0BODY	RI RN RECORDS ON TRACK
DTTCKD				
0	(0) CHARACTER	8	DTTCKD	CNT. KEY & DATA FIELDS ON TRK
0	(0) CHARACTER	8	DTTCNT	COUNT FIELD
0	(0) CHARACTER	5	DTTCCHHR	CCHHR OF RECORD
0	(0) CHARACTER	4	DTTCCCHH	CCHH OF RECORD
0	(0) FIXED	2	DTTCCC	CC OF RECORD
2	(2) FIXED	2	DTTCHH	HH OF RECORD
4	(4) FIXED	1	DTTCRCRD	R OF RECORD
5	(5) FIXED	1	DTTKELEN	KEY LENGTH
6	(6) FIXED	2	DTT0ATLN	DATA LENGTH
0	(0) CHARACTER	6	DTRTLR	
0	(0) CHARACTER	6	DTRSERL	VOLUME SERIAL OF FILE

Name	Hex Offset	Struct Level	Name	Hex Offset	Struct Level
ADRTAPB	0	1	DTPDATA	A	3
BFRPREFIX	0	1	DTPDDDSP	B	3
BPBODY	40	2	DTPDDISP	B	3
BPHWMRK	4	2	DTPDMPID	9	2
BPSCOC	0	2	DTPDSHDR	A	4
BPSCOD	2	2	DTPDSNL	A	4
BPWORKA	8	2	DTPDTRLR	A	4
DTBADDR	4	3	DTPENDCD	B	3
DTBBMID	0	3	DTPFULD	9	3
DTBHDR	0	2	DTPLOGCL	9	3
DTBTMR	0	1	DTPNOSEG	4	3
DTBTRK#	8	3	DTPPARTD	9	3
DTDAIXSP	69	3	DTPPFXLN	8	2
DTDALIAS	6	3	DTPRCFLI	B	2
DTDCAT	F	2	DTPRCIDI	A	2
DTDCATLN	1	2	DTPSCHK	0	2
DTDCDF	69	3	DTPSEGLN	6	2
DTDDSN	3B	2	DTPSEGNO	5	3
DTDDSORG	2	2	DTPSEQNO	0	3
DTDIND	6	2	DTPSPHDR	A	3
DTDIND2	69	2	DTPTHDR	A	4
DTDIPWD	6	3	DTPTRK0	A	3
DTDIRACF	6	3	DTPVHDR	A	3
DTDLEN	0	2	DTPVOLD	A	4
DTDNVOL	5	2	DTPVTOC	A	3
DTDOPTCD	4	2	DTPVTRLR	A	3
DTDPDSE	69	3	DTPVVDS	A	3
DTDPWD	7	2	DTRSERL	0	2
DTDRACFD	6	4	DTRTLR	0	1
DTDRACFG	6	4	D TSAIXNM	0	2
DTDRACFP	6	3	D TSAUTHR	32	2
DTDSHDR	0	1	D TSAUTNM	2A	2
DTDSMS	6	3	DTSCNTLP	8	2
DTDSPER	6	3	DTSCODEN	20	2
DTDTPWD	6	3	DTSERDAY	63	4
DTDVCTD	67	3	DTSEXCNV	65	3
DTDVCTI	68	3	DTSEXPRI	62	3
DTDVOLCT	67	2	DTSEYEAR	62	4
DTHBLKSZ	D	2	DTSINFO	0	1
DTHDAY	0	3	DTSLEN	0	2
DTHDR	0	1	D TSMSTRP	0	2
DTHFCMP	11	3	D TSNUMAT	28	2
DTHGNVI	8	3	DTSPASSW	0	1
DTHGVI	8	3	DTSPATHA	58	2
DTHIND1	11	2	DTSPATHE	59	3
DTHIND2	8	2	DTSPATHF	59	2
DTHLEN	9	2	DTSPATHI	59	3
DTHLVLNO	C	2	DTSPATHN	2C	2
DTHNDS	F	2	DTSPATHP	59	3
DTHSFER	11	3	DTSPHERE	0	1
DTHTIMD	0	2	DTSPRACF	59	3
DTHTIME	4	3	DTSPTHEP	62	2
DTHUNLCD	11	3	DTSPTHON	5A	2
DTHVERNO	B	2	DTSRACFD	59	4
DTLCAT	1	2	DTSRACFG	59	4
DTLDSN	0	1	DTSREADP	18	2
DTLLEN	0	2	DTSUPDAT	10	2
DTM#DSCB	19	2	DTTBODY	18	2
DTM#EXT	1A	2	DTTCC	3	4
DTM#VVRS	18	2	DTTCCC	0	5
DTMCVAF	16	3	DTTCCCCH	0	4
DTMDEVTY	6	2	DTTCCCCH	3	3
DTMIND	16	2	DTTCCCHR	0	3
DTMLOGCY	10	3	DTTCHH	2	5
DTMMAXCB	14	2	DTTCKD	0	1
DTMMINI	16	3	DTTCNT	0	2
DTMMODNO	1B	2	DTTCRCRD	4	4
DTMTRKCP	C	3	DTTDATLN	6	3
DTMTRKCY	12	3	DTTHDR	0	2
DTMVIRT	16	3	DTTHH	5	4
DTMVOL	0	1	DTTINVT	2	4
DTMVOLSZ	C	2	DTTIOER	2	4
DTMVSERL	0	2	DTTKELEN	5	3
DTPBODY	10	2	DTTRODAT	10	2
DTPBTM	A	3	DTTSTAT	2	4
DTPDASDD	9	3	DTTTCMP	2	4
			DTTTRK	0	1
			DTTTRKID	2	3
			DTTTRKLN	0	3

Name	Hex Offset	Struct Level
DTTROVF	2	4
DTVFRST	2	4
DTVSERL	7	3
DTVBLKSZ	C	2
DTVBMSZ	25	2
DTVCVAF	24	3
DTVDAY	17	3
DTVDEVTY	1F	2
DTVFCMP	24	3
DTVGNI	27	3
DTVGI	27	3
DTVIND1	24	2
DTVIND2	27	2
DTVLEN	28	2
DTVLNVI	27	3
DTVLOGCY	8	3
DTVLVF	24	3
DTVLVI	27	3
DTVLVLNO	2B	2
DTVMAXCB	E	2
DTVMINI	24	3
DTVMODNO	23	2
DTVOL	0	1
DTVOLSZ	8	2
DTVSERL	10	2
DTVSMS	27	3
DTVSMI	27	3
DTVTIMD	17	2
DTVTIME	1B	3
DTVTOCB	0	2
DTVTOCE	4	2
DTVTRKCY	A	3
DTVUNLCD	24	3
DTVVERNO	2A	2
DTVVIRT	24	3
DTVVTOCI	16	2
DTVVVDS	24	3
DTVVXTNO	2C	2
DTVVXTOF	2D	2

End of Product Implementation Information

Appendix F. Macro Library as Part of Programming Interfaces for Customers

The macros identified in this appendix are provided to allow a customer installation to write programs that use the services of DFDSS. Only those macros identified in this appendix should be used to request or receive the services of DFDSS.

DFDSS provides a library that contains the macros that are intended to help you write programs that use the services of DFDSS. The name of the library is: **SYS1.SAMPLIB**.

DFDSS provides four product-sensitive programming interfaces and one general-use programming interface intended for use by customers.

Listed below are the product-sensitive and general-use programming interface macros in SYS1.SAMPLIB:

Product-Sensitive Programming Interfaces

Figure 36. Product-Sensitive Programming Interfaces	
Interface Name	Macro
Installation Authorization Exit Routine	ADRUPB
Installation Enqueue Exit Routine	ADRUNQB
Installation Options Exit Routine	ADRUFO
Installation Reblock Exit Routine	ADRRBLKB

End of Product-Sensitive Programming Interfaces

General-Use Programming Interfaces

Figure 37. General-Use Programming Interfaces	
Interface Name	Macro
Application Interface	ADREID0 ADRMCLVL

End of General-Use Programming Interfaces

Glossary of Terms and Abbreviations

This glossary includes definitions of some terms found in this document. Some of the terms defined below are from:

- *The American National Dictionary for Information Processing Systems*, copyright 1982 by the Computer and Business Equipment Manufacturers Association. Copies may be purchased from the American National Standards Institute at 1430 Broadway, New York, New York 10018. These definitions are identified by an asterisk (*).
- *The ISO Vocabulary - Information Processing*, and *The ISO Vocabulary - Office Machines*, developed by the International Standards Organization, Technical Committee 97, Subcommittee 1. Definitions from published sections of this vocabulary are identified by the symbol "(ISO)" preceding the definition. Definitions from draft proposals and working papers under development by the ISO/TC97 vocabulary subcommittee are identified by the symbol "(TC97)," indicating that final agreement has not yet been reached among its participating members.

A

ABEND. Abnormal end of task. End of a task, a job, or a subsystem because of an error condition that cannot be resolved by recovery facilities while the task is performed.

ABENDxxx. The keyword that identifies the abnormal end of DFDSS because of a system-detected error.

ABSTR. A subparameter of the SPACE parameter in a DD statement. It indicates that specified tracks be assigned to a data set.

ACCEPT processing. An SMP/E process necessary for installing the FMIDs. SMP/E ACCEPT processing uses JCL to accept the modules and macros necessary to run the FMIDs. The FMIDs are accepted into the DLIBs from the temporary data sets.

Access method services. A multifunction service program that is used to manage both VSAM and non-VSAM data sets and integrated catalog facility or VSAM catalogs. It is used to define data sets and allocate space for them, convert indexed-sequential data sets to key-sequenced data sets, modify data set attributes in the catalog, reorganize data sets, facilitate data portability between operating systems, create backup copies of data sets, data set records and catalog entries. help make inaccessible data sets accessible, list the records of

data sets and catalogs, define and build alternate indexes, and convert OS CVOLs and VSAM catalogs to integrated catalog facility catalogs.

ACDS. Active control data set.

ACS. Automatic class selection.

AIX. Alternate index.

alias. An alternate name for a member of a partitioned data set.

ALLOC. A space allocation parameter that indicates type, such as cylinders or tracks.

alternate index (AIX). In systems with VSAM, a key-sequenced data set containing index entries organized by the alternate keys of its associated base data records. It provides an alternate means of locating records in the data component of a cluster on which the alternate index is based.

alternate index cluster. In VSAM, the data and index components of an alternate index.

APAR. Authorized Program Analysis Report.

Authorized Program Analysis Report (APAR). A request for correction of a problem caused by a suspected defect in a current unaltered release of a program.

APF. Authorized program facility.

Application Interface. An interface used to invoke DFDSS from another program.

Apply processing. In SMP and SMP/E, the process, initiated by the APPLY command, that places system modifications (SYSMODS) into the target system libraries.

attach. In programming, to create a task that can be performed asynchronously with the performance of the mainline code.

authorization. (1) (TC97) The right granted to a user to communicate with or make use of a computer system. (2) The process of giving a user either complete or restricted access to an object, resource, or function.

automatic class selection (ACS). A mechanism for assigning SMS classes and storage groups.

Index

- A**
- abend dumps, data set passwords 172
 - ABEND=nnn in the EXEC statement 7
 - ABOVE16 in the EXEC statement 7
 - access authority
 - RACF-protected data sets and DASDVOL 175
 - access authority, compatibility with Version 1 162
 - Access Method Services 177
 - addressing mode 143
 - ADRBMB mapping macro
 - output example 220
 - ADRSSU 179
 - ADREID0 data area
 - example 200
 - ADRMCLVL macro 201
 - ADRRBLKB mapping macro
 - explanation 158
 - output example 159
 - ADRREBLK routine 158
 - ADRTAPB mapping macro
 - output example 221
 - ADRUENQ routine 149
 - ADRUFO mapping macro 154
 - output example 155
 - ADRUIM module 182
 - ADRUIXIT routine 154
 - ADRUNQB mapping macro 150
 - output example 151
 - ADRUPB mapping macro 145
 - output example 148
 - ADRUPSWD routine 144
 - AIX (alternate index), recataloging 35
 - ALLDATA keyword
 - COPY command 33
 - DUMP command 63
 - PRINT command 85
 - ALLEXCP keyword
 - COPY command 33
 - DUMP command 69
 - ALLMulti keyword
 - CONVERTV command 25
 - COPY command 34
 - DUMP command 69
 - ALLOC subkeyword, DUMP command 14
 - allocation of space 47, 116
 - ALLX keyword
 - COPY command 33
 - DUMP command 69
 - alternate index, recataloging 35
 - AMODE=31 143, 181
 - AMSGCNT=nnnn in the EXEC statement 7
 - application interface
 - blocks 183
 - application interface (*continued*)
 - example 204-205
 - structure 181
 - application interface summary 196
 - application program process 203
 - asterisks, rules regarding, in a data set qualifier 10
 - authorization checking 144
 - data set 173
 - general description 171
 - volume 173
 - authorization exit routine
 - return codes 147
 - user 144
 - auxiliary commands 128
 - continuation errors 133
 - PARALLEL 128
 - SERIAL 128
 - SET 129
- B**
- backing up (dumping) data 64
 - backup copies, multiple 57, 74
 - BDAM data sets, copying 45
 - BEGIN keyword in VDRL statement (stand-alone DFDSS) 138
 - BELOW16 in the EXEC statement 7
 - BLK subkeyword of TGTALLOC
 - COPY command 47
 - RESTORE command 116
 - BLKSIZE and LRECL keywords, specifying in a PRINT operation 87
 - buffers above 16 megabytes virtual storage 7
 - BY keyword
 - COMPRESS command 20
 - COPY command 34
 - criteria 9
 - DEFRAG command 59
 - DUMP command 69
 - filtering criteria 14
 - RELEASE command 93
 - RESTORE command 104
 - BYPASSACS keyword
 - COPY command 34
 - RESTORE command 105
- C**
- calling block structure 179
 - CANCELERROR keyword
 - COPY command 34
 - DUMP command 69
 - RESTORE command 105

- CATALOG** keyword
 CONVERTV command 25
 COPY command 35
 RESTORE command 105
- catalog search order, standard 16
- CATLG subkeyword, **DUMP** command 14
- changes in DFDSS, this release xii
- characteristics of data sets 9
- class selection, process diagram 120
- codes, condition
 See condition codes
- command syntax, how to read 3
- commands, auxiliary 128
- commands, function
 See also function commands
 example 196
 PARM, information in the EXEC statement used to control DFDSS 6
- command, null 133
- compaction 70
- compatibility
 among releases of DFDSS 163
 between Version 1 and Version 2 of DFDSS 162
 of stand-alone with system DFDSS 135
- COMPRESS** command 19
- COMPRESS** keyword, **DUMP** command 70
- compressing data sets 19
- condition codes
 general description 129
 LASTCC 130
 MAXCC 130
- consolidating unused space in data sets 14
- continuation errors in auxiliary commands 133
- control statement examples
 See examples of DFDSS jobs
- control statements, stand-alone DFDSS 135
- controlling task processing
 description 129
 examples 130, 132
- CONVERT** keyword
 COPY command 36, 54
 converting to SMS, JCL example 54
 partitioned data set 36
 partitioned data set extended 36
- convert to PDSE, JCL example 55
- CONVERTV** command 24
 serialization 174
- COPY** command
 example of data set 51
 example of tracks 50
 examples of full 50
 explanation 28
 non-VSAM data set, results of **COPY** on 56
- COPY** operation
 data mover selection matrix 176
- COPYDUMP** command 57
- copying stand-alone DFDSS restore to tape 139
- COPYVOLID** keyword
 COPY command 35
 RESTORE command 106
- CPYV** keyword
 COPY command 35
 RESTORE command 106
- CREDIT** subkeyword, **DUMP** command 14
- criteria for filtering 9
- CYL** subkeyword of **TGTALLOC**
 COPY command 47
 RESTORE command 116
- D**
- DASD space fragmentation 59
- DASDVOL**, access authority requirements 175
- data integrity
 DYNALLOC 167
 ENQ 166
 example of **RESERVE-ENQUEUE** processing 171
 RESERVE 165
 WAIT option 169
- data mover selection matrix 176
- data set
 authorization checking 173
 BY filtering 14-15
 characteristics (BY criteria) 9
 DUMP
 compatibility with Version 1 162
 examples of 80
 filtering 9
 filtering criteria 15
 names, examples of fully and partially qualified 10
 non-VSAM enqueue options 167
 non-VSAM, results of **COPY** on 56
 qualifier, rules regarding asterisks in 10
 RESTORE
 compatibility with Version 1 162
 example of 122
 SMS-managed data sets
 non-VSAM 118
 physical **RESTORE** actions 118
 VSAM 118
- data-set-changed flag, **DUMP** command 75
- DATALENGTH** keyword, **PRINT** command 86
- DATASET COPY**
 data mover selection matrix 176
- DATASET** keyword
 COPY command 36
 data mover selection matrix for **COPY** 176
 DUMP command 77
 PRINT command 86
 RESTORE command 106
- DCB keywords **LRECL** and **BLKSIZE**, specifying in a **PRINT** operation 87
- DDNAME** keyword
 COMPRESS command 20
 CONVERTV command 25

- DDNAME keyword (*continued*)
 - DEFRAG command 60
 - RELEASE command 93
 - DDNAME list 179
 - DDPTR 179
 - deadlocks during DFDSS runs 165
 - DEFRAG command
 - explanation 59
 - output example 63
 - results of a successful run 63
 - degassing data sets 19
 - DELETE keyword
 - COPY command 36
 - DUMP command 70
 - DELETE, compatibility with Version 1 162
 - detailed notes on using DFDSS
 - data set passwords, abend dumps 172
 - enqueueing versus dynamic allocation 168
 - installation authorization exit 172
 - lockouts during DFDSS runs 165
 - security processing, example of 177
 - device names, generic 139
 - DFDSS
 - application interface
 - structure 181
 - compatibility among releases 163
 - data set authorization checking 173
 - determining version, release, modification level
 - using the ADRMCLVL macro 201
 - DFDSS module, main entry point 179
 - DFDSS target class selection 121
 - exit identification block 183
 - filtering 3, 9-16
 - how authorization is checked for you 171
 - input 6
 - installation exit routines 143
 - JCL 6
 - JCL requirements 5
 - job ending 134
 - lockouts during runs 165
 - output 6
 - PARM information in the EXEC statement 6
 - stand-alone restore 135
 - target class selection 120
 - using PARM information in the EXEC statement 6
 - Version 2, compatibility with Version 1 162
 - DFDSS, stand-alone restore 135
 - DFHSM 75
 - DO-END group of commands 132
 - condition for processing 130
 - syntax 131
 - DS keyword
 - COPY command 36
 - DUMP command 77
 - PRINT command 86
 - RESTORE command 106
 - DSCHA subkeyword, DUMP command 14
 - DS1DSCHA (data-set-changed flag) 75
 - DUMP command 64
 - compatibility with Version 1 162
 - data-set-changed flag 75
 - example of a full 79
 - example of a tracks 79
 - example of data set 80
 - multiple copies 74
 - syntax 64
 - dump data, multiple copies 57, 74
 - dump tapes from DFDSS Version 1 162
 - DYNALLOC keyword
 - COMPRESS command 21
 - COPY command 37
 - DEFRAG command 60
 - DUMP command 71
 - PRINT command 86
 - RELEASE command 94
 - RESTORE command 106
 - DYNAM keyword
 - COMPRESS command 21
 - DEFRAG command 60
 - RELEASE command 94
 - dynamic allocation of data sets versus enqueueing 168
- ## E
- ELSE command 131
 - END command, in DO-END 132
 - END keyword in VDRL statement (stand-alone DFDSS) 138
 - END statement in stand-alone DFDSS 138
 - ending your DFDSS job 134
 - ENQ 166
 - ENQFAILURE subkeyword
 - COPY command 48
 - DUMP command 76
 - PRINT command 88
 - RESTORE command 117
 - enqueue exit routine
 - enqueue scheme 149
 - options for non-VSAM data sets 167
 - user 149
 - enqueue scheme 166
 - enqueueing versus dynamic allocation of data sets 168
 - EOJ command 134
 - ERRORTRACKS keyword, PRINT command 86
 - examples of
 - controlling task processing 130, 132
 - examples of common continuation errors 133
 - examples of DFDSS jobs
 - compressing selected data sets 23
 - CONVERTV command 27
 - COPYDUMP 58
 - data set COPY 51
 - data set COPY for device conversion 51
 - data set DUMP 80

examples of DFDSS jobs (*continued*)

- data set RESTORE 122
- DEFRAG 62
- examples of 62
- full COPY 50
- full DUMP 79
- full RESTORE 121
- PRINT 90
- RELEASE 96
- stand-alone DFDSS Restore 141
- tracks COPY 50
- tracks DUMP 79
- tracks RESTORE 121

examples of qualified data set names 10

example, syntax 4

EXCLUDE criteria 9

EXCLUDE keyword

- COMPRESS command 21
- COPY command 37
- DEFRAG command 60
- DUMP command 71
- RELEASE command 94
- RESTORE command 107

EXEC statement (JCL) 5–8

executing stand-alone DFDSS Restore 139

exit identification block 183

exit interface, structure 183

exit routines, installation

See installation exit routines

EXPDT subkeyword, DUMP command 14

EXPORT/IMPORT (IDCAMS commands) 177

extended architecture usage 181

EXTNT subkeyword, DUMP command 14

F

FDD keyword

- COMPRESS command 21
- COPY command 37
- DUMP command 70
- RELEASE command 94
- RESTORE command 107

FILTERDD keyword

- COMPRESS command 21
- COPY command 37
- DUMP command 70
- RELEASE command 94
- RESTORE command 107

filtering

- BY 14–15
- general description 9–12

FORCE keyword

- COPY command 37
- RESTORE command 107

FRAG keyword, DEFRAG command 61

FRAGMENTATIONINDEX keyword, DEFRAG command 61

free-space fragmentation 59

FROMADDR keyword in RESTORE statement (stand-alone DFDSS) 137

FROMDEV keyword in RESTORE statement (stand-alone DFDSS) 137

FSIZE subkeyword, DUMP command 14

FULL keyword

- COPY command 38
- DUMP command 71
- RESTORE command 107

fully-qualified data set names, examples of 10

function commands

See also commands, function

detailed description of 6

G

GDG filtering 12

generation data group filtering 12

generic device names 139

glossary 226–234

H

HDINGADDR 180

how authorization is checked for you 171

how many subkeywords are allowed 2

how to write installation exit routines 144

I

IDCAMS utility 177

IDD keyword

- COPY command 39
- COPYDUMP command 57
- DUMP command 72
- PRINT command 86
- RESTORE command 108

IDY keyword

- COPY command 39
- DUMP command 72
- PRINT command 87

IEBCOPY utility 176

IEBISAM utility 176

IEHMOVE utility 176

IF-THEN-ELSE group of commands

- description 129
- examples 130, 132
- syntax 131

INCAT keyword

- CONVERTV command 25
- COPY command 38
- DUMP command 72

INCLUDE criteria 9

INCLude keyword

- COMPRESS command 22
- COPY command 39
- DUMP command 72
- RELEASE command 95

INclude keyword (*continued*)

RESTORE command 108

INDDNAME keyword

COPY command 39

COPYDUMP command 57

DUMP command 72

PRINT command 86

RESTORE command 108

INDYNAM keyword

COPY command 39

DUMP command 72

PRINT command 87

input error toleration

COPY command 48

DUMP command 76

PRINT command 88

input to DFDSS (in DD statement) 6

installation authorization exit 172

installation exit routines

authorization checking 144

enqueue 149

installation.options 154

task alterations 153

reblock 158

return codes 147

installation exit routines (overview) 143

installation options exit

user routine 154

invoking DFDSS

via ATTACH, LINK, or CALL macro 182

IOERROR subkeyword

COPY command 48

DUMP command 76

PRINT command 88

J

JCL requirements for DFDSS 5

job ending 134

JOB statement

DFDSS, JCL 5

stand-alone DFDSS Restore 136

K

KEYLENGTH keyword, PRINT command 87

keywords, how to specify 5

L**LASTCC**

definition 130

in IF-THEN-ELSE command sequence 131

in SET command 130

LIDD keyword

COPY command 40

DUMP command 73

LIDY keyword

COPY command 40

DUMP command 73

LINECNT = nnnn in the EXEC statement 6

lockout, avoiding 166

logging, RACF 7.177

LOGICALVOLUME keyword

COPYDUMP command 57

RESTORE command 108

LOGICALVOLUME, compatibility with Version 1 162**LOGINDDNAME** keyword

COPY command 40

DUMP command 73

LOGINDDNAME keyword

COPY command 40

DUMP command 73

LRECL and **BLKSIZE** keywords, specifying in a PRINT operation 87**LVOL** keyword

COPYDUMP command 57

RESTORE command 108

M

macro library 224.2

main storage requirements, stand-alone DFDSS

restore 142

mapping macro

ADRBMB 220

ADREID0 197

ADRTAPB 221

ADRUF0 155

ADRUNQB 151

ADRUPB 148

MAXCC

definition 130

in IF-THEN-ELSE command sequence 131

in SET command 130

maximum number of subkeywords allowed 2

MAXMOVE keyword, DEFrag command 61**MENTITY** keyword

COPY command 40

RESTORE command 108

MGMTCLAS keyword

COPY command 41

RESTORE command 109

MINSECQTY keyword, RELEASE command 95**MINTRACKSUNUSED** keyword, RELEASE command 95**MINTRKS** keyword, RELEASE command 95**MODEL** keyword in RESTORE statement (stand-alone DFDSS) 137

moving data sets (COPY with DELETE) 36

MSG statement in stand-alone DFDSS 136

MULTI subkeyword, DUMP command 14

multiple backup copies 57, 74

multiple subkeywords, how many are allowed 2
 multiple tracks read, on dump 74
 multivolume data sets, copying 34
 MVOLSER keyword
 COPY command 40
 RESTORE command 108

N

non-VSAM data sets, RESTORE actions on 119
 NONSMS keyword
 CONVERTV command 26
 NOPACKing keyword
 COPY command 41
 RESTORE command 109
 NORUN in the EXEC statement 6
 NSC keyword
 COPY command 46
 RESTORE command 115
 null command 133
 NULLMGMTCLAS keyword
 COPY command 41
 RESTORE command 109
 NULLSTORCLAS keyword
 COPY command 46
 RESTORE command 115
 number of subkeywords allowed 2

O

ODD keyword
 COPY command 42
 COPYDUMP command 57
 DUMP command 74
 PRINT command 87
 RESTORE command 109
 syntax 57
 ODY keyword
 COPY command 42
 RESTORE command 110
 operating in serial or parallel mode 128
 OPTIMIZE keyword, on DUMP 74
 option list 179
 OPTPTR 179
 OUTDDNAME keyword
 COPY command 42
 COPYDUMP command 57
 DUMP command 74
 PRINT command 87
 RESTORE command 109
 OUTDYNAM keyword
 COPY command 42
 RESTORE command 110
 output from DFSS (out DD statement) 6
 OUTTRACKS keyword
 COPY command 42
 RESTORE command 110

OUTTRKS keyword
 COPY command 42
 RESTORE command 110

P

page ejection, suppressing 6
 page number list 180
 PAGENO=nnnn in the EXEC statement 6
 PAGEPTR 180
 PAM data sets, compressing 19
 PARALLEL command 129
 parallel mode 129
 PARAM 180
 partially-qualified data set names, examples of 10
 partitioned data sets, compressing 19
 PASSWORD keyword
 COMPRESS command 22
 COPY command 42
 DEFRAG command 61
 DUMP command 74
 PRINT command 87
 RELEASE command 95
 RESTORE command 110
 PCTU keyword
 COPY command 44
 RESTORE command 111
 PDS, compressing 19
 PERCENTUTILIZED keyword
 COPY command 44
 RESTORE command 111
 preallocated target, example 53
 PREPARE keyword
 CONVERTV command 26
 PRG keyword
 COPY command 44
 DUMP command 75
 RESTORE command 111
 PRINT command 84-90
 examples of 90
 output example 91
 syntax 84, 85
 PRINT, compatibility with Version 1 163
 PROCESS keyword
 COPY command 44
 DUMP command 75
 RELEASE command 96, 111
 profile, defining a 43
 programming interfaces 224.2
 PSWD keyword
 COMPRESS command 22
 COPY command 42
 DEFRAG command 61
 DUMP command 74
 PRINT command 87
 RELEASE command 95
 RESTORE command 110

PURGE keyword
 COPY command 44
 DUMP command 75
 RESTORE command 111

Q
 qualified data set names, examples of 10

R
 RACF
 authorization checking 40, 108, 144
 logging 177
 profile definition 43
 RACF-protected data sets
 access authority requirements 175
 RACFLOG=YES in the EXEC statement 7
 reblock exit routine, user 158
 REBLOCK keyword
 COPY command 44
 RESTORE command 111
 RECATALOG keyword
 COPY command 35
 RESTORE command 105
 record processing 196
 REDETERMINE keyword
 CONVERTV command 26
 REFDT subkeyword, DUMP command 14
 relative generation filtering 12
 RELBLKA keyword, COPY command 45
 RELBLOCKADDRESS keyword, COPY command 45
 RELEASE command
 cross-system serialization 94
 example of 96
 syntax 92, 93
 RENAME keyword, RESTORE command 112
 RENAMEUNCONDITIONAL keyword
 COPY command 45
 RESTORE command 113
 RENUNC keyword
 COPY command 45
 RESTORE command 113
 REPLACE keyword
 COPY command 46
 RESTORE command 114
 REPLACE keyword, COPY command
 example 53
 REPRO (IDCAMS command) 177
 RESERVE 165
 RESET keyword, DUMP command 75
 resource serialization 170
 RESTORE actions on non-VSAM data sets 119
 RESTORE command 98
 compatibility with Version 1 162
 cross-system serialization 106
 example of data set 122
 example of full 121

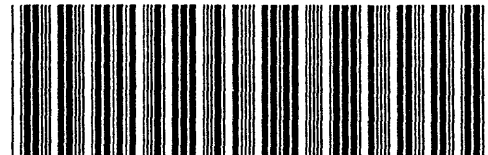
RESTORE command, (continued)
 example of tracks 121
 RESTORE command, RENAME keyword 117
 RESTORE command, RENAMEUNCONDITIONAL
 keyword 117
 RESTORE command, REPLACE keyword 117
 restore limits, stand-alone DFDSS 138
 RESTORE operation
 actions on SMS-managed data sets 118
 compatibility among releases
 stand-alone 164
 system 163
 data set, RENAME, RENUNC, REPLACE
 keywords 117
 RESTORE operation, stand-alone 135
 RESTORE statement in stand-alone DFDSS 136-137
 result from a PRINT command 90
 running stand-alone DFDSS Restore 139

S
 SAF (System Authorization Facility) 173
 passwords, abend dumps 172
 SCAN in the EXEC statement 6
 security processing, example of 177
 SERIAL command 128
 serial mode 128
 serialization
 DYNALLOC 167
 ENQ 166
 lockout, avoiding 166
 RESERVE 165
 resource 170
 WAIT option 169
 SET command 130
 SHARE keyword
 COPY command 46
 DUMP command 75
 PRINT command 88
 RESTORE command 114
 SHR keyword
 COPY command 46
 DUMP command 75
 PRINT command 88
 RESTORE command 114
 SIZE=nnnnK in the EXEC statement 6
 SMS keyword
 CONVERTV command 26
 RESTORE actions on data sets 118
 SOURCE subkeyword of TGTALLOC
 COPY command 47
 RESTORE command 116
 source, definition 6
 space allocation 47, 116
 space fragmentation on DASD 59
 space utilization, percent
 COPY command 44
 RESTORE command 111



TNL SN35-0375

SN35-0375-00



Printed in U.S.A.

Order No. SC26-4389-1

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

If we require further information, may we contact you by phone? ___ Tel. No. _____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail them directly to the address in the Edition Notice on the back of the front cover or title page.)

Reader's Comment Form

Fold and Tape

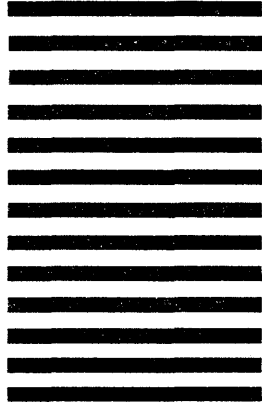
Please Do Not Staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, NEW YORK



POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department 61C
P.O. Box 27155
Tucson, AZ, USA 85775-4401

Fold and Tape

Please Do Not Staple

Fold and Tape





**Data Facility Data Set Services:
Messages
Version 2 Release 5.0**

SC26-4390-2

Third Edition (June 1991)

This edition replaces and makes obsolete the previous edition, SC26-4390-1. Changes from the previous edition are marked with a vertical bar in the left margin. Significant changes are summarized under "Summary of Changes."

This edition applies to Version 2, Release 5, Modification Level 0 of Data Facility Data Set Services, Program Number 5665-327, and to any subsequent releases until otherwise indicated in new editions or technical newsletters.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's program may be used. Any functionally equivalent program may be used instead.

Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality. If you request publications from the address given below, your order will be delayed because publications are not stocked there.

A Reader's Comment Form is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Storage Systems Products Division, Department 61F, Tucson, Arizona, U.S.A. 85744. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1988, 1991. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Special Notices

References in this publication, programs, or services do not imply that IBM intends to market products in all countries in which IBM operates. Any reference to an IBM service is not intended to state or imply that only IBM's service may be used. Any functionally equivalent product, process, or program does not infringe any of IBM's intellectual property rights. Other services may be used instead of the IBM product, program, or service, provided that the proper identification and verification of operation in conjunction with other services, except those expressly designated by IBM, are made.

IBM may have patents or patent applications covering subject matter in this document. This document does not give you any license to these patents. You can apply in writing, to the IBM Director of Commercial Relations, Armonk, NY 10577.

Trademarks

The following terms and abbreviations are used in this publication:

DB2

DFSMS

MVS/DFP

MVS/ESA

MVS/XA

Preface

This book, *Data Facility Data Set Services: Messages*, lists messages issued by DFDSS, with explanations and suggested responses. The audience for this reference is users who encounter DFDSS messages.

Data Facility Storage Management Subsystem (DFSMS) is hereafter referred to as Storage Management Subsystem (SMS).

Major Divisions of This Book

This book contains the following sections:

- **Chapter 1, "How to Read DFDSS and Stand-Alone DFDSS Restore Messages"** provides general information on and examples of DFDSS messages. This chapter explains the differences between DFDSS messages and the stand-alone DFDSS restore program (ADRDMPRS) messages.
- **Chapter 2, "System DFDSS Messages"** lists the messages issued by System DFDSS.
- **Chapter 3, "Stand-Alone DFDSS Restore Messages"** lists the messages issued by ADRDMPRS.
- **Chapter 4, "Problem Determination Table"** provides a table that explains the codes referred to in the problem determination section of each message.

Conventions Used in This Book

The following conventions are used in the text of the actual message:

- Braces ({ }) indicate a required field or parameter. Only one of the items within the braces will appear in the actual message text.
- Brackets ([]) indicate an optional field or parameter. The field or parameter within the bracket may or may not appear in the actual message text.
- Items separated by a vertical bar (|) represent alternative items. Only one of the items separated by a vertical bar may appear in the actual message text.
- Items in quotation marks (' ') indicate the actual key entry the operator must make in response to the message.

The following conventions are used in the explanations of the messages:

- Variables, which are shown in *italics*, symbolically represent either numeric or alphanumeric information that is displayed in the message text.
- Parameters and optional fields referred to in the descriptions of the messages do not necessarily reflect the exact characters that must be entered in a command. See the appropriate publication for the exact characters that must be entered for each command.

Information for each message includes, where applicable:

- **Explanation:** The meaning of the message, the causes, and the variable fields.
- **System Action:** Actions resulting from the conditions causing the message or reason code. For example, the system may be waiting for a response.

- **Programmer Response:** The pertinent response, if a response is necessary, and the effect of that response on the system or program. The programmer can be the application programmer or the system programmer, depending on the problem to be corrected.
- **Operator Response:** The pertinent response, if a response is necessary, and the effect of that response on the system or program.
- **Problem Determination:** If a problem occurs again or is persistent to the point that system efficiency suffers, the actions that can be performed to obtain adequate data for trained hardware or programming-support personnel to diagnose the problem.

Problem determination items refer to Chapter 4, "Problem Determination Table" on page 73. This table does not teach diagnostic techniques but informs the operator or programmer of actions to take when problems occur again. The problem determination instructions are aids to identifying hardware or programming problems and to ensuring that essential programming information is available for qualified support personnel to diagnose and correct the problem.

DFDSS Books

Data Facility Data Set Services: General Information, GC26-4123, introduces you to DFDSS and helps in evaluating its use. It is primarily directed to data processing management.

Data Facility Data Set Services: User's Guide, SC26-4388, gives guidance on how to use DFDSS.

Data Facility Data Set Services: Reference, SC26-4389, describes DFDSS command syntax.

Data Facility Data Set Services: Diagnosis Guide, LY27-9538, tells how to diagnose errors you might encounter while using DFDSS.

Data Facility Data Set Services/Interactive Storage Management Facility: Installation Planning Guide, SC26-4129, tells what to consider when you plan for and install DFDSS/ISMF.

Related Books

Within the text, references are made to some of the publications listed below. Others are publications that may be useful.

MVS/370 Publications:

Short Title	Long Title	Order Number
<i>JCL User's Guide</i>	<i>MVS/370 JCL User's Guide</i>	GC28-1349
<i>JCL Reference</i>	<i>MVS/370 JCL Reference</i>	GC28-1350
<i>Linkage Editor and Loader</i>	<i>MVS/370 Linkage Editor and Loader User's Guide</i>	GC26-4061
<i>System Generation</i>	<i>MVS/370 Installation: System Generation</i>	GC26-4063

MVS/XA Publications:

Short Title	Long Title	Order Number
<i>JCL Reference</i>	<i>MVS/Extended Architecture JCL Reference</i>	GC28-1352
<i>JCL User's Guide</i>	<i>MVS/Extended Architecture JCL User's Guide</i>	GC28-1351
<i>Linkage Editor and Loader</i>	<i>MVS/Extended Architecture Linkage Editor and Loader User's Guide</i>	GC26-4143
<i>System Generation</i>	<i>MVS/Extended Architecture Installation: System Generation</i>	GC26-4148

MVS/ESA Publications:

Short Title	Long Title	Order Number
<i>Application Development Guide</i>	<i>MVS/ESA Application Development Guide</i>	GC28-1821
<i>Application Development Macro Reference</i>	<i>MVS/ESA Application Development Macro Reference</i>	GC28-1822
<i>Catalog Administration Guide</i>	<i>MVS/ESA Catalog Administration Guide</i>	SC26-4502
<i>CVAF Diagnosis Reference</i>	<i>MVS/ESA Common VTOC Access Facility Diagnosis Reference</i>	ZZ26-3712
<i>Data Management Macro Instructions</i>	<i>MVS/ESA Data Management Macro Instructions</i>	SC26-4506
<i>DFP: General Information</i>	<i>MVS/ESA Data Facility Product Version 3: General Information</i>	GC26-4507
<i>JCL Reference</i>	<i>MVS/ESA JCL Reference</i>	GC28-1829
<i>JCL User's Guide</i>	<i>MVS/ESA JCL User's Guide</i>	GC28-1830
<i>Storage Administration Reference</i>	<i>MVS/ESA Storage Administration Reference</i>	SC26-4514
<i>System-Data Administration</i>	<i>MVS/ESA System-Data Administration</i>	SC26-4515
<i>VSAM Administration: Macro Instruction Reference</i>	<i>MVS/ESA VSAM Administration: Macro Instruction Reference</i>	SC26-4517

MVS/DFP Version 3.3 Publications:

Short Title	Long Title	Order Number
<i>Access Method Services Reference</i>	<i>MVS/DFP Version 3 Release 3: Access Method Services for the Integrated Catalog Facility</i>	SC26-4562
<i>MVS/DFP: General Information</i>	<i>MVS/DFP Version 3 Release 3: General Information</i>	GC26-4552

Short Title	Long Title	Order Number
<i>MVS/DFP: Storage Administration Reference</i>	<i>MVS/DFP Version 3 Release 3: Storage Administration Reference</i>	SC26-4566

MVS Message Library Publications:

Short Title	Long Title	Order Number
<i>Dump Output Messages</i>	<i>MVS/ESA Message Library: Dump Output Messages</i>	GC28-1814
<i>Routing and Descriptor Codes</i>	<i>MVS/ESA Message Library: Routing and Descriptor Codes</i>	GC28-1816
<i>System Codes</i>	<i>MVS/ESA Message Library: System Codes</i>	GC28-1815
<i>System Messages Volume 1</i>	<i>MVS/ESA Message Library: System Messages Volume 1</i>	GC28-1812
<i>System Messages Volume 2</i>	<i>MVS/ESA Message Library: System Messages Volume 2</i>	GC28-1813

Device Support Facility Publications:

Short Title	Long Title	Order Number
<i>Device Support Facility User's Guide and Reference</i>	<i>ICKDSF User's Guide and Reference</i>	GC35-0033

System Programming Library Publications:

Short Title	Long Title	Order Number
<i>Application Development Guide</i>	<i>MVS/ESA System Programming Library: Application Development Guide</i>	GC28-1852
<i>Application Development—Extended Addressability</i>	<i>MVS/ESA System Programming Library: Application Development—Extended Addressability</i>	GC28-1854

VM/SP Library Publication:

Short Title	Long Title	Order Number
<i>VM/SP CP General User Command Reference</i>	<i>Virtual Machine/System Product CP General User Command Reference</i>	SC19-6211

Contents

Chapter 1. How to Read DFDSS and Stand-Alone DFDSS Restore Messages	1
How This Book Explains Messages	1
The Format of System DFDSS Messages	1
The Format of Stand-Alone DFDSS Restore Messages	3
Chapter 2. System DFDSS Messages	5
Chapter 3. Stand-Alone DFDSS Restore Messages	67
Chapter 4. Problem Determination Table	73
Table I. Standard Problem Determination Actions	73
Appendix A. Use of Interrupt Key	75
Stand-alone Restore Environment	75
Stand-alone Restore in VM Environment	75
Glossary	77

Summary of Changes

Version 2 Release 5, June 1991

This June 1991 revision is provided to incorporate technical and editorial changes in addition to the changes, Appendix A, "Use of 75 has been added.

Version 2 Release 5, December 1988

DFDSS Version 2 Release 5 provides storage management facilities and enhancements to help you:

- Take advantage of

MVS/DFP Version 5 provides a specially designed data set, called a partitioned data set (PDSE). DFDSS supports the PDSE, which offers significant performance improvements over a partitioned data set (PDS). DFDSS also provides new options to convert a PDS to a PDSE and a PDSE to a PDS.

This support requires DFDSS Release 2 or a later release.

- Use new application options

New options for DFDSS further enhance the interaction between DFDSS and the user at the data set level. DFDSS passes control to the user (UIM) when a data set is selected and when the processing is completed. At these exit points, the UIM can perform various actions. For example, you can customize the UIM to end processing to bypass options such as serialization and security verification.

- Process data sets that have a high-level qualifier of SYS1. By using the PROCESS(SYS1) command:

- Release unused or sequential data sets that have a high-level qualifier of SYS1.
- Move data sets that have a high-level qualifier of SYS1.
- Specify the DEST parameter of the DUMP command to dump and delete data sets that have a high-level qualifier of SYS1.
- Replace preallocated data sets that have a high-level qualifier of SYS1.

- Process data sets that have a high-level qualifier of DSORG. By using the PROCESS(UNDEF) command, you can copy or restore undefined DSORG data sets that have a high-level qualifier of DSORG that has a larger track capacity.

- Specify both the OPTIMIZE and COMPRESS keywords for dumps.

You no longer must specify the COMPRESS keyword for performance (by specifying OPTIMIZE) or saving space (by specifying COMPRESS). You can back up data by using both keywords.

- Reduce CPU time by using the data-compression algorithm during DUMP.

- Use stand-alone RESTORE to restore compressed dump data.
- Reduce I/O to the VSAM volume data set (VVDS) during logical data set DUMP/RESTORE, data set COPY, and CONVERTV.
- Declare a larger block size for a SYSPRINT data set or for the output data set from the PRINT command.
DFDSS now allows a block size greater than 141.
- Perform authorization checking during RESTORE for data sets that have not been preallocated.
- Specify the target status of SMS-managed generation data sets.
- Relieve virtual storage constraints by moving DFDSS internal control blocks above the 16M byte line.

Version 2 Release 4, December 1988

Storage Management Subsystem (SMS) Support

- When processing a data set, DFDSS recognizes and preserves the data, management, and storage classes associated with the data set.
- COPY and DUMP/RESTORE commands move data sets between SMS-managed and non-SMS-managed volumes.
- The new CONVERTV command converts volumes to and from SMS management without data movement.
- DFDSS allows data set selection based on data, management, and storage class names.
- The new INCAT keyword allows DFDSS to locate source data sets cataloged outside the standard search order for COPY, DUMP, and CONVERTV. This keyword can be used in place of JOBCAT/STEPCHAT statements, which are prohibited when processing SMS-managed data sets.

Functional Enhancements

- The new SPHERE keyword allows DFDSS to move an entire VSAM sphere (base cluster and all associated alternate index clusters and paths) with one invocation of COPY, DUMP, or RESTORE.
- DFDSS COPY and RESTORE have the capability to allocate multivolume VSAM target data sets.
- The % character acts as a place holder for DFDSS data set name filtering.
- DFDSS reblocks data sets that have the reblockable indicator set on.
- The REBLOCK keyword (already supported for COPY) is supported for logical RESTORE.
- Multiple arguments for filtering criteria can be specified on the BY keyword.
- A new application interface exit point (EIOP20) has been added. DFDSS passes control to this exit when a data set is allocated by DFDSS during a physical data set RESTORE operation.
- When DFDSS (COPY or logical RESTORE) detects an underallocation of the non-VSAM target data set on an unlike device, it estimates the amount of

additional space needed and extends the data set accordingly. The copy or restore then continues, using the additional space.

Chapter 1. How to Read DFDSS and Stand-Alone DFDSS Restore Messages

DFDSS consists of two programs:

System DFDSS, which runs under the MVS system environment. Chapter 2, "System DFDSS Messages" lists system DFDSS messages.

Stand-alone DFDSS restore, called ADRDMPRS, which runs independently of the MVS system environment. ADRDMPRS is limited to one function—a full or partial (tracks) restore from a physical dump tape. Chapter 3, "Stand-Alone DFDSS Restore Messages" lists ADRDMPRS messages.

How This Book Explains Messages

Each DFDSS message, whether system or stand-alone restore, is accompanied by:

- An explanation
- A description of the system action
- A suggested programmer response
- A suggested operator response
- A reference to items in the problem determination table.

Refer to Chapter 4, "Problem Determination Table" on page 73 for an explanation of the problem determination codes listed in the messages.

The Format of System DFDSS Messages

ADR is the component name for DFDSS messages, and ADRDSSU is the program that produces the messages. For programmers, ADRDSSU produces the messages in the SYSPRINT data set, and for operators, by way of the console. The format of messages written to SYSPRINT:

ADRnnnt (xxx)-mmmm(yy), message text

The format of messages written to the console:

zzADRnnnt (xxx)-mmmm(yy), message text

Figure 1 illustrates the format of a system DFDSS message.

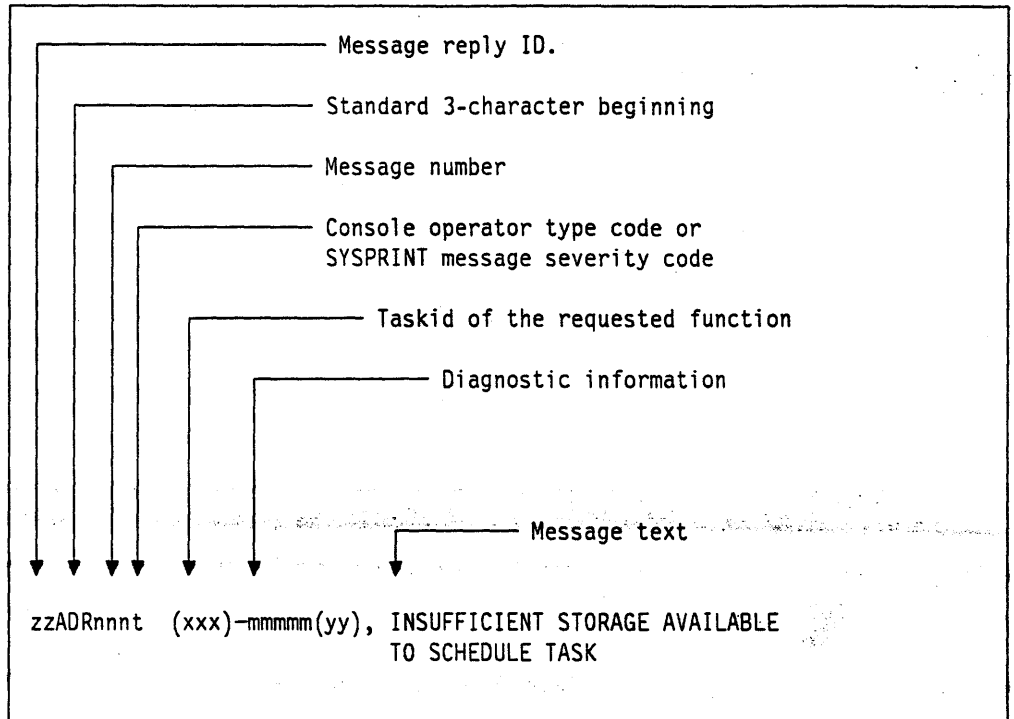


Figure 1. Format of a System DFDSS Message

Standard character representations for system DFDSS messages are:

- zz** Message reply identification (absent if reply is not required).
- ADR** Standard 3-character prefix.
- nnn** Message number. This three-digit number is unique to each message.
- t** Type code. This 1-character code identifies either the console operator type or the SYSPRINT message severity. For console messages, the types are:
- A** Action: Operator must perform a specific action.
 - D** Decision: Operator must choose an alternative action.
 - I** Information: No operator action is required.
 - W** Warning: No operator action is required, but an error occurred.

For SYSPRINT messages, the type code indicates the severity:

- I** Informational message.
- W** Warning message. Task continues, but an error occurred.
- E** Error message. The particular task might end or might continue without completing all requests.
- T** Termination message. DFDSS ends.

(xxx) The taskid of the requested function. For each function requested in the input control statements, a taskid is assigned. Each message related to a task contains this taskid in the message. If the message is general (that is, not relating to a particular function), this field is omitted.

mmmm(yy) Diagnostic information used by IBM service representatives as an aid in problem determination.

m DFDSS Messages

DFDSS messages.
 DFDSS and
 Messages" for an
 mat.

EN FAILED FOR

ame does not exist or is
 nted if a SYSPRINT DD
 sage) is not found or

eturn code is 12.
 error. Verify that a
) DD statement is present
 submit the job.

s 1, 2, 3, 13, and 29.

UFFICIENT MAIN LE TO INITIALIZE

ge was available to
 DFDSS. Message
 cate the storage needed.
 return code is 12.
 error. One of the

B or EXEC statement: to
 sage ADR018I, and

ified, increase the
 not exceed the storage

s 1, 3, 4, 13, and 29.

R ABEND 0001 WILL RRENCE n OF

ontains the ABEND =
 nth occurrence of the

until the nth occurrence
 . If the message is not

SYSIN FILE

(or acceptable
 the job stream.
 return code is 12.
 error. Make certain a
 etry the job.

1, 3, 4, 13, and 29.

ADR006I (xxx)mmmm(yy), date time EXECUTION (BEGINS|ENDS|HALTED)

Explanation: This message gives the date and time a particular task begins or ends processing.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR007E (xxx)mmmm(yy), TASK CONTAINS AN INVALID NUMBER OF I/O FILE REQUESTS

Explanation: The number of input and/or output files specified in the control statements was incorrect. The following are possible conditions:

- A full or tracks RESTORE or COPY did not have exactly one input and one output file.
- A full or tracks DUMP had more than one input volume.
- A DEFRAg function specified more than one volume.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the statement, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR008E (xxx)mmmm(yy), INSUFFICIENT STORAGE AVAILABLE TO SCHEDULE TASK

Explanation: Insufficient storage space was available for this function. Message ADR018I follows this message to indicate the storage for this task.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 3.

Programmer Response: Probable user error. Correct the error as indicated by message ADR018I, and resubmit the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR009T (xxx)mmmm(yy), INVALID {LINECNT | SIZE | TYPRUN | PAGENO | ABEND | AMSGCNT | TRACE | UTILMSG | XABUFF | RACFLOG} VALUE SPECIFIED IN PARM STATEMENT

Explanation: The indicated parameter on the EXEC statement was incorrectly specified.

- A LINECNT value must be between 1 and 9999.
- A SIZE value must be less than or equal to 9999K bytes.
- TYPRUN must be TYPRUN = SCAN or TYPRUN = NORUN.
- A PAGENO value must be between 0 and 9999.
- An ABEND value must be between 0 and 999.
- An AMSGCNT value must be between 1 and 9999.
- A TRACE value must be YES or not specified at all.
- The UTILMSG value must be either YES or NO.
- The XABUFF value must be either ABOVE16 or BELOW16.

- A RACFLOG value must be either YES or NO.

System Action: The job ends. The return code is 12.
Programmer Response: Make sure that the indicated value is valid. Check for keypunch or spelling errors. Correct any errors, and resubmit the job.
Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, and 29.

ADR0101 (xxx)mmmmm(yy), SIZE VALUE OF size WILL BE USED FOR GETMAIN

Explanation: This message indicates the value of the SIZE parameter that the requester specifies. The value does not include the storage acquired by the programs or utilities that DFDSS calls.
System Action: The storage used by DFDSS will not exceed the indicated value.
Programmer Response: None.
Operator Response: None.
Problem Determination: None.

ADR0111 (xxx)mmmmm(yy), LINECNT VALUE OF linecount WILL BE USED

Explanation: This message indicates the value of the LINECNT parameter that the requester specifies.
System Action: SYSPRINT output will be printed at the indicated lines per page. If a value of 9999 is specified, no page eject occurs.
Programmer Response: None.
Operator Response: None.
Problem Determination: None.

ADR0121 (xxx)mmmmm(yy), date time DFDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS rc [FROM:]

Explanation: This message is issued after a DFDSS job step completes. The highest condition code (MAXCC) set during the job step is printed (see message ADR0131) and returned to the DFDSS invoker in register 15. If the highest condition code is not zero, a list of the tasks completed with that code is printed. If the highest condition code is received from the syntax checking of the control statements, the word SYNTAX is printed. If the highest condition code is generated by a SET command, the words SET COMMAND are printed.
System Action: None.
Programmer Response: None.
Operator Response: None.
Problem Determination: None.

ADR0131 (xxx)mmmmm(yy), date time TASK (COMPLETED ABENDED) WITH (RETURN| SYSTEM ABEND| USER ABEND) CODE {nn|yyyy}

Explanation: If the command did not abend, the value nn is the condition code generated during processing of the command. If the command abended, the abend code yyyy is also printed. The system abend code is in hexadecimal, and the user abend code is in decimal.
System Action: If the command abended, LASTCC is set to 8, or it is set to nn. If MAXCC is less than LASTCC, MAXCC is set so it is equal to LASTCC.
Programmer Response: None.
Operator Response: None.
Problem Determination: If the command did not abend on a user-requested 001 abend, see *System Codes*.

ADR0141 (xxx)mmmmm(yy), date time ALL PREVIOUSLY SCHEDULED TASKS COMPLETED. (SERIAL|PARALLEL) MODE NOW IN EFFECT

Explanation: A switch was made from serial mode to parallel mode or from parallel mode to serial mode (at the request of the user in the control statement or by the installation-wide options exit routine).
System Action: All previously scheduled tasks either running or waiting to run have completed. Processing continues.
Programmer Response: None.
Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR0151 (xxx)mmmmm(yy), (MAXIMUM CONDITION CODE|RETURN CODE OF TASK) HAS BEEN RESET FROM rc1 TO rc2

Explanation: A SET command was encountered and successfully reset a condition code. The text indicates which taskid was affected, what the return code was before and after the reset, and to what the return code was reset.
System Action: None.
Programmer Response: None.
Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR0161 (xxx)mmmmm(yy), RACF LOGGING OPTION IN EFFECT FOR THIS TASK

Explanation: RACF logging for volume and data set authorizations by DFDSS will be performed in accordance with the resource profile specifications. The option is in effect because RACFLOG = YES was specified or the installation-wide options exit forced it.
System Action: None.
Programmer Response: None.
Operator Response: None.
Problem Determination: None.

ADR017E (xxx)mmmmm(yy), TASK NOT SCHEDULED DUE TO ERROR. TASK RETURN CODE rc error

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.
Programmer Response: None.
Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR0181 (xxx)mmmmm(yy), STORAGE REQUIRED nK. STORAGE AVAILABLE nK

Explanation: A task was not performed or the job did not start because of lack of storage. This message follows either message ADR002T or ADR008E and indicates the approximate amount of storage required when the job is rerun. The values refer only to the failure indicated in message ADR002T or ADR008E and do not take into account storage required for subsequent task scheduling.
System Action: Message ADR002T or ADR008E indicates the return code and system action.
Programmer Response: None.
Operator Response: None.
Problem Determination: None.

ADR019E (xxx)mmmmm(yy), THE (INPUT|OUTPUT) FILE HAS AN INVALID DEVICE TYPE FOR THE COMMAND SPECIFIED

Explanation: The DASD device is not supported for the command specified.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Probable user error. Correct the error, and resubmit the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR020E (xxx)mmmmm(yy), DDNAME ddname NOT FOUND

Explanation: A ddname specified in a DFDSS control statement has no corresponding ddname in a DD statement. Either a necessary DD statement is missing, or a ddname is misspelled in an existing DD statement.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Probable user error. Correct the error, and resubmit the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR021I (xxx)mmmmm(yy), THE ALLOCATION OF THE MESSAGE DATA SET HAS FAILED. DYNAMIC ALLOCATION ERROR CODE: eeee, INFORMATION CODE: iiiii. MESSAGES WILL BE SPOOLED TO SYSPRINT.

Explanation: DFDSS attempted to allocate a temporary data set to store task-related messages. If the WORKUNIT or the WORKVOL parameters, or both, were specified, they will be used to allocate the temporary data set. The error and information codes are returned by dynamic allocation. The codes are listed in *OS/VS2 MVS System Programming Library: Job Management*, or in *MVS Extended Architecture System Programming Library: System Macros and Facilities*, Volume 1. An undocumented error code may be displayed if an installation-wide validation exit is used to deny dynamic allocation but fails to get a reason code returned by dynamic allocation.

System Action: Messages may not be grouped according to task, if DFDSS is running in parallel mode. The return code is 0.

Programmer Response: Correct the problem as indicated by the error and information codes, or specify a different volume or esoteric name with the WORKUNIT and/or WORKVOL parameters.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR022T (xxx)mmmmm(yy), INVALID DCB SPECIFIED FOR (SYSPRINT|ddname)

Explanation: The DCB for the SYSPRINT file (or its acceptable alternative) must have a RECFM of VB or VBA. The logical record length (LRECL) must have a value between 84 and 137. The BLKSIZE must have a value of at least 4 greater than the LRECL. The actual printed line will be 5 less than the LRECL. If the LRECL is greater than 137, it is changed to 137 and message ADR030W is printed.

System Action: The job ends. The return code is 12.

Programmer Response: Probable user error. Correct the error, and resubmit the job. The BLKSIZE, LRECL, and RECFM need not be specified and can default. If they are specified, RECFM must be VB or VBA; LRECL must be between 84 and 137 (inclusive); and BLKSIZE must be at least 4 greater than the LRECL.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, and 29.

ADR024E (xxx)mmmmm(yy), TRACKS/OUTTRACKS VALUE IS INVALID FOR DEVICE. VALID CYLINDER RANGE IS 0 TO cccc. VALID TRACK RANGE IS 0 TO hhhh

Explanation: Invalid for the specified device types are: (1) the range specified in the TRACKS keyword of a DUMP, COPY, RESTORE, or PRINT, or (2) the range specified for the OUTTRACKS keyword for COPY or RESTORE. The TRACKS values must fall within the cylinder and track range indicated in the message. The cccc and hhhh are high cylinder and head numbers, respectively, in decimal notation.

System Action: The task ends. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the error, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR025E (xxx)mmmmm(yy), (INPUT|OUTPUT) DEVICE TYPE IS INVALID FOR TASK

Explanation: The input or output device type for the command is not allowed. For a DUMP or a CONVERTV, input must be DASD. For a RESTORE, output must be DASD. For a COPY, both input and output must be DASD. For a PRINT, input must be DASD. For a DEFRAG, the volume must be DASD.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the error, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR026E (xxx)mmmmm(yy), (volser,ddname) IS INVALID AS OBJECT DDNAME

Explanation: A reference was made to an incorrect ddname. The following restrictions apply to DD references from the control statements:

- SYSPRINT (or its alternative) can be referred to only from the PRINT command.
- SYSIN (or its alternative) cannot be referred to at all.
- The indicated volser was specified more than once for the INDYNAM parameter on a COPY, CONVERTV, or DUMP command.
- The volser associated with the indicated ddname was specified more than once for the INDDNAME or OUTDDNAME parameter on a COPY, CONVERTV, or DUMP command.

System Action: The task ends. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the error, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR027T (xxx)mmmm(yy), DFDSS IS NOT SUPPORTED BY THE CURRENT SCP

Explanation: DFDSS performs only on MVS/370 with DFP, MVS/XA, and MVS/ESA. The system identification in the nucleus is not one of these.

System Action: The job ends. The return code is 12.

Programmer Response: If the system is one of the above, correct the system identification in the nucleus, and resubmit the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, and 29. Obtain a dump of the CVT in the nucleus, and contact IBM.

ADR028I (xxx)mmmm(yy), TYPRUN = SCAN REQUESTED. ONLY CONTROL CARD SYNTAX CHECKING WILL BE DONE

Explanation: No tasks will be performed. Only checking of the input control statements will be done.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, and 29.

ADR029T (xxx)mmmm(yy), KEYWORD PRECEDING POSITION nn IN PARM STATEMENT IS INVALID

Explanation: In the EXEC statement, the keyword preceding the = sign in the indicated position of the parameter of the PARM keyword is not recognized as a valid keyword option.

System Action: The job ends. The return code is 12.

Programmer Response: Correct the error in the PARM statement, and rerun the job. The valid parameter keywords are shown in the *DFDSS: Reference*.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, and 29.

ADR030W (xxx)mmmm(yy), DCB VALUES HAVE BEEN MODIFIED FOR ddname

Explanation: In the DCB information you specified, either the LRECL or the BLKSIZE value was outside the allowable range. The LRECL and the BLKSIZE are set to default values of 137 and 141, respectively.

System Action: The job continues with the modified DCB for the ddname. The return code is 4.

Programmer Response: In the future, supply a correct DCB or let the DCB for the ddname default to its allowable values.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, and 29.

ADR031I (xxx)mmmm(yy), TYPRUN = NORUN REQUESTED. TASKS WILL EXECUTE IN NORUN MODE

Explanation: TYPRUN = NORUN was specified in the EXEC statement parameter. Subtasks will run in NORUN mode.

System Action: All subtasks except CONVERTV, DEFrag, and data set DUMP and RESTORE will bypass processing. Data set DUMP and RESTORE will perform only data set filtering and will not perform DUMP or RESTORE functions. CONVERTV will perform data set filtering and volume and data set eligibility tests, but will not actually convert any volumes. DEFrag will print the fragmentation index but will not relocate data set extents. Message ADR040I for the scheduled tasks follows this message.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR033W (xxx)mmmm(yy), INSUFFICIENT STORAGE AVAILABLE FOR OPTIMIZE n. OPTIMIZE n-1 WILL BE ATTEMPTED

Explanation: Not enough storage was available to schedule the function with the indicated OPTIMIZE value. The OPTIMIZE value is reduced by 1, and another attempt is made. This message can appear more than once as the OPTIMIZE value is decreased.

System Action: The function performs with a reduced OPTIMIZE value. The return code is 4.

Programmer Response: For the next run, either reduce the OPTIMIZE value in the control statement or increase the available storage.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR035I (xxx)mmmm(yy), INSTALLATION EXIT ALTERED keyword { DEFAULT | REQUEST | OPTION | VALUE } [TO xxxxx]

Explanation: The installation-wide options exit routine has changed the default or requested option or value to that indicated in the message. See the *Data Facility Data Set Services: Reference* under "Coding Installation-wide Exit Routines" for details.

System Action: The DFDSS function will run with the altered options.

Programmer Response: Check with your system programmer for the exit routine operation to get the desired result.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR036W (xxx)mmmm(yy), TASK NOT SCHEDULED AT REQUEST OF INSTALLATION EXIT

Explanation: The installation-wide exit requested that this task not be performed.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Check with your system programmer for restrictions placed by the exit.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR037E (xxx)mmmm(yy), FAILURE WHILE ATTACHING TASK, [ATTACH RETURN CODE IS rc | INSUFFICIENT STORAGE TO ATTEMPT ATTACH]

Explanation: A task could not be attached because one of the following conditions existed:

- The system-attach macro returned a nonzero return code while attaching a task. Probably not enough storage was available to attach the task.
- Not enough storage was available to even attempt the attach for the task.

System Action: Processing is attempted on other tasks. Final return code is set to 8.

Programmer Response: Either increase the region size, specify a smaller size parameter, or both, then rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 13, 29, and 35.

ADR038E (xxx)mmmmm(yy), RELEASE FUNCTION NOT AVAILABLE

Explanation: DFDSS determined from the CVAF table that no CVAF PARTREL macro support is installed on your system.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR040I (xxx)mmmmm(yy), PROCESSING BYPASSED DUE TO NORUN OPTION

Explanation: Further processing is bypassed because TYPRUN = NORUN was specified in an EXEC statement parameter. This option results in the following:

- COPY FULL/TRACKS—Task ends.
- COPY DATA SET—Filtering is performed and the names of the selected data sets are printed; data sets are not copied.
- COPYDUMP—Task ends.
- DEFRAG—Only volume statistics are printed; data set extents are not relocated.
- DUMP FULL/TRACKS—Task ends.
- DUMP DATA SET—Filtering is performed and the names of the selected data sets are printed; data sets are not dumped.
- PRINT—Task ends.
- RESTORE FULL/TRACKS—Task ends.
- RESTORE DATA SET—Filtering is performed and the names of the selected data sets are printed; data sets are not restored.
- COMPRESS and RELEASE—Filtering is performed and the names of the selected data sets are printed; data sets are not restored.

System Action: Further processing is bypassed for the volume or task.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR041I (xxx)mmmmm(yy), I/O BUFFER (ABOVE|BELOW) 16MB VIRTUAL REQUESTED

Explanation: You requested that the DFDSS I/O buffers reside above or below 16 megabytes virtual address.

System Action: If you requested that buffer storage reside above 16 megabytes virtual address, DFDSS will attempt to allocate storage for the buffers above 16 megabytes virtual address.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR042I (xxx)mmmmm(yy), (I/O | APPLICATION INTERFACE) BUFFER WAS OBTAINED BELOW 16MB VIRTUAL ADDRESS

Explanation: Not enough storage was available above 16 megabytes virtual address to satisfy the buffer request. Under MVS/370, this message is issued if you specify ABOVE16.

System Action: The buffer is obtained below 16 megabytes virtual address, and DFDSS processing continues.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR101I (xxx)mmmmm(yy), TASKID xxx HAS BEEN ASSIGNED TO COMMAND 'yyyyyyyy'

Explanation: The indicated taskid was assigned to the indicated command. All future messages related to this command will contain the indicated taskid following the message number.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR102I (xxx)mmmmm(yy), COMMAND BYPASSED DUE TO CONDITION CODES

Explanation: An IF-THEN-ELSE command sequence caused the command to be bypassed. When you specify an IF-THEN-ELSE command sequence, either the THEN or the ELSE clause is performed, and the one not performed is bypassed.

System Action: The bypassed portion of the command sequence is checked for syntax errors but is not performed. DFDSS processing continues.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR103W (xxx)mmmmm(yy), A 'DO' OR 'END' WAS ENCOUNTERED OUTSIDE OF AN IF-THEN-ELSE TEST

Explanation: A DO was encountered without a preceding IF-THEN-ELSE test.

System Action: A DO encountered when there is no IF-THEN-ELSE in effect does not cause DFDSS to take special action. However, it must have a matching END command. DFDSS processing continues. The return code is 4.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR104E (xxx)mmmmm(yy), AN IMPROPERLY PLACED COMMA HAS BEEN ENCOUNTERED

Explanation: An improperly placed comma was found in scanning the command. Commas cannot indicate the omission of positional parameters. The omission of leading positional parameters is not permitted.

System Action: The command is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the command syntax, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR105W (xxx)mmmmm(yy), COMMAND-END DELIMITER APPEARS WITHIN APOSTROPHES

Explanation: A quoted string contains the optional command delimiter (a semicolon). A closing single quotation mark might be omitted.

System Action: The usage is accepted, and the semicolon is treated as a valid character instead of a delimiter. The return code is 4.

Programmer Response: Check the usage of the semicolon, and correct if necessary.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR106W (xxx)mmmmm(yy), TOO MANY RIGHT PARENTHESES FOUND. EXCESS IGNORED

Explanation: Too many closing parentheses were found at the end of the command or following a first-level parameter.

System Action: The excess is ignored, and the processing is continued for the command. The return code is 4.

Programmer Response: Remove the excess parentheses.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR107E (xxx)mmmmm(yy), TOO FEW RIGHT PARENTHESES FOUND AT END OF COMMAND

Explanation: Too few closing parentheses were found at the end of the command to properly close the subparameter lists.

This message may also be issued when the continuation character is omitted in a list of subkeywords.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the command syntax and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR108I (xxx)mmmmm(yy), date time EOJ ENCOUNTERED. ALL TASKS WILL QUIESCE

Explanation: An EOJ command was encountered. A subsequent message, ADR013I, indicates when all tasks are completed.

System Action: After performing tasks before the EOJ command, DFDSS ends.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR109I (xxx)mmmmm(yy), date time INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED

Explanation: All control statements were scanned for correct function name and IF/DO/END balancing.

System Action: Processing phase now begins.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR110W (xxx)mmmmm(yy), WRITE TO OPERATOR MESSAGE HAS BEEN TRUNCATED

Explanation: A WTO command longer than 72 characters was encountered.

System Action: The excess was truncated although the WTO was issued. The return code is 4.

Programmer Response: Shorten the WTO text, or issue multiple WTO commands to accommodate the text.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR120E (xxx)mmmmm(yy), TOO MANY POSITIONAL PARAMETERS AFTER 'xxxxx'

Explanation: Too many positional parameters are specified following the characters xxxx for a parameter list.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Remove the excess parameters from the command, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR121E (xxx)mmmmm(yy), CONSTANT 'xxxxxxx' EXCEEDS CHARACTER LENGTH LIMIT

Explanation: The indicated constant contains more characters than the maximum permitted by the command syntax.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the constant, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR122E (xxx)mmmmm(yy), INCORRECT NUMBER OF CONSTANTS/ENTRIES IN LIST BEGINNING AT 'yyyy'

Explanation: An incorrect number of constants (or entries) was found in the list beginning at yyyy.

System Action: The command is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the command syntax, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR123E (xxx)mmmmm(yy), ITEM 'xxxx' DOES NOT ADHERE TO RESTRICTIONS

Explanation: The indicated parameter does not conform to the required naming conventions. For example, a dname contains invalid characters. Check the password data set for syntax errors.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the parameter error, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR124E (xxx)-mmmm(yy), DELIMITER 'xxxx' IS NOT PROPERLY PRECEDED BY A CONSTANT OR KEYWORD

Explanation: A delimiter was found where a subparameter list or data was expected. The delimiter is improperly used. Parentheses might be improperly positioned, or a positional parameter might be missing.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the syntax error, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR125E (xxx)-mmmm(yy), LEFT PARENTHESIS MISSING FOLLOWING KEYWORD 'xxxx'

Explanation: An opening parenthesis, which should begin the required subparameter list or value associated with the command keyword, does not follow the keyword xxxx.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Check the requirements of the keyword, correct the syntax, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR126E (xxx)-mmmm(yy), RIGHT PARENTHESIS MISSING AFTER 'xxxx'

Explanation: A closing parenthesis was not found where expected, or a subparameter list was not properly delimited.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the command syntax, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR127E (xxx)-mmmm(yy), INVALID PARENTHESIS FOR SPECIFYING REPEATED SUBPARAMETER LIST

Explanation: Parentheses used for delimiting repeated subparameter lists are missing or not matched.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the command syntax, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR128E (xxx)-mmmm(yy), INVALID LEFT PARENTHESIS AFTER 'xxxx'

Explanation: An opening parenthesis that appeared to delimit the positional parameter xxxx was found, but the positional parameter was not a constant or a list of constants.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the command syntax, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR129E (xxx)-mmmm(yy), KEYWORD 'xxxx' IS IMPROPER

Explanation: The command contains a misspelled, improperly specified, or inapplicable keyword.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the command syntax, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR130E (xxx)-mmmm(yy), KEYWORD 'xxxx' APPEARS TOO OFTEN

Explanation: The keyword xxxx appeared too often in the command. A parameter list might be incorrectly specified.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the command syntax by removing the keyword, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR131E (xxx)-mmmm(yy), ABOVE TEXT BYPASSED UNTIL NEXT COMMAND

Explanation: An error was encountered during the syntax check of this command, and syntax checking ended. Messages printed before this message indicate the nature of the error.

System Action: Processing continues with the next control statement. If this task was to be performed, it will not be, and a return code of 8 is set. If the command was being bypassed (see message ADR102I), the return code is not reset by this error.

Programmer Response: Correct the command syntax, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR132E (xxx)-mmmm(yy), IMPROPER PASSWORD ENCOUNTERED AFTER 'xxxxxxx'

Explanation: A password, denoted by a slash (/), was encountered where a password was not allowed; an expected password was missing; or a password longer than 8 characters was specified. This is not a verification of the password but an error in the syntax.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the command syntax, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR133E (xxx)-mmmm(yy), TOO MANY REPEATED SUBPARAMETER LISTS APPEAR IN THE COMMAND

Explanation: Too many repeated subparameter lists appear in the command.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the command syntax, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR134E (xxx)-mmmmm(yy), A HEX OR BINARY CONSTANT IS IMPROPERLY SPECIFIED

Explanation: A hexadecimal or binary constant is not specified in the correct format: X'hh...hh' or B'bb...bb', respectively.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the command syntax, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR135E (xxx)-mmmmm(yy), IMPROPER NUMERIC DIGIT FOUND IN 'xxxx'

Explanation: The constant xxxx contains an invalid character. A decimal number can be specified only with the symbols 0 through 9; a hexadecimal number can be specified only with the symbols 0 through 9 and A through F; and a binary number can be specified only with the symbols 0 and 1.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the command syntax, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR136E (xxx)-mmmmm(yy), CONSTANT 'xxxxxxx' IS NOT WITHIN VALUE RANGE

Explanation: The value of the indicated constant is outside the range of values allowed for the associated parameter.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Check the command syntax for allowed values, correct the error, then reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR137E (xxx)-mmmmm(yy), TOO MANY CONSTANTS/ENTRIES IN LIST BEGINNING AT 'xxxx'

Explanation: Too many constants or entries are specified in the command beginning at xxxx.

System Action: The command is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the command syntax, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR138E (xxx)-mmmmm(yy), REQUIRED (SUB) PARAMETER OF 'xxxx' IS MISSING

Explanation: A required parameter or subparameter that is identified by xxxx is missing.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Add the missing parameter, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR139E (xxx)-mmmmm(yy), INCONSISTENT PARAMETERS INVOLVING 'xxxx'

Explanation: Some commands contain parameters defined as mutually exclusive. For such commands, if one parameter is specified, the other parameter is not allowed.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the command syntax, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR140E (xxx)-mmmmm(yy), INVALID 'BY' CRITERIA

Explanation: A value in the BY keyword is incorrect for the item being tested. For example, a date may have been improperly specified.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the command syntax, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR141E (xxx)-mmmmm(yy), ERROR IN DATA SET NAME dsname

Explanation: A data set name was specified incorrectly for one of the following reasons:

- The length might be greater than 44 characters.
- If the data set name is not in quotation marks, a qualifier might be longer than 8 characters.
- The GDG might be incorrectly specified.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the command syntax, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR142I (xxx)-mmmmm(yy), KEYWORD yyyyyyy HAS BEEN IGNORED

Explanation: The keyword yyyyyyy does not apply and was ignored for one of the following reasons:

- LOGICALNESS—The input data set was created by a logical data set DUMP.
- WORKSIZE—This keyword is no longer required.

System Action: The keyword is ignored.

Programmer Response: Omit the keyword on future runs.

Operator Response: None.

Problem Determination: None.

ADR143E (xxx)-mmmmm(yy), INVALID SYNTAX IN DATA SET SPECIFIED BY DDNAME ddname

Explanation: Invalid data exists in the data set associated with the DD statement specified by the FILTERDD keyword.

System Action: The task ends. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the parameters in error, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR144E (xxx)-mmmmm(yy), INCOMPLETE SPECIFICATION IN DATA SET REFERENCED BY DDNAME ddname

Explanation: An end-of-file condition was encountered in scanning the character string contained in the ddname specified as a PASSWORD, FILTERDD, or EXCLUDE parameter. This condition can be caused by incorrect continuation syntax or by missing records in the input stream.

System Action: The task is not performed. The return code is 8.

Programmer Response: Correct the command syntax, or add the missing records, then resubmit the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR145E (xxx)-mmmmm(yy), KEYWORD keywordname MUST BE SPECIFIED ON DFP LEVELS PRIOR TO Vrelno.

Explanation: The indicated keyword (keywordname) is required when DFDSS is run on levels of DFP before the indicated release (relno).

System Action: Processing of the data set ends. The return code is 8.

Programmer Response: Add the keyword or install the appropriate release of DFP, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR150E (xxx)-mmmmm(yy), TASK TERMINATED DUE TO INSUFFICIENT MAIN STORAGE

Explanation: Not enough storage was available to set up the commands in the input stream for the indicated task.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Either increase REGION size or increase the SIZE parameter, or both, then retry the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR169T (xxx)-mmmmm(yy), NO RECORDS IN SYSIN STREAM

Explanation: The SYSIN stream was empty.

System Action: The job ends. The return code is 12.

Programmer Response: Ensure that the SYSIN (or acceptable alternative ddname) DD statement refers to a nonempty file, and retry the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR170T (xxx)-mmmmm(yy), ERROR OPENING DDNAME ddname

Explanation: The SYSIN (or acceptable alternative ddname) file cannot be opened.

System Action: The job ends. The return code is 12.

Programmer Response: Reestablish the SYSIN file, and resubmit the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR171T (xxx)-mmmmm(yy), MAXIMUM NUMBER OF TASKS EXCEEDED

Explanation: The SYSIN stream contains more than 255 tasks.

System Action: The job ends. The return code is 12.

Programmer Response: If necessary, run multiple job steps to accommodate the number of requested tasks.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR172T (xxx)-mmmmm(yy), ERROR DURING INITIAL SCAN

Explanation: During the initial scan of the control statements, an error was encountered.

System Action: The job ends. The return code is 12.

Programmer Response: Correct the problem as indicated by the previous message, and retry the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR173T (xxx)-mmmmm(yy), I/O ERROR WHILE READING ddname, xxxxxxx

Explanation: An I/O error was encountered in reading the SYSIN file (or acceptable alternative ddname). The ddname is followed by the SYNAD information obtained by the SYNADAF macro.

System Action: The job ends. The return code is 12.

Programmer Response: Correct the cause of the error or reestablish the SYSIN file, and retry the job.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR174T (xxx)-mmmmm(yy), REMAINDER OF COMMAND INPUT STREAM WAS IGNORED

Explanation: An error occurred that prohibits further scanning of the command stream. Messages printed before this message indicate the nature of the error.

System Action: The program ends. The return code is 12.

The condition code (MAXCC) is always set to 12 when this situation is encountered.

Programmer Response: Correct the error, and resubmit the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR175T (xxx)-mmmmm(yy), COMMAND 'xxxxxxx' IS UNKNOWN

Explanation: The command name is not recognized.

System Action: The program ends. The return code is 12.

Programmer Response: Reissue the command with the correct command name.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR176T (xxx)-mmmmm(yy), AN 'ELSE' COMMAND IS IMPROPERLY PLACED

Explanation: An ELSE clause appeared in the command without a corresponding IF statement.

System Action: The remainder of the stream is ignored. DFDSS processing ends. The return code is 12.

Programmer Response: Correct the command syntax, and resubmit the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR177T (xxx)mmmmm(yy), AN 'END' COMMAND IS INVALID

Explanation: An END statement was encountered without a matching DO statement.

System Action: The remainder of the stream is ignored.

DFDSS processing ends. Message ADR172T follows this message. The return code is 12.

Programmer Response: Correct the DO-END sequence, and resubmit the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR178T (xxx)mmmmm(yy), AN 'IF' COMMAND IS INVALID

Explanation: The sequence of the IF-THEN-ELSE statement is incorrect. Only the system variables LASTCC and MAXCC can be specified, and the values must be decimal numbers from 0 through 99999. However, a value greater than 16 will be reduced to 16. If the variable LASTCC is specified, a function command must have been previously encountered in the input command stream.

System Action: The remainder of the stream is ignored.

DFDSS processing ends. Message ADR172T follows this message. The return code is 12.

Programmer Response: Check the requirements of the IF-THEN-ELSE statement sequence, correct the error, then reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR179T (xxx)mmmmm(yy), A 'SET' COMMAND IS INVALID

Explanation: An invalid SET command was encountered. Only the system variables LASTCC and MAXCC can be specified, and the values must be decimal numbers from 0 through 99999. However, a value greater than 16 will be reduced to 16. If the variable LASTCC is specified, a function command must have been previously encountered in the input command stream.

System Action: The remainder of the command stream is ignored. The return code is 12.

Programmer Response: Check the requirements of the SET statement, correct the error, then resubmit the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR180T (xxx)mmmmm(yy), IMPROPER OR MISSING 'THEN'

Explanation: The THEN clause of the IF-THEN-ELSE command sequence is missing or is misspelled.

System Action: The remainder of the command stream is ignored. The return code is 12.

Programmer Response: Correct the error, and reissue the command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR181T (xxx)mmmmm(yy), TOO MANY LEVELS OF 'IF' COMMAND NESTING

Explanation: More than 10 IF statements were nested, or an unended DO group was discovered when the command input stream ended.

System Action: The remainder of the command stream is ignored. The return code is 12.

Programmer Response: Restructure the command stream to avoid the excessive nesting and to end all DO groups, then resubmit the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR182T (xxx)mmmmm(yy), INPUT STREAM END-OF-FILE FOUND BEFORE END OF COMMAND

Explanation: An end-of-file condition was encountered while scanning the command. This condition can be caused by incorrect command-continuation syntax or by missing records in the input stream.

System Action: The job ends. The return code is 12.

Programmer Response: Correct the command syntax, or add the missing records, then resubmit the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR183I (xxx)mmmmm(yy), THE REPLACE KEYWORD REPLACES THE PURGE KEYWORD FOR A DATA SET RESTORE

Explanation: The REPLACE keyword replaces the PURGE keyword for a data set RESTORE.

System Action: Processing continues.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR201E (xxx)mmmmm(yy), VTOC IN ERROR PRIOR TO PROCESSING

Explanation: An error occurred in processing the VTOC. The error condition can be caused by one of the following:

- An I/O error occurred. This is usually accompanied by system I/O error messages.
- A logical error occurred during indexed VTOC processing. When extents are relocated on an indexed volume, a recovery environment is set up by DFDSS. If DEFrag ends before completing the relocation, the audit trail left by the recovery routines indicates to a subsequent DEFrag operation that a mismatch exists between the VTOC index records and the extents.
- A failure occurred during LSPACE(SVC 78) processing.
- The VTOC structure is incompatible. Message ADR210E accompanies this message. More than one of the extents, either data set or free space, points to the same location on the volume.
- A prior DEFrag run or a DADSM function ended before updating the VTOC free-space maps.

System Action: The DEFrag function ends. The return code is 8.

Programmer Response: Do the following when applicable, then rerun the DEFrag function:

- If I/O error messages accompany this message, take corrective action recommended at your site for the type of error.
- If message ADR210E accompanies this message, recover by either deleting one of the overlapping data sets or by using AMASPZAP to change the incorrect extent pointer to a correct known value.
- In all other cases, allocate a temporary data set. This invokes DADSM modules to rebuild the free-space map or to correct the mismatch error. If this procedure does not correct the problem on a nonindexed volume, use AMASPZAP to turn the DIRF bit off and the DOS bit on in the VTOC entry. Then, allocate a temporary data set.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 14, 20, 25b, 29, and 35.

ADR204W (xxx)-mmmmm(yy), ERROR ENCOUNTERED ON EXTENT nnn OF dsname. EXTENT IS BYPASSED.

Explanation: An I/O error occurred in processing an extent of the data set for a DEFRAG operation.

System Action: The extent is not moved. Processing is continued with the next data set extent. The return code is 4.

Programmer Response: Use the PRINT command to print the contents of the track or use IEHATLAS to recover the data and assign an alternate track. If a system I/O error message accompanies this message, take the appropriate action recommended at your site for the error.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR205I (xxx)-mmmmm(yy), EXTENT nnn OF DATA SET dsname IS ALLOCATED BY CYLINDER, BUT IT IS NOT ON A CYLINDER BOUNDARY

Explanation: The VTOC entry indicates that the data set is either allocated by cylinder or has the round attribute. However, the extent does not begin and/or end on a cylinder boundary.

System Action: DEFRAG attempts to relocate the extent so that it begins and ends on a cylinder boundary. If DEFRAG is successful, the VTOC entry remains unchanged. If DEFRAG is unsuccessful, the extent is relocated as though it were originally allocated by tracks, and the VTOC entry is updated accordingly.

If this message is issued because the round bit is set, the round bit is not turned off. In either case, the message can be received on subsequent DEFRAG operations for the same data set.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25, and 29.

ADR206E (xxx)-mmmmm(yy), I/O ERROR OCCURRED WHILE READING VTOC. FUNCTION IS TERMINATED

Explanation: An I/O error occurred in reading the VTOC.

System Action: The function ends. The return code is 8.

Programmer Response: Use IEHATLAS to recover the data, assign an alternate track, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 25a, and 29.

ADR207E (xxx)-mmmmm(yy), ERROR WHILE SETTING DOS BIT IN THE VTOC DURING DEFRAG. TASK IS TERMINATED

Explanation: On a nonindexed VTOC, DEFRAG sets the DOS bit on in the VTOC before it starts processing and resets it at the end. This message appears if the setting or resetting cannot be done.

System Action: DEFRAG ends. The return code is 8.

Programmer Response: If the DOS bit is on, allocate a temporary data set on the volume to allow DADSM to rebuild the free space information, then scratch the temporary data set.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25b, and 29.

ADR208I (xxx)-mmmmm(yy), date time BEGINNING STATISTICS ON volser:

FREE CYLINDERS	cccc
FREE TRACKS	tttt
FREE EXTENTS	nnnn
LARGEST FREE EXTENT (CYL,TRK)	cccc,tttt
FRAGMENTATION INDEX	0.fff
PERCENT FREE SPACE	pp

Explanation: The DEFRAG function issues this message at the beginning of the task. This message is also issued when the CHECKVTOC keyword is specified for DUMP or COPY. The free cylinder value is the number of complete free cylinders. The free track value is the number of free tracks in addition to free cylinders. The cccc and tttt are the cylinder and track values (decimal) respectively. The nnnn indicates the number of free extents on the volume, and (cccc,tttt) indicates the size of the largest free extent in cylinders and additional tracks. The pp is the percentage of free space on the volume. The .fff is the fragmentation index. This index can be a value between 0.000 and 1.000, for which a higher value indicates a higher level of fragmentation.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR209I (xxx)-mmmmm(yy), MOVED EXTENT nnn FROM cc:hh-cc:hh TO cc:hh-cc:hh FOR dsname

Explanation: The nnnth extent of data set dsname was successfully relocated from the original cc:hh-cc:hh to a new cc:hh-cc:hh.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR210E (xxx)-mmmmm(yy), (FREE SPACE|dsname) AND (FREE SPACE|dsname) OVERLAP FROM CC:HH cchh1 TO CC:HH cchh2

Explanation: The indicated data set names and/or FREE SPACES overlap on the volume according to the VTOC. The VTOC is in error.

System Action: If this message is issued during DEFRAG processing, then the DEFRAG function ends with a return code of 8. No extents will be moved by DEFRAG. If this message is issued as a result of the CHECKVTOC keyword during DUMP or COPY processing, then the continuation of the DUMP or COPY processing is determined by the CANCELERROR keyword specification.

Programmer Response: Recover by either deleting one of the overlapping data sets or by using AMASPZAP to change the incorrect extent pointer to a correct value, if known.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, and 29.

ADR211I (xxx)-mmmmm(yy), DATA SET dsname IN USE AND WAS NOT MOVED

Explanation: Another job enqueued or allocated the data set while DEFRAG was trying to relocate one of its extents. The particular extent was not relocated.

System Action: Processing continues.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR212I (xxx)-mmmmm(yy), EXTENT DISTRIBUTION MAP FOR volser:

EXTENT SIZE IN TRACKS	*FREE SPACE BEFORE*		*FREE SPACE AFTER*		*ALLOCATED*	
	NO. EXTS	CUM. PCT/100	NO. EXTS	CUM. PCT/100	NO. EXTS	CUM. PCT/100
ttt	eeee	n.nnn	eeee	n.nnn	eeee	n.nnn
-						

Explanation: The distribution map indicates the free-space fragmentation before and after the run and the distribution by size of data set extents. The map following this message lists both the number of FREE SPACE BEFORE extents (eeee) and the number of FREE SPACE AFTER extents (eeee) for each given size in tracks (ttt), along with the number of ALLOCATED extents (eeee). A cumulative percentage of allocated extents is also provided for each of the three categories (n.nnn).

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, and 29.

ADR213I (xxx)-mmmmm(yy), ENDING STATISTICS ON volser:

DATA SET EXTENTS RELOCATED	nnnn
TRACKS RELOCATED	tttt
FREE CYLINDERS	cccc
FREE TRACKS	tttt
FREE EXTENTS	eeee
LARGEST FREE EXTENT (CYL,TRK)	cccc,tttt
FRAGMENTATION INDEX	.fff
[*** = NON-STANDARD O/S VOLUME]	

Explanation: This message, printed at the end of the DEFrag run, indicates the numbers of relocated data set extents and tracks. Message ADR208I describes the other values. An LSPACE SVC obtains the values indicated in FREE CYLINDERS, FREE TRACKS, FREE EXTENTS, and LARGEST FREE EXTENT. When asterisks (***) appear for these values, the target was a nonstandard OS volume or a DOS/VSE volume.

System Action: None.

Programmer Response: When asterisks appear in the message, verify that the DIRF bit of the volume is off in the VTOC entry.

Operator Response: None.

Problem Determination: None.

ADR220I (xxx)-mmmmm(yy), INTERVAL BEGINS AT CC:HH cc:hh AND ENDS AT CC:HH cc:hh

Explanation: An area of DASD was selected from which a data set extent can be relocated. The interval begins at the first cc:hh and ends at the second cc:hh.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR221E (xxx)-mmmmm(yy), OVERLAP ENCOUNTERED ON EXTENT nnn OF dsname

Explanation: The indicated data set extent overlaps another data set extent on the volume according to the VTOC. Message ADR210E accompanies this message.

System Action: Processing continues. The return code is 8.

Programmer Response: Refer to ADR210E.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, and 29.

ADR226W (xxx)-mmmmm(yy), CCHH RANGE cchh1 TO cchh2 IS UNACCOUNTED FOR IN VTOC.

Explanation: The indicated cchh range is not described in the VTOC. This range will be ignored.

System Action: Processing continues on the remainder of the volume. The return code is 4.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, and 29.

ADR227E (xxx)-mmmmm(yy), CVAF ERROR ON VOLUME volser, CODE fcec

Explanation: An error was encountered during a common VTOC access facility (CVAF) call for the volume. The fc is the CVAF function code, and ec is the CVAF error code. See the *CVAF Diagnosis Reference* for an explanation of the error codes.

System Action: The function ends. The return code is 8.

Programmer Response: If the CVAF function code-error code (fcec) is X'0C00', allocate a temporary data set on the volume to cause DADSM to repair the VTOC; then rerun the job. For other errors, take the appropriate action recommended by CVAF.

Operator Response: None.

Problem Determination: If the CVAF function code-error code (fcec) is not X'0C00', follow Table I, items 1, 2, 3, 4, 13, 20, 25a, 29, and 35.

ADR228W (xxx)-mmmmm(yy), INSUFFICIENT FREE SPACE OR FREE SPACE FRAGMENTATION TO PROCESS DEFrag FUNCTION OR A FREE SPACE EXTENT MEETS THE MAXMOVE SPECIFICATION. TASK IS TERMINATED

Explanation: The DEFrag function determined one of the following:

- The volume was insufficiently fragmented to process the function
- There was insufficient free space to process the function
- There was a free-space extent large enough to meet the MAXMOVE specification.
- If there is fragmented free space on the volume, DEFrag cannot relocate extents with its algorithms or cannot relocate them because they are extents which DEFrag does not process (e.g., VVDS, VTOCIX, etc.) to further reduce the fragmentation.

System Action: DEFrag does not alter the volume. The return code is 4.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR229W (xxx)mmmmm(yy), VOLUME volser HAS TRACKS UNACCOUNTED FOR. THE DOS BIT IN THE VTOC WILL REMAIN ON FOLLOWING PROCESSING

Explanation: On a nonindexed VTOC, the DEFRAG function encountered undefined data or free space. This space will not be processed. However, the DOS bit will remain on following the run. After the DEFRAG run, when the DADSM function allocates or scratches a data set on the volume, the free space information will be rebuilt, reclaiming the unaccountable space as free space.

System Action: The remainder of the volume is processed. The return code is 4.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, and 29.

ADR230I (xxx)mmmmm(yy), MAXMOVE VALUE EXCEEDS THE NUMBER OF FREE TRACKS ON THE VOLUME. IT HAS BEEN ADJUSTED TO THE LOWER VALUE OF nnnn TRACKS

Explanation: The MAXMOVE value must be less than or equal to the number of free tracks on the volume being processed. In this case, the value exceeds the number and is adjusted to the lower value.

System Action: Processing continues with a MAXMOVE value equal to the number of free tracks on the volume.

Programmer Response: The MAXMOVE value, if not specified, defaults to the total number of free tracks on the volume.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, and 29.

ADR231E (xxx)mmmmm(yy), ERROR WHILE ACCESSING VVDS FOR DATA SET dsname ON VOLUME volser. OP = x, VVDS RC = yyyy

Explanation: The VVDS is accessed for the specified call during a DEFRAG, CONVERTV, or data set RESTORE. The operation codes (x) can be:

- 0 Insert VVR,NVR
- 1 Get a VVR,NVR for update
- 2 Put a VVR,NVR for update
- 3 Generic read of components of a cluster
- 4 Read a VVR,NVR.

The yyyy is the return code from the VVDS manager. See the reason code under return code 50 in message IDC3009I in *System Messages*.

System Action: If the operation is DEFRAG, the extent is not relocated. If the operation is data set RESTORE, the affected data set is not restored. Other messages identifying the affected data set accompany this message. The return code is set to 8.

Programmer Response: Take the appropriate action recommended for the reason codes listed under message IDC3009I.

Operator Response: None.

Problem Determination: See return code 50 in message IDC3009I in the *System Messages*. Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR232E (xxx)mmmmm(yy), VVDS DOES NOT MATCH VTOC (EXTENTS|DSORG) FOR COMPONENT componentname, CLUSTER clustname ON VOLUME volser

Explanation: DEFRAG detected a mismatch between the VTOC and the VVDS.

EXTENTS: The number or locations of extents for this component as reported in the VVDS do not match the number or locations of extents reported in the VTOC. A VSAM extend operation on the component might have been interrupted because of a program or system failure.

DSORG: DEFRAG detected a VVDS entry for a VSAM component which is named the same as a VTOC, non-VSAM entry.

System Action: The component is not relocated. Final return code is set to 8.

Programmer Response: EXTENTS: Run Access Method Services DIAGNOSE for the component, and fix the problem by using the procedure recommended in the *Access Method Services Reference*.

DSORG: Dump or copy the contents of the offending cluster. Then run access method services to delete and recreate the offending cluster. Finally, run RESTORE or COPY to replace the contents.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, and 29.

ADR233W (xxx)mmmmm(yy), ACTUAL FRAGMENTATION INDEX IS LESS THAN THE REQUESTED VALUE ON VOLUME volser

Explanation: The volume is not fragmented to the extent indicated in the user input.

System Action: DEFRAG ends. The return code is 4.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR234I (xxx)mmmmm(yy), SEQUENCE CCHH1 - CCHH2 EXTENT DESCRIPTION

Explanation: This is the message header for ADR235I. This message is issued when TYPRUN = NORUN is specified for DEFRAG, or when the CHECKVTOC keyword and TYPRUN = NORUN are specified for DUMP or COPY.

System Action: Processing continues.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR235I (xxx)mmmmm(yy), nnnnnnn cchh1 cchh2 eee FREE SPACE dsname *UNACCOUNTED FOR*****

Explanation: This is the volume extent mapping message. This message is issued when TYPRUN = NORUN is specified for DEFRAG, or when the CHECKVTOC keyword and TYPRUN = NORUN are specified for DUMP or COPY. One message is issued for each extent on the volume. nnnnnnn is the relative sequence in which the extent was encountered during VTOC analysis, cchh1 cchh2 is the range the extent occupies on the volume, and eee is the extent number of the data set or free space or unaccounted for extent.

System Action: Processing continues.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR236W (xxx)mmmmm(yy), VTOC ON VOLUME volser DOES NOT CONTAIN A FREE DSCB

Explanation: The referenced volume does not contain any free DSCBs. Because DEFrag requires a free DSCB for processing protected data sets, protected data set relocation cannot be done.

System Action: Only unprotected data sets are relocated. The final return code is 4.

Programmer Response: If a temporary data set exists on the volume, scratch it (to free a DSCB), and rerun the DEFrag job to relocate protected data sets. If a temporary data set does not exist on the volume, DUMP a data set, specifying DELETE (to free a DSCB), and rerun DEFrag to process protected data sets.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, and 29.

ADR237W (xxx)mmmmm(yy), NOT ENOUGH FORMAT-5 DSCBS AVAILABLE

Explanation: On a nonindexed VTOC, DEFrag cannot find enough space in the VTOC to rebuild the free space information at the end of DEFrag to reflect the free space on the volume.

System Action: The DOS bit is left on in the VTOC. The return code is set to 4.

Programmer Response: Allocate a temporary data set so DADSM can rebuild the Format 5 chain.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, 25b, and 29.

ADR238I (xxx)mmmmm(yy), NO DATA SETS WERE MOVED DURING THIS PASS. PROCESSING CONTINUES.

Explanation: No data sets could be moved by DEFrag. This may be due to data sets that are in use, on the link list, or extents too large for the available free space area.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR240E (xxx)mmmmm(yy), INSUFFICIENT FREE SPACE EXTENTS TO PROCESS DEFrag FUNCTION. TASK IS TERMINATED.

Explanation: There were less than 2 free space extents on the volume. DEFrag requires at least 2 free space extents to process.

System Action: DEFrag terminates without altering the volume. The return code is 4.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR241I (xxx)mmmmm(yy), TARGET VTOC [INDEX] BEGINNING AT cc:hh AND ENDING AT cc:hh IS (BEING USED | OVERLAID)

Explanation: Either the target VTOC or index VTOC in the indicated cc:hh range is being preserved, or the source VTOC location was used for the target VTOC.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR242I (xxx)mmmmm(yy), CHECKVTOC KEYWORD IGNORED. NO INPUT VOLUMES SPECIFIED

Explanation: The CHECKVTOC keyword was specified with the DUMP or COPY command, but no input volumes were specified. No VTOC analysis is performed.

System Action: Processing continues.

Programmer Response: Specify an input volume via the INDDNAME or INDYNAM keywords.

Operator Response: None.

Problem Determination: None.

ADR243W (xxx)mmmmm(yy), VTOC ERRORS FOUND ON VOLUME volser. PROCESSING CONTINUES

Explanation: The CHECKVTOC keyword was specified with the DUMP or COPY command, and VTOC errors were encountered on the specified volume. Because the CANCELERROR keyword was not specified, the DUMP or COPY operation proceeds.

System Action: Processing continues. The return code is 4.

Programmer Response: Refer to the warning and error messages preceding this message.

Operator Response: None.

Problem Determination: Refer to the warning and error messages preceding this message.

ADR244E (xxx)mmmmm(yy), PROCESSING BYPASSED DUE TO CANCELERROR KEYWORD AND VTOC ERRORS FOUND ON VOLUME volser

Explanation: The CHECKVTOC keyword was specified with the DUMP or COPY command, and VTOC errors were encountered on the specified volume. Because the CANCELERROR keyword is specified, the DUMP or COPY operation is terminated.

System Action: Processing terminates. The return code is 8.

Programmer Response: Refer to the warning and error messages preceding this message.

Operator Response: None.

Problem Determination: Refer to the warning and error messages preceding this message.

ADR301E (xxx)mmmmm(yy), AN ERROR WAS ENCOUNTERED WHILE FILTERING DATA SETS ON VOLUME volser. VOLUME WAS NOT PROCESSED.

Explanation: I/O or logical errors were encountered in accessing the VTOC or VVDS on the specified volume during a data set DUMP, COPY, or CONVERTV operation.

System Action: The specified volume is bypassed. Processing continues with the next volume, if any. The return code is 8.

Programmer Response: For I/O errors, follow your site's recommendations for I/O errors, and resubmit the job. For logical errors encountered in the VVDS, run Access Method Services DIAGNOSE against the indicated volume and correct the problem.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 5b, 13, 20, 25b, and 29. If it is a logical error, use the PRINT command to print the VTOC and VVDS tracks before taking action 29.

**ADR302E (xxx)mmmmm(yy), INPUT AND OUTPUT
DEVICE TYPES DO NOT MATCH. UCB
TYPE OF INPUT xxxx, OUTPUT xxxx**

Explanation: The source and target volumes of a TRACKS or FULL volume COPY or RESTORE are not of like or similar device types.

System Action: The task ends. Processing continues with the next control statement. The return code is 8.

Programmer Response: Probable user error. Select a matching device type, correct the control statement in error, and rerun the job. If you want to copy to an unlike device, specify DATASET on your COPY command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, and 20. If this message occurs when the device types are the same, see Table I, items 29 and 35.

**ADR303E (xxx)mmmmm(yy), COMMAND NOT
SUPPORTED WITH SYSTEM RESIDENCE
VOLUME**

Explanation: A full volume RESTORE or COPY onto, or DEFRAg of, the system residence volume is not allowed. The system residence volume was specified for output.

System Action: The task ends. Processing continues with the next control statement. The return code is 8.

Programmer Response: This is a user error. Do not RESTORE, COPY onto, or DEFRAg the system residence volume. Correct the control statement in error, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, and 13. If this message occurs when the system residence volume was not specified as the output volume, see Table I, items 29 and 35.

**ADR304E (xxx)mmmmm(yy), ERROR ENCOUNTERED
WHILE ACCESSING (VTOC|VVDS) IMAGE
OF VOLUME volser ON DDNAME ddname,
xxxxx**

Explanation: An error (logical or I/O) was encountered in analyzing the image of the VTOC or VVDS on the dumped tape during data set RESTORE processing. The types of possible logical errors can be:

- A non-VTOC or non-VVDS record
- An invalid format of the VVDS
- No data on track
- I/O errors encountered on the track during the DUMP
- An end of file during the VTOC or VVDS search.

If physical I/O errors were encountered, other I/O error messages precede this message. The xxxxx in the message is the SYNAD information passed by the access method routines.

System Action: The volume is bypassed. The return code is 8.

Programmer Response: Make sure the function completed normally during the DUMP process. If the error is in the VVDS, non-VSAM data sets can be restored when you specify DSORG NE VSAM in the BY criteria. Use a different backup copy to do the RESTORE.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 5b, 13, and 20. If it is a logical error, list the VVDS and VTOC records on the dump tape and take action 29.

**ADR305E (xxx)mmmmm(yy), UNABLE TO SERIALIZE
(VTOC|VVDS) ON VOLUME volser**

Explanation: DFDSS failed to get exclusive control of the associated DASD volume via enqueueing of the volume's VOLID or VVDS during a DUMP/RESTORE/COPY/CONVERTV/DEFRAg operation, or a TRACKS or VTOC PRINT operation even after a specified or defaulted number of retries. If the resource is VTOC, DADSM functions might have been performing while DFDSS was trying to serialize the use of the DASD volume. If the resource is VVDS, catalog or VSAM functions might have been accessing the VVDS.

System Action: For all operations except DUMP/RESTORE, the function ends with a return code of 8. A logical data set dump with LOGINDD or LOGINDDY specified also ends with a return code of 8. Processing continues with the next control statement. For a data set DUMP/RESTORE operation without LOGINDD or LOGINDDY specified, the volume is bypassed; a return code of 8 is set, but processing continues on any other volumes.

Programmer Response: Either code VOL=(PRIVATE,SER=xxxxxx) in the associated DD statement, or request the operator to issue a mount command for the volume with the USE=PRIVATE subparameter to reduce the possibility of other jobs allocating to (or accessing) the volume. Change the WAIT parameters to increase the WAIT seconds and the retry count. When system activity on the volume is reduced, rerun or resubmit the job.

Operator Response: Mount the volume with USE=PRIVATE before rerunning the job.

Problem Determination: If no other jobs are running at the time, see Table I, items 1, 2, 3, 4, 5a, 13, 20, 29, and 35.

**ADR306E (xxx)mmmmm(yy), UNABLE TO (COPY |
RESTORE) THE VOLUME BECAUSE
OUTPUT VOLUME volser IS IN USE. TASK
IS TERMINATED**

Explanation: A FULL volume COPY or RESTORE was attempted on the volume, but another job is using the volume. It is possible that another job is accessing a data set on the volume. This message can also be issued during a data set COPY when INDD is specified. In this case, the VTOC on the volume was already enqueued.

System Action: The task ends. The return code is 8.

Programmer Response: Either code VOL=(PRIVATE,SER=xxxxxx) in the associated DD statement, or request the operator to issue a mount command with the USE=PRIVATE subparameter to reduce the possibility of other jobs allocating (or accessing) the volume. In the case of data set COPY, remove the INDD keyword and rerun the job.

For information regarding the DFP "F CATALOG, UNALLOCATE" problem, see the *DFDSS User's Guide* under "Restoring Volumes", subheading of "Output Volume Specification."

Operator Response: Mount the volume with USE=PRIVATE before rerunning the job.

Problem Determination: If no other jobs are running at the time, see Table I, items 1, 2, 3, 4, 5a, 13, 20, 29, and 35.

**ADR307E (xxx)mmmmm(yy), UNABLE TO OPEN
VOLUME volser, xx-yy**

Explanation: DFDSS is unable to OPEN the indicated volume for the indicated reason code (xx). The yyy is the return code passed by OBTAIN or RDJFCB. The possible reason codes are:

- 4 OBTAIN failure on VTOC's VTOC entry.

- 6 The VTOC's VTOC entry is not the first record in the VTOC.
- 8 RDJFCB failure
- 12 OPEN failure.

See your *System-Data Administration* book for an explanation of the OBTAIN or RDJFCB error return codes.

System Action: The task ends. Processing continues with the next control statement. The return code is 8.

Programmer Response: If the reason code is 6: DFSS does not support volumes for which the VTOC's VTOC entry is not the first record in the VTOC.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, and 29.

ADR308E (xxx)mmmmm(yy), ACCESS DENIED FOR VOLUME volser

Explanation: One of the following explanations applies:

- The user was unable to obtain the necessary RACF or password access:
 - To the volume during a data set DUMP/COPY/CONVERTV/PRINT/RESTORE operation
 - To the volume during a COMPRESS or RELEASE operation
 - To a data set on the volume for a full or tracks DUMP/RESTORE/COPY/PRINT operation.
- The installation-wide exit routine denied access to the volume.
- The operator denied access to a checkpoint/restart, RACF-protected, or system data set on the volume.
- I/O errors were encountered during checking authorization of the volume, and the operator denied further processing.

System Action: The task ends on a full or tracks operation. The volume is bypassed for a data set operation or for a COMPRESS or RELEASE. The return code is 8.

Programmer Response: Obtain proper access authority, and retry the job.

Operator Response: If the programmer has proper authority, reply U to ADR369D or ADR371D.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25b, and 29.

ADR309E (xxx)mmmmm(yy), SOURCE AND TARGET DEVICE CAPACITIES DO NOT MATCH. CYLINDER CAPACITY OF SOURCE VOLUME xxxx, TARGET VOLUME yyyy

Explanation: The capacity of the source volume is greater than that of the target volume. The xxxx is the highest cylinder number in hexadecimal.

System Action: The task ends. Processing continues with the next control statement. The return code is 8.

Programmer Response: Check whether the source or target volume is:

- An OS/VS-formatted V.M/370 minidisk
- A 3380E (source) and a 3380D (target)
- A 3380K (source) and a 3380E or 3380D (target)
- A 3380E (source) and a 3390-1 (target: in 3380 compatibility mode)
- A 3380K (source) and a 3390-1 or -2 (target: in 3380 compatibility mode)

- A 3390-2 (source) and a 3390-1 (target)

Such conditions are not allowed on a full volume operation, or if the range falls outside the capacity of the volume on a tracks operation. Select the correct input and output volumes, correct the statement in error, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 5a, 13, 20, 25b, 29, and 35.

ADR310W (xxx)mmmmm(yy), ERROR FOUND IN VTOC. UNALLOCATED SPACE WILL BE PROCESSED FOR VOLUME volser

Explanation: This message is printed for the following reasons:

- A volume without an indexed VTOC indicates no valid free space information entries exists in the VTOC.
- On a volume without an indexed VTOC, a nonzero return code was received from OBTAIN because an I/O error occurred, the DSCB was not found, or an invalid work area pointer was passed to OBTAIN.
- A nonzero return code was passed by DFP on a volume with an indexed VTOC.
- A volume with an indexed VTOC was accessed on a system without DFP installed.
- The DIRF bit is ON; or, on a volume without an indexed VTOC, the DOS bit is ON.

System Action: The entire volume, including unallocated space, is dumped or copied. The return code is 4.

Programmer Response: For an I/O error, take the corrective action recommended at your site. If the DOS bit is on for a volume without an indexed VTOC, allocate a temporary data set to let DADSM rebuild the VTOC entry.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25b, and 29.

ADR311E (xxx)mmmmm(yy), TRACKS OPERATION MUST SPECIFY PURGE TO OVERLAY (TRACK ZERO | VTOC | VVDS | VTOCIX)

Explanation: The PURGE parameter was not specified for a TRACKS COPY or RESTORE on track 0, the VTOC, the VVDS, or the VTOC index data set.

System Action: The task ends. Processing continues with the next control statement. The return code is 8.

Programmer Response: If you want to overlay any of these, specify the PURGE keyword in the command, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 5b, 13, 20, 25b, and 29.

ADR312W (xxx)mmmmm(yy), DURING A TRACKS OPERATION, THE VTOC OR TRACK 0 WAS OVERLAID

Explanation: For an attempted TRACKS COPY or RESTORE on the VTOC or track 0, the location and extents of the VTOC on the source and target volumes did not match. **System Action:** The VTOC or track 0 is overlaid. The COPY or RESTORE continues. The return code is 4.

Programmer Response: Make sure that the overlay was intended. If not, restore the output volume from a prior backup.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 5, 13, 20, 25a, and 29.

ADR313E (xxx)mmmmm(yy), OPERATOR DENIED OVERLAY OF {TRACK ZERO | VTOC | VVDS | VTOCIX} ON VOLUME volser

Explanation: A TRACKS COPY or RESTORE was attempted on track 0, VTOC, VVDS, or VTOC index data set. The operator was prompted for the necessary authorization, but the authorization was denied.

System Action: The task ends. Processing continues with the next control statement. The return code is 8.

Programmer Response: Rerun the job, and have the operator allow the VTOC or track 0 to be overlaid.

Operator Response: If you have sufficient authority to overlay track 0 or VTOC, reply U to message ADR345D.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 14, and 29.

ADR314E (xxx)mmmmm(yy), ERROR ENCOUNTERED WHILE ACCESSING {VTOC|VVDS} ON volser. PROCESSING CONTINUES

Explanation: An I/O error was encountered in dumping the VTOC or VVDS of the specified volume.

System Action: A return code of 8 or 16 is set on the input or output error, respectively. Processing continues on input errors and ends on output errors. If CANCELERROR is specified, processing ends on input errors but continues for subsequent volumes on a data set DUMP.

Programmer Response: For a data set operation, use IEHATLAS or Device Support Facilities to recover the bad VTOC or VVDS track and rerun the job. If it is the VVDS, follow the procedure recommended at your site for VVDS errors (run Access Method Services DIAGNOSE and take appropriate action). On a data set DUMP, you can recover the logical volumes that were successfully dumped by running a COPYDUMP operation and specifying the volume serial numbers in the LOGICALVOLUME parameter.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 5, 13, 20, 25a, and 30.

ADR315I (xxx)mmmmm(yy), REQUIRED CHANNEL COMMANDS NOT SUPPORTED ON VOLUME volser. OPTIMIZE(1) WILL BE USED

Explanation: The device does not support the Read Multiple Count, Key, and Data or Read Track channel commands; or I/O errors were encountered in testing whether the Read Multiple Count, Key, and Data and Read Track channel commands are supported on the device. OPTIMIZE(2), OPTIMIZE(3), and OPTIMIZE(4) are applicable only if one of these channel commands is supported. This can also occur if the device can be accessed via multiple control units. Some multiple control units support either channel command and others do not.

System Action: OPTIMIZE(1) is used to run the DUMP function.

Programmer Response: To get the performance advantages, ensure that all control units attached to the device support the Read Multiple Count, Key, and Data or Read Track channel commands.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR316W (xxx)mmmmm(yy), AN {I/O|LOGICAL} ERROR WAS ENCOUNTERED ON THE {VOLUME LABEL TRACK | VTOC | VVDS | DATA TRACKS}, AND ONE OR MORE TRACKS WERE NOT PROCESSED

Explanation: An error occurred on the specified track or tracks on the source volume and CANCELERROR was not specified. Other messages that precede this message indicate the track or tracks that had the error.

System Action: The tracks that had the errors were not processed. If you are doing a DUMP and the error was on VVDS, a data set RESTORE cannot be done for VSAM data sets. If the error was on the VTOC, a data set RESTORE cannot be done for any data set, but a full or tracks RESTORE is possible. If the error was on the volume label track and track zero is to be copied or restored, correct the error and rerun the job. The return code is 4.

Programmer Response: If you are doing a DUMP and the output is intended for a data set RESTORE, correct the error and rerun the job. If it is a logical error on the VVDS, VTOC, or volume label track, do a TRACKS PRINT of the VVDS, VTOC, or volume label track to check their format. Also, if it is the VVDS, run Access Method Services DIAGNOSE on the VVDS to determine the nature of the problem. Correct the error.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 30. If it is a logical error on the VVDS or VTOC, see Table I, items 5b and 35.

ADR317I (xxx)mmmmm(yy), THE DEVICE SIZE FIELD IN THE VTOC IS INCORRECT IF VOLUME volser IS NOT A MINIVOLUME. DEVICE SIZE OF nnnn CYLINDERS WILL BE USED

Explanation: DFDSS uses the device size value passed by the operating system's DEVTYPE macro. To determine whether this is correct, DFDSS does an I/O on the volume using the high cylinder value. If the I/O fails (failure can occur if it is a VM/370 minivolume) and the device size field in the VTOC is less than the one passed by DEVTYPE, the value in the VTOC is used.

System Action: The device size nnnn (in hexadecimal) is used.

Programmer Response: If it is not a minivolume and the nnnn value is incorrect, the DFDSS function completed in error. Run AMASPZAP to correct the device size field in the VTOC to reflect a full volume, then rerun the job. If it is a minivolume, action by DFDSS is correct. Verify that your minivolume size equals the size indicated in the message.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25b, and 29.

ADR318I (xxx)mmmmm(yy), y TRACKS OUT OF x WERE DUMPED FROM VOLUME volser USING NON-OPTIMIZE CHANNEL PROGRAM AFTER ENCOUNTERING TRACK OVERFLOW RECORDS

Explanation: OPTIMIZE channel programs fail on tracks that have overflow records. An attempt was made to dump x tracks using optimize channel program chain. Of these x tracks, y tracks were dumped using a nonoptimized channel program chain because track overflow records were encountered in using optimize channel programs.

System Action: The y tracks are processed using normal channel programs. This results in performance degradation.

Programmer Response: If the number is large, the names of data sets not marked as track overflow data sets in the VTOC must be specified in the TRACKOVERFLOW keyword if OPTIMIZE(2), (3), or (4) is specified. If ALLDATA or

ALLEXCP is specified for a data set, the allocated tracks beyond the end of the used data might have residual track overflow records. If so, do not code ALLDATA or ALLEXCP for these data sets.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR319W (xxx)mmmm(yy), VOLUME SERIAL NOT CHANGED ON VOLUME volser

Explanation: An I/O error was encountered in trying to read track 0 from the target volume at the end of the COPY or RESTORE. Message ADR348E precedes this message. The rest of the tracks were successfully copied or restored.

System Action: Processing ends. The return code is 4.

Programmer Response: The COPY or RESTORE completed successfully but the volume serial did not change. Use AMASPZAP to update the serial number of the volume, or COPY or RESTORE track 0 again. If an alternate track can be assigned to track 0, use Device Support: Facilities to do so. Rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 30.

ADR320I (xxx)mmmm(yy), VOLUME SERIAL xxxxxx ON UNIT cuu IS CHANGED TO yyyyyy

Explanation: The volume serial was changed upon user's request. The operator is notified of the change.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR321E (xxx)mmmm(yy), DATA SET dsname NOT ON VOLUME volser

Explanation: The requested data set is not on the volume.

System Action: Processing continues with the next data set. The return code is 8.

Programmer Response: Correct the data set name or locate the correct volume, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, and 29.

ADR322E (xxx)mmmm(yy), DATA SET dsname WAS NOT SELECTED DUE TO DUPLICATE NAME

Explanation: During a DFDSS DUMP or COPY, an attempt was made to process one or more data sets with the same name.

System Action: The data set is not selected. The return code is 8. Processing continues for other data sets.

Programmer Response: Redefine the data set with a unique name or select only that volume to process by specifying the input volumes. Rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, and 29.

ADR323E (xxx)mmmm(yy), ACCESS DENIED FOR DATA SET dsname ON VOLUME volser

Explanation: The user does not have password or RACF access to the data set, or the operator denied access to the checkpoint/restart or system data set.

System Action: The PRINT task ends. Processing continues with the next control statement. The return code is 8.

Programmer Response: Acquire authorization to access the data set, and rerun the job.

Operator Response: If the user has authorization to access the data set, reply U to message ADR369D or ADR371D.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25b, 29, and 35.

ADR324E (xxx)mmmm(yy), THE VOLUME/DATA SET SPECIFIED BY (DDNAME ddname|VOLUME volser) HAS BECOME UNUSABLE

Explanation: A RESTORE, COPY, or DUMP has not completed successfully, consequently leaving the output volume or data set in questionable condition. For a COPY or RESTORE, the volume serial number defines the device that has become unusable. For a DUMP, the DD statement ddname defines the tape or DASD data set that has become unusable as input to the RESTORE. The volume has become unusable because DFDSS control information was written to the volume yet no valid data sets were copied to it, leaving the volume unusable for any restore copies from that volume. Other messages indicating the cause of the error precede this message.

System Action: For RESTORE and COPY, the current task ends with a return code of 16, and processing continues with the next task. For DUMP, output is halted to the indicated ddname but continues for other ddnames, provided the installation-wide options exit does not force the cancellation of all output upon the occurrence of a single error. If the installation-wide options exit forces such a cancellation, the task ends with a return code of 16, and processing continues with the next task.

Programmer Response: If this message is accompanied by I/O error messages, take corrective action recommended in your site for such errors and rerun the job. If the error was on the output DASD during a COPY or RESTORE, a rerun might fail during authorization checking of the volume. In this case, initialize the volume offline to clean the volume before rerunning the COPY or RESTORE.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25b, and 29.

ADR325E (xxx)mmmm(yy), VOLUME volser HAS UNEXPIRED DATA SETS.

Explanation: A full or tracks COPY or RESTORE tried to overlay unexpired data sets.

System Action: The task ends. Processing continues with the next control statement. The return code is 8.

Programmer Response: If overlay is intended, specify the PURGE option, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25b, and 29.

ADR327W (xxx)mmmmm(yy), NO DATA SETS PROCESSED FROM (VOLUME volser|LOGICAL VOLUME volser)

Explanation: One of the following applies:

- A data set DUMP, COPY, CONVERTV, or RESTORE did not select any data sets that matched the filtering criteria.
- The selected data sets cannot be processed because of failure to ENQ or to allocate.
- The selected data sets cannot be processed because of failure to access the protected data sets.
- For a RELEASE, no data sets were selected because none have releasable space.

System Action: Processing continues on the next volume. The return code is 4.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25b, and 29.

ADR328E (xxx)mmmmm(yy), ERROR DURING DATA SET SELECTION. VOLUME volser IS BYPASSED

Explanation: A data set DUMP, COPY, CONVERTV, or RESTORE with filtering failed on the indicated volume during VTOC or VVDS processing. For a RESTORE, the volser is the volume serial of the source volume.

System Action: Processing continues on the next source volume. The return code is 8.

Programmer Response: Determine the failure from the previous messages associated with this taskid.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR329I (xxx)mmmmm(yy), DATA SET DUMP OF VOLUME volser (BEGINS|ENDS) ON TAPE serial SEQUENCE seq

Explanation: A data set DUMP for the indicated volume started or ended on the specified tape and on the specified data sequence number.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR330W (xxx)mmmmm(yy), OUTPUT VOLUME volser IN USE. PROCESSING CONTINUED

Explanation: The output volume was in use by other jobs in the system during a tracks COPY or RESTORE. Some of the data sets mapped by these tracks might have been in use while DFDSS was processing the volume.

System Action: The tracks operation is continued. If the data sets mapped by these tracks were in use, the contents of these tracks are unpredictable. The return code is set to 4.

Programmer Response: Analyze the job log to check if the data sets mapped by the tracks were in use at the time of the COPY or RESTORE. If they were, rerun the COPY or RESTORE if necessary. For information regarding the DFP "F CATALOG, UNALLOCATE" problem, see the *DFDSS User's Guide* under "Restoring Volumes", subheading of "Output Volume Specification."

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 29, and 35. Display the contents of the UCB when this job is

running to check on the number of data sets allocated or opened to this device.

ADR331E (xxx)mmmmm(yy), OUTPUT BLKSIZE yyyyy FOR DATA SET ON DDNAME ddname IS SMALLER THAN INPUT BLKSIZE zzzzz

Explanation: The block size specified in the JCL (or, if not specified, the defaulted block size) for output on DASD is smaller than the block size of the input volume created during the DUMP. This is not allowed. The yyyyy and zzzzz are in decimal notation.

System Action: The task ends. Processing continues with the next control statement. The return code is 8.

Programmer Response: Do not specify block size in the JCL, or, if a DASD data set is required, use a DASD that allows a larger block size.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29. The first record on the dump tape contains the input block size.

ADR333W (xxx)mmmmm(yy), DATA SET CHANGE INDICATOR NOT RESET FOR {dsname|clustername componentname} ON volser, (I/O ERROR|OBTAIN ERROR|NOT ENQ BY DFDSS, rc)

Explanation: In the VTOC on the indicated volume, the data set change indicator cannot be reset in the data set's VTOC entry. Either DADSM OBTAIN failed, an I/O error occurred while writing the data set VTOC entry, or the data set was not enqueued by DFDSS. *rc* is the error code passed by the OBTAIN macro.

System Action: The data set change indicator was not reset for the data set. The return code is 4.

Programmer Response: Either rerun the job, or use AMASPZAP to reset the indicator.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR335W (xxx)mmmmm(yy), DATA SET CHANGE INDICATOR NOT RESET FOR ANY DATA SET ON A VTOC TRACK ON VOLUME volser

Explanation: An input I/O error was encountered on a VTOC track when the data set change indicator was being reset in the VTOC entries.

System Action: The data set change indicator is not reset in any of the VTOC entries on the track. The return code is 4.

Programmer Response: Wait until the next run to reset the change indicator, or run IEHLIST to list the VTOC and use AMASPZAP to reset the VTOC entries.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25b, and 29.

ADR336I (xxx)mmmmm(yy), LOGICAL ERROR WHILE READING VOLUME volser, TRACK cchh, CCW opcode

Explanation: An unexpected error was encountered during a read operation that used the Read Multiple Count, Key, and Data or Read Track channel command. Normally, you need not take any action; however, see "Programmer Response".

System Action: The track is reread using other channel commands.

Programmer Response: If you get this message too many times in a job, then either the DASD ERP program that retries a temporary data check or DFDSS might be in error.

Operator Response: None.

Problem Determination: Table I, items 35, 5, and 29.

ADR338W (xxx)mmmm(yy), DATA SET CHANGE INDICATOR NOT RESET FOR ANY DATA SET ON VOLUME volser

Explanation: DFDSS was unable to acquire exclusive access of the VTOC on the specified volume for resetting the data set change indicator in the VTOC entries.

System Action: The data set change indicator is not reset for any data set on the volume. The return code is 4.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR339W (xxx)mmmm(yy), I/O ERRORS WERE ENCOUNTERED ON THE FOLLOWING DATA SETS DURING PROCESSING FROM volser

Explanation: A data set DUMP or COPY was specified, but one or more data sets encountered I/O errors. The list of data sets following this message indicates which data sets did not dump or copy successfully because of I/O errors.

System Action: If DELETE or RESET options were indicated in the control statements, the data sets indicated in the list are not deleted and the data set change indicator is not reset. The return code is 4.

Programmer Response: If a RESTORE is attempted for these data sets, the tracks that had the error are not restorable and will be cleared.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 30.

ADR340E (xxx)mmmm(yy), OUTPUT I/O ERROR ENCOUNTERED ON VOLUME volser DURING RESET OF THE DATA SET CHANGE INDICATOR IN THE VTOC

Explanation: A permanent output error was encountered on the input volume while the data set change indicator was being reset during a full DUMP. The output from the DUMP is still usable. An I/O error message indicating the nature of the error precedes this message.

System Action: Processing continues with the next control statement. The return code is 8.

Programmer Response: If the I/O error message indicates that the volume is usable, take the same action as in message ADR338W; otherwise, restore the volume.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25a, 25b, and 30.

ADR341 (xxx)mmmm(yy), DATA SET CHANGE INDICATOR HAS BEEN RESET FOR ALL QUALIFIED DATA SETS ON VOLUME volser

Explanation: For all VTOC entries that had the indicator on, the data set change indicator was reset during a full volume DUMP.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR343E (xxx)mmmm(yy), VTOC IS FULL ON THE SINGLE TARGET VOLUME. NO MORE DATA SETS WILL BE PROCESSED.

Explanation: DFDSS cannot continue to allocate data sets on the volume specified by the user because its VTOC is full.

System Action: The job ends. The return code is 8.

Programmer Response: Use ICKDSF or a similar product to expand the VTOC of the target volume, and rerun the job or specify additional target volumes.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, and 29.

ADR344I (xxx)mmmm(yy), VOLSER ON UCB cuu IS A DUPLICATE. VOLUME MADE UNAVAILABLE.

Explanation: A RESTORE or COPY resulted in a duplicate volume serial number in the UCB.

System Action: The volume on the specified UCB is made unavailable to the system. Message ADR320I precedes this message.

Programmer Response: None.

Operator Response: To make the new volume available to the system, the old volume with this volume serial must be unloaded or varied offline before the new volume is made available to the system.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR345D (xxx)mmmm(yy), REQUEST AUTHORIZATION TO OVERLAY (TRACK ZERO | VTOC | VTOC INDEX | VVDS) ON UCB cuu, jobname, stepname, procname. REPLY U OR T

Explanation: A TRACKS COPY or RESTORE is being attempted on a portion of the VTOC or on track zero.

System Action: Processing continues according to the operator's response. The return code is 0.

Programmer Response: Instruct the operator to let DFDSS continue processing, and rerun the job.

Operator Response: A response of R zz. U allows DFDSS to overlay track 0 or the VTOC. A response of R zz, T ends the operation, and processing continues with the next function.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR346I (xxx)mmmm(yy), VTOC INDEX WILL BE DISABLED ON VOLUME volser

Explanation: The installation-wide exit requested that the VTOC index not be reconstructed.

System Action: The DFDSS function continues processing. The SYS1.VTOCIX.volser data set is not reconstructed by ICKDSF.

Programmer Response: The ICKDSF program can be performed separately from DFDSS.

Operator Response: None.

Problem Determination: None.

ADR347E (xxx)mmmm(yy), PERMANENT I/O ERROR ON DDNAME ddname,xxxxxxxxxx

Explanation: A permanent I/O error occurred on the indicated DD statement. The ddname is followed by the SYNAD information passed by EXCP.

System Action: A DUMP ends with a return code of 16.

Continuation of a RESTORE depends on the CANCELERROR option coded by the user. If CANCELERROR is coded, the RESTORE ends with a return code of 16; otherwise, recovery is attempted but results are

unpredictable. Processing continues with the next input control statement if a return code of 16 is set.

Programmer Response: For a DUMP, take appropriate corrective action, and rerun the job. For a RESTORE, use a different version of the backup copy.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 5, 13, 20, and 30.

ADR348E (xxx)-mmmmm(yy), PERMANENT
(INPUT|OUTPUT) ERROR ON VOLUME
volser cchh opcode csw sense

Explanation: The DASD device contained a permanent I/O error. The *cchh*, *opcode*, *csw*, and *sense* are printed in hexadecimal digits.

System Action: Another message explaining the ending action, if any, follows this message. The return code is 8.

Programmer Response: Take appropriate action recommended at your site for the error identified by the sense information in the message.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 30.

ADR349W (xxx)-mmmmm(yy), ERROR ENCOUNTERED
WHILE REBUILDING F5-DSCB ON
VOLUME volser

Explanation: An error occurred when a temporary DFDSS data set, &&DFDSS.TEMPDS.volser, was to be allocated.

System Action: The DOS converted VTOC bit for the indicated volume is set on. The free space information should be accurate. The return code is 4.

Programmer Response: Use the IEHLIST utility program to determine the VTOC accuracy. If there were other DFDSS or allocation error messages, you might need to perform the COPY or RESTORE job again.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, and 25b.

ADR350E (xxx)-mmmmm(yy), (I/O LOGICAL ERROR)
ENCOUNTERED ON VVDS|VVDS IMAGE)
ON volser

Explanation: One of the following applies:

- I/O errors were encountered while the VVDS tracks were being read.
- Logical errors were encountered while the VVDS was being scanned for authorization checking on a TRACKS or FULL volume operation.
- Logical errors were encountered while data sets were being selected during a data set DUMP or RESTORE.

System Action: If this error occurs during a FULL volume or TRACKS operation while checking authorization, other messages accompany this message. If this error occurs during data set selection, the volume is bypassed. The return code is 8.

Programmer Response:

- For a logical error during DUMP: Run Access Method Services DIAGNOSE on the volume to determine the problem, and correct it by using the procedure recommended by the *Access Method Services Reference*.
- For a logical or physical error during RESTORE: Use a previous backup copy to restore the desired VSAM data set.

- For a physical I/O error during DUMP: Take the appropriate action recommended in your site, or contact your system programmer.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR351E (xxx)-mmmmm(yy), UNEXPECTED END OF
FILE ON DDNAME ddname

Explanation: An unexpected end of file was encountered before the normal end of processing. The input used might have been created in a DUMP that abnormally ended.

System Action: The task ends. The output volume or data sets may be unusable. Processing continues with the next control statement. The return code is 8 or 16.

Programmer Response: Use a good copy of the backup, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR352W (xxx)-mmmmm(yy), VOLID ON UNIT cuu,
THAT HAD VSAM DATA SETS, HAS BEEN
CHANGED FROM volser1 TO volser2
DURING A TRACKS OPERATION

Explanation: The output volume contains VSAM data sets. The VOLID was changed on the volume. If the tracks mapping these data sets were not copied or restored, these data sets might be inaccessible because of the change in the VOLID.

System Action: The VOLID was changed. The return code is 4.

Programmer Response: If the data sets are inaccessible, reset the VOLID to the original value.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR353I (xxx)-mmmmm(yy), THE FOLLOWING DATA
SETS WERE SELECTED FROM VOLUME
volser

Explanation: TYPRUN=NORUN was specified in the EXEC statement parameter, and in data sets, the list of names that follows this message met the filtering criteria on the volume.

System Action: The data sets are not processed.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25b, and 29.

ADR354W (xxx)-mmmmm(yy), ERRORS ENCOUNTERED
WHILE PROCESSING {VVDS|VTOC} ON
VOLUME volser

Explanation: Errors were encountered while the VVDS or VTOC were being processed.

System Action: If the error is on the VTOC, you cannot do a data set RESTORE on VSAM data sets. Only full or tracks RESTORE can be done from the DUMP output. The return code is 4.

Programmer Response: Other error messages accompany this message. If you want to do a data set RESTORE from the DUMP output, take the appropriate action and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR355W (xxx)mmmmm(yy), CHANGE VOLID ON UNIT cuu FROM volsr1 TO volsr2

Explanation: A full COPY or RESTORE onto the volume was done from the source volume without specifying COPYVOLID. The VVDS data set name in the VTOC and the volume serial of the VVR for the VVDS on the output volume were carried over from the input volume. This does not agree with the volume serial of the output volume.

System Action: The VSAM data sets on the receiving volume cannot be accessed until the volume serial of the receiving volume is changed. The return code is 4.

Programmer Response: If you want to access VSAM data sets, use Device Support Facilities to change the volume serial of the receiving volume. Do not rerun the COPY or RESTORE job until the volume serial is changed because authorization checks may fail.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 13, 20, 25b, and 29.

ADR356E (xxx)mmmmm(yy), TASK TERMINATED BY UIM EXIT (nn)

Explanation: The indicated User Interaction Module exit point, nn, ended the related DFDSS function before normal completion.

System Action: The return code is 8.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 5, 13, 20, 29, and 35.

ADR357W (xxx)mmmmm(yy), PROCESSING VSAM DATA SETS CATALOGED IN THE INTEGRATED CATALOG FACILITY IS NOT SUPPORTED UNDER THIS OPERATING SYSTEM.

Explanation: The input volume contained VSAM data sets cataloged in the Integrated Catalog Facility Catalog, or a request (specific or implied) to process these data sets was made. Such processing is not supported by DFDSS under this operating system. You can process these data sets only in the MVS/370 or MVS XA Data Facility Product environment with the REALLOC Macro Support installed.

System Action: If requested, non-VSAM data sets are processed. The return code is 4.

Programmer Response: If non-VSAM data set processing is desired, but not requested, rerun the job requesting only non-VSAM data set processing.

Operator Response: None.

Problem Determination: If the system is running in the DFP/370 or DFP XA with REALLOC Macro Support installed, take a SYSABEND DUMP on message ADR357W. Table I, items 1, 2, 3, 4, 5, 13, 20, 29, and 35.

ADR358W (xxx)mmmmm(yy), PROPER LEVEL levelnum OF productname NOT ON THE SYSTEM. {COMMAND | VOLUME volsr | DATA SET dsname} WILL BE PROCESSED, rc.

Explanation: The proper level of the named licensed product is not installed. The reason code, rc, indicates:

- 1 VSAM linear data set support requires MVS/XA Data Facility Product Version 2 Release 3.0. The target data set is allocated as ESDS.
- 2 Locked user catalog support requires MVS/XA Data Facility Product Version 2 Release 3.0. The LOCK status of the user catalog is ignored.

- 3 VSAM VRRDS data set support requires MVS/XA Data Facility Product Version 3 Release 3.0. The target data set is allocated as KSDS.

System Action: Processing continues. The return code is 4.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR359W (xxx)mmmmm(yy), UTILITY PROCESSING FOR DATA SET dsname COMPLETED WITH WARNINGS.

Explanation: The utility successfully processed the data set but issued warning messages during processing. These messages are not spooled to SYSPRINT unless the DFDSS parameter UTILMSG= YES is specified.

System Action: Processing continues for other data sets. Return code is 4.

Programmer Response: Specify DFDSS parameter UTILMSG= YES, and rerun the job to view utility messages.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, and 13.

ADR360I (xxx)mmmmm(yy), PROCESSING LOCKED USER CATALOG catname

Explanation: The corresponding VVR for the integrated catalog facility user catalog being processed had the lock indicator set on.

System Action: The function continues processing.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR361E (xxx)mmmmm(yy), REQUIRED LEVEL levelnum OF productname IS NOT ON SYSTEM. {FUNCTION|VOLUME volsr|DATA SET dsname|KEYWORD keywordname} NOT PROCESSED, rc.

Explanation: The indicated level of a system product is not installed, but it is required to process the particular DFDSS function, volume, or data set. The reason codes are:

- 1 CONVERT keyword with PDSE subkeyword requires MVS/DFP Version 3 Release 2 and MVS/SP Version 3 Release 1.0E.
- 3 SMS function keyword requires MVS DFP Version 3 Release 1.

System Action: The return code is 8. The function|volume|data set: keyword is not processed.

Programmer Response: Verify that the indicated level of the system product is installed and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, 25b, and 29.

ADR362T (xxx)mmmmm(yy), NO BUFFER AVAILABLE & NONE IN WRITE

Explanation: The DFDSS I/O modules cannot find a free buffer. There are outstanding requests for waits or checks.

System Action: The job ends. The return code is 12.

Programmer Response: Contact IBM for programming support.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 5, 13, and 29.

ADR363E (xxx)mmmmm(yy), SOURCE DATA SET dsname {IS EMPTY | HAS AN UNSUPPORTED DSORG}. IT WILL NOT BE COPIED TO PREALLOCATED TARGET DATA SET.

Explanation: The data set will not be copied to the preallocated target data set for one of the following reasons:

- It is empty.
- It has a BLKSIZE of 0 and the target volume is of unlike device.
- It has an unsupported DSORG and the target volume is of unlike device.

System Action: The data set was not copied. Processing continues. The return code is 8.

Programmer Response: To have the data set copied, do one of the following:

- For an empty data set, use ALLDATA or ALLEXCP to force copy of all allocated space in the data set. Make sure that the selected target volume is of like device and that the data set is not being reblocked.
- For a data set with a BLKSIZE = 0, make sure that the selected target volume is of like device.
- For a data set with an unsupported DSORG, make sure that the target volume is of like device. For a data set with an unsupported DSORG going to an unlike target device, specify the PROCESS keyword with the UNDEFINED subkeyword. The specified or chosen target device must have a track capacity equal to or greater than the source.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR364W (xxx)mmmmm(yy), TRACK CANNOT BE RESTORED DUE TO DATA CHECK DURING DUMP. CCHH cchh, DDNAME ddname

Explanation: An error was encountered on a DASD volume during the DUMP. Because CANCELERROR was not coded, a dummy record was dumped. This record is recognized during the RESTORE.

System Action: The indicated track is not restored. Processing continues for other tracks. The return code is 4.

Programmer Response: Use IEHATLAS to re-create the data.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR365E (xxx)mmmmm(yy), THE CATALOG catname SPECIFIED IN {RECATALOG | INCAT} IS NOT AVAILABLE. THE {COPY | DUMP | RESTORE CONVERTV} TASK IS TERMINATED.

Explanation: The catalog specified in the RECATALOG or INCAT parameter does not exist or is not available on the system.

System Action: Performance of the task ends. The return code is 8.

Programmer Response: Ensure that the catalog name is correct and available, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR366W (xxx)mmmmm(yy), TRACK cchh WRITTEN TO VOLUME volser WITH INVALID TRACK FORMAT

Explanation: The track identified by the cchh had invalid track format on the source volume during DUMP or COPY. It was restored or copied with the same condition to the indicated target volume. When this track is accessed on the target volume, the invalid-track-format condition will recur. Or, a BDAM data set that could not be accessed by relative block address (that is, it had a standard user label or its record format was not fixed or fixed block) was being copied from a smaller-capacity DASD to a larger-capacity DASD. A track of data from the smaller-capacity device could not fit on the track of the larger-capacity device. This condition can occur when the data set without fixed or fixed-block record format has a large maximum blocksize but the actual blocks are so small that the track of the larger-capacity device cannot contain all the smaller blocks because of the increased inner-block gap size of the larger-capacity device. ADR366W is issued for each track copied that will not fit on the track of the larger-capacity device, and the copy will continue. (You can stop the copying of such data sets by using the CANCELERROR keyword. Refer to the use of CANCELERROR in the *DFDSS Reference* and in Message ADR367E in this manual.)

System Action: Operation continues and the remaining tracks, if any, are restored. The return code is 4.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, and 29.

ADR367E (xxx)mmmmm(yy), {R0 COUNT FIELD|INVALID TRACK FORMAT} ERROR ON {INPUT|OUTPUT} VOLUME volser cchh opcode csw sense

Explanation: Either the cchh track had an invalid track format, or a search ID for R0 on this track failed.

System Action: If the R0 count field error is encountered on an input volume and CANCELERROR was specified, the function ends with a return code of 16; otherwise, processing continues and a return code of 4 is set.

If the invalid-track-format condition was encountered, all the data that DFDSS was able to read from this track is written to the output volume (COPY and DUMP), and the track is marked as having an invalid-track-format condition on the output tape (DUMP only). In this case, the return code is set to 8. Also, this message is issued instead of ADR366W when copying BDAM data sets when an invalid-track-format error occurs on a write to an output volume if the keyword CANCELERROR has been specified. The COPY of the data set receiving the invalid-track-format error is stopped, and the target data set is deleted. Because of the concurrent processing of the output buffers on the WRITES, as many as three invalid tracks might be written before processing of the data set can be ended (that is, three ADR367E messages might be issued before the copy can be ended). The COPY job will continue to process any subsequent data sets.

Programmer Response: If the function ends because of the R0 count field error, correct the R0 record using Device Support Facilities and rerun the job. For an invalid-track-format condition, the user needs to reconstruct the track in error after the RESTORE.

If CANCELERROR has been specified, the invalid track format track is erased from the target and the target data set is deleted during COPY or RESTORE processing.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR368E (xxx)mmmmm(yy), I/O ERROR ON DDNAME ddname WHILE (PROCESSING VTOC|ADVANCING TO NEXT TRAILER RECORD) xxxxxx

Explanation: An I/O error was encountered in processing the VTOC or in bypassing the file for a data set RESTORE. The xxxxxx is the SYNAD information passed by the access method routines.

System Action: If the operation was to process the VTOC, the restore ends and no further processing is performed. If the operation was to advance to the next trailer record, the restore continues because the I/O error occurred on a record that was not needed for this restore. The return code is 8.

Programmer Response: If the operation was to process the VTOC, use a different version of the backup copy. If the operation was to advance to the next trailer record, permanent errors will result if a future RESTORE is attempted to restore the record that had the I/O error.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 30.

ADR369D (xxx)mmmmm(yy), AUTHORIZE FOR (READ|WRITE) ACCESS (A CHECKPOINT/RESTART DATA SET|A RACF DATA SET ON NON-RACF SYSTEM|A VVDS|A VTOCIX|ICF VSAM DATA SET ON NON-ICF SYSTEM) ON volser,jobname,stepname, REPLY U OR T

Explanation: The requested authorization is for:

- A checkpoint/restart data set
- A RACF data set on a non-RACF system
- A VVDS or VTOCIX data set
- An integrated catalog facility-cataloged VSAM data set on a nonintegrated catalog facility system.

System Action: Processing continues according to the operation action. If the operator replies T, a full or tracks operation ends and, on a data set operation, the data set is not processed. The return code is 0. A reply of U authorizes the data set to be processed.

Programmer Response: Instruct the operator to authorize the processing via a reply of U, and rerun the job.

Operator Response: Respond R zz,U to cause DFDSS to continue processing the data set or volume. Respond R zz,T to end the function and cause processing to continue for the next data set or next control statement.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR370E (xxx)mmmmm(yy), INVALID SEQUENCE NUMBER ON DDNAME ddname LAST yyyyyyy NEXT zzzzzzz

Explanation: During a RESTORE, the sequence number of the record read (zzzzzzzz in hexadecimal) did not match the previous record processed (yyyyyyy in hexadecimal). If there is an end of volume involved, a tape might have been mounted in the wrong sequence.

System Action: Processing continues with the next control statement. If the output volume was updated, a return code of 16 is set; otherwise, a return code of 8 is set.

Programmer Response: Probable user error. Mount the proper restore tape, or use the correct dump data set.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, and 28. If tapes were not involved or tapes were mounted in the right sequence, take actions 5b, 29, and 35.

ADR371D (xxx)mmmmm(yy), I/O ERROR ON volser DURING AUTH CHECKING FOR (INPUT|OUTPUT) ACCESS, jobname, stepname, REPLY U OR T

Explanation: An I/O error was encountered while the VTOC was accessed for authorization checking.

System Action: Processing continues, depending on the response from the operator. A reply of T ends the function with a return code of 8.

Programmer Response: Instruct the operator to let DFDSS continue processing, and rerun the job.

Operator Response: If the user has authorization to access the volume, reply R zz,U to continue the function. Otherwise, reply R zz,T to end the function.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR372W (xxx)mmmmm(yy), ALIAS dsname NOT RESTORED FOR USER CATALOG catname, rc

Explanation: The user catalog alias was not restored for the restored user catalog. The reason code, rc, indicates why, as follows:

- 00 User catalog aliases are not restored when the user catalog is restored to a nonempty preallocated target.
- 04 The user catalog for this alias was not restored successfully.
- 08 A duplicate name exists in the master catalog for the alias.
- 12 A catalog error other than duplicate data set name occurred while the alias was being restored.

System Action: The user catalog alias is not restored. Processing continues with the next data set. The system return code is 4.

Programmer Response: The user catalog was restored. To define the alias, do the following according to the reason code:

- 00 Use the IDCAMS DEFINE ALIAS command to define the alias if it does not exist and is desired.
- 04 Other messages indicate why the user catalog was not restored. Correct the problem and rerun the restore.
- 08 If the duplicate name in the master catalog is an alias for the restored user catalog, no action is necessary. If the duplicate name is not an alias, correct the problem. If the alias is required, define it using the IDCAMS DEFINE ALIAS command.
- 12 Determine the error from message ADR497E that precedes this message and correct the problem. Then define the alias using the IDCAMS DEFINE ALIAS command.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, and 29.

ADR373E (xxx)mmmmm(yy), REQUESTED TRACK (cchh | (SOME)) NOT ON INPUT DDNAME ddname

Explanation: The track requested for the RESTORE is not on the input file. It was probably not dumped. If the cchh is not printed, then some of the requested tracks are not on the input volume.

System Action: The task ends. Processing continues with the next control statement. The return code is 8 if the task is a data set RESTORE or a full or tracks RESTORE that does not result in an unusable output volume. The return code is 16 if a full or tracks RESTORE results in an unusable output volume. Subsequent tracks are not restored.

Programmer Response: Either change the track range requested, or RESTORE from a different copy of the backup.
Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29. List the second and third records on the input file to determine the tracks that are on the input file. See the *Data Facility Data Set Services: Diagnosis Guide* under "Format of the DFDSS Dump Data Set" for descriptions of these records.

ADR374E (xxx)mmmmm(yy), UNABLE TO OPEN
DDNAME ddname, xx-yyy

Explanation: DFDSS was unable to OPEN the indicated data set for the indicated reason code (xx). The yyy is the return code passed by RDJFCB. The possible reason codes are:

- 8 RDJFCB failure.
- 10 Invalid parameters passed in JCL. The following are the probable reasons:
 - SYSIN/SYSOUT data set
 - DISP = SHR coded for the output data set
 - BLKSIZE less than 7892 bytes
 - DSORG other than PS
 - BUFNO specified.
- 12 OPEN failure.
- 14 Invalid file format passed in the JCL. One of the following is true:
 - RECFM is not FB or F
 - RECLEN is not 80
 - BLKSIZE is not a multiple of the record length.

System Action: The task ends. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the parameters in error, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 5b, 13, 20, 29, and 35.

ADR375E (xxx)mmmmm(yy), DATA SET dsname IS
PARTITIONED, BUT HAS NO DIRECTORY.
IT CAN ONLY BE {COPIED|RESTORED}
TO A LIKE DEVICE

Explanation: The data set is partitioned, but has no directory. It can be copied or restored only to a like device, but no like devices were available or usable.

System Action: Function fails for that data set. The return code is 8.

Programmer Response: Rerun the job, ensuring that a like device with sufficient space is available.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 14, 15, 20, and 29.

ADR376E (xxx)mmmmm(yy), UNABLE TO ACQUIRE
ADDITIONAL STORAGE FOR THE TASK

Explanation: Additional storage was required to continue the function, but it was not available.

System Action: The task ends. Processing continues with the next control statement. The return code is 8. An abend 80A might occur during DFDSS postprocessing.

Programmer Response: Provide adequate storage by increasing either the REGION size, the SIZE parameter, or both, or respecify the filtering criteria to reduce the number of data sets for the operation. Rerun the job. See the *Data Facility Data Set Services: User's Guide* under "Storage Requirements" for more information on storage estimates.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR377W (xxx)mmmmm(yy), COMMAND IS NOT
SUPPORTED FOR DATA SET dsname ON
VOLUME volser

Explanation: The COPY, DUMP, or data set RESTORE command does not support the data set organization of the indicated data set. The volume indicated is the logical volume during a RESTORE. The PRINT command does not support split-cylinder data sets or nonintegrated catalog facility VSAM data sets.

System Action: Processing continues for other data sets. The return code is 4.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25b, and 29.

ADR378I (xxx)mmmmm(yy), THE FOLLOWING DATA
SETS WERE SUCCESSFULLY PROCESSED
FROM VOLUME volser

Explanation: A list of data set names follows this message. The volume indicated is the logical volume for a RESTORE.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR379E (xxx)mmmmm(yy), {OBTAIN ERROR xxxx|NO
F2/F3 DSCB|NOT F2/F3 DSCB|ERROR IN F1
DSCB|ERROR IN F3 DSCB|TOO MANY
EXTENTS|NO EXTENTS} [AT CCHH cchh]
ON VOLUME volser FOR
{DSNAME|CLUSTERNAME name
COMPONENTNAME name}

Explanation: Errors were encountered when the extents for the data set were accessed. The possible errors are:

- OBTAIN failed while the extents for the data set are retrieved.
- The VTOC entry indicated that the data set had more than three extents, but the additional extent descriptions could not be located. The VTOC entry for an ISAM data set did not include the ISAM-specific information.
- The record retrieved from the VTOC was not of the expected type.
- The description for one of the first three extents had errors.
- The descriptions for extents beyond the third extent had errors.
- The number of extents was more than the number supported (16 for a non-VSAM data set and 123 for a VSAM data set).
- The extent information in the VTOC entry is null.

During a RESTORE, the volume serial is the logical volume.

System Action: For non-VSAM data sets, only the retrieved extents are processed. If the error occurred during a data set DUMP and the data set is restored during a subsequent data set RESTORE, only the successfully dumped extents are restored. If the error occurred on a VSAM data set, it is not processed. The return code is 8.

Programmer Response: Use IEHLIST to verify the DSCBs for the VTOC. If the VTOC does not contain errors, rerun the job to dump the data set. Use a prior backup copy of the dump to restore the data set if it occurs during a RESTORE.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25a, 25b, and 29.

ADR380E (xxx)mmmm(yy), DATA SET dsname NOT PROCESSED, (rc/rc-xxx)

Explanation: The data set was not copied, restored, converted, or released. The reason codes, rc, are:

- | | |
|---|---|
| <p>0 An unmovable/ABSTR/ISAM data set was not preallocated. Either the system does not support allocation of these data sets, or you are attempting physical restore of a nonpreallocated unmovable/ABSTR/ISAM data set.</p> <p>1 Replace was not specified for the unmovable data set.</p> <p>2 There were unmatched extents on input and output for an unmovable/ABSTR/ISAM data set. The extents of the source unmovable data set are printed.</p> <p>3-xxx DADSM was unable to allocate a data set. The 'xxx' indicates DADSM (SVC 32) return codes. See your <i>System-Data Administration</i> book or <i>MVS/370 DADSM Diagnosis Reference</i> or <i>MVS/370 Facility Product Version 2: Diagnosis Reference</i> for more information.</p> <p>3-004 The dsname of request already exists on this volume; initial allocation was not possible under the given name.</p> <p>3-008 No room was available in the VTOC.</p> <p>3-012 A permanent I/O error or an error returned by CVAF was encountered.</p> <p>3-016 Direct access to the absolute track was unavailable.</p> <p>3-020 The requested quantity was unavailable.</p> <p>3-024 The average record length is greater than 65535 bytes.</p> <p>3-048 The parameter list was invalid.</p> <p>3-052 The JFCB was invalid or a partial DSCB pointer was specified.</p> <p>3-072 The DOS VTOC cannot be converted to an OS VTOC.</p> <p>3-076 No space parameter was given for a new data set, or zero space was requested at absolute track zero.</p> <p>3-100 CYL and CONTIG requests conflict.</p> <p>3-104 The space subparameter was invalid.</p> <p>3-116 User labels were not supported.</p> <p>3-120 DSSIZE=0 and minimum allocation unit is greater than 0.</p> <p>3-124 DSSIZE is not a multiple of the minimum allocation unit.</p> <p>3-132 Space request must be ABSTR for a DOS volume.</p> <p>3-148 Overlapping extents were in the VTOC.</p> <p>3-152 Overlapping DOS split-cylinder extents were in the VTOC.</p> <p>3-156 DADSM allocation ended because of possible VTOC errors.</p> <p>3-164 Allocation ended because of DOS stacked pack format.</p> <p>3-168 RACDEF failed; data already was defined.</p> | <p>3-172 The user was not authorized to define the data set.</p> <p>3-176 The installation-wide exit rejected the request with a return code of 8.</p> <p>3-180 The installation-wide exit rejected the request with a return code of 4.</p> <p>3-184 RACF was defined with modeling specified; the model was not found.</p> <p>4 The authorization check failed.</p> <p>5 The block sizes of the target and source data sets do not match. This can be caused by attempting to move a data set created in an N*512 environment to a non-N*512 environment, or vice versa.</p> <p>6 The volume sequence number did not match.</p> <p>7 DSORG did not match.</p> <p>8 There was an error in the rename filter.</p> <p>9 The movable data set on the receiving volume is not the same size as the data set on the source volume. The data set is in use and cannot be scratched and reallocated.</p> <p>10 The movable data set on the receiving volume is smaller than the data set on the source volume. Failure occurred in the scratch; the data set cannot be scratched and reallocated.</p> <p>11-xx The VVDS calls failed. See return code 50 under message IDC3009I in the <i>System Messages</i> for 'xx.'</p> <p>12 The extents mapped by the VTOC and VVDS on DASD do not match the VVDS on the input logical volume.</p> <p>13-xx Retrieving the extents from the VTOC failed. xx is the obtain error code.</p> <p>14 Component names do not match. Either the component was redefined since the data set was backed up; or it is a multivolume data set, and the input and output volumes are different volumes of a multivolume data set.</p> <p>15 The number of components differs on input and output. Either the component was redefined since the data set was backed up; or it is a multivolume data set, and input and output volumes are different volumes of a multivolume data set.</p> <p>16 The beginning RBA for the components does not match on input and output. The input and output volumes are different volumes of a multivolume data set.</p> <p>17 Primary/secondary indicators do not match on input and output. The input and output volumes are different volumes of a multivolume data set.</p> <p>18 REPLACE or RENAME (conditional or unconditional) was not specified for the movable non-VSAM data set, or REPLACE was not specified for a VSAM data set.</p> <p>19 The extents mapped by the VTOC do not match the VVDS on the output volume.</p> <p>20-xxxx,yyyy The data set on the output volume is smaller than on the input logical volume. xxxx = data component size in tracks (hexadecimal). yyyy = index component size in tracks (hexadecimal).</p> <p>21 The VVDS, BCS, or VTOCIX cannot be restored to a different volume.</p> |
|---|---|

- 22 The location of where the unmovable data set is to be allocated is not free, and FORCE is not specified.
- 23 The indicated data set is a duplicate data set and exists on more than one input volume.
- 24 The preallocated target data set was defined with a larger control area size than the source data set.
- 25 DFDSS cannot restore data sets with a volume serial of *MIGRAT*.
- 26 CVOL cannot be replaced.
- 27 Input/output block cannot be built because one or more of the following conditions exists:
 - Ddname is not found.
 - Ddname is a SYSIN or SYSPRINT.
 - Device is not DASD.
 - Device is not supported.
 - Incorrect ranges are specified for DASD.
- 28 The catalog return code was unexpected.
- 29 No output volumes were available because one of the following conditions exists:
 - There are no volumes in the output volume list that fulfill the requirements of the target volumes (that is, same or like device type).
 - The data set is preallocated, and its attributes do not match those of the source data set. (The target data set is single volume and the source data set is multivolume, or vice versa.)
 - All volumes of a preallocated multivolume data set are not included in the output volume list.
 - An attempt is made to restore an unmovable data set while output volumes are specified without at least as much space as is required by each corresponding source volume.
 - The volumes on which the preallocated VSAM clusters reside do not match the volumes listed in the catalog entry for the cluster. This can happen, for example, if the preallocated cluster is cataloged outside the standard order of search or is not cataloged at all, while another cluster with the same name is cataloged in the standard order of search.
 - A VSAM data set was preallocated on the target volumes, but no catalog entry was found.
- 30 The volume serials of the preallocated, cataloged multivolume data set do not match those of the first volume.
- 31 The VSAM data set was preallocated on source volumes, but the catalog entry was not found.
- 32 Space on the output volumes was insufficient to restore the data set.
- 33 Allocation errors were unexpected.
- 34 Data set not processed. Catalog entry for an SMS-managed data set was found but there was no corresponding VTOC entry.
- 35 The VSAM define failed.
- 36 Processing of a VSAM user catalog is not supported when INDDNAME or INDYNAM is specified.
- 37 The new name specified with RENAME or RENAMEU already exists on the volume. The REPLACE parameter cannot be specified with the RENAME or RENAMEU keywords. REPLACE is ignored if it is specified.
- 38 The VVR for a component of the data set was found, but no corresponding DSCB was found.
- 39 RENAME/RENAMEU is not supported for unmovable data sets.
- 40 A nonexistent catalog is specified in the RECATALOG parameter, but the target data set is not SMS managed.
- 41 During physical restore, the target volume is not SMS managed, but one of the following situations occurs:
 - The ACS routine returned an SMS storage class.
 - The user specified BYPASSACS with STORCLAS.
 - The user specified BYPASSACS without STORCLAS, but the source data set was SMS managed.
- 42 The NOPACKING keyword was specified for the data set, but the preallocated target was on an unlike device.
- 43 During physical restore, the target volume is SMS managed, but one of the following situations occurs:
 - The ACS routine returned NULL storage class.
 - The user specified BYPASSACS with NULLSTORCLAS.
 - The user specified BYPASSACS without NULLSTORCLAS, but the source data set was not SMS managed.
 - The data set is not supported by SMS.
- 44 The source component is multivolume, and the preallocated target is either a different size or is now single volume.
- 45 The preallocated target is too small and is not the first part of a multivolume data set.
- 46 Allocation of resource failed because DFDSS used a utility to perform data movement. Processing ends.
- 50 DFDSS is unable to determine the number of blocks per track for the target device because of an invalid source data set blocksize.
- 52 The requested data set is a PDSE, but MVS/DFP Version 3 Release 2 or later is not available on the system.
- 54 The requested data set is a PDSE, but MVS SP Version 3 Release 1.0E or later is not available on the system.
- 56 The requested data set is a PDSE but SMSX is not currently available and/or the PDSE cannot be allocated and accessed at this time.
- 58 The requested data set is a PDSE, but the target volume selected is not SMS-managed. A PDSE can reside only on SMS-managed volumes.
- 60 The requested data set is a PDSE but there is a preallocated target data set that is not a PDSE.
- 70 Dynamic allocation failed for DFDSS export data set.
- 71 The SMS indicator in the VTOC entry is off, stating that the data set is not SMS, but the volume is SMS managed.

System Action: The return code is 8. The data set is not processed.

Programmer Response: Take the following action, depending on the reason code (rc):

- 0 Preallocate the data set with the size and location of the extents to match those on the source volume, or specify FORCE. Rerun the job.
- 1 Specify REPLACE and rerun the job.
- 2 Copy/Restore of unmovable/ABSTR/ISAM data sets is allowed only if the size and location of the extents match and the data set is preallocated. Initialize a new DASD volume with the VTOC extent matching the input logical volume, do a TRACKS COPY/RESTORE to copy the VTOC and TRACK 0, and do a data set COPY/RESTORE of the failing data set. Use the appropriate utility for the type of data set to move the data set from the scratch volume to the failing volume.
- 3 Take appropriate action according to the DADSM return code.
- 4 Acquire adequate authorization (RACF or password). If it is a checkpoint/restart or system data set, instruct the operator to allow the function to continue. You can also change the user authorization exit routine to allow access to the data set. Rerun the job.
- 5 Copy or restore the data set in the same environment in which it was created.
- 6 For a multivolume non-VSAM data set, the volume sequence number must match. For a VSAM data set, the beginning RBA of the data set on the volume must match.
- 7 If it is the same data set, use AMASPZAP to modify the DSORG field in VTOC entry. Rerun the job.
- 8 Correct the RENAME filter specification. Rerun the job.
- 9 Rerun the job when the data set is not in use.
- 10 Scratch the data set and rerun the job.
- 11 See return code 50 under message IDC3009 in *System Messages*.
- 12 Use a different backup copy to restore the data set.
- 13 Contact your system programmer.
- 14 If it is not a multivolume data set, delete the data set on DASD and resubmit the job. If it is a multivolume data set and a RESTORE in which the data set is not restored from the correct input logical volume, resubmit the job to restore from the proper logical volume. If it is a COPY, pass the proper input and/or output volumes and rerun the job.
- 15 See reason code number 14.
- 16 See reason code number 14.
- 17 See reason code number 14.
- 18 Specify REPLACE, RENAME, or RENAME-UNCONDITIONAL (RESTORE only) if it is a non-VSAM data set. Specify REPLACE if it is a VSAM data set.
- 19 Run the DIAGNOSE command of Access Method Services to determine the error.
- 20 Delete the data set and redefine it.
- 21 For non-SMS-managed data sets, restore to the correct DASD volume, or use the LVOL parameter to restore from the correct logical volume. For an SMS-managed user catalog, specify the correct DASD volume in the OUTDD/OUTDY parameter, and ensure that the volume is in a Storage Class with the guaranteed space attribute.
- 22 Specify FORCE or free the corresponding location on the target volume, and rerun the job.
- 23 If it is a multivolume data set, correct the last volume indicator and the volume sequence number fields in the VTOC, or catalog the data set in the proper sequence in the appropriate catalog. Rerun the job.
- 24 Specify RENAME or RENAMEUNCONDITIONAL or delete the preallocated target data set, then rerun the job.
- 25 Recall the data set and retry the restore operation.
- 26 Delete the CVOL from the target volume, then rerun the job.
- 27 Take one or more of the appropriate actions listed below, then rerun the job:
 - Ensure that a DDNAME statement is present in job step.
 - Ensure that the ddname is not that of a SYSIN or SYSPRINT.
 - Ensure that the device specified is a DASD.
 - Ensure that the device is supported on the system.
 - Ensure that the correct ranges are specified for DASD.
- 28 Refer to message ADR497E, printed before this message, and take the appropriate action.
- 29 Take one of the actions listed below; then rerun the job:
 - Supply a new choice of output volumes.
 - Ensure that the attributes of the source and the preallocated target match.
 - Ensure that the data set is cataloged and the catalog volumes and output volumes on which it is preallocated match.
 - Restore as a movable data set by specifying the FORCE keyword.
 - Supply the name of the catalog in which the preallocated cluster resides, either by using JOBCAT/STPCAT or the RECATALOG keyword. If the preallocated target cluster is not cataloged in any catalog, then either catalog it or delete it.
 - Ensure that the preallocated target data set is cataloged in the standard order of search.
- 30 Ensure that the correct target volumes for the multivolume data set are specified in the output DD statements, then rerun the job.
- 31 Ensure that a STEPCAT DD or a JOBCAT DD statement exists for the catalog with the VSAM data set on it. If a STEPCAT DD or JOBCAT DD statement already exists, ensure that it points to the correct catalog, then rerun the job.
- 32 Provide a different choice of output volumes, then rerun the job.
- 33 Provide a different choice of output volumes, then rerun the job.
- 34 Delete the catalog entry for the failing data set. Full volume dump/restore does not allocate data sets. However, it does restore all of the data tracks as well as the VTOC. Thus, full volume restore can be used to restore cataloged SMS-managed data sets with no corresponding VTOC entries.

- 35 Choose a different name for the target data set or specify REPLACE, then rerun the job.
- 36 Ensure that the catalog to be moved meets the INCLUDE, EXCLUDE, or BY filtering criteria, then rerun the job.
- 37 Ensure that the new name specified with RENAME or RENAMEU does not already exist on the volume. If the new name already exists, it must be deleted or another new name must be specified before retrying the command. REPLACE is ignored if RENAME or RENAMEU is used.
- 38 Delete the VVR for the failing component, and rerun the job.
- 39 Specify REPLACE to replace the unmovable data set under the old name, and rerun the job.
- 40 Ensure that the catalog name is correct and available if the target data set is not SMS-managed, and rerun the job.
- 41 Specify either an SMS target or a NULLSTORCLAS.
- 42 Delete and reallocate the target partitioned data set on a like device, or rerun the job without specifying NOPACKING for the data set.
- 43 Specify a non-SMS target.
- 44 Delete the data set on DASD, and resubmit the job.
- 45 Delete the data set on DASD, and resubmit the job.
- 46 Rerun the job when the data set is not in use.
- 50 Ensure that the source data set has a valid blocksize or preallocate a target data set with a valid blocksize. Specify the REPLACE keyword.
- 52 Process the PDSE using a system that has MVS/DFP Version 3 Release 2 or greater installed or do not select the PDSE.
- 54 Process the PDSE using a system that has MVS/SP Version 3 Release 1.0E or greater installed or do not select the PDSE.
- 56 Ensure that SMSX is available and that the PDSE can be allocated and accessed on the system. Rerun the job.
- 58 Ensure that the target volume for the PDSE is SMS-managed or do not select the PDSE.
- 60 Process the PDSE using the RENAME/RENAMEUNCONDITIONAL keywords, or rename the preallocated target data set, or delete the preallocated target data set.
- 70 Determine the condition that caused the allocation to fail (for example, authorization failure, and so on), correct the error condition, and rerun the job. To bypass the problem, the user must define a generic profile with 'HLQ.*'
- 71 Correct the VTOC entry and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 25b. Take actions 29 and 35 on 3-43, 3-52, 3-76, 3-100, 3-104, 3-116, 3-120, and 3-124. On a RESTORE, dump the VTOC track records on the input that are at the beginning of the file.

ADR381E (xxx)mmmm(yy), ALL OUTPUT VOLUMES INACCESSIBLE. DATA SETS NOT PROCESSED

Explanation: The OUTDD subparameter is missing, or the output volume cannot be opened for a data set COPY or RESTORE. Message ADR307E might precede this message.

System Action: The task ends. The return code is 8.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, and 29.

ADR382W (xxx)mmmm(yy), THE FOLLOWING DATA SETS WERE NOT PROCESSED FROM VOLUME volser DUE TO ERRORS

Explanation: The data sets in the list were not processed.

Other messages that precede this message identify the error.

System Action: The return code is 4.

Programmer Response: Correct the problem according to preceding messages.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR383W (xxx)mmmm(yy), DATA SET dsname NOT SELECTED

Explanation: The data set was not selected. This condition can occur for several reasons including:

- The fully qualified data set name did not pass the EXCLUDE, or BY filtering criteria.
- The data set name was specified more than once in the INCLUDE list.
- Logical input volumes were specified during DUMP or COPY and DFDSS was unable to locate the catalog in which the VSAM data set resides.
- A VSAM data or index component name was incorrectly specified instead of the cluster name. DFDSS only processes VSAM data sets at the cluster name level.
- The data set has been migrated (*volser = migrat*).

System Action: The data set is not processed. The return code is 4.

Programmer Response: Rerun the job after doing one of the following:

- Use the proper filtering criteria.
- Do not duplicate data set names in the INCLUDE list.
- Ensure any VSAM data sets to be processed are cataloged in the standard order of search.
- Ensure that VSAM cluster names were specified in the filtering criteria.
- If the data set you want is not cataloged, use the correct volume serial number. If you want to operate on the migrated data set, issue a DFHSM RECALL command for the data set.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25b, and 29. If it is a RESTORE, print the VTOC track records on the dump tape.

ADR384E (xxx)mmmmm(yy), UNCATALOG FAILED FOR data set ON volser, (rc|rc-xx)

Explanation: The UNCATALOG operation failed for the indicated data set. The return code from the system uncatolog operation is given in rc and the reason code, if any, in xx.
System Action: Processing continues. The return code is 8.
Programmer Response: See the *Catalog Administration Guide* under "Using Catalog Management Macro Instructions for OS CVOLs" for an explanation of return and reason codes, and take the appropriate action.
Operator Response: None.
Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR385E (xxx)mmmmm(yy), CATALOG FAILED FOR data set ON volser, (rc|rc-xx)

Explanation: The CATALOG operation failed for the indicated data set. The return code from the system catalog operation is given in rc and the reason code, if any, in xx.
System Action: Processing continues. The return code is 8.
Programmer Response: See the *Catalog Administration Guide* under "Using Catalog Management Macro Instructions for OS CVOLs" for an explanation of return and reason codes, and take the appropriate action.
Operator Response: None.
Problem Determination: Table I, 1, 2, 3, 4, 13, 20, and 29.

ADR386W (xxx)mmmmm(yy), UNIT NAME xxxxxx OF LOGICAL VOLUME volser DOES NOT MATCH ANY OUTPUT VOLUME DEVICE TYPE

Explanation: None of the output DASD volumes matched the device type of the source volume on the input file. The xx is the esoteric unit name. This message is also issued if you try to restore to an unlike device from a physical data set dump tape.
System Action: Processing continues with the next source volume, if any, on the input file. The return code is 4.
Programmer Response: Supply the matching output DASD volume, and rerun the job. If you want to restore to an unlike device, use a logical data set dump tape.
Operator Response: None.
Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR387E (xxx)mmmmm(yy), THE FOLLOWING LOGICAL VOLUMES DID NOT PROCESS FROM DUMP FILE:

Explanation: The user requested a data set RESTORE from the listed logical volumes from the input file, but the logical volumes are not on the input file. A list of volume serials follows the message.
System Action: The return code is 8.
Programmer Response: If no source volume was processed, use the correct RESTORE file or the correct source volume serial numbers, and rerun the job.
Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR388W (xxx)mmmmm(yy), DATA SET dsname NOT ON ANY VOLUME.

Explanation: The requested data sets are not on the source volumes on the input file. Either the wrong input file was used, an incorrect data set name was passed, or all the VSAM data sets or all the non-VSAM data sets were rejected because of the DSORG filtering criteria. This message is printed if fully qualified names are passed and there are no allocated data sets by that name on those volumes.
System Action: The return code is 4.

Programmer Response: Correct the error, and rerun the job.
Operator Response: None.
Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29. Use INCL(**) and the NORUN keyword to list all the data sets on the volume.

ADR389E (xxx)mmmmm(yy), INVALID INPUT ON DDNAME ddname, (INPUT DATA SET NOT PRODUCED BY DFDSS | DATA SET RESTORE NOT FEASIBLE FROM TRACKS DUMP | VSAM DATA SET RESTORE NOT FEASIBLE FROM RELEASE 1 DUMP | ONLY VSAM DATA SETS ON INPUT | ONLY NONVSAM DATA SETS ON INPUT | FULL RESTORE NOT FEASIBLE FROM TRACKS OR DATA SET DUMP | EXPECTED INPUT RECORD NOT FOUND | INPUT VOLUMES MOUNTED OFF SEQUENCE)

Explanation: Either the input is invalid for the type of RESTORE desired or the input was not produced by DFDSS. (For example, a dump tape was created using a data set DUMP and you tried to restore it via a full RESTORE.) If unlabeled tapes were used, you might have mounted them in the wrong sequence.
System Action: The task ends. Processing continues with the next control statement. The return code is 8.
Programmer Response: Take the appropriate action, depending on the error, and rerun the job.
Operator Response: None.
Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR390I (xxx)mmmmm(yy), DATA SET dsname WAS SCRATCHED FROM volser BECAUSE OF UNMATCHED SIZE. IT WILL BE REALLOCATED.

Explanation: The movable data set's SIZE (in tracks) on the DASD volume was not equal to the data set SIZE on the tape. Because REPLACE was specified instead of RENAME or RENAMEUNCONDITIONAL (on RESTORE), the data set was scratched to allocate the extents mapped in the input logical volume.
System Action: Operation continues. If allocation is feasible on this volume, message ADR396I follows this message. If allocation is not feasible on this volume, message ADR380E and/or message ADR472E follows this message.
Programmer Response: If reallocation fails, preallocate the data set with adequate space, and rerun the job.
Operator Response: None.
Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR391E (xxx)mmmmm(yy), RESTORE OF dsname REQUIRES OUTPUT VOLUMES

Explanation: The named data set was not cataloged when dumped. Because neither the dump catalog information nor the original source volumes are available for allocation, the output volumes must be specified to restore the data set.
System Action: Processing ends for the data set. The return code is 8.
Programmer Response: Specify at least one output volume using OUTDD or OUTDYNAM, and rerun the restore.
Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, and 29.

ADR392E (xxx)mmmmm(yy), dsname EXISTS ON volser WITH NEWNAME newname

Explanation: Allocation of the data set using the name generated from the RENAME parameter was attempted, but a data set with the new name exists on the volume.

System Action: Message ADR380E might follow this message. The return code is 8.

Programmer Response: Either scratch the existing data set with the name created from the RENAME parameter, or use another name in the RENAME parameter to generate a data set name that does not exist on the volume. Rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25a, and 29.

ADR393I (xxx)mmmmm(yy), ALIAS dsname RESTORED FOR USER CATALOG catname

Explanation: The user catalog alias was restored for the catalog.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR394W (xxx)mmmmm(yy), UNABLE TO ALLOCATE dsname ON volser WITH NEWNAME dsname

Explanation: Allocation of the data set using the name generated from the RENAME parameter was attempted, but the allocation failed.

System Action: This message is preceded by another error message, indicating the reason for the failure. The return code is 4.

Programmer Response: Appropriate responses are listed under the preceding message.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR395I (xxx)mmmmm(yy), DATA SET dsname ALLOCATE (AS A PDS) WITH NEWNAME dsname [, IN CATALOG catname], ON VOLUME(S): (volser | volser list)

Explanation: The data set has been allocated on the volume or volumes listed with the NEWNAME generated from the RENAME parameter. If the data set is a VSAM cluster, the catalog name may be listed as well. If the target data set of a PDSE is allocated as a PDS, the new target type is listed as well, and the data set is converted to the new target type.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR396I (xxx)mmmmm(yy), DATA SET (dsname | CLUSTER clusname COMPONENT compname) ALLOCATED [AS A PDS][, IN CATALOG catname] , ON VOLUME(S): (volser | volser list)

Explanation: The data set has been allocated on the volume or volumes listed. If the data set is a VSAM cluster, the catalog name may be listed as well. If the target data set of a PDSE is allocated as a PDS, the new target type is listed as well, and the data set is converted to the new target type.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR397E (xxx)mmmmm(yy), DATA SET dsname NOT (COPIED | RESTORED). DATA BLOCKSIZE OF blksize EXCEEDS TARGET DEVICE TRACK CAPACITY OF trksize.

Explanation: DFDSS allocated or located the target data set on a DASD device with the indicated track capacity. This was done because the block size of the source data set was less than the track capacity of the target device. However, during processing, DFDSS encountered a data block from the source data set that was larger than the track capacity of the target device. The data block cannot be placed on the target device, so the DFDSS function fails for the data set.

This situation can occur only when the track capacity of the target device is less than that of the source device. DFDSS allocates to such a device only when target volumes of same, like, or larger track capacity are not specified or are specified but do not have space available.

System Action: The specified data set is not processed. If DFDSS allocated the target data set, it is deleted. The return code is 8.

Programmer Response: Perform one of the following actions, and rerun the job:

- Provide DFDSS with a target DASD device with a track capacity large enough to contain the block size specified in the message, by using OUTDD or OUTDYNAM. Specify only target volumes that have a track capacity larger than the block size specified in the message. This includes volumes of the same or a like device type as the source and any volumes of a device type with a large track capacity. You must also eliminate the REPLACE keyword if a preallocated target is located on a device with a smaller track capacity than the block size indicated in the message.
- Preallocate a target data set on a volume with a track capacity large enough to contain the block size specified in the message. Specify the volume to DFDSS by using OUTDD or OUTDYNAM and use the REPLACE keyword to force DFDSS to use the preallocated target. The target device should meet the conditions specified in the preceding solution.
- Specify the REBLOCK keyword. If the source data set can be reblocked and the source data set logical record length is less than the target track capacity, DFDSS reblocks the data set as it is copied or restored. Certain data sets cannot be reblocked, so one of the solutions described above must be used. Whether or not the data set can be reblocked can be tested by specifying REBLOCK and rerunning. If the function fails again, use one of the preceding solutions.
- If you were performing a COPY, you can update the source VTOC entry to reflect the block size specified in the message by using AMASPZAP if you are authorized to use it. When the job is rerun, DFDSS will select a usable target volume based on the updated block size.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25b, 25c, and 29.

**ADR398W (xxx)mmmmm(yy), DATA SET dsname
BLOCKSIZE OF blocksize IS INCORRECT.
LARGEST BLOCKSIZE IS blocksize**

Explanation: While performing a COPY or RESTORE of the data set, DFDSS encountered at least one data block whose size exceeded the maximum block size specified in the VTOC. The data set is copied or restored with the block size intact. The largest block size found is indicated.

System Action: Processing continues for the specified data set. The return code is 4.

Programmer Response: Take appropriate actions to correct this situation.

Operator Response: None.

Problem Determination: None.

**ADR401W UNEXPECTED RETURN CODE xxxx
REASON CODE yyyy FROM UCBLLOOK
MACRO. VOLUME ON UCB zzzz IS MADE
UNAVAILABLE.**

Explanation: UCBLLOOK macro returned unexpected return code while DFDSS was trying to check for a duplicate VOLSER. Return and reason code are in decimal. zzzz represents the device number.

System Action: Processing continues. Volume on UCB zzzz is made unavailable. The return code is 4.

Programmer Response: This message is not normally expected and indicates a programming or system problem. Ensure that no duplicate VOLSER exists.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 5, 13, 20, 29, and 35.

**ADR402E (xxx)mmmmm(yy), AUTHORIZATION CHECK
FAILED FOR dsname, [ON VOLUME volser,]
rc:IN CATALOG catname)**

Explanation: The authorization check failed for the indicated reason code, rc. The possible reason codes are:

- 1 The operator specified *T* to message ADR369D, which denies access to one of the following:
 - A checkpoint/restart data set
 - A RACF data set on a non-RACF system
 - A VVDS or VTOCIX data set or an integrated catalog facility-cataloged VSAM data set on a nonintegrated catalog facility system, or

Input/output errors were encountered on the VTOC or VVDS.

- 2 RACF UPDATE authorization failed for the data set name.
 - 3 RACF ALTER authorization failed for the data set name.
 - 4 RACF READ authorization failed for the data set name.
 - 5 The user authorization exit rejected the volser.
 - 6 The user authorization exit rejected the data set name.
 - 7 The password was not supplied for the data set name.
 - 8 READ password was specified, but WRITE password was required.
 - 9 The password was incorrect for the data set name.
 - 10 An error occurred while the PASSWORD data set was being accessed.
 - 11 OPEN failed while verifying the password for the VSAM data set.
-

12 Duplicate or missing cluster entries for the data set in the VVDS.

13 The expiration date did not pass for the data set name.

14-xxx-yyy The authorization check failed for the VSAM data set. The xxx and yyy are the return codes and reason codes from catalog services. See return code/reason code under message IDC3009I in *System Messages* for an explanation.

15 The return code from the user authorization exit is invalid.

16 No access to the volume was passed.

32 One of the following conditions occurred:

- Permanent error while reading the VTOC, VVDS, or the password data set and the operator denied permission to continue
- Unable to get the storage required
- Permanent error on the password data set.

System Action: Other messages indicate the actions taken for the various conditions. The return code is 8.

Programmer Response: For all reason codes except code 14, acquire proper authorization for the data set, and rerun the job. (For code 2, 3, or 4, authorization may have failed because of other attributes, for example, SECLEVEL.) For code 14, see return code/reason code under message IDC3009I in the *System Messages* manual.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 5b, 13, 20, and 29.

**ADR404E (xxx)mmmmm(yy), COMMAND NOT
ALLOWED ON AN OPEN CATALOG,
CATALOG catname**

Explanation: A COPY or RESTORE of a catalog is not allowed on an open Integrated Catalog Facility catalog.

System Action: The data set is not processed. The return code is 8.

Programmer Response: Run the job when the catalog is not active on the system.

Operator Response: None.

Problem Determination: If the catalog is not open, take a SYSABEND on message ADR404E. Table I, items 1, 3, 4, 5, 13, 20, 29, and 35.

**ADR405E (xxx)mmmmm(yy), DYNAMIC ALLOCATION
OF VOLUME volser FAILED. ERROR CODE
eeee. INFORMATION CODE iiiii**

Explanation: DFDSS attempted to dynamically allocate the indicated volume and failed. The error and information codes are returned by dynamic allocation and listed in the *System Programming Library: Application Development Guide*. Error code X'049C' can result if: (1) the volume is online and not mounted, and (2) the unit name was not passed because DFDSS tried to allocate the volume as a virtual volume after allocation as a real volume had failed. An undocumented error code can be displayed if an installation-wide validation exit is used to deny allocations for dynamic allocation but fails to get the reason code returned by dynamic allocation.

System Action: The task or volume is bypassed. The return code is 8.

Programmer Response: Correct the problem indicated by the error and information codes listed in the *System Programming Library: Application Development Guide*.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR406E (xxx)mmmm(yy), dsn CANNOT BE RESTORED FROM ddname

Explanation: The specified data set, dsn, cannot be restored from the logical dump data set assigned by ddname. This situation occurs when you attempt to restore one of the following:

- An ISAM data set
- A CVOL to an unlike device
- A BDAM data set to an unlike device with a target track capacity smaller than the source track capacity without specifying the FORCE keyword.

System Action: The indicated data set is not restored. The return code is 8.

Programmer Response: If the data set is an ISAM data set or a CVOL, rerun the job, specifying a like device type as the target of RESTORE. If the data set is a BDAM data set, specify the FORCE keyword and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR407E (xxx)mmmm(yy), UNABLE TO FIND MODEL VOLSER FOR MODEL ENTITY {model entity name | vsam catalog name}

Explanation: The model entity (MENTITY) supplied was not found in the standard catalog search order or is a VSAM data set cataloged in a nonintegrated catalog facility VSAM catalog.

System Action: The COPY or RESTORE is ended. The return code is 8.

Programmer Response: Verify that the spelling is correct for the MENTITY supplied and that it is cataloged in one of the catalogs used in the standard catalog search order; or ensure that the MENTITY is not cataloged in a VSAM catalog. Then rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR408E (xxx)mmmm(yy), LOCATE FAILED FOR CATALOG catname

Explanation: An error occurred in attempting to locate the volsr of the catalog name containing the VSAM model entity (MENTITY) entry.

System Action: The COPY or RESTORE is ended. The return code is 8.

Programmer Response: Use IDCAMS to define the catalog specified in this message, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR409E (xxx)mmmm(yy), CLUSTER clusname HAS PRE DFP310 VVRS WHICH ARE INCOMPATIBLE WITH SMS

Explanation: Cluster was created by a version of DFP prior to DFP 3.1.0. Therefore, the VVRS for the cluster do not contain all the fields necessary for SMS, even though an SMS subcell was added to the VVRS.

System Action: The volume is left in its initial status. The return code is 8.

Programmer Response: REPRO the data out of the data set, delete the data set, then recreate the data set on DFP version 3.1.0 or later system. If you have a large number of these data sets, contact your IBM service representative.

Operator Response: None.

Problem Determination: None.

ADR411W (xxx)mmmm(yy), DATA SET {dsname | clustname COMPONENT componentname} [IN CATALOG catname] ON VOLUME volsr WAS NOT SERIALIZED ON REQUEST

Explanation: An ENQ(EXCLUSIVE), ENQ(SHARE), or dynamic allocation with DISP=OLD or SHR failed after a specified (or defaulted) number of retries. TOL(ENQF) was specified by the installation-wide exit or by the user.

System Action: If you were doing a COPY and DFDSS is moving the data itself, the data set is processed without the enqueue, and the return code is 4. If you were doing a COPY and DFDSS invokes a utility to move the data, the data set is not processed further, and the return code is 8. If you were doing a data set DUMP, the data set is processed without the enqueue, and the return code is 4. If you specified RESET on the data set dump, the data set is processed, but the data set changed indicator is not reset in the data set's VTOC entry.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR412E (xxx)mmmm(yy), DATA SET {dsname|clustname COMPONENT componentname} [IN CATALOG catname] ON VOLUME volsr FAILED SERIALIZATION

Explanation: An ENQ(EXCLUSIVE), ENQ(SHARE), or dynamic allocation with DISP=OLD or SHR failed after a specified (or defaulted) number of retries. TOL(ENQF) was not specified by the user.

System Action: The data set is not processed. The return code is 8.

Programmer Response: Rerun the job when the data set is not in use by other users. You can increase the WAIT parameters, and if you are not using the COMPRESS command to compress a PDS, you can specify TOL(ENQF).

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR413E (xxx)mmmm(yy), DATA SET dsname ON VOLUME volsr WAS NOT (DELETED | UNALLOCATED), {xx|xx-yyyy : zzzz}

Explanation: The data set cannot be scratched, uncataloged, or unallocated via dynamic allocation. The error codes (xx) are:

- 0 dynamic allocation installation-wide validation routine denied the request. The reason code is zzzz.
- 1 dynamic allocation information error code is yyyy. The reason code is zzzz. The codes are explained in the *System Programming Library: Application Development Guide*, GC28-1852.
- 2 IEFAB4C3 was called to release data set integrity but it failed.

System Action: The return code is 8.

Programmer Response: Scratch the data set by other means.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR414E (xxx)mmmm(yy), DATA SET dsname ON VOLUME volsr NOT SCRATCHED, {xx|xx-yy,zzz}

Explanation: The data set cannot be scratched. The error codes (xx) are:

- 0 DEQ on the data set failed.
- 1 The initiator enqueued the data set.

- 2 The scratch failed. The yy is the return code and/or status code. The zzz is the reason code from DADSM SCRATCH, as described in the *System-Data Administration*.

System Action: The return code is 8.

Programmer Response: Scratch the data set by other means.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR415W (xxx)-mmmmm(yy), NO DATA SETS WERE COPIED, DUMPED, OR RESTORED FROM (ANY VOLUME | VOLUME volser | LOGICALVOLUME volser)

Explanation: A data set COPY, data set DUMP, or data set RESTORE did not select any data sets. One of the following applies:

- No data sets matched the filtering criteria.
- Data sets matched the filtering criteria but failed serialization or authorization and were not processed.
- Permanent I/O errors occurred on the output tape during the DUMP process.

System Action: If I/O errors occurred on output during DUMP, the return code is set to 16; otherwise, it is set to 4. If there were no I/O errors, and if the failure was on one volume, other volumes will be processed.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25b, and 29. If it was a RESTORE, print the VTOC tracks on the dump tape.

ADR416E (xxx)-mmmmm(yy), DATA SET dsname [IN CATALOG catname] WAS NOT DELETED, rc

Explanation: The data set was not deleted for one of the following reasons:

- 0 It is a VSAM data set. DFDSS does not support DELETE for VSAM.
- 1 I/O or other errors, such as authorization or enqueue failure, occurred on the data set, or no catalog entry was found for the data set.
- 2 Either the data set is not a single-volume data set, or it cannot be determined if the data set is single volume or multivolume and it is not presently cataloged.
- 3 The data set was not enqueued by DFDSS.
- 4 The data set was not processed. (It is probably an empty data set.)
- 5 The data set has not expired by the expiration date. If this return code is received for a data set COPY, the dsname appearing in the message is a temporary name that DFDSS generated to properly name the data set on the target volume. You must now delete the data set with the temporary name from the source volume.

System Action: The data set is not deleted. The return code is 8.

Programmer Response: Delete the data set by other means. For reason code 5, specify PURGE to delete the unexpired data set.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25a, and 29.

ADR417W (xxx)-mmmmm(yy), COPY/RESTORE OF DATA SET dsname IS INCOMPLETE, rc

Explanation: The data set was copied or restored, but errors occurred during processing; or the data tracks were not copied or restored because the VTOC entry could not be reset; or the RACF indicator could not be set for the data set. The reason code (rc) is as follows:

- 0 The protected data set was copied or restored, but definition of the RACF discrete profile failed.
- 1 The data set was copied or restored, but the VTOC entry cannot be retrieved for updating.
- 2 The data set was copied or restored but errors occurred while the VTOC entry was being updated.
- 3 For physical data set RESTORE, the data set is restored but I/O errors occurred on one or more tracks of the data set (resulting from errors during dump). For logical data set RESTORE and data set COPY, no further data is copied or restored. If the data set was not preallocated, it is deleted.
- 4 A RACF-protected data set was copied or restored to a system that does not support RACF.
- 5 The data set was not copied or restored because the VTOC entry cannot be retrieved to set the RACF indicator or be updated for the data set.
- 6 The data set was not copied or restored because errors occurred while the VTOC entry was either being rewritten to reset the RACF indicator or being updated for the data set.
- 7 The target VSAM data set cannot be protected with a discrete RACF profile because it was not renamed, the target catalog resides on the same volume as the source catalog, and a discrete RACF profile already exists for the source data set.
- 8 The data set was not copied because a track overflow record was found in the data set.

System Action: Processing continues. The return code is 4.

Programmer Response: Take the following action, depending on the reason code (rc):

- 0 Define the data set to RACF.
 - 1 Use AMASPZAP to reset the relevant fields in the VTOC entry for the data set, or scratch the data set and rerun the COPY or RESTORE.
 - 2 See reason code 1.
 - 3 On input errors, restore the data set from a backup copy. On output errors, take corrective action recommended in your site, and rerun the job.
 - 4 Password-protect the data set.
 - 5 Use AMASPZAP to reset the RACF indicator, and rerun the job.
 - 6 Correct the problem for the system I/O error message that precedes this message, and rerun the job.
 - 7 The data set does not have the RACF indicator set. If the source data set was deleted, define the target data set to RACF. If the source data set was not deleted, you can define the target data set to RACF with a generic profile. However, in an always call (RACF 1.5 or greater) environment, the data set is protected by the discrete RACF profile.
 - 8 Use an appropriate system utility to copy the data set.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25b, and 29.

ADR418I (xxx)-mmmm(yy), THE FOLLOWING COMPONENTS FOR CLUSTER clustname ON volser MAY HAVE TO BE CATALOGED IN CATALOG catname

Explanation: The cluster that did not exist on the volume was copied or restored to the volume. VTOC and VVDS entries were created on the volume for the cluster. If the indicated catalog does not have entries for the cluster, you must recatalog the cluster before accessing the data set.

System Action: None.

Programmer Response: Use the Access Method Services DEFINE RECATALOG command to recatalog the components and cluster in the indicated catalog before accessing the data set.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25b, and 29.

ADR419W (xxx)-mmmm(yy), CLUSTER clustname IN CATALOG catname ON DASD WITH {INHIBIT UPDATE | OPEN} INDICATOR WILL BE COPIED OR RESTORED

Explanation: A fully qualified name was passed for the data set. It is copied or restored although it has the indicated condition. The inhibit-update indicator means that the data set is a read-only copy of a data set. Open-for-update means that the data set was in the OPEN status. Either it was not closed correctly or it was in use by another job on the same processing unit.

System Action: If the data set had the inhibit-update indicator in the VVR, this indicator is left on. If the data set had the OPEN indicator, this indicator is reset. If the input copy had the OPEN indicator, this is carried over to output DASD. The return code is 4.

Programmer Response: If message ADR411W accompanies this message, the data set was in use when the COPY or RESTORE was done. The data set on the target volume may therefore be invalid. Reprocess the data set without specifying TOL(ENQF).

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25b, and 29.

ADR420E (xxx)-mmmm(yy), {VVDS|VTOCIX} DATA SET CANNOT BE RESTORED WITH {OTHER VSAM | ANY OTHER} DATA SET(S)

Explanation: Restore of the VVDS data set is not allowed with restore of other VSAM data sets from the same logical volume. Restore of the VTOCIX data set is not allowed with restore of other data sets from the same logical volume.

System Action: The volume is not processed. The return code is 8.

Programmer Response: If the VVDS data set and other VSAM data sets need to be restored from the same input logical volume, rerun the job by passing two RESTORE commands. Restore the VVDS on the first RESTORE command, and restore other VSAM data sets on the second RESTORE command. If VTOCIX data set and other data sets need to be restored from the same input logical volume, rerun

the job by passing two RESTORE commands. Restore the VTOCIX on the first RESTORE command, and restore other data sets on the second RESTORE command. The return code is 8.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR421E (xxx)-mmmm(yy), DATA SET dsname NOT PROCESSED, FULLY QUALIFIED NAME NOT PASSED, n

Explanation: A fully qualified name was not passed for a data set COPY, DUMP, or RESTORE. The n in the message can be:

- 1 For a VVDS during RESTORE
- 2 For a catalog during COPY, DUMP, or RESTORE
- 3 For a read-only data set during COPY or RESTORE
- 4 For a data set that has the OPEN-for-update indicator during DUMP or COPY
- 5 For the VTOCIX data set during RESTORE
- 6 For a fully qualified cluster name not passed when only the index component exists on the volume
- 7 For SYS1.STGINDEX data set during RESTORE
- 8 For a multivolume data set during DUMP.

System Action: The data set is not processed. The return code is 8.

Programmer Response: Resubmit the job by passing the fully qualified name.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25b, and 29.

ADR422I (xxx)-mmmm(yy), LOGICAL VOLUME volser BYPASSED BECAUSE IT ONLY HAS {VSAM | NON-VSAM} DATA SETS

Explanation: The requested type of data sets is not on the logical volume.

System Action: The volume is bypassed. Data set RESTORE continues with other logical volumes.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29. Print the first record on the dump tape. It identifies the type of DUMP and whether VSAM or non-VSAM data sets exist on the volume.

ADR423W (xxx)-mmmm(yy), SELECTED DATA SET dsname IS A MULTIVOLUME DATA SET

Explanation: A fully qualified name was passed in the INCLUDE parameter of a data set DUMP or RESTORE command. This is a multivolume data set. The data set from all of the volumes might not have been processed.

System Action: The return code is 4.

Programmer Response: Make sure that the entire data set from all of the volumes on which it resides was processed. If not, rerun the job by passing the LVOL parameter and corresponding DASD volumes to process the entire data set.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR424E (xxx)mmmm(yy), AN ERROR OCCURRED WHILE RETRIEVING (VVDS|VTOCIX) EXTENTS FROM VOLUME volser, rc

Explanation: DADSM OBTAIN returned an error code (rc) while the VVDS or VTOC extents were being retrieved from the indicated volume. See your *System-Data Administration* book for an explanation of the OBTAIN error codes.

System Action: If it is a data set COPY or DUMP, the volume is bypassed. The command is ended on a data set RESTORE. The return code is 8.

Programmer Response: Take corrective action for the error, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR425E (xxx)mmmm(yy), CONFLICTING BY CRITERIA SPECIFIED

Explanation: Either VSAM and non-VSAM data sets were not wanted or VSAM data set processing was not supported on the system.

System Action: The command ends. The return code is 8.

Programmer Response: Correct the BY criteria in the command, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR426W (xxx)mmmm(yy), EXTENT 1 SIZE OF OUTPUT PARTITIONED DATA SET dsname IS SMALLER THAN EXTENT 1 SIZE OF INPUT

Explanation: The first extent of a newly allocated partitioned data set is smaller than the one that existed during DUMP. For partitioned data sets, the directory must be in the first extent. This message is issued because DFDSS cannot determine the size of the directory to verify that it can fit in the first allocated extent.

System Action: The data set is restored. The return code is 4.

Programmer Response: Check that the directory is in the first extent. Do this by running IEHLIST with the LISTPDS option. If the members cannot be listed, scratch the data set and try to obtain a larger contiguous extent by running DEFRAg, then restore the data set.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25c, and 29.

ADR427E (xxx)mmmm(yy), ERROR IN (VVDS|VVDS IMAGE) ON (VOLUME|LOGICAL VOLUME) volser WHILE ACCESSING COMPONENTS OF CLUSTER clustname IN CATALOG catname

Explanation: DFDSS found errors while retrieving components for the cluster during a COPY, DUMP, or RESTORE. Some components cannot be located for the cluster in the VVDS or VVDS image, or, if it was a RESTORE, the VVDS was not dumped successfully during a full volume DUMP. Inconsistencies exist in the VVRs, or the VVRs have invalid formats.

System Action: The data set is not copied, dumped, or restored. The return code is 8.

Programmer Response: If it is a RESTORE, use a different backup copy of the data set, and rerun the job. If it is a

DUMP, run Access Method Services DIAGNOSE against the data set to determine the nature of the error, and take the action recommended.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 29, and 36b.

ADR428E (xxx)mmmm(yy), VTOC ENTRIES NOT FOUND FOR CLUSTER clustname IN CATALOG catname ON (VOLUME volser|LOGICAL VOLUME volser)

Explanation: VTOC entries are missing for one or more components of the cluster on the volume. For a logical volume, the error is in the source volume and was not detected during DUMP.

Programmer Response: If the operation was not a RESTORE, run Access Method Services DIAGNOSE to detect the error and take the actions recommended. If the operation was a RESTORE, use a previous backup copy, and resubmit the job. The return code is 8.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29. If it is a COPY or DUMP, 36a and 36b.

ADR429W (xxx)mmmm(yy), DATA SET dsname WAS RESTORED TO AN SMS-MANAGED VOLUME BUT IT WAS NOT CATALOGED BECAUSE THE CATALOG KEYWORD WAS NOT SPECIFIED

Explanation: The named non-VSAM data set was restored to an SMS-managed volume. Because the catalog keyword was not specified on the RESTORE command, DFDSS did not catalog the data set when it was restored.

System Action: The data set is restored. The return code is 4.

Programmer Response: You must catalog the data set before it can be accessed because it is now under SMS management.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR430E (xxx)mmmm(yy), ERROR IN CLUSTER clustname IN CATALOG catname ON VOLUME volser

Explanation: A failure occurred during definition of the cluster components in the VVDS or VTOC during a data set COPY or RESTORE.

System Action: The data set is not processed. The return code is 8.

Programmer Response: Run Access Method Services DIAGNOSE on the data set on the volume, and take the appropriate action before rerunning the job to process the data set.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25a, 25b, 29, 36a, and 36b.

ADR431I (xxx)mmmm(yy), DATA SET dsname [IN CATALOG catname] HAS BEEN DELETED

Explanation: The data set was deleted on user request.

System Action: The data set was deleted from the volume.

Programmer Response: If the data set was cataloged, it must be uncataloged by using other utilities.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25a, 25b, and 29.

ADR432E (xxx)mmmmm(yy), VVR FOR COMPONENT componentname [IN CATALOG catname] IS IN ERROR

Explanation: A data set cell, AMDSB cell, or volume cell cannot be located in the VVR for the component in the VVDS or VVDS image (input to data set RESTORE). The format of the VVR may be in error.

System Action: The data set to which the component belongs is not processed. Other messages indicating the affected cluster that is not processed accompany this message. The return code is 8.

Programmer Response: If the failure is on a VVDS on DASD, run Access Method Services DIAGNOSE to determine the nature of the error, and take the recommended action. If it is the input to the RESTORE, use a previous backup copy to restore the data set.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 5, 13, 20, 29, 36a, and 36b.

ADR433E (xxx)mmmmm(yy), LOGICAL ERROR IN BUFFER ON TRACK cchh

Explanation: A logical error was discovered in the track data after it had been read into the buffer and before writing it to DASD. The length of the track data does not match the sum of the count, key, and data fields of all records on the track in the buffer. This problem can arise if the track was not reconstructed correctly by DFDSS during the RESTORE or if the track was not written correctly during the DUMP or, similarly, during a COPY.

System Action: The track is not processed. The return code is 8.

Programmer Response: Contact IBM to correct the problem.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 5b, 13, 20, and 29.

ADR434E (xxx)mmmmm(yy), DATA SET dsname [IN CATALOG catname] WAS NOT UNCATALOGED, rc

Explanation: The data set was not uncataloged for one of the following reason codes (rc):

- 0 It is a VSAM data set. (DFDSS does not support UNCATALOG for VSAM.)
- 1 I/O or other errors, such as authorization or enqueue failure, occurred on the data set, or no catalog entry was found for the data set.
- 2 Either the data set is not a single-volume data set, or it cannot be determined if the data set is single volume or multivolume and it is not presently cataloged.
- 3 The data set was not enqueued by DFDSS.
- 4 The data set was not processed. It is probably a null data set.
- 5 It is an SMS-managed data set. UNCATALOG is ignored for SMS-managed data sets.

System Action: Processing continues. The return code is 8.

Programmer Response: Uncatalog the data set by other means.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25a, and 29.

ADR435E (xxx)mmmmm(yy), DATA SET dsname WAS NOT CATALOGED

Explanation: The data set was not cataloged because either the data set is not a single-volume data set, or it cannot be determined whether the data set is a single-volume or multivolume data set.

System Action: Processing continues. The return code is 8.

Programmer Response: Catalog the data set by other means.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25a, and 29.

ADR436E (xxx)mmmmm(yy), LOGICAL ERROR IN BUFFER ON VOLUME volser, TRACK cchh

Explanation: A logical error was found in the track data after it had been read into the buffer and before writing it to tape.

System Action: The track is not processed. The command is ended. The return code is 8.

Programmer Response: Contact IBM to correct the problem.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 5b, 13, 20, and 29.

ADR437E (xxx)mmmmm(yy), CLUSTER clustername [IN CATALOG catname] CANNOT BE {OPENED | CLOSED} retcode rsncode

Explanation: The specified cluster cannot be opened or closed for VSAM I/O processing. VSAM responded with a return code and a reason code after processing had failed.

System Action: The indicated data set is not restored. The return code is 8.

Programmer Response: Use your *VSAM Administration: Macro Instruction Reference* to identify the problems indicated by the return code and the reason code. Take corrective actions, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR438E (xxx)mmmmm(yy), VSAM I/O FAILED FOR cluster retcode rsncode

Explanation: VSAM I/O processing failed for the specified cluster name. VSAM provided a return code and a reason code after processing had failed.

System Action: The indicated data set is not restored. The return code is 8.

Programmer Response: Use your *VSAM Administration: Macro Instruction Reference* to identify the problems indicated by the return code and the reason code. Take corrective actions, then rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR439E (xxx)mmmmm(yy), PREALLOCATED DATA SET dsname WAS SELECTED BUT IS NOT USABLE, rc

Explanation: DFDSS found a preallocated data set but determined that it was unusable for the following reason (rc):

- 04 The preallocated cluster type is not equal to the target cluster type.
- 08 An AIX is defined over the preallocated cluster.
- 12 The source data RECORDSIZE is incompatible with the target CISIZE and/or control area size attributes.
- 16 The preallocated target relative key position and/or key length is not equal to the source cluster values.

- 20 The preallocated keyed VSAM cluster is not empty or REUSABLE.
- 24 The preallocated KSDS KEYRANGES do not match those of the source cluster.
- 28 The index component of the preallocated target cluster does not have sufficient allocated space for the source index component.
- 32 At least one data component of the preallocated target cluster does not have sufficient allocated space for the corresponding source data component.
- 40 At least one component of the preallocated target cluster spans volumes.
- 44 An update of the target VTOC entry failed. The target must be updated to make the target data set look empty if a utility will do the data transfer.
- 48 The VSAM data set must be copied or restored with utilities, which requires that both the source and the target data sets be cataloged in the standard order of search. However, because they both have the same name, this is not possible.
- 52 The preallocated, unmovable data set is not on the same or a like device type as the source data set.
- 56 The preallocated DSORG is not equal to the target DSORG.
- 60 The preallocated data set space is insufficient.
- 64 The preallocated PDS has an insufficient number of directory blocks to contain the source directory.
- 68 The preallocated data set checking could not be completed.
- 72 The preallocated data set extents do not match the source data set extents, and the source is an unmovable data set.
- 76 The BDAM data set cannot be preallocated to an unlike device type.
- 80 The preallocated data set has no SUL extent.
- 84 The preallocated SUL data set has only one extent.
- 88 An I/O error was encountered on the directory of the preallocated partitioned data set.
- 92 The data set being restored is a PDS with no directory blocks and cannot be restored to a preallocated data set.
- 96 The preallocated sphere does not match the source sphere.
- 100 The preallocated data set is unusable because the source block size is greater than the target device track capacity.
- 104 The NOPACKING keyword was specified for the data set, but the preallocated target was on an unlike device.
- 108 The high-used page value (HURPN) of the preallocated PDSE cannot be retrieved. DFDSS needs the HURPN to calculate the size of the preallocated target data set to ensure that it is usable.
- System Action:** The specified data set is not processed. The return code is 8.
- Programmer Response:** If use of a preallocated data set is not required and more than one output volume is specified, rerun the job without specifying REPLACE. If use of the preallocated data set is required, correct the problem as indicated by the reason code (rc), and rerun the job.
- 04 Delete and redefine the target cluster, ensuring that the cluster types are the same.
- 08 Delete any AIX and PATH associations defined over the target cluster.
- 12 Delete and redefine the target cluster using the source cluster RECORDSIZE attributes.
- 16 Either delete and redefine the target cluster using the source cluster KEY attributes, or ALTER the target cluster KEY attributes to equal the source.
- 20 Delete and redefine the target cluster. If the target is to be copied to repeatedly and is not a key range cluster, specify the REUSE attribute on the DEFINE.
- 24 Delete and redefine the target cluster, ensuring that the KEYRANGES(lowkey highkey) for target cluster are equal to the KEYRANGES(lowkey highkey) for the source cluster.
- 28 Delete and redefine the target cluster with at least as much primary index space as the source index has, or force the target cluster index to extend to at least as much space as the source index.
- 32 Delete and redefine the target cluster with at least as much primary data space as the largest source data component, or force the target cluster data components to extend to at least as much space as the source data components.
- 40 Delete and redefine the target cluster, ensuring that no components span volumes (CANDIDATE volumes excepted).
- 44 One of the following actions should be taken: delete the target and rerun DFDSS, allowing it to perform the allocation; delete the existing target and reallocate a new one, leaving it empty; or use AMASPZAP to alter the target data set VTOC entry to make the data set appear empty.
- 48 Do not preallocate the data set.
- 52 Allocate an unmovable target data set on the same or a like device type, or specify the FORCE keyword.
- 56 Ensure that the preallocated DSORG is equal to the source DSORG.
- 60 Ensure that the amount of space allocated for the preallocated data set is sufficient to contain the source data set.
- 64 Ensure that the number of directory blocks defined for the preallocated PDS is sufficient to contain the source directory blocks.
- 68 Refer to the previous DFDSS message for the probable reason.
- 72 Ensure that the preallocated data set extents match the source data set extents.
- 76 Ensure that the REBLOCKADDRESS keyword was specified and that the source:
 - Is not a standard user label data set
 - Has a RECFM of fixed or fixed block.
Or scratch and reallocate the target on the same device as the source or on a like device.
- 80 Allocate a preallocated data set with standard user labels.
- 84 Allocate a standard user label data set with more than just the standard user label extent.
- 88 Scratch and reallocate the target partitioned data set to either a different location on the target volume or a different target volume.
- 92 Scratch the target data set and allow DFDSS to allocate it during the restore operation.

- 96 Ensure that the preallocated sphere's AIX names, number of AIXs, number of paths, and path names match the source sphere's AIX names, number of AIXs, number of paths, and path names.
- 100 Either reblock the data set or specify a target device that is large enough for the source block size.
- 104 Delete and reallocate the target partitioned data set on a like device, or rerun the job without specifying NOPACKING for the data set.
- 108 Ensure that the preallocated PDSE is usable and accessible.
- Operator Response:** None.
- Problem Determination:** Table I, items 1, 3, 4, 13, 20, 25b, 25c, and 29.

ADR440E (xxx)mmmmm(yy), UNEXPECTED RETURN CODE FROM externalname: xxx-yyy, WHILE PROCESSING DATA SET dsname

- Explanation:** In processing the specified data set, the named external routine was called, and it returned an unexpected return and/or reason code.
- System Action:** Processing is ended for the specified data set. The return code is 8.
- Programmer Response:** This message is not normally expected and indicates a programming or system problem. Rerun the job with PARM = 'ABEND=440' on the EXEC statement, and a SYSUDUMP DD statement. Contact IBM for programming support.
- Operator Response:** None.
- Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

ADR441W (xxx)mmmmm(yy), LOGICAL ERROR IN BUFFER ON TRACK cchh

- Explanation:** A logical error was discovered in the track data after it had been read into the buffer and before writing it to DASD. The length of the track data does not match the sum of the count, key, and data fields of all records on the track in the buffer. This message is issued instead of ADR433E when CANCELERROR is not specified.
- System Action:** The track is not processed. The return code is 4.
- Programmer Response:** Use DFDSS PRINT to examine the track data. Contact IBM to correct the problem if unable to determine the cause.
- Operator Response:** None.
- Problem Determination:** Table I, items 1, 3, 4, 5b, 13, 20, and 29.

ADR442I (xxx)mmmmm(yy), DATA SET dsname PREALLOCATED [, IN CATALOG catname], ON VOLUME(S): (volser | volser list)

- Explanation:** The data set was selected for replacement as a preallocated target data set. The target catalog name, if known, is printed. The target volume or optional volume list is always printed with this message.
- System Action:** None.
- Programmer Response:** None.
- Operator Response:** None.
- Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

ADR443W (xxx)mmmmm(yy), DUE TO ERRORS, TARGET COMPONENT name IS NOW NAMED name. CLUSTER cluster name, CATALOG catname

- Explanation:** During COPY processing, the target cluster and components can be allocated using generated names. After the COPY, the target cluster name and component names are altered to match the source names. This message indicates that the target cluster name was successfully altered, but the alter for a component failed.
- System Action:** Processing continues with the next data set. Message ADR469W follows this message. The return code is 4. Run Access Method Services LISTCAT against the target catalog to determine the reason for the Access Method Services ALTER failure. You can alter the component name to match the source by using Access Method Services ALTER NEWNAME.
- Operator Response:** None.
- Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, 26, and 29.

ADR444E (xxx)mmmmm(yy), DATA SET dsname (IS EMPTY | HAS AN UNSUPPORTED DSORG). IT WILL NOT BE COPIED

- Explanation:** The data set cannot be copied to an *unlike* device for one of the following reasons:
- It has an unsupported DSORG.
 - It has a BLKSIZE of 0.
 - It is empty (DS1LSTAR = 0) but does not have an EOF as its first record.

The data set cannot be copied to a *like* device if it is both

- Empty and is being reblocked, and
- Its first record is not an EOF record.

System Action: The data set was not copied. Processing continues. The return code is 8.

Programmer Response: To have the data set copied, do one of the following:

- For a data set with an unsupported DSORG, make sure that the selected target volume is of a like device type. For a data set with an unsupported DSORG going to an unlike target device, specify the PROCESS keyword with the UNDEFINED subkeyword. The target device specified or chosen must have a track capacity equal to or greater than the source.
- For a data set with a BLKSIZE = 0, make sure that the selected target volume is of a like device type.
- For an empty data set, use ALLDATA or ALLEXCP to force copy of all allocated space in the data set. Make sure that the selected target volume is of a like device type and the data set is not being reblocked.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25a, and 29.

ADR4451 (xxx)mmmmm(yy), SOURCE VSAM DATA SET dsname [IN CATALOG catname] IS EMPTY

Explanation: The source VSAM data set is empty; therefore the target data set is allocated, but no data movement is performed. If the data set is preallocated, the existing data set is deleted then reallocated using the source as the model.

System Action: Processing continues. The return code is 0.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR446E (xxx)mmmmm(yy), DATA SET dsname [IN CATALOG catname] CONTAINS NO DIRECTORY BLOCKS

Explanation: A partitioned data set with no directory was encountered. This was caused by not specifying at least one directory block at the time the data set was created.

System Action: The data set is not processed. The return code is 8.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 2, 3, 4, 13, 20, and 29.

ADR447I (xxx)mmmmm(yy), VSAM DATA SET dsname [IN CATALOG catname] WAS DELETED FROM VOLUME volser, rc

Explanation: The target VSAM data set was deleted. The reason codes (rc) are:

- 1 The target KSDS/VRRDS on an unlike device was not reusable.
 - 2 The data set organization of the target did not match the source.
 - 3 The CI size of the target did not match the source.
 - 4 The IMBED attribute of the target did not match the source.
 - 5 The KEY length of the target did not match the source.
 - 6 The record length of the target did not match the source.
 - 7 The REPLICATE attribute of the target did not match the source.
 - 8 The SPANNED attribute of the target did not match the source.
- B The source data set is empty.

System Action: Processing continues. For reason codes 2 and 4 to 8, the source attributes override those of the preallocated target. The return code is 0.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR448I (xxx)mmmmm(yy), DATA SET dsname DUMPED USING NON-OPTIMIZE CHANNEL PROGRAMS AFTER TRACK OVERFLOW RECORDS WERE ENCOUNTERED

Explanation: The source data set contained track overflow records and OPTIMIZE(2), (3), or (4) was specified for the dump.

System Action: The source data set tracks were processed by using nonoptimized channel programs. This results in performance degradation if the data set is not flagged as track

overflow in the VTOC entry or is not in the TRACKOVERFLOW keyword list.

Programmer Response: No direct programmer action is required, because the data set was dumped successfully. Optimized channel programs fail on tracks containing overflow records, thus forcing a channel retry with nonoptimized channel programs in order to read the tracks. Therefore, data sets containing track overflow records that are not flagged as such in the VTOC entry must be prevented from using an optimized channel program. If ALLDATA or ALLEXCP is specified for the data set, and it does not contain any track overflow records, the allocated tracks beyond the end of the used data may have residual track overflow records that cause the message to be issued.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR449I (xxx)mmmmm(yy), MESSAGE DATA SET USED TO STORE TASK RELATED MESSAGES FOR THIS TASK IS FULL AND CANNOT BE EXTENDED FURTHER. MESSAGES WILL BE SPOOLED TO SYSPRINT. SUBSEQUENT MESSAGES FOR THIS TASK WILL BE STORED AND GROUPED TOGETHER

Explanation: The message data set used to store messages for this task (in order to keep messages grouped by task when DFDSS is running in parallel mode) has run out of space. All the messages stored thus far for this task are spooled to SYSPRINT. If DFDSS is indeed running in parallel mode, this can cause the messages for this task to be broken into groups with messages from other tasks located in between them.

System Action: The message data set is reinitialized, and messages continue to be stored and grouped together until the task ends or the message data set becomes full again.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR450E (xxx)mmmmm(yy), THE NUMBER OF INPUT AND OUTPUT TRACK RANGES DOES NOT MATCH

Explanation: The number of input track ranges does not correspond to the number of output track ranges. Each input track range requires a corresponding output track range. Each output track range requires a corresponding input track range.

System Action: The task is ended. The return code is 8.

Programmer Response: Specify the same number of input and output track ranges.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR452E (xxx)mmmmm(yy), (IEHMOVE | IEBCOPY | IEBISAM | IDCAMS | ICKDSF) UTILITY FAILED WHILE PROCESSING (DATA SET dsname VOLUME volser), (ABEND | RETURN) CODE IS nnn

Explanation: During processing of a DFDSS function, a system utility was invoked to process the specified data set. The utility encountered errors, and processing failed.

System Action: The data set is not processed. The return code is 8.

Programmer Response: Take one of the following actions:

- For abending utilities, see the *System Codes* for an explanation of the abend code. If the abend is because of insufficient storage (DASD or virtual), you must run the utility itself (not through DFDSS), as described in your *Utilities* manual. If IEHMOVE abends with code 0C4, it

could be because DFDSS called it with a task I/O table (TIOT) referring to more DD statements than IEHMOVE can handle. In this case, change the DFDSS filtering to select fewer data sets.

- For IEHMOVE, IEBCOPY, or IEBISAM error return codes, run the job by specifying the DFDSS *UTILMSG = YES* parameter. See the *MVS/JXA System Messages* or *MVS/370 Utilities Messages* for an explanation of the utility messages.
- For IDCAMS error return codes, run the job by specifying the DFDSS *UTILMSG = YES* parameter. See the *System Messages* for an explanation of the messages.
- For ICKDSF error return codes, run the job and specify the DFDSS *UTILMSG = YES* parameter. See the *Device Support Facilities User's Guide and Reference* for an explanation of the messages.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 5, 13, 20, and 29.

ADR453E (xxx)-mmmmm(yy), USER REBLOCK EXIT SPECIFIED AN INVALID BLOCK SIZE FOR DATA SET dsname. DATA SET WILL NOT BE PROCESSED

Explanation: The data set was eligible for reblocking, and the user reblock exit was called. However, the block size returned to DFDSS from the exit was invalid for the data set.

System Action: The data set is not allocated or copied. The return code is 8.

Programmer Response: Correct the block size for the data set in the exit routine.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR454I (xxx)-mmmmm(yy), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED

Explanation: The list of successfully processed data sets follows this message.

System Action: Processing continues.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR455W (xxx)-mmmmm(yy), THE FOLLOWING DATA SETS WERE NOT SUCCESSFULLY PROCESSED

Explanation: The list of data sets that were not successfully processed follows this message.

System Action: Processing continues. The return code is set to 4.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR456I (xxx)-mmmmm(yy), THE NUMBER OF DATA SETS PROCESSED ON VOLUME volser IS nnnnn

Explanation: The number, nnnnn, is the number of data sets successfully processed. For COMPRESS, the data set names follow this message.

System Action: Processing continues.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR457I (xxx)-mmmmm(yy), THE NUMBER OF TRACKS MADE AVAILABLE ON VOLUME volser IS nnnnn

Explanation: RELEASE processing released nnnnn tracks.

System Action: Processing continues.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR458I (xxx)-mmmmm(yy), (TOTAL | USED) NUMBER OF TRACKS ON VOLUME volser IS nnnnn

Explanation: The number, nnnnn, indicates either the total number of tracks on the volume or the number of used tracks on the volume after RELEASE processing.

System Action: Processing continues.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR459E (xxx)-mmmmm(yy), ERROR ENCOUNTERED WHILE FILTERING ON THE CATALOG.

Explanation: This message is preceded by message ADR497E or ADR724E. When this message is preceded by ADR497E, refer to IDC3009I in *System Messages* for an explanation and corrective action. When preceded by ADR724E, refer to IGW01zzz in *System Messages* for an explanation and corrective action.

System Action: The current volume is bypassed. Processing continues with the next volume, if any. The return code is 8.

Programmer Response: For I/O errors, follow recommendations by your installation for I/O errors, and resubmit the job. For logical errors encountered in the VVDS, run Access Method Services DIAGNOSE against the indicated volume, and take necessary action to correct the problem.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 5b, 13, 20, 25b, and 29. If it is a logical error, use the PRINT command to print the VTOC and VVDS tracks before taking action 29.

ADR460I (xxx)-mmmmm(yy), UTILITY GENERATED MESSAGES FOLLOW FOR {DATA SET dsname | VOLUME volser}

Explanation: During processing of a DFDSS function, a system utility was invoked that generated SYSPRINT data. The utility messages are printed following this message when UTILMSG = YES is specified in the PARM information of the DFDSS EXEC statement.

System Action: The function continues processing.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR461W (xxx)-mmmmm(yy), UNMOVABLE DATA SET dsname WITH FOLLOWING EXTENTS CANNOT BE PROCESSED

Explanation: The unmovable data set that has the listed extents cannot be processed. Either the system does not support allocation of unmovable data sets or the extents are not free on the target volume.

System Action: The data set is not processed. The return code is set to 4.

Programmer Response: Either specify FORCE if you want to move the unmovable data set, or free the required extents and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 25a, and 29.

ADR462E (xxx)mmmmm(yy), NON-VSAM DATA SET dsname [IN CATALOG catname] IS NOT PROCESSABLE

Explanation: The CVOL catalog entry is not processed because the DELETE subparameter was not specified and the RENAMEUNCONDITIONAL subparameter was specified.
System Action: Processing continues. The return code is 8.
Programmer Response: Copy the CVOL catalog, specifying DELETE, and remove the RENAMEUNCONDITIONAL subparameter.
Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR463E (xxx)mmmmm(yy), CLUSTER clustname WAS NOT (DELETED|DEFINED|ALLOCATED) BECAUSE VOLUME(S) WERE NOT AVAILABLE

Explanation: The allocation of the volumes failed because the volumes are not available. The cluster was not deleted, defined, or allocated.
System Action: The return code is 8.
Programmer Response: Ensure that the volumes where the cluster resides are available.
Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR464I (xxx)mmmmm(yy), DATA SET dsname HAS BEEN UNCATALOGED FROM CATALOG catname

Explanation: The named data set was uncataloged at the user's request.
System Action: None.
Programmer Response: None.
Operator Response: None.
Problem Determination: None.

ADR465I (xxx)mmmmm(yy), DATA SET dsname HAS BEEN CATALOGED IN CATALOG catname

Explanation: The named data set was cataloged at the user's request.
System Action: None.
Programmer Response: None.
Operator Response: None.
Problem Determination: None.

ADR466E (xxx)mmmmm(yy), UNEXPECTED RETURN CODE nn FROM PARTREL MACRO IN MODULE modname WHILE PROCESSING DATA SET dsname

Explanation: A PARTREL macro was issued to release all unused DASD space from a sequential or partitioned data set, and an unexpected return code was received during RELEASE processing. A return code of 3 indicates that a data set was open. Any other return code probably indicates an invalid VTOC.
System Action: Processing of the command is ended. The return code is 8.
Programmer Response: See the *System-Data Administration* for an explanation of PARTREL return codes.
Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, 25b, and 29.

ADR467E (xxx)mmmmm(yy), DURING A TARGET TRACKS COPY, THE OUTPUT TRACKS FALL WITHIN THE INPUT TRACK RANGE; AND THE INPUT AND OUTPUT VOLUMES ARE THE SAME

Explanation: A COPY request was issued with the OUTTRACKS option. The input volume serial number and output volume serial number are the same, and the output tracks overlap the input tracks.
System Action: Processing of the command is ended. The return code is 8.
Programmer Response: Select nonoverlapping tracks for input and output.
Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, and 29.

ADR468E (xxx)mmmmm(yy), VSAM DATA SET dsname [IN CATALOG catname] IS NOT PROCESSABLE

Explanation: The VSAM entry was not processed because one or more of the following conditions exist:

- If REPLACE was specified, either the data set does not qualify for preallocation or a preallocated target does not exist, and one of the following conditions applies:
 - If DELETE is specified and the entry name is a SYS1., page, or swap data set, the RENAMEU subparameter was not specified.
 - If the entry name is a cluster name and DELETE was not specified, then: (1) the RENAMEU subparameter was not specified, or (2) the RECAT subparameter was not specified.
 - If the entry name is an alternate index or a user catalog name: (1) the DELETE subparameter was not specified, or (2) the RENAMEU subparameter was specified.
- If REPLACE was not specified, one of the following conditions applies:
 - If DELETE is specified and the entry name is a SYS1., page, or swap data set, the RENAMEU subparameter was not specified.
 - If the entry name is a cluster name and DELETE was not specified: (1) the RENAMEU subparameter was not specified or (2) the RECAT subparameter was not specified.
 - If the entry name is an alternate index or a user catalog name: (1) the DELETE subparameter was not specified or (2) the RENAMEU subparameter was specified.
 - If the entry name is a user catalog name, INDDNAME or INDYNAM was specified.

System Action: Processing continues. The return code is 8.
Programmer Response: Take one of the following actions:

- If REPLACE was specified:
 - If the data set is eligible for preallocation, ensure that a qualifying target is available on the target volumes.
 - If the data set is not eligible for preallocation or a REPLACE operation is not required, then determine further action from the following lists.
- If the entry name is a SYS1., page, or swap data set and DELETE is specified, copy the entry specifying RENAMEU(newname).

- If the entry name is a cluster name: (1) copy the cluster by specifying DELETE or (2) copy the cluster by specifying RENAMEU(newname) or (3) copy the cluster by specifying RECAT(catname).
- If the entry name is an alternate index or a user catalog name, copy the alternate index or user catalog, specifying DELETE, and remove the RENAMEU subparameter.
- If the entry name is a user catalog name, do not specify INDDNAME or INDYNAM.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR469W (xxx)mmmm(yy), THE FOLLOWING DATA SETS WERE COPIED, BUT ENCOUNTERED POSTPROCESSING ERRORS

Explanation: After a data set was copied, a postprocessing error was encountered. The error occurred during an attempt to catalog, uncatalog, scratch, correct a VSAM cluster component name, or RACF/password-protect the source or target data set.

System Action: Processing continues. The return code is 4.

Programmer Response: See earlier messages for specific errors.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR470W (xxx)mmmm(yy), NO DATA SETS SELECTED FOR COPY

Explanation: No data sets were selected for processing. The data sets found in the catalog or on the input volumes did not pass INCLUDE, EXCLUDE, or BY filtering or cannot be copied.

System Action: Processing continues with the next control statement. The return code is 4.

Programmer Response: When input volumes are specified, ensure that all components of a cluster and all pieces of a multivolume data set are contained on the input volumes. Change filtering criteria, add a STEPCAT, or specify different input volumes, then rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR471E (xxx)mmmm(yy), DUE TO ERRORS, SOURCE {CLUSTER,COMPONENT} name IS NOW NAMED name. CLUSTER clustname, CATALOG catname

Explanation: During COPY processing, the source cluster and components can be renamed in order to allow the COPY target cluster to be altered to the source name. If the target-to-source alter fails, the target data set is deleted, and restoring the source cluster to its original state is attempted. This message is issued for each alter rename failure. The cluster name and catalog name indicate the name and location of the failing data set.

System Action: Processing continues with the next data set. Message ADR455E follows this message. The return code is 8.

Programmer Response: Run Access Method Services DIAGNOSE against the source and target catalogs to determine the reason for the Access Method Services ALTER failures. Restore the original cluster name, using Access Method Services ALTER NEWNAME.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 26, 29, and 36.

ADR472E (xxx)mmmm(yy), UNABLE TO SELECT A TARGET VOLUME FOR DATA SET dsname [IN CATALOG catname], rc

Explanation: DFDSS was unable to select a target volume from the available target volumes for one of the following reasons (rc).

- 04 No target volume has enough space to accommodate the target data set.
- 08 At least one duplicate data set was encountered on a target volume, and the REPLACE keyword was not specified. If the duplicate data set is named SYS1.xxxx, you must also specify PROCESS(SYS1). However, DFDSS will not replace a SYS1.xxxx data set that is a CVOL, Integrated Catalog Facility user catalog, VVDS, or VTOCIX, regardless of the REPLACE and PROCESS(SYS1) keywords.
If the target volume did not contain a duplicate data set, then this reason code means the target volume did not have enough space for the data set. For an ISAM data set, the matching space in the target volume was not available.
- 12 The data set is non-VSAM and is allocated as either unmovable or absolute track. DFDSS was able to select only unlike targets for target allocation and the FORCE keyword was not specified.
- 16 The data set is non-VSAM and either unmovable or absolute track allocated. An absolute allocation failure, or lack of system support for absolute allocation, precludes the use of absolute allocation; however, the FORCE keyword was not specified.
- 20 A directory read failed for the partitioned data set.
- 24 An error was encountered while the VVR records for the VSAM data set were being read.
- 28 Duplicate VTOC entries were encountered on the only or all available target volumes for the VSAM data set.
- 32 A duplicate entry for the VSAM data set was encountered in the target catalog.
- 36 An error was encountered during the define of the target VSAM data set.
- 40 GETMAIN errors were encountered while the internal control blocks for the data set were being built.
- 44 The data set was a direct access (BDAM) data set, and only target devices of smaller track capacity than the source were available. The data set was not specified in the RELBLOCKADDRESS keyword list, or it was specified in RELBLOCKADDRESS, but the data set is not accessed by relative block address (i.e., it has a standard user label or its record format is not F or FB).
DFDSS does not allocate or attempt to copy the data set.
- 48 The source data set block size was greater than the track capacity of all volumes DFDSS was able to select for allocation. The data set was not indicated as track overflow in the VTOC entry.
- 52 An indexed VTOC was not available.
- 56 Not enough space is available in the VTOC on the target volume.
- 60 The DADSM installation exit rejected the allocation request.
- 64 No output volume was specified for the COPY function, and the data set is not SMS managed.

- 68 Usable space on all the target volumes combined was insufficient to define the target data set.
- 72 During a non-SMS allocation, no target volumes were available and at least one output volume was not selected because it was SMS managed.
- 76 The NOPACKING keyword was specified for the data set, but no space was available or no like devices were specified as targets.
- 80 No target volume that has device track capacity equal to or greater than the source is available for selection.
- 84 The data set was a direct access (BDAM) data set and the only available target devices were unable to contain the source track's worth of data due to the blocksize of the data set. For example, if the source device is a D/T3380 and the only available devices are D/T3390S, and the data set's blocksize is 190 with a key length of 0, then a track on the target cannot contain as much data as a track on the source.

The data set was not specified in the RELBLOCKADDRESS keyword list, or it was specified in the REBLOCKADDRESS but it was not accessed by a relative block address (i.e., it has a standard user label or it's record format is not F or FB). DFDSS does not allocate or attempt to copy the data set.
- 88 During a non-SMS allocation, no usable target volumes were available.
- 92 During DFDSS data set COPY, an attempt was made to copy the data set back to its source volume with DELETE specified, but neither RENAME nor RENAMEUNCONDITIONAL was specified.

System Action: The data set is not copied. The return code is 8.

Programmer Response: Depending on the reason code, take the following action and rerun the job:

- 04 Either increase the value for PERCENTUTILIZED, or specify additional target volumes.
- 08 Specify additional target volumes, delete the duplicate data set, specify REPLACE, or specify PROCESS(SYS1). For ISAM data sets, specify a volume that has free space on the matching tracks.
- 12 Either specify FORCE if the unlike target volumes are desired, or specify additional target volumes of the same device type that the data set currently resides on.
- 16 Specify FORCE or specify additional target volumes.
- 20 Ensure that the partitioned data set is usable.
- 24 Ensure that the cluster and its components are usable.
- 28 Specify additional target volumes.
- 32 If a target catalog was specified by using RECATALOG, either delete the existing data set in that catalog, or specify a different catalog. If a target catalog was not specified, the data set is being defined in a catalog set by the STEPCAT/JOBCAT/MASTERCAT structure. Either determine where the duplicate entry exists and delete it, or modify the catalog structure using a STEPCAT statement in the job JCL.
- 36 Message ADR497E accompanies this message. Determine the cause of the error and correct it.
- 40 Increase the region size.
- 44 Specify target devices of the same or larger track capacity for the direct access (BDAM) data set.

- 48 Specify target devices of the same or larger track capacity for the data set, or use an appropriate system utility to copy the data set.
- 52 Specify a volume with an indexed VTOC.
- 56 Enlarge the VTOC on the target volume or specify additional target volumes.
- 60 Specify additional target volumes.
- 64 Specify OUTDDNAME or OUTDYNAM.
- 68 Either specify additional target volumes or, if PERCENTUTILIZED was specified, increase the PERCENTUTILIZED value.
- 72 Specify non-SMS-managed volumes in the OUTDD or OUTDYN.
- 76 Either rerun the job without specifying NOPACKING for the data set, or rerun the job specifying the same or like devices with enough room to allocate the data set.
- 80 If the target data set is SMS, make sure that the storage class selected maps to a storage group that has devices with track capacity equal to or greater than the source. If the target data set is non-SMS, specify a target volume that has track capacity equal to or greater than the source.
- 84 Specify target devices of the same or larger track capacity for the direct access (BDAM) data set.
- 88 Specify usable non-SMS target volumes in the OUTDD or OUTDYN statement.
- 92 Choose a target volume different from the data set's source volumes, or use the RENAMEU keyword to rename the data set.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25b, 26, and 29.

ADR473E (xxx)mmmm(yy), FAILURE WHILE ATTACHING UTILITY MODULE, RC = xx

Explanation: DFDSS was unable to attach a utility module. The reason codes, xx, are documented in the *Application Development Guide*.

System Action: No data sets are processed. The task ended. Processing continues with the next control statement. The return code is 8.

Programmer Response: Correct the problem as indicated by rcxx, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR474I (xxx)mmmm(yy), DATA SET dsname IS EMPTY, BUT IT WILL BE (COPIED |RESTORED)

Explanation:

During a COPY operation:

- For an empty non-VSAM data set, the data set is allocated if the source data set has an EOF in its first record. Only an EOF record is written to the target data set.
- For an empty VSAM data set, the data set is allocated but no data is copied.

During a RESTORE operation:

- The target data set is allocated but no data is restored.

System Action: Processing continues. The return code is 4.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25b, 26, and 29.

ADR475I (xxx)-mmmmm(yy), FOLLOWING DATA SETS WERE SELECTED

Explanation: TYPRUN = NORUN was specified in the EXEC statement parameter, and the following list of data set names met the filtering criteria.

System Action: The data sets are not dumped, copied, or restored.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR476E (xxx)-mmmmm(yy), UTILITY PROCESSOR TERMINATED BECAUSE OF SYSPRINT/SYSIN PROCESSING FAILURES

Explanation: Errors in utility SYSPRINT or SYSIN data set processing occurred, causing the utility processor task to end.

System Action: Utility processing is ended. The return code is 8.

Programmer Response: Rerun the job, ensuring that adequate DASD temporary space is available.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR477E (xxx)-mmmmm(yy), VVDS/VVR/CELL ERRORS ENCOUNTERED FOR CLUSTER name

Explanation: Data set COPY or RESTORE encountered errors while reading or updating VVDS/VVR/CELL information.

System Action: The data set is not copied or restored. Processing continues. The return code is 8.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR478W (xxx)-mmmmm(yy), VSAM CLUSTER cluster name NOT DUMPED, COMPONENT component name SIZE IS GREATER THAN A SINGLE SOURCE VOLUME

Explanation: The VSAM cluster was not dumped because it had a multivolume component that was larger than an entire source volume. To restore the data set, it would be necessary to allocate the component on multiple volumes, but the system does not support the allocation of VSAM data sets on multiple volumes.

System Action: The data set was not dumped. Processing continues. The return code is 4.

Programmer Response: Take one of the following actions:

- Use DFDSS DUMP, specifying the input volumes that contain the VSAM cluster. DFDSS RESTORE can restore the data set from the dump tape in physical dump format.
- Use Access Method Services EXPORT to dump the VSAM cluster. Use Access Method Services IMPORT to restore from the EXPORT-created dump tape.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR479E (xxx)-mmmmm(yy), LOGICAL DATA SET DUMP TAPE DOES NOT CONTAIN {VSAM|NON-VSAM} DATA SETS

Explanation: The requested type of data set is not on the logically formatted dump tape. Either the BY criterion is incorrect or the wrong tape was specified in the RESTORE command.

System Action: No data sets are restored. Processing continues. The return code is 8.

Programmer Response: Either change the BY criterion DSORG or use a different dump tape, and rerun the job. For an empty VSAM data set, no action is required.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR480W (xxx)-mmmmm(yy), THE FOLLOWING DATA SETS WERE NOT PROCESSED FROM THE LOGICALLY FORMATTED DUMP TAPE DUE TO ERRORS:

Explanation: The data sets in the list were not processed. The error is identified by other messages that precede this message.

System Action: The return code is 4.

Programmer Response: Correct the problem according to preceding messages.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR481E (xxx)-mmmmm(yy), UNABLE TO ALTER VSAM CLUSTER clustername [IN CATALOG catname]

Explanation: The VSAM ALTER NEWNAME failed while attempting to alter the source cluster name.

System Action: Processing continues. The return code is 3.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR482E (xxx)-mmmmm(yy), I/O ERROR ON DDNAME ddname WHILE PROCESSING taperec

Explanation: I/O errors were encountered while the indicated type of dump tape record was being read during logical data set RESTORE processing.

System Action: The data set is bypassed. The return code is 8.

Programmer Response: Use an alternate backup copy to restore the data set.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR483W (xxx)-mmmmm(yy), DATA SET dsname NOT CATALOGED

Explanation: The data set was not cataloged. RECATALOG(*) was specified, but the source data set was not cataloged.

System Action: Processing continues. The return code is 4.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR484W (xxx)mmmmm(yy), COMMAND IS NOT SUPPORTED FOR DATA SET dsname DURING LOGICAL DATA SET PROCESSING

Explanation: Data set RESTORE from a logically formatted dump tape does not support the data set organization of the indicated data set.

System Action: Processing continues for other data sets. The return code is 4.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR485E (xxx)mmmmm(yy), CATALOG catname IS NOT IN STEPCAT/JOBCAT/MASTERCAT STRUCTURE. DATA SET dsname WILL NOT BE PROCESSED

Explanation: The cluster named in the message required DFDSS to use IDCAMS or VSAM I/O to perform the COPY or RESTORE. This requires that both the source and target cluster (allocated by DFDSS) be accessible via the catalog structure.

System Action: Processing continues with the next data set. The return code is 8.

Programmer Response: Add a STEPCAT or JOBCAT JCL statement specifying the named catalog, or concatenate it to your existing STEPCAT/JOBCAT JCL statement, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR486I (xxx)mmmmm(yy), UNMOVABLE DATA SET dsname RESTORED TO FREE SPACE

Explanation: The unmovable data set specified in the message cannot be restored to the cylinder/track location from which it was dumped. Because FORCE was specified in the RESTORE command, this data set was restored to free space.

System Action: Processing continues.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR487W (xxx)mmmmm(yy), DATA SET dsname NOT ON LOGICALLY FORMATTED DUMP TAPE

Explanation: During a data set RESTORE from a logically formatted dump tape, the data set was not found on the tape. Possibly the data set name is contained in the data set name list at the beginning of the dump tape, but the data set was not dumped. (This can occur if the data set could not be serialized at dump time.) Either the wrong input file was used, an incorrect data set name was specified, or the data set was rejected because of the DSORG filtering criteria. This message is printed if fully qualified names are specified.

System Action: Processing continues. The return code is 4.

Programmer Response: Correct the error, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR488E (xxx)mmmmm(yy), ERROR ENCOUNTERED WHILE ACCESSING (VTOC|VVDS) FOR DATA SET dsname ON VOLUME volser ON DDNAME ddname

Explanation: An inconsistency on the VTOC or VVDS was encountered during data set RESTORE from a logically formatted dump tape.

System Action: Processing continues for other data sets. The return code is 8.

Programmer Response: Use an alternate backup copy to restore the data set.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 5, 13, 20, 25a, 29, 36a, and 36b.

ADR489I (xxx)mmmmm(yy), {dsname|clustername|component name} WAS {SELECTED|RESTORED}

Explanation: After a data set was successfully restored or after a data set was successfully selected (when TYPRUN=NORUN), the data set name is printed for a non-VSAM data set. Component names are printed for a VSAM data set.

System Action: Processing continues.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR490W (xxx)mmmmm(yy), ERROR {OPENING|READING} UTILITY SYSPRINT MESSAGES. UTILITY MESSAGES WILL BE LOST

Explanation: An error occurred either in opening or in reading the utility SYSPRINT file.

System Action: DFDSS continues processing for the task, but utility SYSPRINT messages are lost. The return code is 4.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR491E (xxx)mmmmm(yy), TASK TERMINATED DUE TO INSUFFICIENT STORAGE FOR UIM I/O BUFFER AND/OR EXIT IDENTIFICATION BLOCK

Explanation: If the application interface is used, the system requires a buffer to pass I/O records to the user interaction module (UIM). If virtual storage cannot be obtained for the buffer or the UIM exit identification block, the related DFDSS function is ended.

System Action: None. The return code is 8.

Programmer Response: Increase either the SIZE parameter, the region size, or both, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 29, and 35.

ADR492E (xxx)mmmmm(yy), ERROR OCCURRED DURING COPY OF CATALOG catname, RC-xx. [EXPORT DATA SET dsname IS ON VOLUME volser]

Explanation: The catalog was being moved using Access Method Services EXPORT or IMPORT when IDCAMS errors occurred. Corrective action may be required to restore the source catalog to a usable condition. The reason code describes the error as follows:

01 An error occurred when the catalog was exported to a temporary sequential data set. The COPY is discontinued, and the source catalog remains in usable condition. For this reason code, the second portion of the message is not printed.

02 A failure occurred when the target catalog was imported. The source catalog was deleted and must be recovered by the user.

System Action: Processing continues with the next COPY. The return code is 8.

Programmer Response: Ensure that the DFDSS parameter option *UTILMSG = YES* is specified when running jobs that perform catalog moves. All reason codes are accompanied by IDCAMS messages that clarify the remedial actions necessary. See the *System Messages* for an explanation of the messages.

Take the following actions, depending on the reason code:

- 01 Use the IDCAMS messages as a guide in correcting the error. Rerun the job.
- 02 Restore the source catalog before attempting to rerun the job. If the catalog was backed up before the COPY was attempted, the backup can be used to restore the catalog. Otherwise, the catalog can be restored by importing the catalog from the export data set described in the message. See the *Access Method Services Reference* for an explanation of the IMPORT command. Delete the export data set described in the message before rerunning the job.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25b, 26, and 29.

ADR493E (xxx)mmmmm(yy), UNCATALOG/DELETE IS NOT ALLOWED FOR DATA SET dsname [IN CATALOG catname]

Explanation: Either exclusive control of the data set is not obtained or it is a *SYS1*, system, page, or swap data set that cannot be uncataloged.

System Action: Processing continues. The return code is 8.

Programmer Response: For exclusive control, either remove the SHARE keyword or ensure exclusive control of the data set, unless the data set is a VTOCIX or a VVDS. For *SYS1*, system, page, or swap data sets, specify PROCESS(SYS1) to uncatalog or delete the data set.

Note: PROCESS(SYS1) does not lift processing restrictions for VTOCIXs and VVDSs.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR494W (xxx)mmmmm(yy), {OPEN|CLOSE|I/O} ERROR ON MESSAGE DATA SET FOR TASK. TASK RELATED MESSAGES MAY BE LOST

Explanation: An OPEN, CLOSE, or I/O error occurred on a message data set used to store task-related messages.

System Action: DFDSS continues processing for the task, but messages for the task may be lost. The return code is 4.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR495E (xxx)mmmmm(yy), ECB UNEXPECTEDLY POSTED IN MODULE modname, ECB CONTENTS ARE xxxxxxxx

Explanation: An ECB related to an attached task was unexpectedly posted.

System Action: Command processing is ended. The return code is 8.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, 25b, and 29.

ADR496E (xxx)mmmmm(yy), UNEXPECTED RETURN CODE nn FROM DETACH ISSUED FROM MODULE modname

Explanation: An attempt to detach a subtask failed.

System Action: The return code is 8.

Programmer Response: None.

Operator Response: None.

Problem Determination: See the *Application Development Guide* manual for an explanation of DETACH return codes.

ADR497E (xxx)mmmmm(yy), A CATALOG ERROR OCCURRED WHILE action-description. RETURN CODE IS xx, REASON CODE IS nn-yyy

Explanation: Catalog Management module IGG0CLnn returned this return code, xx, and reason code, yyy, as the result of a catalog error or an exception condition while DFDSS was performing the action described by *action-description*.

Action-description can be any one of the following:

- RENAMING CLUSTER *clusname* TO *clusname2*
- RENAMING COMPONENT *compname* TO *compname2*
- RENAMING AIX *aixname* TO *aixname2*
- RACF PROTECTING *dsname*
- RACF UNPROTECTING *dsname*
- ALTERING ACCESS TO USER CATALOG *catname*
- RENAMING DATA SET *dsname* TO *dsname2*
- CONVERTING CLUSTER *clusname* TO SMS
- CONVERTING CLUSTER *clusname* TO NONSMS
- REMOVING CANDIDATE VOLUMES FROM COMPONENT *compname*
- DEFINING CLUSTER *clusname*
- CATALOGING DATA SET *dsname*
- DEFINING AIX *aixname*
- DEFINING PAGESPACE *dsname*
- DEFINING USER CATALOG *catname*
- DEFINING PATH *pathname* TO *clusname*
- DEFINING ALIAS *aliasname* TO *clusname*
- SEARCHING FOR DATA SETS BEGINNING WITH *prefix*
- OBTAINING INFORMATION FOR DATA SET *dsname*
- OBTAINING VOLUME INFORMATION FOR DATA SET *dsname*
- DETERMINING TARGET CATALOG FOR DATA SET *dsname*

- DELETING CLUSTER *clusname*
- DELETING DATA SET *dsname*
- DELETING USER CATALOG *catname*
- DELETING AIX *aixname*
- DELETING PATH *pathname*
- EXPORTING CATALOG *catname*
- IMPORTING CATALOG *catname*
- RECATALOGING CLUSTER *clusname*
- RECATALOGING AIX *aixname*
- RECATALOGING DATA SET *dsname*
- READING CATALOG *catname*
- ALTERING THE VOLUME LIST FOR DATA SET *dsname*.

System Action: Command processing is ended. The return code is 8.

Programmer Response: See message IDC3009I in the *System Messages* for specific return code and reason code information. For return code 40 or 44, change the DFDSS filtering to reduce the number of selected CVOL-cataloged data sets to fewer than 1455, or increase the region size if you are selecting fewer than 1455; then rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR498E (xxx)mmmmm(yy), DATA SET dsname CAN NOT BE COPIED/RESTORED. PREALLOCATE THE DATA SET WITH trks TRACKS

Explanation: DFDSS cannot determine the amount of space required to contain the data set before performing the COPY/RESTORE to an unlike device.

System Action: The data set is not processed. The return code is set to 8.

Programmer Response: Preallocate the data set with an allocation of trks tracks on the volumes listed in message ADR396I or ADR442I, or COPY/RESTORE the data set to a like device.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 26, 29, and 36.

ADR500 through ADR699

See Chapter 3, "Stand-Alone DFDSS Restore Messages"

ADR701E (xxx)mmmmm(yy), UNEXPECTED RETURN CODE FROM externalname: xxx-yyy.

Explanation: The named external routine was called, and it returned an unexpected return and/or reason code.

System Action: Processing is ended. The return code is 8.

Programmer Response: This message is not normally expected and indicates a programming or system problem. If *externalname* is STORAGE MANAGEMENT SUBSYSTEM, ensure that SMS is active. If SMS is not active, activate SMS and rerun the job. If SMS is already active, rerun the job with PARM = 'ABEND = 701' on the EXEC statement and a SYSUDUMP DD statement, and contact IBM for programming support.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR702E (xxx)mmmmm(yy), JOBCAT/STEPDAT NOT ALLOWED WHEN PROCESSING SMS MANAGED DATA SET dsname.

Explanation: A JOBCAT or STEPCAT DD statement was specified, and one of the data sets being dumped or copied was SMS managed.

System Action: Task ends with return code 8.

Programmer Response: Eliminate JOBCAT/STEPDAT in the JCL and rerun the job. For those data steps that are not cataloged in the standard search order, specify INCAT and give one catalog name or a list of catalog names.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, and 29.

ADR703E (xxx)mmmmm(yy), DATA SET dsname NOT SELECTED DUE TO MISSING NVR

Explanation: The data set, which is non-VSAM and SMS managed, was not selected because the non-VSAM volume record (NVR) for it cannot be found.

System Action: The data set is not processed. The return code is 8.

Programmer Response: Run Access Method Services DIAGNOSE for the data set, fix the error, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25b, and 29.

ADR704E (xxx)mmmmm(yy), DATA SET dsname ORGANIZATION IS NOT SUPPORTED

Explanation: The following types of data sets are not supported by DFDSS in an SMS environment:

- Data set cataloged in more than one catalog
- CVOLs
- ISAM data sets
- Absolute track allocation
- Unmovable data sets
- VTOCIX/VVDS
- PAGE/SWAP data sets
- SYS1.STGINDEX data set
- BDAM data sets with OPTCD = A.

System Action: The data set is not processed, and the return code is set to 8.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, and 29.

ADR705E (xxx)mmmmm(yy), INVALID (STORAGE CLASS | MANAGEMENT CLASS) NAME {sname | mname} SPECIFIED

Explanation: A nonexistent storage class or management class was specified in the STORCLAS or MGMTCLAS keyword.

System Action: Task ends with return code 8.

Programmer Response: Specify a valid storage class/management class and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR706E (xxx)mmmmm(yy), DUE TO ERRORS, SOURCE DATA SET dsname IS NOW NAMED name

Explanation: During COPY processing, the source data set can be renamed in order to allow allocation of the SMS-managed target data set. If the target allocation fails, an attempt is made to restore the source data set to its original state. If the ALTER of the source data set fails, this message is issued. The data set name indicates the name of the failing data set.

System Action: Processing continues with the next data set. The return code is 8.

Programmer Response: Run Access Method Services DIAGNOSE against the source catalog to determine the reason for the Access Method Services ALTER failure. Restore the original data set name using Access Method Services ALTER NEWNAME.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 26, and 29.

ADR707E (xxx)-mmmm(yy), NOT AUTHORIZED TO USE {BYPASSACS | CONVERTV | INCAT | PROCESS(SYS1) | STORAGE CLASS | MANAGEMENT CLASS} {KEYWORD | sname | mcname} FOR DATA SET dsname

Explanation: Proper RACF authority is required as follows:

- BYPASSACS keyword—RACF FACILITY class authority to bypass ACS.
- CONVERTV keyword—RACF FACILITY class authority to convert volume.
- INCAT keyword—RACF FACILITY class authority for direct catalog call.
- PROCESS(SYS1) keyword—RACF FACILITY class authority to remove restrictions on processing of SYS1 system data sets.
- STORCLAS(sname)—RACF STORCLAS class authority to use storage class sname. The RESOWNER of the data set must have the authority to use the indicated storage class (which may have been chosen for the data set by the ACS routines or by the user if BYPASSACS was specified for the data set).
- MGMTCLAS(mcname)—RACF MGMTCLAS class authority to use management class mcname. The RESOWNER of the data set must have the authority to use the indicated management class (which may have been chosen for the data set by the ACS routines or by the user if BYPASSACS was specified for the data set).

System Action: Other message will indicate action taken. The return code is 8.

Programmer Response: Acquire the proper RACF authority and rerun the job. For BYPASSACS, you must be permitted access to either the STGADMIN.ADR.COPY.BYPASSACS or the STGADMIN.ADR.RESTORE.BYPASSACS profiles. For CONVERTV, you must be permitted access to the STGADMIN.ADR.CONVERTV profile. For INCAT, you must be permitted access to either the STGADMIN.ADR.COPY.INCAT or the STGADMIN.ADR.DUMP.INCAT profiles. For PROCESS(SYS1), you must be permitted access to the STGADMIN.ADR.COPY.PROCESS.SYS, STGADMIN.ADR.DUMP.PROCESS.SYS, or STGADMIN.ADR.RELEASE.PROCESS.SYS profiles. For STORCLAS and MGMTCLAS, the RESOWNER of the data set must be permitted access to the appropriate storage class or management class profile.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 5b, 13, 20, and 29.

ADR708E (xxx)-mmmm(yy), DATA SET dsname WAS NOT PROCESSED BECAUSE IT IS SMS-MANAGED AND JOBCAT/STEP CAT WAS SPECIFIED

Explanation: DFDSS COPY or RESTORE cannot create an SMS-managed data set because JOBCAT or STEP CAT was specified.

System Action: The data set is not processed. The return code is 8.

Programmer Response: Eliminate JOBCAT/STEP CAT in the JCL and rerun the job. For source data sets that are not cataloged in the standard search order, specify INCAT and give one or a list of catalog names.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, and 29.

ADR709E (xxx)-mmmm(yy), AN ERROR OCCURRED IN sms-service WHILE action-description. sms-service MESSAGES FOLLOW

Explanation: An error occurred in the specified SMS service while DFDSS was performing the action described by action description. Error messages produced by the service in error are printed following this message.

SMS service can be any one of the following:

- Automatic Class Selection (ACS)
- VTOC/Data Set Services (VDSS).

Action description can be any one of the following:

- DETERMINING SMS CONSTRUCTS FOR DATA SET dsname [WITH NEWNAME dsname2]
- ALLOCATING DATA SET dsname [WITH NEWNAME dsname2]
- RENAMING DATA SET dsname1 TO dsname2
- SCRATCHING DATA SET dsname.

System Action: The data set is not processed. The return code is 8.

Programmer Response: Refer to the messages following this message to determine the actual error and to correct the problem.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, and 29.

ADR710E (xxx)-mmmm(yy), COPYVOLID REQUIRED TO {RESTORE|COPY} INPUT VOLUME volser1 TO OUTPUT VOLUME volser2

Explanation: The source volume of a FULL volume COPY or RESTORE is SMS managed. In order to perform this operation, the COPYVOLID keyword must be specified.

System Action: The function is ended with a return code of 8.

Programmer Response: If you want to COPY or RESTORE an SMS-managed volume, specify the COPYVOLID keyword in the control statement, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR711I (xxx)-mmmm(yy), DATA SET dsname HAS BEEN ALLOCATED [AS A PDS | AS A PDSE] [WITH NEWNAME dsname2] USING STORCLAS sname {DATA CLAS dname|NO DATA CLAS}, AND {MGMTCLAS mcname|NO MGMTCLAS}

Explanation: If the RENAME parameter was specified, the data set has been allocated with the new name in the storage class listed. The data class and management class assigned to the data set are also listed, if they exist. If the target data set of a PDS or a PDSE is allocated with a different type than the

source, the new target type is listed as well, and the data set is converted to the new target type.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR712W (xxx)mmmmm(yy), THE CATALOG catname SPECIFIED IN RECATALOG IS NOT AVAILABLE

Explanation: The catalog specified in the RECATALOG parameter does not exist or is not available on the system.

System Action: If the target data set is SMS managed, the RECATALOG parameter is ignored, and the system determines the catalog. If the target data set is not SMS managed, then message ADR380E is issued, and the task continues with the next data set. The return code is 4.

Programmer Response: If the target data set is not SMS managed, ensure that the catalog name is correct and available, then rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR713E (xxx)mmmmm(yy), UNABLE TO ALLOCATE SMS MANAGED DATA SET dsname BECAUSE NEITHER DELETE NOR RENAMEU WAS SPECIFIED

Explanation: An attempt to copy a cataloged data set into system-managed storage failed because the source data set was not deleted and the target data set was not renamed.

System Action: The data set is not copied. The return code is 8.

Programmer Response: Specify either DELETE to delete the source data set or RENAMEU to rename the target data set, then rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR714E (xxx)mmmmm(yy), UNABLE TO RESTORE USER CATALOG ucatname BECAUSE THE SYSTEM-SELECTED TARGET VOLUME targetvol DOES NOT MATCH THE SOURCE VOLUME sourcevol

Explanation: A user catalog must be restored to a volume with the same serial number as the volume from which it was dumped. Because the user catalog is SMS-managed, the target volume is determined by the system.

System Action: The user catalog is not restored. The return code is 8.

Programmer Response: Specify the correct DASD volume in the OUTDD,OUTDY parameter, and ensure that the volume is in a storage class with the guaranteed space attribute.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR715W (xxx)mmmmm(yy), UNABLE TO ALTER STATUS OF SMS MANAGED GENERATION DATA SET dsname TO (ACTIVE | ROLLED-OFF). IT WILL BE LEFT IN DEFERRED STATUS.

Explanation: DFDSS was attempting to update the status of an SMS-managed generation data set from *deferred* to either *active* or *rolled-off*, but it was unable to do so because of a VVDS manager error. Message ADR231E describing the VVDS manager error will precede this message.

System Action: The data set is left in *deferred* status. The return code is 4.

Programmer Response: Take the action recommended by message ADR231E. To change the status to *active*, use the Access Method Services command ALTER ROLLIN.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR716W (xxx)mmmmm(yy), UNABLE TO CONVERT USER CATALOG catname (TO | FROM) SMS MANAGEMENT. VOLUME volser HAS BEEN PLACED IN INITIAL STATUS.

Explanation: DFDSS physical data set restore has restored a non-SMS-managed Integrated Catalog Facility user catalog to an SMS-managed volume or an SMS-managed user catalog to a non-SMS-managed volume without performing the appropriate conversion to or from SMS management. To avoid contaminating the volume, it was placed in INITIAL status.

System Action: The user catalog is restored, and the output volume is placed in INITIAL status. The return code is 4.

Programmer Response:

- Connect the user catalog to the master catalog.
- Use the DFDSS CONVERTV command to return the volume to its status before the restore. CONVERTV will also perform the appropriate conversion on the user catalog.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, and 29.

ADR717E (xxx)mmmmm(yy), SYSTEM SERVICES ERROR OCCURRED WHILE action-description. (RETURN INFORMATION IS vv-ii-ww-zzz | ABEND CODE IS nnn) [AND MESSAGES FOLLOW:]

Explanation:

- The system services' version number (vv), module ID (ii), return code (ww), and reason code (zzz) were returned as the result of an error or exception condition while DFDSS was performing the action described by "action-description."
- If the system services abended, nnn is the abend code. See *System Codes* for an explanation of the abend code. DFDSS was performing the action described by "action-description" at the time of the abend.

Action-description can be any one of the following:

- RACF PROTECTING *dsname*
- RACF UNPROTECTING *dsname*
- RENAMING DATA SET *dsname* TO *dsname2*
- OBTAINING INFORMATION FOR DATA SET *dsname*
- OBTAINING VOLUME INFORMATION FOR DATA SET *dsname*
- DETERMINING TARGET CATALOG FOR DATA SET *dsname*
- ALTERING THE CHECKPOINT ATTRIBUTE OF DATA SET *dsname*
- DUMPING DATA SET *dsname*
- COPYING DATA SET *dsname*
- RESTORING DATA SET *dsname*

System Action: The data set is not processed. The return code is 8.

Programmer Response: Refer to IGW01zzz in the *System Messages* following this message to determine the actual error and the measures to be taken to correct the problem.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR718W (xxx)-mmmmm(yy), SYSTEM SERVICES ENCOUNTERED A WARNING CONDITION WHILE action-description. RETURN INFORMATION IS vv-ii-ww-zzz [AND MESSAGES FOLLOW:]

Explanation: The system services' version number (vv), module ID (ii), return code (ww), and reason code (zzz) were returned as the result of a warning condition while DFDSS was performing the action described by "action-description."

Action-description can be any one of the following:

- COPYING DATA SET *dsname*
- RESTORING DATA SET *dsname*
- DUMPING DATA SET *dsname*

System Action: Processing continues with the next data set. The return code is 4.

Programmer Response: Refer to IGW01zzz in *System Messages* following this message to determine the actual error and the measures to be taken to correct the problem.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR719E (xxx)-mmmmm(yy), DATA SET *dsname* CANNOT BE CONVERTED TO { A PDSE | A PDS }.

Explanation: CONVERT keyword with PDS or PDSE subkeyword was specified in the COPY command for the data set. However, it cannot be converted because of the following reason code (rc):

- 01 The target data set is preallocated, and its data set type conflicts with the CONVERT PDS or CONVERT PDSE keyword. For example, if CONVERT PDSE is specified, the preallocated target data set must be a PDSE.
- 02 The target volume for the data set is not SMS-managed.
- 03 PDSE support is not available.
- 04 Conversion to a PDSE was requested with CONVERT PDSE keyword, but the source PDS was a load module. Load modules cannot be converted to a PDSE with DFP Version 3 Release 2.
- 05 Conversion to PDS was requested with the CONVERT PDS keyword, but the target PDS was preallocated as a load module. A PDSE cannot be converted to load modules with DFP Version 3 Release 2.

System Action: The data set is not processed. The return code is 8.

Programmer Response: Correct the problem as indicated by the reason code (rc) and reprocess the data set:

- 01 Rename or delete the preallocated data set.
- 02 Specify an SMS-managed target volume or modify the ACS routine to direct the target allocation to an SMS-managed volume.
- 03 Do one of the following and rerun the job:
 - Use INCLUDE/EXCLUDE/BY filtering so that the data set is not selected
 - Delete the data set.
 - Ensure that SMSX is available on your system.

04 Change or remove the CONVERT PDSE keyword specification so that conversion to a PDSE is not attempted for load modules.

05 Rename the PDSE or delete the preallocated target PDS so that conversion to a load module is not attempted for the PDSE.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR724E (xxx)-mmmmm(yy), ERROR DURING function FOR DATA SET *dsname* [IN CATALOG *catname*], vv-ii-ww-zzz.

Explanation: The version number (vv), module ID (ii), return code (ww), and reason code (zzz) were returned by the catalog services as the result of a catalog error or an exception condition.

System Action: Command processing is ended. The return code is 8.

Programmer Response: See IGW01zzz in *System Messages* for specific return code and reason code definitions.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR725E (xxx)-mmmmm(yy), INTERNAL BUFFERING ERROR ENCOUNTERED FOR DATA SET *dsname* [IN CATALOG *catname*] WHILE INTERFACING WITH SYSTEM SERVICES FOR DATA MOVEMENT

Explanation: An error was detected in the I/O buffer table. This is probably an internal error in DFDSS.

System Action: Processing ends for the data set. Processing continues with the next data set that is not a PDSE. The return code is 8.

Programmer Response: Contact IBM for programming support.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR727E (xxx)-mmmmm(yy), DATA SET *dsname* WAS NOT PROCESSED BECAUSE IT IS NOT CATALOGED IN THE STANDARD ORDER OF SEARCH

Explanation: The data set, which is SMS-managed, was not selected because it was found to be cataloged outside the standard order of search.

System Action: The data set is not processed. The return code is 8.

Programmer Response: Correct the situation and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 26, and 29.

ADR729E (xxx)-mmmmm(yy), UNABLE TO ALLOCATE SUFFICIENT SPACE TO PROCESS DATA SET *dsname*. IT CANNOT BE COPIED/RESTORED

Explanation: DFDSS is unable to allocate sufficient space to contain the data set.

System Action: The data set is not copied/restored. The return code is 8.

Programmer Response: If possible, preallocate the target data set with the appropriate amount of space. Otherwise, contact IBM for programming support.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 26, 29, and 36.

ADR730W (xxx)mmmm(yy), CLUSTER clustname IS OPEN

Explanation: The specified VSAM cluster is in open-for-update status. This condition will not prevent the data set from being processed.

If the function this message was issued against was either data set dump or data set copy, the cluster will be backed up or copied.

Note: The data set may be unusable if the source was in the process of being updated.

If the function this message was issued against was data set restore, the backup copy being restored was open at the time of dump. The restored data set may or may not be usable.

System Action: The cluster will be dumped, copied, or restored if it is selected. The return code is 4.

Programmer Response: For dump or copy, close the cluster and rerun the job if a complete copy is necessary. For restore, use the access method services verify or examine commands to ensure that the cluster is usable.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR731E (xxx)mmmm(yy) THE SIZE OF THE BLOCK OF DATA ACTUALLY READ, NNNNN BYTES, IS NOT A MULTIPLE OF LRECL, MMMMM. DATA SET dsname IS NOT PROCESSED.

Explanation: During a DFDSS data set COPY or RESTORE with REBLOCK specified, an inconsistency was detected between the size of a block of data actually read from the user's data set and the LRECL obtained from the VTOC entry. For a data set with RECFM = FB to be legitimate, all blocks of data in the data set must be an integer multiple of LRECL. All blocks do not have to be the same length, but they must all be an integer multiple of LRECL. One or more blocks of data in the data set named in the message does not meet this criteria.

System Action: The data set is not processed. The return code is 8. Processing continues for other data sets.

Programmer Response: Inspect the data set for any LRECL anomalies, and correct any that you find. Retry without specifying REBLOCK.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, and 29.

ADR732E (xxx)mmmm(yy) DATA SET dsname WAS NOT SELECTED BECAUSE IT IS NOT CATALOGED

Explanation: During DFDSS logical DUMP or COPY with or without INCAT specified, the VSAM data set is not selected because its entry was deleted from the catalog before the DFDSS job was run.

System Action: The data set is not selected. The return code is 8. Processing continues for other data sets.

Programmer Response: Inspect the catalog for the entry of the VSAM data set listed in the message. Use IDCAMS to recatalog the VSAM if the data set is not found.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, and 29.

ADR733I (xxx)mmmm(yy) DATA SET dsname DOES NOT HAVE A RECORD FORMAT OF FIXED OR FIXED BLOCK/HAS A STANDARD USER LABEL. RELATIVE BLOCK ADDRESS PROCESSING WILL NOT BE USED.

Explanation: The keyword RELBLOCKADDRESS was specified, but the data set did not have a record format of fixed or fixed block, or the data set had a standard user label. The data set cannot be processed using relative block address.

System Action: Processing continues without the use of relative block address.

Programmer Response: None.

Operator Response: None.

ADR744W (xxx)mmmm(yy), NO VALID MEMBERS WERE FOUND FOR PDS dsname. ONLY THE DIRECTORY WILL BE UPDATED

Explanation: No members were found in the PDS or all members found had invalid starting TTRs. This message is preceded by message ADR745W for each member with an invalid start TTR.

System Action: Only the directory of the data set is processed. The return code is 4.

Programmer Response: For a data set with no members, no user action is required. For a data set with all invalid member start TTRs, see message ADR745W for direction.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, and 29.

ADR745W (xxx)mmmm(yy), STARTING TTR ttttrr FOR MEMBER memname IN PDS dsname IS INVALID. REASON CODE IS xx

Explanation: The starting TTR of the named member is invalid for one of the following reasons:

- 04 The TTR points to a record before the end of the directory.
- 08 The TTR points to a record after the end of the data set as given in the DSCB1 DS1LSTAR field.
- 12 The record portion of the TTR points to a block after the last valid block on the indicated track.

System Action: The member data is not processed. The directory entry for the member is transferred unchanged to the target data set. The return code is 4.

Programmer Response: The data set does not conform to standard conventions for a PDS. If the data set must be moved as is, rerun the job by specifying the data set in the NOPACKING keyword list and by providing at least one target volume of a like device as the volume on which the data set resides.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, and 29.

ADR746W (xxx)mmmm(yy), END OF DIRECTORY RECORD MISSING FOR PDS dsname

Explanation: The PDS specified does not have an end-of-file (EOF) record for the directory or the EOF is not in the correct place.

System Action: An EOF record is supplied for the directory if the NOPACKING keyword is not specified. Processing continues for this data set. The return code is 4.

Programmer Response: The data set does not conform to standard conventions for a PDS. If the data set must be moved as is, rerun the job by specifying the data set in the

NOPACKING keyword list and by providing at least one target volume of a like device as the volume on which the data set resides.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, 25c, and 29.

ADR747W (xxx)-mmmmm(yy), HIGH MEMBER ENTRY MISSING FOR PDS dsname

Explanation: The PDS specified does not have a high member entry (member name = all 'FF's). This entry signifies the position of the last active member in the directory.

System Action: The high member entry is supplied for the directory if the NOPACKING keyword is not used.

Processing continues for this data set. The return code is 4.

Programmer Response: The data set does not conform to standard conventions for a PDS. If the data set must be moved as is, rerun the job by specifying the data set in the NOPACKING keyword list and by providing at least one target volume of a like device as the volume on which the data set resides.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, 25c, and 29.

ADR748W (xxx)-mmmmm(yy), TTR ttttrr IN MEMBER member OF PDS dsname IS INVALID. BYTE OFFSET IS nnnn IN {DIRECTORY ENTRY | CCHHR cchhrr}. REASON CODE IS xx

Explanation: The user or notelist TTR found in the given PDS was invalid and cannot be translated by DFDSS track-packing logic. The TTR points to a record outside the bounds of the current member being processed or past the end of valid records on a track within the member. The absolute disk address of the source TTR (if in a notelist) and its offset relative to zero within the record or directory entry are given. The reason code describes the specific condition for the invalid TTR as follows:

- 04 The TTR points to a block before the starting block for the member as indicated in the member directory entry.
- 08 The TTR points to a block after the ending block for the member.
- 12 The record portion of the TTR points to a block after the last valid block on the indicated track.
- 16 The record portion of the TTR is zero. Valid blocks begin at one.

System Action: The TTR is copied without change. Processing for this data set continues. The return code is 4.

Programmer Response: The data set does not conform to standard conventions for a PDS. If the data set must be moved as is, rerun the job by specifying the data set in the NOPACKING keyword list and by providing at least one target volume of a like device as the volume on which the data set resides.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, 25c, and 29.

ADR749W (xxx)-mmmmm(yy), END OF FILE RECORD MISSING FOR MEMBER member OF PDS dsname AT TTR ttttrr

Explanation: During the copy or restore of a PDS, an expected end of file record for the member specified was not found before the starting TTR of the next member. The directory shows the next member should start at the given TTR.

System Action: An end of file record is supplied for the member at the TTR given. Processing continues with the next member in the data set. The return code is 4.

Programmer Response: The data set does not conform to standard conventions for a PDS and may contain additional errors that cannot be detected by DFDSS. If the data set must be moved as is, rerun the job by specifying the data set in the NOPACKING keyword list and by providing at least one target volume of a like device as the volume on which the data set resides.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, 25c, and 29.

ADR750E (xxx)-mmmmm(yy), PROCESSING FOR DATA SET dsname IS TERMINATED TO PREVENT DELETION OF THE SOURCE DATA SET

Explanation: An invalid condition or structure was detected in the data set during processing. The DELETE keyword was also specified. In order to preserve the source data set and preclude possible data integrity problems, DFDSS ends the processing after data movement and deletes the target data set. The invalid condition might be:

- Invalid TTRs detected in the directory or member entry.
- Missing EOF record for a member.

System Action: The specified data set was not copied, and the return code is set to 8. Processing continues with the next data set.

Programmer Response: Correct the invalid conditions to have the PDS copied.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, 25c, and 29.

ADR751E (xxx)-mmmmm(yy), (OPEN | WRITE | CLOSE) FAILED WITH RETURN CODE (rc) FOR data set IN AN ATTEMPT TO EXTEND THE DATA SET

Explanation: DFDSS cannot extend the data because the OPEN, WRITE, or CLOSE failed for the data set.

System Action: The data set is not processed. The return code is 8.

Programmer Response: See the *Data Management Macro Instructions* manual for an explanation of return codes.

Operator Response: None.

Problem Determination: See the *Data Management Macro Instructions* manual.

ADR752E (xxx)-mmmmm(yy), (OPEN | WRITE | CLOSE) ABENDED WITH ABEND CODE (rc) FOR data set IN AN ATTEMPT TO EXTEND THE DATA SET

Explanation: DFDSS cannot extend the data because the OPEN, WRITE, or CLOSE for the data set abnormally ended.

System Action: The data set is not processed. The return code is 8.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR753E (xxx)mmmm(yy), UNABLE TO ESTABLISH ESTAE EXIT

Explanation: DFDSS cannot register an ESTAE exit to the system and, therefore, cannot process abends.
System Action: The data set is not processed. The return code is 8.
Programmer Response: None.
Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

Programmer Response: Do one of the following, then rerun the job:

- Specify the DELETE or RECATALOG keyword.
- Add or change the RENAMEU keyword specification to rename all data sets in the sphere.

Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, 25b, 26, and 29.

ADR760W (xxx)mmmm(yy), DATA SET dsname WAS NOT DUMPED WITH SPHERE KEYWORD SPECIFIED

Explanation: An attempt to restore a data set at the sphere level that was not dumped at the sphere level failed.
System Action: Only the data set will be restored (no other sphere information will be restored). The return code is 4.
Programmer Response: Rerun the job without specifying SPHERE.
Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR764E (xxx)mmmm(yy), DATA SETS IN SPHERE OF BASE CLUSTER clusname [WITH NEWNAME clusname2] WERE DIRECTED TO MORE THAN ONE CATALOG

Explanation: The data sets in the sphere were directed by the standard order of search to more than one catalog. The VSAM base cluster, data and index components, AIXs, AIX components, and path names of a sphere must all resolve to the same catalog.
System Action: Processing continues with the next data set. The return code is 8.
Programmer Response: Ensure that all data sets in the sphere are directed to the same catalog, using one or more of the following techniques:

- Adding or deleting user catalog aliases
- Renaming the data sets with the RENAME or RENAMEU keyword.

Rerun the job.
Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, 25b, 26, and 29.

ADR761W (xxx)mmmm(yy), CLUSTER cluster name IN CATALOG catalog name WAS RESTORED WITHOUT SPHERE INFORMATION

Explanation: DFDSS tried to restore a VSAM cluster that was dumped as part of a VSAM sphere. The cluster will be restored, but no relationship information will be created. If the cluster is a base cluster, then any existing alternate indexes might have to be rebuilt. Likewise, if the cluster is an alternate index, it might not properly reflect the current state of its base cluster.
System Action: The data set is restored. The return code is 4.
Programmer Response: If the intent was to restore the entire sphere, rerun the job, specifying the SPHERE keyword.
Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR765W (xxx)mmmm(yy), SPHERE OF BASE CLUSTER clusname IN CATALOG catname IS NOT COMPLETE ON THE DUMP TAPE

Explanation: DFDSS sphere processing detected that not all of the sphere components are on the dump tape. DFDSS is unable to locate all of the alternate indexes related to the named base cluster on the dump tape because of errors encountered when the sphere was dumped. If the sphere was preallocated, the existing alternate indexes that were not on the dump tape will not reflect the current state of the restored base cluster and will need to be rebuilt. The named catalog is the source catalog of the sphere.
System Action: This error is not detected until after the base cluster of the sphere is processed. All alternate indexes on the dump tape are restored, and processing continues with the next data set. The return code is 4.
Programmer Response: Take the appropriate action:

- For non-preallocated data set RESTORE, check the DFDSS messages for the AIXs that were restored, use Access Method Services commands to define the missing AIXs, define paths for them, and build (BLDINDEX) the AIXs.
- For preallocated data set RESTORE, check the DFDSS messages for the AIXs that were restored, and use the Access Method Services command BLDINDEX to rebuild the existing alternate indexes that were not restored.

Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, 20, 25b, 25c, and 29.

ADR762W (xxx)mmmm(yy), AIX aixname OF BASE CLUSTER clusname ALREADY EXISTS RELATED TO BASE CLUSTER clusname IN CATALOG catname

Explanation: During DFDSS sphere processing in which REPLACE was specified but the preallocated target was not, an alternate index with the same name was found in the target catalog. This alternate index is related to a base cluster that is different from the source base cluster.
System Action: The base cluster was successfully processed, but the specified AIX was not processed. Processing continues with the next AIX. The return code is 4.
Programmer Response: Use Access Method Services commands to DEFINE the AIX with a different name, DEFINE a path, and build (BLDINDEX) the alternate index.
Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, 20, 25b, 25c, and 29.

ADR763E (xxx)mmmm(yy), DATA SETS IN SPHERE OF BASE CLUSTER clusname [WITH NEWNAME clusname2] WERE NOT ALL RENAMED.

Explanation: When copying a sphere without specifying DELETE or RECATALOG, you must rename all parts of the sphere. If the target sphere is to be SMS managed, the data sets must be renamed even if RECATALOG was specified, because RECATALOG is ignored for SMS-managed data sets.
System Action: Processing continues with the next data set. The return code is 8.

ADR766E (xxx)mmmmm(yy), CLUSTER clusname [WITH NEWNAME clusname2] CANNOT BE MOVED TO {AN SMS | A NONSMS} TARGET BECAUSE IT IS PART OF {AN SMS | A NONSMS} MANAGED SPHERE [IN A DIFFERENT STORAGE GROUP]

Explanation: When DFDSS copies with DELETE between SMS and NONSMS or between different storage groups in SMS, all parts of a VSAM sphere must be included.

System Action: Processing continues with the next data set. The return code is 8.

Programmer Response: Do one or more of the following, then rerun the job:

- Specify RENAMEU or RECATALOG instead of DELETE.
- Specify SPHERE to include all parts of the sphere.
- Specify all parts of the sphere in the data set INCLUDE parameter.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25b, 25c, 26, and 29.

ADR768W (xxx)mmmmm(yy), THE FOLLOWING SPHERE(S) COULD NOT BE COMPLETELY PROCESSED DUE TO ERRORS

Explanation: During DFDSS sphere processing, one or more components of a sphere could not be successfully processed. The message lists the status of each piece of the sphere.

System Action: If possible, processing continues with the next sphere component for COPY and RESTORE, or the next data set for DUMP. If the command was DUMP, processing for each individual sphere halts if an error was detected with an AIX. The return code is 8.

Programmer Response: Check other DFDSS messages for the components that failed. After correcting the error conditions, run the job again.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25b, 25c, and 29.

ADR769E (xxx)mmmmm(yy), PROCESSING OF DATA SET dsname TERMINATED BY UIM EXIT (nn)

Explanation: The indicated User Interaction Module exit nn has ended processing for the named data set before normal completion. If the SPHERE keyword is specified and the named data set is a base cluster, processing for all related AIXs is also ended.

System Action: Processing continues with the next data set. The return code is 8.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, 25b, 26, and 29.

ADR770W (xxx)mmmmm(yy), AIX aixname RELATED TO BASE CLUSTER clustname WAS NOT TERMINATED AS REQUESTED BY UIM EXIT (nn)

Explanation: The indicated User Interaction Module exit (nn) requested that processing for the named alternate index be ended. The request was denied because the SPHERE keyword was specified and the alternate index is related to the named base cluster that was selected for processing.

System Action: Processing continues for the named alternate index. The return code is 4.

Programmer Response: Take one or more of the following actions:

- Copy only the base cluster by omitting the SPHERE keyword.
- Change the UIM exit so that termination is not attempted for the alternate index.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, 25b, 26, and 29.

ADR772W (xxx)mmmmm(yy), UIM EXIT (nn) IS NOT AUTHORIZED TO BYPASS SECURITY PROCESSING FOR DATA SET dsname

Explanation: The indicated User Interaction Module exit (nn) attempted to bypass RACF and security processing for the named data set. The request was denied because the application program was not authorized to bypass security because the JSCBPASS flag was off in the job step control block.

System Action: Processing continues with normal security processing for the data set. The return code is 4.

Programmer Response: Alter the application so that it is authorized to bypass security processing by ensuring that the JSCBPASS flag is on in the job step control block, or change the UIM exit so that it does not attempt to bypass security processing.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, 25b, 26, and 29.

ADR773E (xxx)mmmmm(yy), PREALLOCATED TARGET DATA SET dsname [IN CATALOG catname] ON VOLUME volser WAS NOT SCRATCHED AND REALLOCATED BECAUSE DFDSS DID NOT SERIALIZE THE DATA SET, rc

Explanation: DFDSS did not delete and reallocate a preallocated target data set because DFDSS did not serialize the target data set. DFDSS needed to delete and reallocate the preallocated target because one or more target data set attributes did not match the source. The reason code (rc) is as follows:

- 1 The target KSDS on an unlike device was not reusable.
 - 3 The CI size of the target did not match the source.
 - 4 The IMBED attribute of the target did not match the source.
 - 5 The KEY length of the target did not match the source.
 - 6 The Record length of the target did not match the source.
 - 7 The REPLICATE attribute of the target did not match the source.
 - 8 The SPANNED attribute of the target did not match the source.
 - 9 The size of the target data set was smaller than the size of the source data set.
- B The source data set is empty.

DFDSS probably bypassed serialization because UIM exit 22 requested that serialization be bypassed for the data set.

System Action: The data set is not processed. Processing continues with the next data set. The return code is 8.

Programmer Response: Do one of the following, then rerun the job:

- Delete the preallocated target and allocate it with attributes that match the source data set.
- Delete the preallocated target and let DFDSS allocate it.

- Change UIM exit 22 so that it does not bypass serialization.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, 25b, 26, and 29.

ADR774E (xxx)-mmmmm(yy), DATA SET dsname CANNOT BE PROCESSED BECAUSE CONVERSION TO A (PDS |PDSE) WAS NOT REQUESTED, rc

Explanation: The named data set cannot be processed because the CONVERT keyword was not specified. The possible reason codes (rc) are:

- 04 The source data set is a PDSE, but there is a usable preallocated PDS data set on the target volume, and CONVERT with PDS subkeyword was not specified for the data set.
- 08 The source data set is a PDS, but there is a usable preallocated PDSE data set on the target volume, and CONVERT with PDSE subkeyword was not specified for the data set.
- 12 The data set being processed is a PDSE, and the target volume for the data set is not SMS-managed. A PDSE must be SMS-managed.
- 16 A PDSE is being copied to an MVS/DFP Version 3 Release 2 (or subsequent releases) system that does not have MVS/SP Version 3 Release 1.0E. Allocation of PDSE requires at least MVS/DFP Version 3 Release 2 and MVS/SP Version 3 Release 1.0E.

System Action: The data set is not processed. The return code is 8.

Programmer Response: Correct the problem as indicated by the reason code (rc) and reprocess the data set:

- 04 If the source PDSE is to be copied to the preallocated target PDS, specify CONVERT with PDS subkeyword; otherwise, delete or rename the preallocated target PDS.
- 08 If the source PDS is to be copied to the preallocated target PDSE, specify CONVERT with PDSE subkeyword; otherwise, delete or rename the preallocated target PDSE.
- 12 If the PDSE is to be copied to a non-SMS managed volume, specify CONVERT with PDS subkeyword; otherwise, specify an SMS-managed target volume, or modify the ACS routine to direct the target allocation to an SMS-managed volume.
- 16 Specify CONVERT with PDS subkeyword to copy and convert the PDSE to PDS, or copy the data set on a system that has at least MVS/DFP Version 3 Release 2 and MVS/SP Version 3 Release 1.0E.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR780I (xxx)-mmmmm(yy), THE INPUT DUMP DATA SET BEING PROCESSED IS IN (FULL VOLUME | TRACKS | PHYSICAL DATA SET | LOGICAL DATA SET) FORMAT

Explanation: The dump data set now being processed was created using the indicated format.

System Action: Processing continues.

Programmer Response: None. For information on logical and physical processing see *Data Facility Data Set Services: User's Guide*.

Operator Response: None.

Problem Determination: None.

ADR781I (xxx)-mmmmm(yy), DATA SET dsname [IN CATALOG catname] HAS AN UNDEFINED DSORG. TRACK LEVEL I/O WILL BE USED TO PROCESS THE DATA SET TO AN UNLIKE DEVICE

Explanation: The data set has an undefined data set organization and is being moved to an unlike target device. Track image data movement will be used. This results in an exact track-for-track image of the source data set on the target volume.

System Action: Processing continues.

Programmer Response: None

Operator Response: None

Problem Determination: None

ADR782W (xxx)-mmmmm(yy), DATA SET dsname HAS AN INVALID DSORG. VSAM PROCESSING IS BYPASSED

Explanation: The VVDS data set has an invalid data set organization, and VSAM processing will not be performed.

System Action: Processing continues. The return code is 4.

Programmer Response: Notify Systems Support.

Operator Response: None

Problem Determination: None

ADR799E (xxx)-mmmmm(yy), AN UNEXPECTED ERROR HAS OCCURRED [WHILE PROCESSING (DATA SET datasetname | VOLUME volser)]

Explanation: DFDSS detected an unexpected error during processing. It was possibly an internal error.

System Action: If possible, DFDSS attempts to continue processing; otherwise, the task ends. The return code is 8.

Programmer Response: Contact IBM for programming support.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 26, and 29.

ADR860I (xxx)-mmmmm(yy), PROCESSING BEGINS ON VOLUME volser

Explanation: DFDSS began processing the named volume.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR861T (xxx)-mmmmm(yy), REQUIRED SUBSYSTEM subsys IS UNAVAILABLE. TASK IS TERMINATED

Explanation: The indicated subsystem is not installed. It is required to perform the CONVERTV function.

System Action: The job is ended. The return code is 12.

Programmer Response: Verify that the indicated subsystem is installed, and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 1, 3, 4, 13, 20, 25a, 25b, and 29.

ADR862I (xxx)mmmmm(yy), THE REDETERMINE KEYWORD WAS SPECIFIED. ALL DATA SETS WILL BE REPROCESSED

Explanation: The REDETERMINE keyword was specified in the SMS job stream. The SMS constructs for all of the data sets will be reset, including those data sets already SMS managed.

System Action: None.
Programmer Response: None.
Operator Response: None.
Problem Determination: None.

ADR863I (xxx)mmmmm(yy), THE TEST KEYWORD WAS SPECIFIED. NO DATA SETS OR VOLUMES WILL BE CONVERTED

Explanation: The TEST keyword was specified in the SMS job stream. Subtasks will run in the NORUN mode.

System Action: No volumes or data sets will be converted, but volume and data set eligibility checking will be done. A report will be issued indicating whether SMS processing would be successful.

Programmer Response: None.
Operator Response: None.
Problem Determination: None.

ADR864E (xxx)mmmmm(yy), JOBCAT/STPCAT IS NOT ALLOWED WHEN CONVERTING VOLUMES TO OR FROM SMS MANAGEMENT. TASK IS TERMINATED

Explanation: A JOBCAT or STPCAT DD statement was encountered. These statements are not allowed in the SMS environment.

System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

Programmer Response: Remove the JOBCAT or STPCAT DD statement. Use the INCAT keyword if it is necessary to specify an input catalog. Rerun the job.
Operator Response: None.
Problem Determination: Table I, items 1, 3, 4, 13, 20, and 29.

ADR868I (xxx)mmmmm(yy), THE FOLLOWING DATA SETS ON VOLUME volser HAD ('*') CANDIDATE VOLUMES WHICH WERE {DELETED| CONVERTED TO '*')}

Explanation: During conversion to SMS, non-VSAM data sets with candidate volumes have the candidate volumes converted to nonspecific '*' volumes. During conversion from SMS, both VSAM and non-VSAM data sets have their nonspecific '*' volumes deleted.

System Action: None.
Programmer Response: None.
Operator Response: None.
Problem Determination: None.

ADR869I (xxx)mmmmm(yy), THE FOLLOWING DATA SETS ON VOLUME volser WERE NOT CATALOGED IN THE STANDARD ORDER OF SEARCH. THEY {WERE|WILL BE} CATALOGED IN THE NAMED CATALOG

```
dsname      CATALOG: new catalog name
            [DELETED FROM: old catalog name]
```

Explanation: Catalog entries were not found for each of the listed data sets in the standard order of search. The listed VSAM data sets were deleted from their original catalog and recataloged in the standard order of search. For non-VSAM

data sets, if the INCAT keyword was specified and an entry for the non-VSAM data set was found in one of the catalogs listed with this keyword, then that entry was deleted from this catalog and the data set was recataloged in the standard order of search.

If INCAT was not specified or no entry was found in any of the catalogs listed, then the system assumed that the non-VSAM data set was uncataloged and cataloged it in the named catalog. In this case, the DELETED FROM line will not be issued.

System Action: None.
Programmer Response: None.
Operator Response: None.
Problem Determination: None.

ADR870I (xxx)mmmmm(yy), PREPARE PROCESSING {WILL BE|WAS} SUCCESSFUL. VOLUME volser {WILL NOW BE|IS NOW} IN INITIAL STATUS

Explanation: PREPARE processing for the named volume completed successfully, and the volume is now in INITIAL status. No data sets can be allocated or deleted on the volume. If TEST was specified in the job stream, this message indicates whether the task will be successful if actually run. The volume status is not changed with TEST.

System Action: None.
Programmer Response: None.
Operator Response: None.
Problem Determination: None.

ADR871W (xxx)mmmmm(yy), VOLUME volser MUST BE ASSIGNED TO A STORAGE GROUP BEFORE SMS CONVERSION CAN PROCEED

Explanation: This message is issued with a PREPARE task if the named volume is eligible for SMS conversion but is not assigned to an SMS storage group. The volume will be successfully placed in INITIAL status but must be assigned to a storage group before the conversion to SMS management can be completed.

System Action: None. The return code is 4.
Programmer Response: Change the active control data set so that the volume is defined to a storage group before attempting to convert the volume to SMS management.
Operator Response: None.
Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR872E (xxx)mmmmm(yy), VOLUME volser IS ALREADY IN {INITIAL|CONVERTED} STATUS. PREPARE PROCESSING {WILL FAIL|FAILED}

Explanation: PREPARE processing for the named volume failed because the volume was already in INITIAL status.

System Action: Processing of that volume is ended. Processing continues with the next volume in the input volume list. The return code is 8.
Programmer Response: None.
Operator Response: None.
Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25a, 25b, and 29.

ADR873I (xxx)mmmmm(yy), VOLUME volser [IN STORAGE GROUP storgrp] IS ELIGIBLE FOR CONVERSION (TO|FROM) SMS MANAGEMENT

Explanation: DFDSS determined that the named volume is eligible for SMS/non-SMS processing. Conversion of the data sets on that volume can now proceed. *Storgrp* is the SMS storage group in which the volume is defined. The storage group is listed only when a volume is converted to SMS management.

System Action: The system proceeds with converting all of the data sets on the named volume either to or from SMS management.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR874E (xxx)mmmmm(yy), VOLUME volser IS NOT ELIGIBLE FOR CONVERSION (TO|FROM) SMS MANAGEMENT, rc

Explanation: The named volume failed the eligibility requirements for conversion either to or from SMS management. *rc* is the reason code for the ineligible volume:

- 04 The volume is not a DASD device.
- 08 The volume has a nonindexed VTOC.
- 12 The volume is not defined to an SMS storage group.
- 16 The volume is in unknown status. The SMS-converted flag is on but the SMS-initial flag is off in the VTOC index map.
- 20 The volume is in unknown status. The SMS-converted flag is on but the SMS-initial flag is off in VTOC entry.
- 24 SMS indicators for VTOC index map and VTOC entry do not match.

System Action: Conversion of the volume failed. Processing continues with the next volume in the input list. The return code is 8.

Programmer Response: Take the appropriate action for each reason code:

- 04 Do not attempt to convert non-DASD volumes to SMS management.
- 08 Define an index for the VTOC, and rerun the job.
- 12 Change the ACDS so that the volume is defined to a storage group, and rerun the job.
- 16 Contact IBM for programming support.
- 20 Contact IBM for programming support.
- 24 Contact IBM for programming support.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25a, 25b, and 29.

ADR875E (xxx)mmmmm(yy), VOLUME volser IS ALREADY IN CONVERTED STATUS AND REDETERMINE WAS NOT SPECIFIED. SMS PROCESSING (WILL FAIL|FAILED)

Explanation: The named volume was already in SMS status, and the REDETERMINE keyword was not specified in the DFDSS job stream. Therefore, SMS processing of that volume is ended.

System Action: Processing continues with the next volume in the input volume list. The return code is 8.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR876E (xxx)mmmmm(yy), VOLUME volser IS NOT IN INITIAL OR CONVERTED STATUS. NONSMS PROCESSING (WILL FAIL|FAILED)

Explanation: Because the named volume was already in non-SMS status, non-SMS processing of the volume is ended.

System Action: Processing continues with the next volume in the input volume list. The return code is 8.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25a, 25b, and 29.

ADR877I (xxx)mmmmm(yy), THE FOLLOWING DATA SETS ON VOLUME volser (WILL BE|WERE) SUCCESSFULLY PROCESSED

```

dsname                                CATALOG: catname
[STORCLAS: scname                    MGMTCLAS: mcname]
[VOLUMES: vol001 vol002 vol003 . . .]
[                                vol100x vol100y vol100z . . .]
[AIX: aixname]
[PATH: pathname]
" " " " " "
    
```

Explanation: The listed data sets were successfully processed either to or from SMS management.

- At the end of processing they are cataloged in the named catalog.
- If the task was conversion to SMS management, the storage and management classes in which the data set is defined are listed.
- If the data set is multivolume, the volumes on which the data set resides are listed.
- If the data set has any AIXs or PATHs defined for it, these are listed. If any of the AIXs are multivolume, then the volumes on which they reside are listed.

If TEST was specified, the data sets were not actually processed. The message indicates that data sets would be successfully processed if the job was run.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR878E (xxx)mmmmm(yy), THE FOLLOWING DATA SETS ON VOLUME volser (WILL NOT BE|WERE NOT) SUCCESSFULLY PROCESSED

```

dsname                                CATALOG: catalog name
REASON: rc,mmmmm(yy)[,volser][,reason]
[.,catlg func,catlg rsn code,catlg mod id,
 parmlist rsn code,catlg service id]
[.,VVR func,VVDS manager rsn code]
" " " " " "
AIX: aix name rc[,volser][,reason]
[.,catlg func,catlg rsn code,catlg mod id,
 parmlist rsn code,catlg service id]
[.,VVR func,VVDS manager rsn code]
" " " " " "
PATH: path name rc
[.,catlg func,catlg rsn code,catlg mod id,
 parmlist rsn code,catlg service id]
" " " " " "
" " " " " "
    
```

Explanation: The listed data set (or its associated AIX or PATH) failed conversion either to or from SMS management, where "rc" is the reason code for the data set failing. If the data set is multivolume and the error occurred on a volume other than the one being processed by CONVERTV at that time, then the volser of this other volume is printed. If a multivolume data set fails conversion because one of the other volumes that the data set is on is ineligible for SMS processing, then this reason is printed after the volser. If the error occurred in a catalog call, then the catalog function code, catalog reason code, catalog module ID, and parmlist reason code are printed. Refer to MSGIDC3009I in *System Messages* for an explanation of the error, where:

```
MSGIDC3009I return code = MSGADR878E catlg
rsn code
MSGIDC3009I reason code = MSGADR878E
parmlist rsn code
MSGIDC3009I module ID = MSGADR878E catlg
mod ID
```

If the error occurred in a VVR REQUEST function call, then the VVR function code that failed and the VVDS manager reason code are printed. Refer to Message IDC3009I in *System Messages* under return code 50 for an explanation of the VVDS manager reason code.

The reason codes are:

- | | |
|---|---|
| <p>02 The data set is not supported in an SMS environment (ISAM, ABSTR, unmovable, model DSCB, CVOL, nonintegrated catalog facility VSAM catalog, or nonintegrated catalog facility VSAM data set).</p> <p>03 The data set is not supported in a non-SMS environment (PDSE data set).</p> <p>04 The data set is an uncataloged GDS.</p> <p>06 The data set is a cataloged GDS, but it is not cataloged in the standard order of search.</p> <p>08 The data set is a GDS with nonspecific volumes.</p> <p>10 The data set is a GDS with candidate volumes.</p> <p>12 ACS routines did not return SMS constructs for the data set.</p> <p>14 A null storage class was returned for the data set.</p> <p>16 The storage class definition cannot be retrieved for the data set.</p> <p>18 The data set is multivolume, extents on volumes <i>volser</i>s are not specified in the volume list, and ALLMULTI is not specified.</p> <p>20 The data set is multivolume and the first extent of the data set is on the volume <i>volser</i> that was not specified in the DDNAME/DYNAM list.</p> <p>21 The data set is multivolume and the first extent of an AIX of the data set is on the volume <i>volser</i> that was not specified in the DDNAME/DYNAM list.</p> <p>22 The data set is multivolume and has extents on volumes <i>volser</i>s that are in different storage groups.</p> <p>24 The data set has candidate volumes <i>volser</i>s that are in different storage groups.</p> <p>26 The data set is multivolume and has extents on volumes <i>volser</i>s that are ineligible for conversion for the reasons given. (Refer to message ADR874E for volume reason code.)</p> <p>28 The data set has candidate volumes <i>volser</i>s that are ineligible for conversion for the reasons given. (Refer to message ADR874E for volume reason code.)</p> | <p>30 The data set is multivolume and is not cataloged.</p> <p>32 The data set is not cataloged in the standard search order, and the CATALOG keyword was not specified.</p> <p>34 The data set is not cataloged in the standard search order, and a duplicate entry was found in another catalog.</p> <p>36 The data set is not cataloged in the standard search order, and a duplicate entry was found in a standard order of search catalog.</p> <p>38 The VSAM base cluster is in a sphere where all data sets are not convertible to SMS management.</p> <p>40 The catalog to which the AIX's sphere is directed is different from the catalog to which the AIX's related base cluster is directed. All components must be in the same catalog.</p> <p>42 The catalog to which the path's sphere is directed is different from the catalog to which the path's related base cluster is directed. All components must be in the same catalog.</p> <p>44 ALTER failed during an attempt to update the catalog entry for the data set.</p> <p>46 DELETE NOSCRATCH failed during an attempt to delete the catalog entry for the data set.</p> <p>48 DEFINE RECATALOG failed during an attempt to recatalog the data set.</p> <p>49 DEFINE ALIAS failed during an attempt to define the aliases for the data set.</p> <p>50 DEFINE CATALOG failed during an attempt to catalog the data set.</p> <p>52 CATALOG NAME LOCATE failed during an attempt to determine where the data set should be cataloged according to standard order of search.</p> <p>54 DEFINE PATH failed during an attempt to define a path.</p> <p>56 LOCATE failed.</p> <p>58 The VSAM base cluster is in a sphere that includes data sets with catalog errors.</p> <p>60 Read non-VSAM volume record (NVR) failed in the VVDS manager.</p> <p>62 Get for update of non-VSAM volume record (NVR) failed in the VVDS manager.</p> <p>64 Put for update of non-VSAM volume record (NVR) failed in the VVDS manager.</p> <p>66 Delete of non-VSAM volume record (NVR) failed in the VVDS manager.</p> <p>68 Insert of non-VSAM volume record (NVR) failed in the VVDS manager.</p> <p>70 Read VSAM volume record (VVR) failed in the VVDS manager.</p> <p>72 Get for update of VSAM volume record (VVR) failed in the VVDS manager.</p> <p>74 Put of VSAM volume record (VVR) failed in the VVDS manager.</p> <p>76 The VSAM base cluster is in a sphere that includes data sets with VVR errors.</p> <p>78 Read of the VTOC failed during an attempt to update the SMS indicator for the data set.</p> |
|---|---|

- 80 Rewrite of the VTOC failed during an attempt to update the SMS indicator for the data set.
- 82 The VSAM base cluster is in a sphere that includes data sets with VTOC entry update errors.
- 84 The VSAM component does not have a corresponding VTOC entry.
- 86 The non-VSAM data set is SMS managed, but its NVR cannot be found.
- 88 Storage class authorization for the data set failed.
- 90 Management class authorization for the data set failed.
- 92 Both storage class and management class authorization for the data set failed.
- 94 Unable to get extent information from the VTOC.
- 96 The data set failed enqueue.

System Action: Processing continues with the next data set. The volume will not be successfully converted to or from SMS management. The return code is 8.

Programmer Response:

- 02 - 10 Move or delete the data set from the volume.
- 12 Update ACS routines to return SMS constructs for the data set.
- 14 - 16 Verify ACS routines to ensure that the data set is defined to a storage class.
- 18 (1) Add the other volumes that the data set resides on to the volume list, or (2) specify the ALLMULTI keyword, or (3) move or delete the data set off the volume you are trying to convert.
- 20 (1) Add the primary volume of the data set to the volume list, or (2) move or delete the data set off the volume you are trying to convert.
- 21 (1) Add primary volume of the AIX to the volume list, or (2) move or delete the data set off the volume you are trying to convert.
- 22 Change your active control data set so that all the volumes that the data set resides on are defined to the same storage group.
- 24 (1) Delete the candidate volumes from the data set's catalog entry, or (2) change your ACDS so that all the candidate volumes are defined to the same storage group as the volumes on which the data set resides.
- 26 - 28 Refer to message ADR874E for required action.
- 30 (1) Catalog the data set, or (2) move or delete the data set from the volumes that you are trying to convert.
- 32 (1) Specify the CATALOG keyword, or (2) recatalog the data set into the standard order of search.
- 34 (1) Catalog the data set into the standard order of search, or (2) rename the data set, or (3) move or delete the data set from the volumes that you are trying to convert.
- 36 (1) Rename the data set, or (2) move or delete the data set from the volumes that you are trying to convert.
- 38 - 42 (1) Rename the offending base cluster, the data or index component, AIX, AIX data or index component, or PATH to be directed to the same catalog as the VSAM base cluster, or (2) delete or

add catalog aliases so all parts of the sphere are alias-directed to the same catalog.

- 44 - 86 Contact IBM for programming support.
- 88 - 92 (1) Make sure that the RESOWNER of the data set is authorized to use the storage and/or management class derived for the data set, or (2) update the ACS routines to select a different storage and/or management class for the data set.
- 94 See previous error message associated with the data set for specific action.
- 96 Ensure that no other jobs are accessing data sets on volumes to be converted, then rerun the CONVERTV job.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25a, 25b, and 29.

ADR879I (xxx)mmmm(yy), THE FOLLOWING DATA SETS ON VOLUME volser WERE ALREADY IN (NON-)SMS STATUS

```
dsname          CATALOG: catname
[STORCLAS: scname  MGMTCLAS: mcname]
[VOLUMES: vol001 vol002 vol003 . . . ]
[          vol00x vol00y vol00z . . . ]
[AIX: aixname]
[PATH: pathname]
```

Explanation: The listed data sets were not processed because they were already in the desired SMS status.

- The data sets are cataloged in the named catalog.
- If the task was conversion to SMS management, the storage and management classes in which the data set is defined are listed.
- ***** Asterisks (*) shown in the place of storage and management class names indicate that no primary VVR was found for the cluster.
- Dashes (-) shown in the place of storage and management class names indicate incorrectly converted old format VVRs. Refer to message ADR409E.
- If the data set is multivolume, the volumes on which the data set resides are listed.
- If the data set has any AIXs or PATHs defined for it, these are listed. If any of the AIXs are multivolume, then the volumes on which they reside are listed.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR880I (xxx)mmmm(yy), VOLUME volser IS EMPTY. NO DATA SETS WERE CONVERTED.

Explanation: Data set conversion was not required because the named volume did not have any data sets defined on it.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR881I (xxx)mmmm(yy), VOLUME volser CONTAINS THE FOLLOWING TEMPORARY DATA SETS

dsname
"
"

Explanation: The named volume contains the temporary data sets listed.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR882W (xxx)mmmm(yy), THE FOLLOWING DATA SETS ON VOLUME volser (WILL NOT BE|WERE NOT) SUCCESSFULLY PROCESSED, BUT WERE ALREADY IN (NON)SMS STATUS.

```
dsname          CATALOG: catalog name
REASON: rc,mmmm(yy)[,volser][,reason]
              [,catlg func,catlg rsn code,catlg mod id,
              parmlist rsn code,catlg service id]
              [,VVR func,VVDS manager rc]
              "
AIX: aix name rc[,volser][,reason]
      [,catlg func,catlg rsn code,catlg mod id,
      parmlist rsn code,catlg service id]
      [,VVR func,VVDS manager rc]
      "
PATH: path name rc
      [,catlg func,catlg rsn code,catlg mod id,
      parmlist rsn code,catlg service id]
      "
"
```

Explanation: An error was found while the listed data set (or its associated AIX or PATH) was being processed, but the data set was already converted, so this error will not affect the volume being successfully converted. "rc" is the reason code for the data set's failing. Refer to message ADR878E for a full explanation of the data set error information returned and the reason codes.

System Action: Processing continues with the next data set. The return code is 4.

Programmer Response: Refer to message ADR878E.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25a, 25b, and 29.

ADR885I (xxx)mmmm(yy), VOLUME volser (WILL BE|HAS BEEN) SUCCESSFULLY CONVERTED (TO|FROM) SMS MANAGEMENT

Explanation: All data sets on the named volume were successfully converted to either SMS or non-SMS status. The SMS flags in the VTOC entry and the VTOC index map for the volume were successfully updated to the appropriate status. If TEST was specified, this message indicates that processing would have been successful if the job had actually been run. No changes occur to the volume under TEST.

System Action: Processing continues with the next volume in the input volume list.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR886E (xxx)mmmm(yy), DATA SETS EXIST ON VOLUME volser WHICH ARE NOT CONVERTIBLE (TO|FROM) SMS MANAGEMENT. THE VOLUME (WILL BE|WAS) (PLACED|LEFT) IN INITIAL STATUS

Explanation: The named volume failed SMS/non-SMS conversion because of data sets that cannot be converted existing on the volume. The volume is left in INITIAL status at the end of processing. If TEST was specified, this message indicates that processing would have failed if the job had actually been run. No changes occur to the volume under TEST.

System Action: Processing continues with the next volume in the input volume list. The return code is 8.

Programmer Response: Appropriate responses are:

- For SMS processing, the problem can be solved by one of the following:
 - Adding the CATALOG, INCAT, and/or ALLMULTI keywords to the DFDSS job stream, and rerunning the job.
 - Running a non-SMS job to place the volume back in non-SMS status, deleting or moving the data sets that caused the processing to fail, and rerunning the SMS job.
- For non-SMS processing, the problem can be solved by one of the following:
 - Adding the ALLMULTI keyword to the DFDSS job stream, and rerunning the job.
 - Running an SMS job to place the volume back in SMS status, deleting or moving the data sets that caused the processing to fail, and rerunning the non-SMS job.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25a, 25b, and 29.

ADR891E (xxx)mmmm(yy), (CONVERSION|PREPARATION) OF VOLUME volser FAILED, rc

Explanation: The SMS/NONSMS/PREPARE job failed on the named volume because of an error in trying to update the VTOC entry or VTOC index map SMS status flags. rc is the reason code for the failure.

- 04 Read of the VTOC entry failed during an attempt to update the SMS indicators to INITIAL.
- 08 Read of the VTOC entry failed during an attempt to update the SMS indicators after processing data sets.
- 12 Rewrite of the VTOC entry failed during an attempt to update the SMS indicators to INITIAL.
- 16 Rewrite of the VTOC entry failed during an attempt to update the SMS indicators to NONSMS.
- 20 Rewrite of the VTOC entry failed during an attempt to update the SMS indicators to SMS.
- 24 Read of the VTOC index map failed during an attempt to update the SMS indicators to INITIAL.
- 28 Read of the VTOC index map failed during an attempt to update the SMS indicators after processing data sets.
- 32 Rewrite of the VTOC index map failed during an attempt to update the SMS indicators to INITIAL.
- 36 Rewrite of the VTOC index map failed during an attempt to update the SMS indicators to NONSMS.

- 40 Rewrite of the VTOC index map failed during an attempt to update the SMS indicators to SMS.
- 44 An attempt to determine if a VVDS exists on the volume failed.
- 48 An attempt to add a VVDS to the volume failed.

System Action: Processing continues with the next volume in the input volume list. The return code is 8.

Programmer Response: Contact IBM for programming support.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, and 29.

ADR892I (xxx)-mmmmm(yy), THE STATUS OF EACH VOLUME (WILL BE) AS FOLLOWS

VOLUME	FINAL STATUS	REASON FOR FAILURE
volser - PREPARED	INITIAL	
volser - CONVERTED	SMS NONSMS INITIAL	
volser - FAILED	SMS NONSMS INITIAL	VOLUME NOT ELIGIBLE
volser - FAILED	SMS NONSMS INITIAL	DATA SETS NOT ELIGIBLE
volser - FAILED	SMS NONSMS INITIAL	VTOC UPDATE FAILED
volser - FAILED	SMS NONSMS INITIAL	VVDS ERROR

Explanation: After all of the volumes in the CONVERTV job are processed and individual volume and data set processing messages are printed, DFDSS issues this report summary message, which lists all of the volumes and their final status. If conversion of a volume failed, the reason for failure is also indicated.

System Action: None.

Programmer Response: If any of the volumes failed processing, refer to that volume's specific processing messages to determine the reason for failure and corrective action for the error.

Operator Response: None.

Problem Determination: None.

ADR893I (xxx)-mmmmm(yy), THE FOLLOWING VOLUMES CONTAIN PIECES OF MULTIVOLUME DATA SETS WHICH WERE PROCESSED DURING THE CONVERTV TASK. THE VOLUMES ARE IN THE INDICATED STATUS

volser - SMS|INITIAL|NONSMS|UNKNOWN

Explanation: Some data sets processed during the CONVERTV job were multivolume and had extents on volumes not in the input volume list. For SMS processing, the SMS status of these volumes is not altered. For non-SMS processing, the volumes are set to INITIAL if at least one data set with extents on those volumes was successfully processed.

Note: If this message is issued, the user may have pieces of SMS-managed data sets on non-SMS volumes. These volumes must also be converted to SMS management.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR894E (xxx)-mmmmm(yy), ERRORS OCCURRED WHILE TRYING TO PLACE THE FOLLOWING VOLUMES IN INITIAL STATUS

volser - reason code

Explanation: Some data sets processed during the NONSMS CONVERTV job were multivolume and had extents on volumes not in the input volume list. An error occurred in trying to set the SMS status of these listed volumes to INITIAL. Refer to message ADR891E for the explanation of the reason code.

System Action: None. The return code is 8.

Programmer Response: Contact IBM for programming support.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 20, 25a, 25b, and 29.

Chapter 3. Stand-Alone DFDSS Restore Messages

This chapter describes stand-alone DFDSS restore messages. See Chapter 1, "How to Read DFDSS and Stand-Alone DFDSS Restore Messages," for an explanation of the message format.

ADR501W INVALID CARD CODE. CORRECT ERROR. DEPRESS INTERRUPT KEY.

Explanation: An invalid card code appeared in the last card processed.

System Action: The system enters a wait state, awaiting operator response.

Programmer Response: None.

Operator Response: Probable user error. Check for and correct keypunch or multipunch errors. Reset the card reader. Press the interrupt key and continue processing. Refer to Appendix A, "Use of Interrupt Key" on page 75 for information on interrupt key usage in stand-alone environments.

Problem Determination: Table I, items 2, 20, and 29.

ADR502A CONTROL STATEMENT ERROR. JOB TERMINATED.

Explanation: A control statement contains an incorrect keyword, parameter, or name field.

System Action: The job is ended.

Programmer Response: Probable user error. Check for keypunch errors. Correct keyword, parameter, or name fields. Resubmit the job.

Operator Response: None.

Problem Determination: Table I, items 2, 20, and 29.

ADR503A STATEMENT SEQUENCE ERROR. JOB TERMINATED.

Explanation: Either the statements are not in proper sequence or unnecessary statements are present.

System Action: The job is ended.

Programmer Response: Probable user error. Correct control statement sequence, or remove unnecessary statements. Resubmit the job.

Operator Response: None.

Problem Determination: Table I, items 2, 20, and 29.

ADR504W SVC INTERRUPT. JOB TERMINATED

Explanation: An unknown SVC interrupt occurred. This program does not issue SVCs.

System Action: The job ends.

Programmer Response: None.

Operator Response: Rerun the program.

Problem Determination: Table I, items 11, 15, 20, and 29.

ADR505A DEFINE INPUT DEVICE.

Explanation: The input device must be identified to the system by the operator, as specified under "Operator Response."

System Action: The system waits for the operator's response.

Programmer Response: None.

Operator Response: Prepare the input device by placing control statements in it. Load the source tape on the source device and prepare it. Enter the following message from the console:

```
INPUT=dddd, cuu
```

where dddd is the device type, and cuu is the channel and unit address of the input device.

Problem Determination: None.

ADR506A THE VOLID IN CONTROL STATEMENT DOES NOT AGREE WITH ID IN VOL LABEL WHICH FOLLOWS. VOLID = xxx

Explanation: The VOLID parameter in the control statement did not match the volume serial number, xxx, found on the receiving volume.

System Action: The job is ended.

Programmer Response: Probable user error. Correct the error and resubmit the job.

Operator Response: Mount the correct volume, and rerun the program.

Problem Determination: Table I, items 2, 15, 20, 28, and 29.

ADR507A ENTER CONTROL CARD STATEMENT

Explanation: The input control statements are being entered from a console; this message is a prompt.

System Action: None.

Programmer Response: None.

Operator Response: Enter the control statements for JOB, MSG, RESTORE, and, optionally, VDRL, and then END.

Problem Determination: None.

ADR510I BAD TRACK cchh

Explanation: A defective track was found at the location specified by cc, the cylinder number, and hh, the head number.

System Action: None.

Programmer Response: Make sure that message ADR511I was issued for each bad track.

Operator Response: None.

Problem Determination: None.

ADR511I ALTERNATE cchh | NONE

Explanation: An alternate track at the location specified by cc, the cylinder number, and hh, the head number, is the track that replaces the defective track.

System Action: None.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR552W PROGRAM INTERRUPT. JOB TERMINATED.

Explanation: A program interrupt occurred.

System Action: The job is ended.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 2, 11, 20, and 29.

ADR553A CONSOLE FAILED TO READ LAST MESSAGE. DEPRESS INTERRUPT KEY.

Explanation: The console failed to read the input message.
System Action: The system waits for the operator's response.
Programmer Response: None.
Operator Response: Press the interrupt key, and attempt to enter the input message again. Refer to Appendix A, "Use of Interrupt Key" on page 75 for information on interrupt key usage in stand-alone environments.
Problem Determination: Table I, item 30.

ADR554A READY READER cuu. DEPRESS INTERRUPT KEY.

Explanation: The reader on channel c, unit uu, has a card or transport jam or is out of cards.
System Action: The system waits for the operator's response.
Programmer Response: None.
Operator Response: Correct the faulty condition, ready the reader, and press the interrupt key to continue the program. If the card or transport jam occurs again, reproduce the jammed cards and try again. Refer to Appendix A, "Use of Interrupt Key" on page 75 for information on interrupt key usage in stand-alone environments.
Problem Determination: Table I, item 30.

ADR555A READY PRINTER cuu. DEPRESS INTERRUPT KEY.

Explanation: The printer on channel c, unit uu, is not ready. This can be caused by a forms check, an open interlock, or the STOP key being pressed.
System Action: The system waits for the operator's response.
Programmer Response: None.
Operator Response: Correct the faulty condition, ready the device, and press the interrupt key. Refer to Appendix A, "Use of Interrupt Key" on page 75 for information on interrupt key usage in stand-alone environments.
Problem Determination: Table I, item 30.

ADR556A READY TAPE cuu. DEPRESS INTERRUPT KEY.

Explanation: The tape drive on channel c, unit uu, is not ready.
System Action: The system waits for the operator's response.
Programmer Response: None.
Operator Response: Correct the faulty condition, ready the device, and press the interrupt key. Refer to Appendix A, "Use of Interrupt Key" on page 75 for information on interrupt key usage in stand-alone environments.
Problem Determination: Table I, item 30.

ADR557A READY DASD cuu. DEPRESS INTERRUPT KEY.

Explanation: The direct-access device on channel c, unit uu, is not ready.
System Action: The system waits for the operator's response.
Programmer Response: None.
Operator Response: Correct the faulty condition, ready the device, and press the interrupt key. Refer to Appendix A, "Use of Interrupt Key" on page 75 for information on interrupt key usage in stand-alone environments.
Problem Determination: Table I, item 30.

ADR558A WRONG TAPE ON cuu. COND = x. MOUNT PROPER TAPE.

Explanation: The tape on the drive on channel c, unit uu, does not pertain to this job. The values of "x" are:

- 1 This is not a DFDSS-created dump tape, or, if it is, the stand-alone program does not support the format on the tape. (The tape may have been created by an incompatible DFDSS release.)
- 2 An incorrect tape was mounted in response to ADR561A.
- 3 The dump tape is not compatible with the function specified. A full volume RESTORE is being attempted from a tracks or physical data set DUMP tape.
- 5 A RESTORE is being attempted from a DFDSS Version 2 logical data set dump tape.

System Action: The system waits for the operator's response.
Programmer Response: None.
Operator Response: Probable user error. Mount the correct tape, and press the interrupt key to continue. Refer to Appendix A, "Use of Interrupt Key" on page 75 for information on interrupt key usage in stand-alone environments.
Problem Determination: Table I, items 2, 11, 20, 28, and 29.

ADR559A READER CHECK. CORRECT ERROR. DEPRESS INTERRUPT KEY.

Explanation: A reader check occurred.
System Action: The system waits for the operator's response.
Programmer Response: None.
Operator Response: Correct the faulty condition, and clear the reader check. Ready the reader and continue the program by pressing the console interrupt key. Refer to Appendix A, "Use of Interrupt Key" on page 75 for information on interrupt key usage in stand-alone environments. If reader checks occur frequently, check the input cards with a card guide for off-punched cards. Ensure that the cards are both punched correctly and in good condition.
Problem Determination: Table I, item 30.

ADR560A PRINT CHECK. CORRECT ERROR. DEPRESS INTERRUPT KEY.

Explanation: A print check occurred.
System Action: The system waits for the operator's response.
Programmer Response: None.
Operator Response: Clear the print check by pressing Check Reset on the printer. Press the interrupt key on the console to continue processing. Refer to Appendix A, "Use of Interrupt Key" on page 75 for information on interrupt key usage in stand-alone environments.
Problem Determination: Table I, item 30.

ADR561A END OF TAPE. MOUNT TAPE ON cuu. DEPRESS INTERRUPT KEY.

Explanation: It is the end of the present tape reel on channel c, unit uu.
System Action: The system waits for the operator's response.
Programmer Response: None.
Operator Response: Mount the next tape volume on the active tape device, the FROMDEV device. Press the interrupt key on the console to continue the program. Refer to Appendix A, "Use of Interrupt Key" on page 75 for information on interrupt key usage in stand-alone environments.
Problem Determination: None.

ADR563A END OF JOB.

Explanation: A normal end-of-job condition occurred.
System Action: The job is ended.
Programmer Response: None.
Operator Response: None.
Problem Determination: None.

ADR565A ATTEMPT TO RESTORE TO WRONG DEVICE. JOB TERMINATED.

Explanation: The program attempted to restore data to a device type other than the type from which it was dumped.
System Action: The job is ended.
Programmer Response: Probable user error. Correct the TODEV operand to reflect the device from which the data was dumped, and rerun the job.
Operator Response: None.
Problem Determination: Table I, items 2, 15, 20, 28, and 29.

ADR568I TRACK 0 HAS AN ALTERNATE ASSIGNED. THIS VOLUME HAS BECOME NON-IPL-ABLE

Explanation: Track 0 was flagged as a defective track. (This volume is usable as a work volume but may not be usable as a system residence volume.)
System Action: The job is ended.
Programmer Response: None.
Operator Response: If the volume is to be used as an IPLable volume, try IPLing from that volume. If it fails, mount another volume and rerun the job.
Problem Determination: None.

ADR569A PRINTER EQUIPMENT CHECK. CORRECT ERROR. DEPRESS INTERRUPT KEY.

Explanation: A printer equipment check occurred.
System Action: The system waits for the operator's response.
Programmer Response: None.
Operator Response: Correct the error at the printer. Press the interrupt key and continue processing. Refer to Appendix A, "Use of Interrupt Key" on page 75 for information on interrupt key usage in stand-alone environments.
Problem Determination: Table I, item 30.

ADR570A SEQUENCE ERROR ON RESTORE TAPE. JOB TERMINATED.

Explanation: The restore tape has a missing, incorrect, or extra record.
System Action: The program is unable to continue with the RESTORE. The job is ended.
Programmer Response: None.
Operator Response: Probable error during DUMP.
Problem Determination: Table I, items 11, 20, 28, and 29.

ADR571I TRACK NOT RESTORED DUE TO I/O ERROR DURING DUMP, TRACK cchh

Explanation: An I/O error was encountered in reading the track during DUMP.
System Action: The track on the disk is not written on.
Programmer Response: Verify that the track is not necessary.
Operator Response: None.
Problem Determination: None.

ADR601W COMMAND REJECT. device, opcode, status, sense bytes

Explanation: The specified channel rejected an incorrect channel command word (CCW) list.
System Action: The job is ended.
Programmer Response: None.
Operator Response: Rerun the program.
Problem Determination: Table I, items 11 and 29.

ADR602A INTERV. REQUIRED. cuu

Explanation: The device on channel c, unit uu, is not ready.
System Action: The system waits for the operator's response.
Programmer Response: None.
Operator Response: The specified device requires operator intervention to ready it. If for no apparent reason the device is not ready, initiate the following problem determination.
Problem Determination: Table I, item 30.

ADR603W BUS OUT CHECK. c

Explanation: A bus out check occurred on channel c. Probable hardware error.
System Action: The job is ended.
Programmer Response: None.
Operator Response: None.
Problem Determination: Table I, items 12 and 30.

ADR604W EQUIPMENT CHECK.

Explanation: An equipment failure occurred. Probable hardware error.
System Action: The job is ended.
Programmer Response: None.
Operator Response: None.
Problem Determination: Table I, item 30.

ADR605W DATA CHECK. device, opcode, status, sense bytes

Explanation: A solid data check occurred on the specified device. Probable hardware error.
System Action: The job is ended.
Programmer Response: None.
Operator Response: None.
Problem Determination: Table I, item 30.

ADR606W OVERRUN. c

Explanation: An overrun check occurred on the specified channel.
System Action: The job is ended.
Programmer Response: None.
Operator Response: Rerun the program.
Problem Determination: Table I, item 30.

ADR608W DATA CONV. CHECK. device, opcode, status, sense bytes

Explanation: A data converter check occurred on the specified device.
System Action: The job is ended.
Programmer Response: None.
Operator Response: Move data to another device, and rerun the job.
Problem Determination: Table I, item 30.

ADR609W END OF CYLINDER. device, opcode, status, sense bytes

Explanation: An unusual end-of-cylinder condition occurred on the specified device.
System Action: The job is ended.
Programmer Response: None.
Operator Response: Rerun the job.
Problem Determination: Table I, item 30.

ADR610W INVALID ADDRESS. device, opcode status, sense bytes, cchh

Explanation: An invalid address was issued to the specified device.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Determine if the cchh address shown in the message is valid for the indicated device.

Problem Determination: Table I, items 2, 13, and 29.

ADR611W NOT AVAILABLE. device, opcode, status, sense bytes

Explanation: The specified device is not attached to the system.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Ensure that the ENABLE/DISABLE switch on the device and/or the tape control is in the ENABLE position.

Problem Determination: Table I, item 30.

ADR612W READ DATA CHECK. device, opcode, status, sense bytes

Explanation: A permanent read data check was detected on the specified tape unit.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Clean the read/write heads of the tape unit, and rerun the job.

Problem Determination: Table I, item 30.

ADR614W TRACK OVERRUN.

Explanation: A track overrun condition occurred.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Rerun the program.

Problem Determination: Table I, items 11 and 29.

ADR616I ASYN UC

Explanation: DFDSS received an interrupt containing a unit check to a device for which a Start I/O was initiated but not started.

System Action: The task is terminated.

Programmer Response: None.

Operator Response: Rerun the job.

Problem Determination: Table I, items 11 and 29.

ADR617W NO RECORD FOUND

Explanation: Two index markers were detected during a CCW chain, and the record being sought was not found.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Probable hardware error. Execute Device Support Facilities to reinitialize the DASD volume. Rerun the job.

Problem Determination: Table I, item 30.

ADR618W INDETERMINATE ERROR.

Explanation: An indeterminate I/O error occurred.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Verify that the control unit and channel are not shared. Rerun the program.

Problem Determination: Table I, item 29.

ADR619W UNEXPECTED ERROR. device, opcode, status, sense bytes

Explanation: The error return is valid, but it is not associated with the specified device.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Rerun the program.

Problem Determination: Table I, item 29.

ADR620W CHAN. CTRL ERROR. cxx, opcode, status, sense bytes

Explanation: A channel control check occurred on the channel specified by cxx.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Probable hardware error. Rerun the job.

Problem Determination: Table I, items 12 and 30.

ADR621W INTERFACE ERROR. cxx, opcode, status, sense bytes

Explanation: An interface control check occurred on the channel specified by cxx.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Probable hardware error. Rerun the job.

Problem Determination: Table I, items 12 and 30.

ADR622W CHAN. DATA CHECK. cxx, opcode, status, sense bytes

Explanation: A channel data check occurred on the channel specified by cxx.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Probable hardware error. Rerun the job.

Problem Determination: Table I, items 12 and 30.

ADR624W PROGRAM CHECK.

Explanation: A program check occurred because of an incorrect CCW.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Rerun the program.

Problem Determination: Table I, items 11, 13, 20, and 29.

ADR625W PROTECTION CHECK. (device, cuu), opcode, status, sense bytes

Explanation: A protection check occurred on the specified device.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Rerun the program.

Problem Determination: Table I, items 11 and 29.

ADR626W UNIT EXCEPTION. cuu, device, opcode, status, sense bytes

Explanation: A unit exception occurred on the specified unit.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Rerun the program.

Problem Determination: None.

ADR627W INCORRECT LENGTH. cuu, opcode, status, sense bytes

Explanation: An incorrect length record condition occurred on the specified unit.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Rerun the program.

Problem Determination: Table I, item 29.

ADR628W CHAINING CHECK. cxx, opcode, status, sense bytes

Explanation: A chaining check occurred on the channel specified by cxx.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Probable hardware error. Rerun the job.

Problem Determination: Table I, items 11 and 30.

ADR629W COMMAND SEQ. ERR.

Explanation: An invalid sequence of CCWs was issued.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Probable hardware error. Rerun the job.

Problem Determination: Table I, items 11 and 30.

ADR630W SEEK CHECK ERROR.

Explanation: An invalid SEEK address was issued, or a unit malfunction caused a SEEK check.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Probable hardware error. Rerun the job.

Problem Determination: Table I, items 2, 20, and 30.

ADR632W TAPE-LOAD POINT. cuu, opcode, status, sense bytes

Explanation: A tape at load point condition occurred on the specified tape unit.

System Action: The job is ended.

Programmer Response: None.

Operator Response: None.

Problem Determination: Table I, items 11 and 30.

ADR633W NOISE RECORD. cuu, opcode, status, sense bytes

Explanation: A noise record was found on the specified tape unit.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Clean the read/write heads of the tape unit, and rerun the job.

Problem Determination: Table I, items 28 and 30.

ADR634W MISSING ADR-MARK. cuu, opcode, status, sense bytes

Explanation: A missing address marker occurred on the specified device.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Rerun the job.

Problem Determination: Table I, item 30.

ADR635W RESTORE TERMINATED. OUTPUT DEVICE (cccc) TOO SMALL FOR INPUT (cccc)

Explanation: The number of cylinders on the target DASD volume is less than either:

- The number of cylinders dumped from the original input volume
- The highest cylinder number to be restored, as specified on a VDRL statement.

System Action: The RESTORE operation is ended.

Programmer Response: Supply an appropriate volume that matches the input for a full volume RESTORE, or correct the VDRL statement and rerun the job.

Operator Response: None.

Problem Determination: Table I, items 2 and 29.

ADR642W INVALID TRK FMT. cuu, opcode, status, sense bytes

Explanation: An attempt was made to write data exceeding track capacity on the specified device.

System Action: If the input tape contains the invalid track, the RESTORE continues. Otherwise, the job is ended.

Programmer Response: None.

Operator Response: Rerun the program.

Problem Determination: Table I, items 11 and 29.

ADR643W WRITE INHIBITED.

Explanation: The WRITE INHIBIT switch is probably on in the control unit to inhibit processing of write commands.

System Action: The job is ended.

Programmer Response: None.

Operator Response: Check the status of the WRITE INHIBIT switch:

- If the WRITE INHIBIT switch is on to inhibit execution of write commands and the disk pack should be written on, set the switch off and rerun the job.
- If the WRITE INHIBIT switch is set on to intentionally inhibit processing of write commands, follow the procedures established by your installation.
- If the WRITE INHIBIT switch is off to allow processing of write commands, rerun the job.

Problem Determination: Table I, items 2, 13, and 30.

ADR649W I/O ERROR, JOB TERMINATED.

Explanation: This message follows all messages that describe input/output error conditions.

System Action: The job is ended.

Programmer Response: None.

Operator Response: None.

Problem Determination: None.

ADR655I ADDITIONAL SENSE xxxxxxxx (xxxxxxx)

Explanation: If the device supplies more than 24 bytes of sense data, the additional bytes, beginning with byte 25, will be printed in groups of six up to the 64th byte.

System Action: See previously issued message.

Programmer Response: See previously issued message.

Operator Response: None.

Problem Determination: See previously issued message.

Chapter 4. Problem Determination Table

As a DFDSS user, you determine whether you or IBM is responsible for resolving a problem. You do this by using IBM procedures to identify a failing hardware unit or program.

In most instances, one or more numbers appear after the heading, "Problem Determination," for a given message or code. These numbers refer to the numbered items that appear in Problem Determination Table I, and they require you to take specific actions before calling IBM for support.

Table I. Standard Problem Determination Actions

Note: Table I contains only procedures applicable to determining DFDSS errors. For additional diagnosis information, see the appropriate manuals listed in "DFDSS Books" on page vi.

1. Ensure that MSGLEVEL=(1,1) was specified in the JOB statement.
2. Save the console sheet from the primary console. In systems with Multiple Console Support (MCS), save a copy of the hard-copy log.
3. Save the job stream associated with the job.
4. Save the system output (SYSOUT) associated with the job.
5. Ensure that the failing job step includes one of the following:
 - a. SYSABEND DD statement
 - b. SYSUDUMP DD statement.
6. Not used.
7. Run the LISTIDR function of the AMBLIST service aid program to obtain a list of all members with a program temporary fix (PTF) or local fix, and save the output. Run the program against the:
 - a. SYS1.LINKLIB data set
 - b. SYS1.SVCLIB data set
 - c. Library containing the program that issued the message.
8. Not used.
9. Not used.
10. Not used.
11. Run the AMDSADMP service aid program to dump the contents of real storage and page data sets on magnetic tape. After restarting the system, run the GO function of the AMDPRDMP service aid program to print the real storage portion of the dump tape produced by AMDSADMP. Save both the tape from AMDSADMP (in case further information from the tape is required) and the listing from AMDPRDMP.
12. Run the SEREP program, and save the resulting output.
13. Save all the associated output.
14. In the normal response to this message, the programmer/operator was requested to run a specific program. Save all output from that program.

15. Save the program listing associated with the job.
16. Not used.
17. Have the system generation (SYSGEN) output available from:
 - a. Stage I
 - b. Stage II.
18. Not used.
19. Not used.
20. Save the control statements associated with the job.
21. Not used.
22. Not used.
23. Not used.
24. Not used.
25. Run the IEHLIST system utility program to obtain a list of the:
 - a. Volume table of contents of the associated volume by specifying the FORMAT option
 - b. Volume table of contents of the associated volume by specifying the DUMP option
 - c. Directory of the associated data set
 - d. System catalog.
26. Run the Access Method Services LISTCAT ALL command to list the source and target catalog entries.
27. Not used.
28. Save the associated volume.
29. Contact IBM for programming support.
30. Contact IBM for hardware support.
31. Not used.
32. Not used.
33. Not used.
34. Not used.
35. Rerun the job with the parameter ABEND = nnn, where nnn is the three-digit message number on which the dump must be taken, and insert a SYSUDUMP DD statement in the JCL. This causes a User 0001 abend at the time the message ADRnnnx is issued. (Not applicable to stand-alone DFDSS restore messages.)
36. Run the DIAGNOSE command of Access Method Services to determine the cause of the error on the:
 - a. VVDS on the volume
 - b. Data set on the volume and the catalog.

(Not applicable to stand-alone DFDSS restore messages.)

Appendix A. Use of Interrupt Key

Some of the Stand-alone DFDSS Restore Messages and Operator Response descriptions in Chapter 3, "Stand-Alone DFDSS Restore Messages" on page 67 refer to the interrupt key. Depending on the environment in which you are running, the required action varies as described below.

Stand-alone Restore Environment

In the native mode (real stand-alone), you press the external interrupt key on the processor console. The processor console is the console from which you IPLed the stand-alone restore program.

Stand-alone Restore in VM Environment

When running the Stand-alone DFDSS Restore program in a VM environment, you do the equivalent of pressing an external interrupt key. This is accomplished by issuing the VM CP command, EXTERNAL, from the same console that you are using to send commands to the stand-alone restore program. (In VM, you do everything from the same console.) The EXTERNAL command can be executed by entering:

```
#CP EXTERNAL
```

The #CP is entered to ensure that the processor (VM) receives the command and doesn't pass it along to the stand-alone restore program. If stand-alone restore is in full-screen mode, you may have to press PA1 to put the terminal into CP READ mode before issuing the EXTERNAL command. This information is documented in *VM/SP CP General User Command Reference*.

Glossary

This glossary includes definitions of some terms found in this document. Some of the terms and definitions are taken from *Dictionary of Computing*, copyright 1987 by International Business Machines Corporation. It can be obtained from IBM under Form Number SC20-1699.

Some of the terms defined are taken from:

- *The American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions are identified by the symbol (A) after the definition.
- *The Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

A

ABEND. Abnormal end of task. End of a task, a job, or a subsystem because of an error condition that cannot be resolved by recovery facilities while the task is performed.

ABENDxxx. The keyword that identifies the abnormal end of DFDSS because of a system-detected error.

ABSTR. A subparameter of the SPACE parameter in a DD statement. It indicates that specified tracks be assigned to a data set.

ACCEPT processing. An SMP/E process necessary for installing the FMIDs. SMP/E ACCEPT processing uses JCL to accept the modules and macros necessary to run the FMIDs. The FMIDs are accepted into the DLIBs from the temporary data sets.

Access method services. A multifunction service program that is used to manage both VSAM and non-VSAM data sets and integrated catalog facility or VSAM catalogs. It is used to define data sets and allocate space for them, convert indexed-sequential data sets to key-sequenced data sets, modify data set

attributes in the catalog, reorganize data sets, facilitate data portability between operating systems, create backup copies of data sets, data set records and catalog entries. help make inaccessible data sets accessible, list the records of data sets and catalogs, define and build alternate indexes, and convert OS CVOLs and VSAM catalogs to integrated catalog facility catalogs.

ACDS. Active control data set.

ACS. Automatic class selection.

AIX. Alternate index.

alias. An alternate name for a member of a partitioned data set.

ALLOC. A space allocation parameter that indicates type, such as cylinders or tracks.

alternate index (AIX). In systems with VSAM, a key-sequenced data set containing index entries organized by the alternate keys of its associated base data records. It provides an alternate means of locating records in the data component of a cluster on which the alternate index is based.

alternate index cluster. In VSAM, the data and index components of an alternate index.

APAR. Authorized Program Analysis Report.

Authorized Program Analysis Report (APAR). A request for correction of a problem caused by a suspected defect in a current unaltered release of a program.

APF. Authorized program facility.

Application Interface. An interface used to invoke DFDSS from another program.

Apply processing. In SMP and SMP/E, the process, initiated by the APPLY command, that places system modifications (SYSMODS) into the target system libraries.

attach. In programming, to create a task that can be performed asynchronously with the performance of the mainline code.

authorization. (1) (TC97) The right granted to a user to communicate with or make use of a computer system. (2) The process of giving a user either complete or restricted access to an object, resource, or function.

automatic class selection (ACS). A mechanism for assigning SMS classes and storage groups.

B

base cluster. In systems with VSAM, a key-sequenced or entry-sequenced data set over which one or more alternate indexes are built.

basic catalog structure (BCS). The name of the catalog structure in the integrated catalog facility environment. An integrated catalog facility catalog consists of a BCS and its related VSAM volume data sets (VVDSs).

basic direct access method (BDAM). An access method used to directly retrieve or update particular blocks of a data set on a direct access device.

basic partitioned access method (BPAM). An access method that can be applied to create program libraries in direct access storage for convenient storage and retrieval of programs.

basic sequential access method (BSAM). An access method for storing or retrieving data blocks in a continuous sequence, using either a sequential access or a direct access device.

BCS. Basic catalog structure.

BDAM. Basic direct access method.

BLK. A subparameter of the SPACE parameter in a DD statement. It specifies that space is allocated by blocks.

block length. Synonym for block size.

block size. (1) (TC97) The number of data elements in a block. (2) * A measure of the size of a block, usually specified in units such as records, words, computer words, or characters. (3) Synonymous with block length.

BPAM. Basic partitioned access method.

bpi. bits per inch.

BSAM. Basic sequential access method.

C

call. (ISO) * The action of bringing a computer program, a routine, or a subroutine into effect, usually by specifying the entry conditions and jumping to an entry point.

card image. * A one-to-one representation of the hole patterns of a punched card; for example, a matrix in which a one represents a punch and a zero represents the absence of a punch.

CCHHR. Cylinder, cylinder, head, head, record.

CCW. Channel command word.

CDE. Contents directory entry.

CDS. Control data set.

channel command word (CCW). A doubleword at the location in main storage specified by the channel address word. One or more CCWs make up the channel program that directs data channel operations.

CI. Control interval.

CLIST. Command list.

component identification keyword. The first keyword, represented as a number, in a set of keywords used to describe a DFDSS program failure.

COMPRESS command. The DFDSS function that reduces partitioned data sets by taking unused space and consolidating it at the end of the data set.

concatenation. An operation that joins two characters or strings in the order specified, forming one string whose length is equal to the sum of the lengths of the two characters or strings.

control interval (CI). A fixed-length area of auxiliary storage space in which VSAM stores records. It is the unit of information transmitted to or from auxiliary file, storage by VSAM.

control volume (CVOL). A volume that contains one or more indexes of the catalog.

constructs. A collective name for data class, storage class, management class, and storage group.

CONVERTV command. The DFDSS function that converts volumes to and from Storage Management Subsystem management without data movement.

COPY command. The DFDSS function that performs data sets, volumes, and tracks movement.

CP. Control Program.

CREDT. Creation date.

CSW. Channel status word.

CVAF. Common VTOC access facility.

CVOL. Control volume.

CVT. Communication vector table.

CYL. A subparameter of the SPACE parameter in a DD statement. It specifies that space is allocated by cylinders.

D

DADSM. The direct access space management program that maintains the VTOC, VTOCIX, and space on a volume.

DAM. Direct access method.

DASD. Direct access storage device.

DASD ERP. DASD error recovery procedure.

data class. A list of data set allocation parameters and their values that are used when allocating a new SMS-managed data set.

data compression (run-length). A method of encoding repetitive series of identical characters so that they occupy less space on a dump tape. Data compression is supported by both physical dump and logical dump processing.

Data Facility Data Set Services (DFDSS). A DASD data and space management tool for backup and recovery, space management functions, and copying data between like and unlike devices. DFDSS is the primary data mover of the Data Facility family.

Data Facility family. A family of products, consisting of DFP, DFDSS, DFHSM, and DFSORT, that performs key data and storage management functions.

Data Facility Hierarchical Storage Manager (DFHSM). A facility that automates space and availability management by backing up and migrating data within a hierarchy of storage.

Data Facility Product (DFP). The key architectural component of the Storage Management and non-Storage Management Subsystem environments. DFP manages programs, data, and devices in the MVS environment.

Data Facility Sort (DFSORT). A data and storage management tool that handles sorting, merging, and copying applications.

Data Facility Storage Management Subsystem (DFSMS). The complementary functions of MVS/DFP and other individual products of the Data Facility family, which, together with RACF provide a system-managed, administrator-controlled storage environment.

data set backup. Backup to protect against the loss of individual data sets.

data set change indicator. A bit that is set by OPEN when the data set is opened for processing other than input. This flag is supported on MVS systems that have data-set-changed flag support installed.

DAU. Direct access unmovable.

DB2*. IBM DATABASE 2.

DCB. Data control block.

DEFRAG command. The DFDSS function that consolidates the free space on a volume to help prevent out-of-space abends on new allocations.

DEQ. An Assembler language macro instruction used to remove control of one or more serially reusable resources from the active task.

DFDSS. Data Facility Data Set Services.

DFHSM. Data Facility Hierarchical Storage Manager.

DFP. Data Facility Product.

DFSMS*. Data Facility Storage Management Subsystem

DFSORT. Data Facility Sort.

DGTHIX50. The panel identification number for the DFDSS Help Index. It is used to verify that DFDSS/ISMF is installed correctly.

DIAGNOSE. An access method services command that scans an integrated catalog facility basic catalog structure (BCS) or a VVDS to validate the data structure.

DIRF. DADSM interrupt recording facility. If a system fails, or a permanent I/O error occurs during allocation of space or during performance of a routine that updates the VTOC, the VTOC may be in error. To ensure that an error is recorded, the DADSM routines turn on a bit in the VTOC upon entry to a DADSM function, and, if no errors occur during processing, turn off that bit upon exiting from that function.

distribution libraries. IBM-supplied partitioned data sets on tape containing one or more components that the user restores to disk for subsequent inclusion in a new system.

DLIB. Distribution library.

DOC. In diagnosing program failures, the keyword that identifies an error in the documentation of a program.

DOS. Disk Operating System.

DOS bit. On a volume without an indexed VTOC, a bit that indicates that the free space map is invalid.

DOS/VSE. DOS/Virtual Storage Extended.

DSCB. Data set control block.

DSCHA. A DFDSS keyword that is used in BY filtering. It indicates that the data set is to be selected if the data set has been changed.

dsn. Data set name

DSORG. Data set organization. It is specified in the JCL as "DSORG = :"

DUMP command. The DFDSS function used to back up data sets, tracks, and volumes.

dynamic allocation. Assignment of system resources to a program when the program is performed rather than when it is loaded main storage.

E

Early Warning System (EWS). A microfiche copy of the information contained in the Software Support Facility (SSF), organized by component identification number and indexed by APAR symptom code. EWS is published monthly and available to customers of IBM licensed programs.

ECB. Event control block.

EC mode. Engineering change mode.

empty data set. A data set in which the pointer to the last-used block is 0.

ENQ. An Assembler language macro instruction that requests the control program to assign control of one or more serially reusable resources to the active task. It is also used to determine the status of a resource; that is, whether it is immediately available or in use, and whether control has been previously requested for the active task in another ENQ macro instruction.

entry-sequenced data set (ESDS). In VSAM, a data set whose records are loaded without respect to their contents and whose RBAs cannot change. Records are retrieved and stored by addressed access, and new records are added at the end of the data set.

EOF. End-of-file.

EOJ. End of job.

ESDS. Entry-sequenced data set.

ESTAE. Extended specify task abnormal exit.

EQ. Equal to.

EWS. Early Warning System.

EXCP. Execute channel program.

Execute channel program (EXCP). A macro used to access a data set without specifying the organization.

EXPDT. Expiration date.

Extended specify task abnormal exit (ESTAE). A task recovery routine that provides recovery for those programs that run enabled, unlocked, and in task mode.

extent. A continuous space on a DASD volume occupied by a data set or portion of a data set. An extent of a data set contains a whole number of control areas.

F

FC. CVAF function code.

FCEC. CVAF function-error code.

filtering. The process of selecting data sets based on specified criteria. These criteria consist of fully- or partially-qualified data set names and/or of certain data set characteristics.

FMID. Function modification identifier.

fully-qualified data set name. A data set in which all the qualifiers are completely spelled out.

function modification identifier (FMID). A code that identifies the release levels of a program product.

FVL. function vector list.

G

GDG. Generation data group.

GDS. Generation data set.

Generation data group (GDG). A collection of historically related non-VSAM data sets that are arranged in chronological order; each data set is a generation data set.

Generation data set. One generation of a generation data group.

GT. Greater than.

I

ICKDSF. Device Support Facilities.

IDCAMS. Access Method Services.

IMS/VS. Information Management System/Virtual Storage.

INCORROUT. In diagnosing program failures, the keyword that identifies incorrect or missing program output.

incremental backup. A process in which data sets are backed up only if they have changed since their last backup.

Installation-wide exit. The means specifically described in an IBM software product's documentation by which an IBM software product may be modified by a customer's system programmers to change or extend the functions of the IBM software product. Such modifications consist of exit routines written to replace one or more existing modules of an IBM software product, or to add one or more modules or subroutines to an IBM software product, for the purpose of modifying (including extending) the functions of the IBM software

integrated catalog facility. A facility by which VSAM data set volume-related fields are separated from the catalog and maintained in the VVDS on the volume on which the data set resides.

Interactive Storage Management Facility (ISMF). A DFP component that provides an interactive interface to DFDSS and DFHSM. ISMF uses menu-driven panels to build jobs streams for most DFDSS functions. DFDSS/ISMF performs the following functions: COMPRESS, CONVERTV, COPY, DEFRAG, DUMP, RELEASE, and RESTORE.

Interactive System Productivity Facility (ISPF). An IBM licensed program used to develop, test, and run application programs interactively. ISPF is the interactive interface for all storage management functions.

I/O. Input/output.

IPL. Initial program load.

ISMF. Interactive Storage Management Facility.

ISAM. Indexed sequential access method.

ISMF. Interactive Storage Management Facility.

ISPF. Interactive Systems Productivity Facility.

ISPF/PDF. Interactive Systems Productivity Facility/Program Development Facility.

J

JCL. Job control language.

JES. Job entry subsystem.

JES2. An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for operation, processes their output, and purges them from the system. In an installation site with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing.

JES3. An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for operation, processes their output, and purges them from the system. In complexes that have several loosely-coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them via a common job queue.

JFCB. Job file control block.

job control language (JCL). A problem-oriented language used to identify the job or describe its requirements to an operating system.

job entry subsystem (JES). A system facility for spooling, job queuing, and managing I/O.

JSCB. Job step control block.

K

KB. Kilobyte (1 024 bytes).

key-sequenced data set. A VSAM file or data set whose records are loaded in ascending key sequence and controlled by an index. Records are retrieved and stored by keyed access or by addressed access, and new records are inserted in key sequence by means of distributed free space. Relative byte addresses can change because of control interval or control area splits.

keyword. A symptom that describes one aspect of a program failure.

KSDS. Key-sequenced data set.

L

LASTCC. Last condition code.

LDS. Linear data set.

like devices. Devices that have the same track capacity and number of tracks per cylinder (for example, 3380 Standard and 3380 Model J or K).

LINK. An Assembler language macro instruction that causes control to be passed to a specified entry point. The linkage relationship established is the same as that created by a BAL instruction.

link-pack area (LPA). An area of virtual storage that contains re-enterable routines that are loaded at IPL (initial program load) time and can be used concurrently by all tasks in the system.

load module. (TC97) A computer program in a form suitable for loading into main storage for operation.

load module library. A partitioned data set used to store and retrieve load modules.

logical (data set) DUMP operation. A DUMP operation in which logical processing is performed.

logical (data set) processing. Processing that treats each data set and its associated information as a logical entity. As an example, DFDSS processes an entire data set before beginning with the next one.

logical (data set) RESTORE operation. A RESTORE operation that uses as input a data set produced by a logical DUMP operation.

logical volume. The output produced from a physical DUMP operation, for which all data is derived from a single DASD volume.

LOOP. In diagnosing program failures, the keyword that identifies a program failure in which some part of the program repeats endlessly.

LPA. link-pack area.

LT. Less than

LRECL. Logical record length.

LVOL. logical volume.

M

maintenance-level keyword. In diagnosing program failures, a keyword that identifies the maintenance level of DFDSS.

management class. A list of the migration, backup and retention parameters and their values for an SMS-managed data set.

map record. The record that maps the tracks that were dumped by DFDSS.

MAXCC. Maximum condition code.

MCS. Multiple Console Support.

MENTITY. Model entity.

minivolume. In an MVS system running on VM/370, an OS/VS-formatted VM/370 minidisk whose size is equal to or less than that of the real volume. DFDSS uses the device size specified in the in their VTOC. Minivolumes are supported only by the system version of DFDSS.

MSGADRnnnt. In diagnosing program failures, the DFDSS message keyword that tells of an error, or seems itself to be in error.

MSS. Mass storage system.

MVS. Multiple virtual storage.

N

NVR. Non-VSAM volume record.

O

Operating System (OS). Software that controls the execution of programs; an operating system may provide services such resource allocation, scheduling, input/output control, and data management.

OS. Operating System.

OS/VS2. Operating System/Virtual Storage 2.

P

pageable link-pack area (PLPA). Link-pack area.

PAM. Partitioned access method.

partially-qualified data set name. A data set name in which the qualifiers are not spelled out. Asterisks and percent signs are used in place of the undefined qualifier(s).

partitioned data set (PDS). A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

partitioned data set extended (PDSE). A SMS-managed, page-formatted data set on direct access storage. A PDSE contains an indexed directory and members similar to the directory and members of partitioned data sets. A PDSE can be used instead of a partitioned data set.

PDS. Partitioned data set.

PDSE. Partitioned data set extended.

PERFM. In diagnosing program failures, the keyword that identifies degradation in program performance.

physical (data set) DUMP operation. A DUMP operation in which physical processing is performed.

physical (data set) processing. Processing that moves data at the track-image level and can operate against volumes, tracks, and data sets. As an example, DFDSS may only process one volume of a multivolume data set.

PLPA. Pageable link-pack area.

POU. Partitioned organization unmovable.

PRB. Program request block.

private library. A user-owned library that is separate and distinct from the system library.

PSU. Physical sequential unmovable.

PSW. Program status word.

PTF. Program temporary fix.

Q

QSAM. Queued sequential access method.

qualified name. A data set name consisting of a string of names separated by periods; for example, "TREE.FRUIT.APPLE" is a qualified name.

qualifier. Each component name in a qualified name other than the rightmost name. For example, "TREE" and "FRUIT" are qualifiers in "TREE.FRUIT.APPLE."

Queued sequential access method (QSAM). An extended version of the basic sequential access method (BSAM). Input data blocks awaiting processing or output data blocks awaiting transfer to auxiliary storage are queued on the system to minimize delays in I/O operations.

R

RACF. Resource Access Control Facility.

RB. Request block.

RBA. Relative byte address.

RDJFCB. Read job file control block.

RECEIVE processing. An SMP/E process necessary to install new product libraries. During this process, the code, organized as unloaded partition data sets, is loaded into temporary SMPTLIB data sets. SMP/E RECEIVE processing automatically allocates the temporary partitioned data sets that correspond to the files on the tape, and loads them from the tape.

RECFM. Record format.

REFDT. A DFDSS keyword used in BY filtering. It indicates the last-referenced date.

relative byte address (RBA). The displacement (expressed as a fullword binary integer) of a data record or a control interval from the beginning of the data set to which it belongs, independent of the manner in which the data set is stored.

Relative record data set (RRDS). A VSAM data set whose records are loaded into fixed-length slots.

RELEASE command. The DFDSS function that releases the unused space in sequential and partitioned data sets for use by other data sets.

RESERVE. A method of serializing DADSM update accesses to the VTOC. It is also a method of serializing processor accesses to a shared DASD volume.

Resource Access Control Facility (RACF). An IBM program product that provides for access control by identifying and verifying users to the system, authorizing access to DASD data sets, logging the detected unauthorized attempts to enter the system, and logging the detected accesses to protected data sets.

RESTORE command. The DFDSS function used to recover data sets, tracks, and volumes.

RMID. Replacement module identifier.

RRDS. Relative record data set.

run-length data compression. Data compression (run-length).

S

SAF. System Authorization Facility.

SAM. Sequential access method.

scheduler task. A DFDSS subtask that interprets and schedules commands.

SCP. System control program.

SEQ. Sequential or sequential processing

SEREP. System environmental recording, editing, and printing

SMF. System management facilities.

SML. MVS Storage Management Library.

SMP. System Modification Program.

SMP/E. System Modification Program/Extended.

SMP/E. A cataloged procedure that includes the required DD statements for running SMP/E and is used in the RECEIVE, APPLY, and ACCEPT steps of SMP/E processing.

SMS. Storage Management Subsystem.

Software Support Facility (SSF). An IBM online data base that allows for storage and retrieval of information about all current APARs and PTFs.

SP. System Product.

sphere. A VSAM cluster with one or more associated alternate indexes and paths. The VSAM cluster (sometimes called the base cluster), alternate indexes (AIXs), and paths are sometimes referred to as sphere components.

SSF. Software Support Facility.

stand-alone DFDSS restore program. One of two DFDSS programs. Stand-alone DFDSS restore runs independently of the MVS system environment and is limited to one function—a full or partial (tracks) RESTORE from a dump tape.

storage class. A named list of data set storage attributes that SMS uses to identify performance and availability requirements.

storage constructs. The group of predefined models (data class, management class, storage class, and storage group) that are used to classify storage management needs and procedures for data sets under the Storage Management Subsystem. Each data set has construct names associated with it by explicit specification or defaulting.

storage group. A named collection of DASD volumes that have been grouped to meet a defined service strategy.

storage management. The task of managing auxiliary storage resources for an installation.

Storage Management Subsystem (SMS). An MVS subsystem that helps automate and centralize the management of storage. To manage storage, the storage management subsystem provides the storage administrator with control over data class, storage class, management class, storage group, and automatic class selection routine definitions.

subtask. A task initiated and ended by a higher order task.

SVC. Supervisor call instruction.

SVRB. Supervisor request block.

SYSRES. System residence disk.

system library. A collection of data sets or files in which the parts of an operating system are stored.

system link library. System library.

System Modification Program (SMP). A program used to install software and software changes on the MVS system.

System Modification Program Extended (SMP/E). An IBM licensed program used to install software and software changes on the MVS system. In addition to providing the services of SMP, SMP/E consolidates installation data, allows more flexibility in selecting changes to be installed, provides a dialog interface, and supports dynamic allocation of data sets.

T

TCB. Task control block.

Time sharing option (TSO). An option on the operating system for a System/370 that provides interactive time sharing from remote terminals.

TIOT. Task input/output table.

TLIBs. Target data sets.

track packing. A technique used by DFDSS that builds target tracks for any DASD device using input physical record information.

TRK. A subparameter of the SPACE parameter in a DD statement. It specifies that space is to be allocated by tracks.

TSO. Time-sharing option.

TSO/E. TSO/Extensions.

TTR. track-track-record.

type-of-failure keyword. In diagnosing program failures, a keyword that identifies the type of program failure that has occurred in DFDSS.

U

UACC. Universal access authority.

UCB. Unit control block.

UIM. User interaction module.

unlike devices. Devices that have different track capacities, for example, 3350 and 3380.

used tracks. Tracks from the beginning of data sets to the last-used track.

user exit. A programming service provided by an IBM software product that may be requested by an application program for the service of transferring control back to the application program upon the later occurrence of a user-specified event.

V

VDRL. Volume restore limits.

VDSS. VTOC/Data Set Services.

virtual storage access method (VSAM). An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by the relative-record number.

VM. Virtual machine.

VOLID. Volume ID.

VOLSER. Volume serial number.

volume backup. Backup of an entire volume to protect against the loss of the volume.

volume header record. The record in the DFDSS dump tape that identifies and contains data pertinent to the whole volume, and identifies the type of operation that created a dump.

volume trailer record. The record in the DFDSS dump tape that identifies the end of the data for a DASD volume.

VRRDS. A VSAM variable record RRDS.

VSAM. Virtual storage access method.

VSAM volume data set (VVDS). A data set that describes the VSAM and SMS-managed non-VSAM data sets on a volume. The name of the data set is SYS1.VVDS.Vvolser.

VSE. Virtual storage extended.

VTOC. Volume table of contents.

VTOCIX. The data set on which the location of the data set VTOC entries are kept in an index for quick access by DADSM. The name of the data set is SYS1.VTOCIX.Vvolser.

VVDS. VSAM volume data set.

VVR. VSAM volume record.

W

WAIT. In diagnosing program failures, the keyword that identifies DFDSS suspended activity, while waiting for some condition to be satisfied. DFDSS does not issue a message to tell why it is waiting.

WTO. Write to operator

X

XA. Extended Architecture.

Order No. SC26-4390-2

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, FAX number, and date:

If we require further information, may we contact you by phone?

___ Tel. No. _____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail them directly to the address in the Edition Notice on the back of the front cover or title page.) You may also FAX your comments to: (602) 799-6487. Contact department is 61C/060.

Reader's Comment Form

Fold and Tape

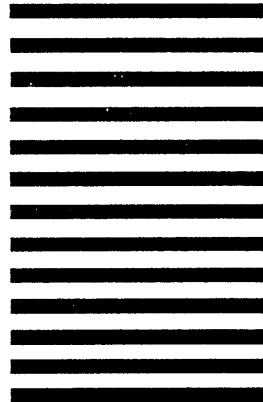
Please Do Not Staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, NEW YORK



POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department 61C
P.O. Box 27155
Tucson, AZ, USA 85775-4401

Fold and Tape

Please Do Not Staple

Fold and Tape



Order No. SC26-4390-2

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, FAX number, and date:

If we require further information, may we contact you by phone? ___ Tel. No. _____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

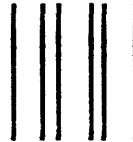
Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail them directly to the address in the Edition Notice on the back of the front cover or title page.) You may also FAX your comments to: (602) 799-6487. Contact department is 61C/060.

Reader's Comment Form

Fold and Tape

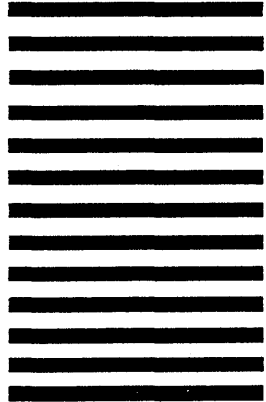
Please Do Not Staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, NEW YORK



POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department 61C
P.O. Box 27155
Tucson, AZ, USA 85775-4401

Fold and Tape

Please Do Not Staple

Fold and Tape





User's Guide

Version 2 Release 5.0

Second Edition (December 1989)

This is a major revision of SC26-4388-0, which is now obsolete. Changes from the previous edition are marked with a vertical bar in the left margin. Significant changes are summarized under "Summary of Changes."

This edition applies to Version 2, Release 5, Modification Level 0 of the Data Facility Data Set Services, Program Number 5665-327, and to all subsequent releases and modification levels until specified otherwise in new editions or technical newsletters.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's program may be used. Any functionally equivalent program may be used instead.

Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality. If you request publications from the address given below, your order will be delayed because publications are not stocked there.

A Reader's Comment Form is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, General Products Division, Department 61C, Tucson, Arizona, U.S.A. 85744. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1984, 1985, 1986, 1987, 1988, 1989. All Rights Reserved.

Note to U.S. Government Users—Documentation related to restricted rights—Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corporation.

Preface

This book describes how to use Data Facility Data Set Services (DFDSS) to perform various storage management tasks. It is intended primarily for system programmers. This book also describes functions that should be used with the interfaces provided herein. Unless specifically stated otherwise, the information in this book must not be used for programming purposes. However, this book does provide the following type of information which is explicitly identified where they occur:

General-Use Programming Interface: General-use programming interfaces are provided to allow the customer to write programs that use the services of DFDSS.

Note: Examples provided in this publication contain sets of program statements offered to users as an illustration of how to obtain product service and are not to be used as programming interfaces.

For information about the syntax of DFDSS commands, see *Data Facility Data Set Services: Reference*. For information about DFDSS messages, see *Data Facility Data Set Services: Messages*.

Data Facility Storage Management Subsystem (DFSMS) is hereafter referred to as Storage Management Subsystem (SMS).

Trademarks

The following terms are trademarks of IBM and are used in this publication:

DB2

DFSMS

MVS/DFP

MVS/ESA

MVS/SP

MVS/XA

Major Divisions of This Book

This book is organized as follows:

Chapter 1, Introduction to DFDSS presents a brief overview of DFDSS functions.

Chapter 2, Requirements for Running DFDSS describes the requirements for running DFDSS.

Chapter 3, Logical/Physical Processing and Data Set Filtering explains the differences between logical and physical processing, and how to use data set filtering to select data sets for DFDSS processing.

Chapter 4, Changing Command Specifications for DFDSS Version 2 Release 5 explains how to change the specification of commands to make a smooth transition to DFDSS Version 2 Release 5.0.

Chapter 5, Invoking DFDSS explains the different ways you can invoke DFDSS.

Chapter 6, Protecting DFDSS Functions discusses how you can protect various DFDSS and DFDSS/ISMF functions.

Chapter 7, Managing Availability with DFDSS describes how to use DFDSS DUMP and RESTORE commands to fulfill your data availability requirements.

Chapter 8, Managing Data Movement with DFDSS describes how to use DFDSS to perform data movement.

Chapter 9, Converting Data to and from SMS Management describes how to use DFDSS to convert data to and from Storage Management Subsystem (SMS) control.

Chapter 10, Managing Space with DFDSS describes how to use DFDSS RELEASE, COMPRESS, and DEFRAg commands to manage DASD space.

Appendix A, ACS Routine Variables lists the variables available to ACS routines during DFDSS COPY, RESTORE, and CONVERTV.

Appendix B, Changing Command Specifications for DFDSS Version 2 Release 4 explains what the new keywords were for Version 2 Release 4 and how they affect DFDSS processing.

Appendix C, Stand-Alone Restore Program lists storage and other devices for the stand-alone DFDSS restore program.

Required Product Knowledge

To use this manual you need some knowledge of the Data Facility Product (DFP), Data Facility Hierarchical Storage Manager (DFHSM), Resource Access Control Facility (RACF), and job control language (JCL).

Prerequisite Books

Data Facility Data Set Services: General Information, GC26-4123, introduces you to DFDSS and helps in evaluating its use.

Data Facility Data Set Services: User's Guide, SC26-4388, gives guidance on how to use DFDSS.

Data Facility Data Set Services: Reference, SC26-4389, describes DFDSS command syntax.

Data Facility Data Set Services: Messages, SC26-4390, lists DFDSS messages.

Data Facility Data Set Services: Diagnosis Guide, LY27-9538, tells how to diagnose errors you might encounter while using DFDSS.

Data Facility Data Set Services/Interactive Storage Management Facility: Installation Planning Guide, SC26-4129, describes considerations when you plan for and install DFDSS/ISMF.

Corequisite Books

To use this book effectively, you should be familiar with whichever of the following books are associated with your system:

MVS/370

Long Title	Short Title	Order Number
<i>MVS/370 JCL User's Guide</i>	<i>JCL User's Guide</i>	GC28-1349
<i>MVS/370 JCL Reference</i>	<i>JCL Reference</i>	GC28-1350
<i>MVS/370 Linkage Editor and Loader User's Guide</i>	<i>Linkage Editor and Loader User's Guide</i>	GC26-4061
<i>MVS/370 Installation: System Generation</i>	<i>System Generation</i>	GC26-4063

MVS/XA

Long Title	Short Title	Order Number
<i>MVS/Extended Architecture JCL User's Guide</i>	<i>JCL User's Guide</i>	GC28-1351
<i>MVS/Extended Architecture JCL Reference</i>	<i>JCL Reference</i>	GC28-1352

MVS/XA Version 2

Long Title	Short Title	Order Number
<i>MVS/Extended Architecture Linkage Editor and Loader User's Guide</i>	<i>Linkage Editor and Loader User's Guide</i>	GC26-4143
<i>MVS/Extended Architecture Installation: System Generation</i>	<i>System Generation</i>	GC26-4148

MVS/ESA

Long Title	Short Title	Order Number
<i>MVS/ESA Application Development Guide</i>	<i>Application Development Guide</i>	GC28-1821
<i>MVS/ESA Application Development Macro Reference</i>	<i>Application Development Macro Reference</i>	GC28-1822
<i>MVS/ESA Data Facility Product Version 3: General Information</i>	<i>General Information</i>	GC26-4507
<i>MVS/ESA System Generation</i>	<i>System Generation</i>	GC28-1825
<i>MVS/ESA Catalog Administration Guide</i>	<i>Catalog Administration Guide</i>	SC26-4502
<i>MVS/ESA: Storage Administration Reference</i>	<i>Storage Administration Reference</i>	SC26-4514
<i>MVS/ESA Storage Management: Managing Data Sets</i>	<i>SML: Managing Data Sets</i>	SC26-4657

MVS/DFP Version 3 Release 2

Long Title	Short Title	Order Number
<i>MVS/DFP Version 3 Release 2: General Information</i>	<i>General Information</i>	GC26-4552
<i>MVS/DFP Version 3 Release 2: Interactive Storage Management Facility User's Guide</i>	<i>ISMF User's Guide</i>	SC26-4563
<i>MVS/DFP Version 3 Release 2: Storage Administration Reference</i>	<i>Storage Administration Reference</i>	SC26-4566

Related Books

For a detailed list of related publications, see the following reader's guides:

	Title	Order Number
MVS/370	<i>MVS/370 Data Facility Product: Reader's Guide</i>	GC26-4055
MVS/Extended Architecture	<i>MVS Storage Management Library: Reader's Guide</i>	GC26-4326
MVS/XA Version 1	<i>MVS/Extended Architecture Data Facility Product: Reader's Guide</i>	GC26-4326
MVS/XA Version 2	<i>MVS/Extended Architecture Storage Management Library: Reader's Guide</i>	GC26-4265
MVS/ESA Version 3 Release 1	<i>MVS/ESA Storage Management Library: Reader's Guide</i>	GC26-4403
MVS/ESA Version 3 Release 2	<i>MVS/ESA Storage Management Library: Reader's Guide</i>	GC26-4654

Other Related Publications

Long Title	Order Number
<i>Using the IBM 3380 Direct Access Storage in an MVS Environment</i>	GC26-4492
<i>Device Support Facilities User's Guide and Reference</i>	GC35-0033
<i>Data Facility Hierarchical Storage Manager: General Information</i>	GH35-0092
<i>Data Facility Hierarchical Storage Manager: System Programmer's Command Reference</i>	SH35-0083
<i>MVS/DFP Version 3 Release 2: Customization</i>	SC26-4560
<i>Resource Access Control Facility (RACF) Security Administrator's Guide</i>	SC28-1340
<i>Resource Access Control Facility (RACF) General Information Manual</i>	GC28-0722
<i>Job Control Language User's Guide</i>	GC28-1830

Contents

Chapter 1. Introduction to DFDSS	1
The Role of DFDSS	1
Managing User Data with SMS	1
Managing Availability with DFDSS	2
Backing Up Volumes and Data Sets	3
Using DFHSM for Backup	3
Managing Data Movement with DFDSS	3
Moving Data	3
Moving Data in an SMS Environment	4
Converting Data to and from SMS Management	4
Conversion of Data Sets by Data Movement	4
Conversion of Volumes without Data Movement	5
Managing Space with DFDSS	5
Chapter 2. Requirements for Running DFDSS	7
Operating Environment	7
Storage Requirements	7
DASD Requirements	9
Stand-Alone DFDSS Restore Program	9
Volume Formats	10
Data Set Organizations	10
Chapter 3. Logical/Physical Processing and Data Set Filtering	11
Logical and Physical Processing	11
Logical Processing	11
Physical Processing	12
Choosing Data Sets for Processing—Filtering	13
Filtering by Data Set Names	14
Filtering by Data Set Characteristics	15
Some Uses of Filtering	16
Chapter 4. Changing Command Specifications for DFDSS Version 2 Release 5 ..	19
Converting to and from a PDSE	19
Restrictions for COPY, DUMP, and RESTORE	19
Processing SYS1 Data Sets	19
Processing Undefined DSORG Data Sets	19
Omitting the TGTALLOC Keyword with COPY or RESTORE Commands ..	20
Specifying Both the OPTIMIZE and COMPRESS Keywords for Dumps ..	20
Selecting Arguments for Filtering by Data Set Characteristics ..	20
Specifying the Status of the Target Generation Data Set	20
Chapter 5. Invoking DFDSS	21
Invoking DFDSS with ISMF	21
How to Invoke ISMF	21
ISMF Data Set Application	23
ISMF Volume Application	27
Invoking DFDSS with JCL	31
The JCL You Need	31
Controlling DFDSS with the EXEC Statement	33
Examples of Invoking DFDSS with JCL	34
Invoking DFDSS with the Application Interface	36
User Interaction Module Exit Functions	36

Chapter 6. Protecting DFDSS Functions	37
Protecting DFDSS/ISMF Functions with RACF	37
ISMF Functions You Might Want to Protect	37
Setting Up the Authorization Structure	37
DFDSS Keywords You Can Protect with RACF	39
Chapter 7. Managing Availability with DFDSS	41
Planning an Availability Strategy	41
Backup and Recovery	42
Disaster Recovery	43
Vital Records	44
Archiving	45
Backing Up Data Sets	45
Logical Data Set DUMP	46
Physical Data Set DUMP	47
A Backup Scenario	47
Backing Up Data Sets with Special Requirements	48
Backing Up SMS-Managed Data Sets	51
Backing Up Volumes	51
Logical Volume DUMP	51
Physical Volume DUMP	52
Efficiency Considerations for Dumping Data	52
Space Considerations	52
Performance Considerations	53
Shared DASD Considerations	53
Restoring Data Sets	53
Logical Data Set Restore	54
Physical Data Set Restore	58
Restoring Data Sets with Special Requirements	59
Restoring Multivolume Data Sets	59
Restoring Integrated Catalog Facility Catalogs	60
Restoring ISAM, Unmovable, DAM, and Absolute Track Data Sets	61
Restoring an Undefined DSORG Data Set	62
Restoring a VSAM Sphere	62
Restoring the VVDS and the VTOCIX	63
Restoring a PDSE	63
Restoring a Damaged PDS	63
Restoring Data Sets in an SMS Environment	64
Restoring SMS-Managed Data Sets	64
Restoring GDG Data Sets	68
Restoring Non-SMS-Managed Data Sets	69
Restoring Volumes	69
Chapter 8. Managing Data Movement with DFDSS	73
Preparing for Data Movement	73
Evaluating the Use of Logical and Physical COPY	74
What DFDSS Copies	74
Moving Data Sets	74
Specifying Input Volumes	75
Output Volume Selection	75
Renaming Data Sets	75
Moving to Preallocated Data Sets	76
Rules for Moving to Preallocated Target Data Sets	76
Specifying Multiple Target Volumes	77
How Keywords Work with Preallocated Targets	77
Defining RACF Profiles	78

Moving Data Sets with Special Requirements	79
Moving System Data Sets	79
Moving User Catalogs	80
Moving Multivolume Data Sets	80
Moving VSAM Data Sets	80
Moving a PDSE	81
Moving a Damaged PDS	81
Moving Unmovable Data Sets	82
Moving ISAM Data Sets	82
Moving BDAM Data Sets	82
Moving GDG Data Sets	83
Moving Volumes	84
Logical Volume COPY	84
Physical Volume COPY	84
VTOC Considerations	85
Moving Volumes to Like Devices of Equal Capacity	86
Moving Volumes to Like Devices of Greater Capacity	86
Moving Volumes to Unlike Devices	86
Considerations for Moving Data Sets in an SMS Environment	86
Moving SMS-Managed Data Sets	86
Moving Non-SMS-Managed Data Sets	88
Considerations for Moving Volumes in an SMS Environment	89
Chapter 9. Converting Data to and from SMS Management	91
Evaluating Conversion to SMS Management	91
Data Sets Ineligible for Conversion to SMS	92
Volume Eligibility for Conversion to SMS	92
Conversion by Data Movement	92
Converting to SMS Management by Data Movement	92
Converting from SMS Management by Data Movement	93
Conversion Without Data Movement	93
Simulating Conversion	93
Preparing a Volume for Conversion	94
Converting to SMS Management without Data Movement	94
Special Data Set Requirements for Conversion to SMS	96
VSAM Sphere Eligibility	96
Multivolume Data Sets	97
GDG Data Sets	97
Temporary Data Sets	98
VTOC and VVDS	98
Converting from SMS Management without Data Movement	98
Special Data Set Requirements for Conversion from SMS	98
Multivolume Data Sets	98
GDG Data Sets	99
Temporary Data Sets	99
VTOC, VTOCIX, and VVDS	99
Chapter 10. Managing Space with DFDSS	101
Reclaiming DASD Space	101
Releasing Unused Space in Data Sets	101
Compressing a PDS	102
Deleting Unwanted Data Sets	102
Combining Data Set Extents	103
Consolidating Free Space on Volumes	104
When to Run DEFrag	104
Data Sets Excluded from DEFrag Processing	104

DEFRAG Options	105
Serialization	105
Maximizing Track Utilization by Reblocking Data Sets	106
Security Considerations	107
Appendix A. ACS Routine Variables	109
ACS Variables Available during COPY	109
ACS Variables Available during RESTORE and CONVERTV	110
Appendix B. Changing Command Specifications for DFDSS Version 2 Release 4	113
Volume Specification	113
JCL Changes	114
Processing VSAM Spheres	114
Considerations for Processing VSAM Spheres	114
Reblocking Data Sets with RESTORE	115
Appendix C. Stand-Alone Restore Program	117
Storage Devices	117
Other Devices	117
Restrictions	118
Compatibility with DFDSS	119
Glossary of Terms and Abbreviations	121
Index	131

Figures

1.	Storage Requirements for DFDSS Operations with I/O Buffers below 16-Megabyte Virtual	7
2.	Minimum Storage Requirements for DFDSS Operations with I/O Buffers above 16-Megabyte Virtual	8
3.	Minimum Storage Requirements for a Restore to an Unlike Device	9
4.	Starting an ISMF Session from the ISPF Master Application Menu	22
5.	ISMF Primary Option Menu	22
6.	Complete Page 1 of the Data Set Selection Entry Panel	23
7.	The Resulting Data Set List	24
8.	Completing Page 1 of the COPY Command Entry Panel	24
9.	Completing Page 2 of the COPY Command Entry Panel	25
10.	Verifying the COPY Job Statement Information	25
11.	The DFDSS Execute Statement Entry Panel	26
12.	The List with Successful Submission of the COPY Command	26
13.	Complete Page 1 of the Volume Selection Entry Panel	27
14.	The Volume List	28
15.	Page 1 of the DUMP Entry Panel	28
16.	Page 2 of the DUMP Entry Panel	29
17.	Completing the DUMP Entry Panel	29
18.	DUMP Job Submission Entry Panel	30
19.	DFDSS EXECUTE Statement Entry Panel	30
20.	The DUMP Job Stream is Successfully Built	31
21.	Module Names for DFDSS/ISMF Line Operators	38
22.	Module Names for DFDSS/ISMF Data Set Application Commands	38
23.	RACF Facility Class Profile Names for DFDSS Keywords	39
24.	Output From a Dumped Integrated Catalog Facility User Catalog	50
25.	RACF-Protected Data Sets in a Logical Data Set RESTORE	57
26.	Output from Restored Integrated Catalog Facility User Catalog	61
27.	Defining Profiles—with RACF 1.5 or Later Installed	78
28.	SMS Report	96
29.	Data Set Erase Table for DEFRAG (with RACF Version 1 Release 7)	108
30.	Variables Passed to ACS Routines during DFDSS COPY	109
31.	Variables Passed to ACS Routines during DFDSS RESTORE and CONVERTV	110

Summary of Changes

Version 2 Release 5, December 1989

DFDSS Version 2 Release 5 provides additional storage management facilities and enhancements to help you:

- Take advantage of a new type of data set.

MVS/DFP Version 3 Release 2 now provides a specially designed data set, called a partitioned data set extended (PDSE). DFDSS supports the PDSE, which offers significant advantages over a partitioned data set (PDS). DFDSS also provides new keywords to convert a PDS to a PDSE and a PDSE to a PDS.

This support requires MVS/DFP Version 3 Release 2 or a later release.

- Use new application interface exit points.

New options for application interface exits further enhance the interaction between DFDSS and its invoking program at the data set level. DFDSS passes control to the user interaction module (UIM) when a data set is selected and when the processing of that data set is completed. At these exit points, the UIM can perform a number of tasks. For example, you can customize the UIM to end processing of the data set or to bypass options such as serialization and security verification.

- Process data sets with a high-level qualifier of SYS1.

By using the PROCESS(SYS1) keyword, you can:

- Release unused space in partitioned or sequential data sets with a high-level qualifier of SYS1.
- Move data sets with a high-level qualifier of SYS1.
- Specify the DELETE keyword with the DUMP command to dump and delete data sets with a high-level qualifier of SYS1.
- Replace preallocated data sets that have a high-level qualifier of SYS1.

- Process data sets that have undefined DSORG.

By using the PROCESS(UNDEFINEDSORG) keyword, you can copy or restore undefined DSORG data sets to an unlike device that has a larger track capacity.

- Specify both the OPTIMIZE and COMPRESS keywords for dumps.

You no longer must choose between performance (by specifying OPTIMIZE) or saving space (by specifying COMPRESS). You can back up data by using both keywords.

- Reduce I/O to the VVDS during logical data set DUMP/RESTORE, data set COPY and CONVERTV.

- Use stand-alone RESTORE to restore compressed dump data.

- Declare a larger block size for a SYSPRINT data set or the output data set from the PRINT command.

DFDSS now allows a block size greater than 141.

- Perform authorization checking during RESTORE for non-preallocated data sets.
- Reduce CPU time by improving the data compression algorithm during DUMP.
- Specify the target status of SMS-managed generation data sets.

Version 2 Release 4, December 1988

Storage Management Subsystem (SMS) Support

- When processing a data set, DFDSS recognizes and preserves the data, management, and storage classes associated with the data set.
- COPY and DUMP/RESTORE commands move data sets between SMS- and non-SMS-managed volumes.
- The new CONVERTV command converts volumes to and from SMS management without data movement.
- DFDSS allows data set selection based on data, management, and storage class names.
- The new INCAT keyword allows DFDSS to locate source data sets cataloged outside the standard search order for COPY, DUMP, and CONVERTV. This keyword can be used in place of JOBCAT/STEPSCAT statements, which are prohibited when processing SMS-managed data sets.

Functional Enhancements

- The new SPHERE keyword allows DFDSS to move an entire VSAM sphere (base cluster and all associated alternate index clusters and paths) with one invocation of COPY, DUMP, or RESTORE.
- DFDSS COPY and RESTORE have the capability to allocate multivolume VSAM target data sets.
- The % character acts as a place holder for DFDSS data set name filtering.
- DFDSS reblocks data sets that have the reblockable indicator set on.
- The REBLOCK keyword (already supported for COPY) is supported for logical RESTORE.
- Multiple arguments for filtering criteria can be specified on the BY keyword.
- A new application interface exit point (EIOP20) has been added. DFDSS passes control to this exit when a data set is allocated by DFDSS during a physical data set RESTORE operation.
- When DFDSS (COPY or logical RESTORE) detects an under-allocation of the non-VSAM target data set, it estimates the amount of additional space needed and extends the data set accordingly. The copy or restore then continues, using the additional space.

Version 2 Release 3, June 1987

Information is added to support DFDSS Version 2 Release 3 enhancements, which include:

- Data Facility Family enhancements. With MVS/Extended Architecture Data Facility Product (MVS/XA DFP) Version 2 Release 3 installed:
 - The Interactive Storage Management Facility (ISMF) supports volume-oriented functions, for example, DEFRAAG and full-volume DUMP.
 - DFDSS supports enhanced integrated catalog facility catalog recovery. You can restore a user catalog without bringing your application program down. This allows continuous applications such as IBM DATABASE 2 (DB2) and IMS/VS to stay up while a damaged user catalog is replaced.
 - DFDSS supports VSAM linear data sets.
 - DFDSS supports data set expiration dates beyond the year 1999.
 - DFDSS preserves aliases when VSAM user catalogs are copied or dumped and restored. They no longer need to be re-created.
- Faster copying of DASD data.
- The ability to do logical processing at a volume level, thus allowing you to restore to an unlike device and to incrementally back up volume pools. The new keywords are LOGINDDNAME, LOGINDYNAM, and ALLMULTI.
- Automatic rebuilding of the VTOC free space chain and VTOC index after a full-volume COPY or RESTORE to a like device of larger capacity.
- Choice in target volume space allocation, allowing you to allocate by cylinders, blocks, or tracks, or to preserve the source space allocation. The new keyword is TGTALLOC.
- An improved algorithm to reblock partitioned and sequential data sets. This requires MVS/XA DFP Version 2 Release 2 or a later release.
- PAM data sets with NOTE lists are updated.
- DFDSS messages now provide the name of the issuing module, to improve serviceability.
- Optional RACF logging to SMF data sets.

Chapter 1. Introduction to DFDSS

Data Facility Data Set Services (DFDSS) is a direct access storage device (DASD) data and space management tool. You can use DFDSS to:

- Copy and move data sets between volumes of like and unlike device types¹
- Dump and restore data sets, entire volumes, or specific tracks
- Convert data sets and volumes to and from SMS management
- Compress partitioned data sets
- Release unused space in data sets
- Reduce or eliminate DASD free-space fragmentation by consolidates free space on a volume.

Stand-Alone DFDSS Restore Program: With the stand-alone DFDSS restore program, you can restore:

- A full volume or ranges of tracks
- Your system residence (SYSRES) volume, if your operating system fails to IPL.

Refer to Appendix C, "Stand-Alone Restore Program" on page 117 for more information on the stand-alone restore program.

The Role of DFDSS

DFDSS can help you perform many of your storage management tasks. The role DFDSS plays at your site depends on what other Data Facility products you use. In particular, the way you use DFDSS depends on whether or not you also use the Data Facility Product, Storage Management Subsystem (SMS), and the Data Facility Hierarchical Storage Manager (DFHSM).

To understand the role of DFDSS in an SMS environment, you need a basic understanding of SMS.

Managing User Data with SMS

SMS can help you efficiently manage space, availability, and performance to meet user requirements.

SMS allows you to match the needs of users' data (like data set organization, size, and format) to the characteristics of storage devices without requiring users to know or understand the hardware configuration at your site. With SMS, users can store and retrieve data without being aware of space limitations, device characteristics, or volume serial numbers.

¹ Like devices have the same track capacity and number of tracks per cylinder (for example, 3380 Standard and 3380 Model D or E). Unlike DASD devices have different track capacities, a different number of tracks per cylinder (for example, 3350 and 3380), or both.

Using SMS, you can define allocation management criteria for the different types of data at your site. The values you specify identify your users' requirements for space, availability, and performance. You define these values to SMS as:

Data Class A named list of data set allocation and space attributes that SMS assigns to a data set when it is created.

Management Class A named list of management attributes that SMS uses to control DFHSM action for data set retention, migration, backup, and release of allocated but unused space.

Storage Class A named list of data set storage service attributes that SMS uses to identify performance and availability requirements.

Storage Group A named collection of DASD volumes that have been grouped to meet a defined service strategy.

Automatic class selection (ACS) is the SMS mechanism for assigning SMS classes and storage groups (otherwise known as constructs). Depending on the DFDSS command you are using, SMS invokes some or all of the ACS routines in the following order:

1. Data class ACS routine
2. Storage class ACS routine
3. Management class ACS routine
4. Storage group ACS routine.

SMS uses the assigned constructs to automatically place and manage data and storage. For example, you can use a storage class to keep performance-sensitive data on high-speed storage devices and management classes to control the movement of less active data to tape. Because SMS controls data placement, you are no longer required to perform the time-consuming task of hand-placing this sensitive data.

For a more complete discussion of SMS and how to use it, see *MVS/ESA Storage Administration Reference*.

Managing Availability with DFDSS

DFDSS availability management consists of backing up data on DASD to tape and restoring the backup if the original is lost, damaged, or inadvertently changed.

The two general forms of backup are:

Data set backup Protects against the loss of individual data sets.

Volume backup Protects against the loss of a volume.

For data set backup, you should consider doing incremental backups to help reduce processing time while still meeting your backup requirements. Incremental backup means that data sets are backed up only if they have changed since their last backup. With this form of backup, data sets are backed up only if they require it.

Volume backups are used to protect against media failure. You can use volume backups in conjunction with incremental data set backups to recover a volume. As a result, you need not do volume dumps as frequently. Incremental data set backups should be done on a daily basis and volume backups on a weekly basis. If a volume is lost for some reason, you can restore the most recent volume backup and apply incremental data set backups to the volume to bring it back to its most current status.

Backing Up Volumes and Data Sets

You use the DFDSS DUMP command to back up volumes and data sets, and you use the RESTORE command to recover them. By doing a data set DUMP with RESET specified and filtering on the data-set-changed flag, you can make incremental backups of your data sets with DFDSS.

In an SMS environment, DFDSS saves the class names for the data sets it dumps. When you restore the data set to an SMS volume, DFDSS invokes ACS and passes it the class names saved with the data set. Based upon this and other input from DFDSS (for example, class names specified with the STORCLAS or MGMTCLAS keywords), ACS assigns SMS constructs to each data set.

Because DFDSS restore invokes ACS, you can restore the data sets to SMS volumes. Conversely, data sets backed up as SMS data sets can be restored as non-SMS data sets.

In addition to providing for routine backup requirements, you can use DFDSS to back up application data for disaster recovery and vital records purposes. By using DFDSS logical data set DUMP and filtering on data set names, you can back up all the data sets associated with a particular application for disaster recovery or vital records. If you do not want to perform a separate DUMP operation for disaster recovery, you can specify more than one OUTDDNAME to create up to 255 separate backup copies when you do your routine backup. These extra copies can then be used for disaster recovery or vital records purposes. The DUMP command can also be used to archive data sets that have not been accessed for long periods of time.

Using DFHSM for Backup

DFHSM provides automated incremental backup, interactive recovery, and inventory of what it backs up. You can use DFDSS to complement DFHSM backup and recovery or, if you do not have DFHSM, to provide for all your backup and recovery requirements.

If DFHSM is being used, you should use DFDSS for volume backup and the backup of data sets not supported by DFHSM. If DFHSM is not installed, you can use DFDSS for all of your volume and data set backup.

Managing Data Movement with DFDSS

DFDSS can help you move data to replace devices, add capacity, and meet performance requirements. There are three general types of data movement: data set, volume, and tracks movement.

Moving Data

Using the DFDSS COPY command, you can perform data set, volume, and track movement. To perform data set movement, specify COPY with DELETE. This causes DFDSS to delete the source data set after it successfully copies the data set.

Full-volume COPY is useful for moving data between like devices. If you are moving volumes to like devices of larger capacity, generally you need to have a larger VTOC because the larger device can hold more data sets. DFDSS rebuilds indexed VTOCs and recognizes larger VTOCs on target volumes when the VTOCs are moved to a like device of larger capacity. However, if you are moving data

between unlike devices, you must use logical data set COPY for all the data sets on the volume.

When moving data sets to unlike devices, DFDSS fills the tracks as completely as possible instead of just copying track for track. In addition, if the reblockable indicator is set on, the data set is reblocked to a system-determined block size efficient for the device.

Moving Data in an SMS Environment

Data set movement is much different in an SMS environment. ACS routines and VTOC/Data Set Services determine the target volume in an SMS environment.

DFDSS moves data sets to different volumes if their storage groups change. However, if their storage groups do not change, DFDSS might move the data sets to a different location on the same volume or to different volumes.

If a new, empty volume is added to a storage group, data sets moved into that storage group are likely to be placed on that volume. This can help you when you replace volumes in a storage group.

If a data set's storage class has the guaranteed space attribute and the user specifies input volumes, the data set is placed on the specified volume(s) if:

- All volumes specified with OUTDDNAME or OUTDYNAM are SMS managed and belong to the same storage group.
- The ACS storage group routine assigns the data set to the storage group that contains the specified volumes.

Note: SMS-managed data sets must be cataloged in the standard order of search.

Converting Data to and from SMS Management

DFDSS is the primary tool for converting data to and from SMS management.

There are two ways of converting data to and from SMS management:

- Conversion of data sets with data movement
- Conversion of volumes without data movement.

The following sections briefly describe these two kinds of conversion.

Conversion of Data Sets by Data Movement

To convert data sets by data movement, use DFDSS COPY or DUMP/RESTORE. When moving data sets to SMS-managed volumes from non-SMS-managed volumes, DFDSS invokes ACS, which may assign class names to the data sets. Alternatively, you can specify the BYPASSACS and STORCLAS keywords on COPY or RESTORE to force the data sets to be SMS managed.

When moving data sets out of SMS management, specify the BYPASSACS and NULLSTORCLAS keywords on COPY or RESTORE. This causes DFDSS to bypass ACS and drop the data set's class names. You should also be aware that ACS can make data sets non-SMS managed.

Conversion of Volumes without Data Movement

To convert volumes to and from SMS management without data movement, you can use the DFDSS CONVERTV command. This command allows you to:

- **Prepare a volume for conversion.** Using the PREPARE keyword, you can stop new allocations and data set extensions to another volume while still allowing access to the data on the volume.
- **Convert a volume to SMS management.** Using the SMS keyword, you can convert a volume and all its data sets to SMS management.
- **Convert a volume from SMS management.** Using the NONSMS keyword, you can remove a volume and its data sets from SMS management.
- **Simulate conversion.** Using the TEST keyword, you can verify that the volume and its data sets are eligible for conversion and see what class names ACS would assign to the data sets.

Managing Space with DFDSS

DFDSS has four functions to help you manage your DASD space:

COMPRESS	Compresses your partitioned data sets by taking unused space and consolidating it at the end of the data set. To make the unused space available for other data sets, you must use the RELEASE command.
RELEASE	Releases the unused space in sequential and partitioned data sets for use by other data sets.
DEFRAG	Consolidates the free space on a volume to help prevent out-of-space abends on new allocations.
DUMP/RESTORE	Deletes unwanted data sets and combines data set extents. (The COPY command can also be used to combine data set extents.)

Chapter 2. Requirements for Running DFDSS

Certain requirements must be met in order to run DFDSS. This chapter describes these various requirements.

Operating Environment

You can run DFDSS in an MVS system environment. This version of DFDSS runs under:

- MVS/370 DFP Version 1 Release 1 and its corequisite MVS/System Product (MVS/SP) Version 1 Release 3
- MVS/XA DFP Version 2 Release 1 and its corequisite MVS/SP Version 2 Release 1.2
- MVS/ESA DFP Version 3 Release 1 and its corequisite MVS/SP Version 2 Release 2.3

and any subsequent releases of these system control programs unless otherwise indicated.

To take full advantage of the new or improved functions of DFDSS Version 2 Release 5, you must have MVS/ESA DFP Version 3 Release 2 and MVS/SP Version 3 Release 1.0E.

Storage Requirements

In most cases, you can allow DFDSS to determine the amount of storage it uses for an operation. Sometimes, however, you might want closer control of the amount of storage DFDSS uses. Use the storage estimates in this section as a starting point for determining minimum region sizes in which DFDSS can run. Figure 1 and Figure 2 on page 8 show the minimum storage requirements, in bytes, to run each DFDSS operation. Figure 3 on page 9 shows the minimum storage requirements, in bytes, to restore PAM and VSAM data sets to unlike devices. The values include the storage required to load the DFDSS program into the region.

Storage requirements depend on your operating system configuration and your device and data set characteristics. The storage requirement estimates shown for COPY, DUMP, and RESTORE are for full volume COPY, DUMP, or RESTORE only; they might vary for a data set operation.

If DFDSS determines that the storage requirements are greater than the storage available during processing, DFDSS issues error message ADR376E to indicate this. The out-of-storage condition might cause an abend 80A during DFDSS post-processing.

If you use buffers above 16 megabytes virtual storage, the buffer size is allocated independently of the region size.

If you use the PARALLEL command to run two or more DFDSS tasks concurrently, the total storage required is the sum of the storage required for all functions to be run in parallel.

DFDSS Command	Storage Requirements
COMPRESS	dssize & copysize + (2 * trksize of largest device) + (1152K * number of additional volumes, up to four)
CONVERTV	dssize + (buffersize*5)
COPY (FULL)	dssize + (trksize * 5)
COPYDUMP	dssize + (buffersize * 5)
DEFRAG	dssize + (trksize * 5) + (16K if VSAM present)
DUMP (FULL) OPT(1)	dssize + (buffersize * 7)
DUMP (FULL) OPT(2)	dssize + (buffersize * 6)
DUMP (FULL) OPT(3)	dssize + (buffersize * 15)
DUMP (FULL) OPT(4)	dssize + (buffersize * (3 * trk/cyl))
PRINT	dssize + (3 * trksize)
RELEASE	dssize + trksize
RESTORE (FULL)	dssize & copysize + (buffersize * 6)

Figure 1. Storage Requirements for DFDSS Operations with I/O Buffers below 16-Megabyte Virtual

DFDSS Command	Storage Below 16 Mb Virtual	Storage Above 16 Mb Virtual
COMPRESS	dssize & copysize + 1152K * number of additional volumes, up to four	2 * trksize of largest device
CONVERTV	dssize	buffersize * 5
COPY (FULL)	dssize	trksize * 5
COPYDUMP	dssize	buffersize * 5
DEFRAG	dssize + 16K if VSAM present	trksize * 5
DUMP (FULL) OPT(1)	dssize	buffersize * 7
DUMP (FULL) OPT(2)	dssize	buffersize * 6
DUMP (FULL) OPT(3)	dssize	buffersize * 15
DUMP (FULL) OPT(4)	dssize	buffersize * (3 * trk/cyl)
PRINT	dssize	3 * trksize
RELEASE	dssize	trksize
RESTORE (FULL)	dssize	buffersize * 6

Figure 2. Minimum Storage Requirements for DFDSS Operations with I/O Buffers above 16-Megabyte Virtual

Type of Data Set	Storage Requirements
PAM Data Sets	$dssize + (((trksize + 64) * 5) + 8K)$
VSAM Data Sets	$dssize + (((trksize + 64) * 5) + (2 * \text{maximum record size}) + (3 * \text{buffspace}))$

Figure 3. Minimum Storage Requirements for a Restore to an Unlike Device

Legend

K	1024 bytes
dssize	DFDSS load module size, 1240K
copysize	IEBCOPY load module size
buffersize	The maximum of trksize and default maximum block size (32K)
buffspace	Buffer space specified in DEFINE command when data set was allocated
trksize	The size, in bytes, of a track on your DASD volume
trks/cyl	The number of tracks per cylinder on your DASD volume

DASD Requirements

You can use DFDSS with the following IBM DASD:

- 2305 Fixed Head Storage Model 2
- 3330 Disk Storage Models 1 and 11
- 3340 Direct Access Storage Facility
- 3344 Direct Access Storage
- 3350 Direct Access Storage (in native mode and in IBM 3330 Models 1 and 11 compatibility mode)
- 3375 Direct Access Storage
- 3380 Direct Access Storage (all models)
- 3330 virtual volumes residing in an IBM 3850 Mass Storage System.

In addition, you can use DFDSS with all magnetic tape devices, system consoles, printers, and card readers supported by Data Facility Product.

Stand-Alone DFDSS Restore Program

Refer to Appendix C, "Stand-Alone Restore Program" on page 117 for a list of devices you can use with the stand-alone DFDSS restore program.

Volume Formats

You can use DFDSS with the following DASD volume formats:

- Volumes with indexed VTOCs
- Volumes with nonindexed VTOCs
- OS/VS minivolumes in a VM environment.

All DASD volumes used by DFDSS must be initialized by Device Support Facilities (ICKDSF).

Data Set Organizations

DFDSS can copy, dump, and restore data sets of the following organization types:

- SAM (sequential access method)
- PAM (partitioned access method)
 - PDS (partitioned data set)
 - PDSE (partitioned data set extended)
- BDAM (basic direct access method)
- ISAM (indexed sequential access method)
- VSAM² (virtual storage access method), including:
 - ESDS (entry-sequenced data set)
 - RRDS (relative record data set)
 - KSDS (key-sequenced data set)
 - KSDS with key ranges
 - LDS (linear data set)
- DATABASE 2 (DB2)
- DATABASE 2 (DB2) data sets, supported
- EXCP (execute channel program)
- Unmovable data set types (PSU, POU, DAU, ABSTR, ISU, BDAM with OPTCD = A).

² VSAM data sets that are cataloged in an integrated catalog facility catalog

Chapter 3. Logical/Physical Processing and Data Set Filtering

Before you begin using DFDSS, you should understand the difference between logical and physical processing and how to use data set filtering to select data sets for processing. The following sections describe these two aspects of DFDSS.

Logical and Physical Processing

DFDSS can perform two kinds of processing when executing COPY, DUMP, and RESTORE commands:

- *Logical processing* operates against data sets independently of physical device format.
- *Physical processing* moves data at the track-image level and operates against volumes, tracks, and data sets.

Each type of processing offers different capabilities and advantages.

During a RESTORE, the data is processed the same way it was dumped because physical and logical dump tapes have different formats. If a data set is dumped logically, it is restored logically. If it is dumped physically, it is restored physically.

Logical Processing

A logical COPY, DUMP, or RESTORE treats each data set and its associated information as a logical entity, and processes an entire data set before beginning the next one. Each data set is moved as a set of data records, allowing data movement between devices with different track and cylinder configurations.

DFDSS performs logical processing if:

- You specify the DATASET keyword with COPY. A data set COPY is always a logical operation, regardless of how or whether you specify input volumes.
- You specify the DATASET keyword with DUMP, and either no input volume is specified, or LOGINDDNAME or LOGINDYNAM is used to specify an input volume(s).
- The RESTORE command is performed and the input tape was created by a logical DUMP.

Both catalogs and VTOCs are used to select data sets for logical processing. The catalogs are used to select data sets for COPY and DUMP if you do not specify input volumes. If you specify input volumes for COPY, the VTOCs are used to select data sets. If you specify input volumes for DUMP with LOGINDDNAME or LOGINDYNAM, the VTOCs are used for selecting data sets.

Note: To copy or dump entire multivolume data sets, you need not specify all the volumes in the LOGINDDNAME or LOGINDYNAM volume list when you also specify the ALLMULTI keyword.

When to Use Logical Processing

You should use logical processing when:

- Data is copied to an unlike device type.

Logical processing is the only way to move data between unlike device types.

- Data that may need to be restored to an unlike device is dumped.

Data must be restored the same way it was dumped. This is particularly important to bear in mind when making backups that you plan to retain for a long period of time (such as vital records backups). If a backup is retained for a long period of time, it is possible that the device type it originally resided on will no longer be in use at your site when you want to restore it. This means you will have to restore it to an unlike device, which can be done only if the backup was made logically.

- Aliases of VSAM user catalogs are to be preserved during copy and restore.

Aliases are not preserved for physical processing.

- Unmovable data sets or data sets with absolute track allocation are moved to different locations.

- Multivolume data sets are processed.

- VSAM and multivolume data sets are to be cataloged in a different catalog.

- Data sets are to be deleted from the source volume after a successful dump or copy.

- Both non-VSAM and VSAM data sets are to be renamed after a successful copy or restore.

- You want to control the percentage of space allocated on each of the output volumes for copy and restore.

- You want to copy and convert a PDS to a PDSE or vice versa.

- You want to copy or restore a data set with an undefined DSORG to an unlike device.

Physical Processing

Physical processing moves data based on physical track images. Because data movement is carried out at the track level, only target devices with track sizes equal to those of the source device are supported. Physical processing operates on volumes, ranges of tracks, or data sets. For data sets, it relies only on volume information (in the VTOC and VVDS) for data set selection, and processes only that part of a data set residing on the specified input volumes.

DFDSS performs physical processing if:

- You specify the FULL or TRACKS keyword with COPY or DUMP. This results in a physical volume or physical tracks operation.
- You specify the DATASET keyword on the DUMP command and input volumes with the INDDNAME or INDYNAM parameter. This produces a physical data set dump.
- The RESTORE command is executed and the input volume was created by a physical DUMP.

When to Use Physical Processing

You should use physical processing when:

- Backing up system volumes that you might want to restore with stand-alone DFDSS restore.

Stand-alone DFDSS restore supports only physical dump tapes.

- Performance is an issue.

Generally, the fastest way—measured by elapsed time—to copy or dump an entire volume is with a physical full volume command. This is primarily because minimal catalog searching is necessary for physical processing.

- Substituting one physical volume for another or recovering an entire volume.

With a COPY or RESTORE—full volume or track—the volume serial number of the input DASD volume can be copied to the output DASD volume.

- Dealing with I/O errors.

Physical processing provides the capability to COPY, DUMP, and RESTORE a specific track or range of tracks.

Choosing Data Sets for Processing—Filtering

You can select data sets for DFDSS processing by filtering on specified criteria. DFDSS can filter on fully- or partially-qualified data set names (by using the INCLUDE or EXCLUDE keyword) and on various data set characteristics (by using the BY keyword).

You can filter data sets with any of the following commands:

- Logical DUMP
- Logical RESTORE
- Physical data set DUMP
- Physical data set RESTORE
- Data set COPY
- COMPRESS
- DEFrag
- RELEASE.

At least one of the INCLUDE, EXCLUDE, or BY parameters must be specified with the above commands.

The following sections briefly describe what you can filter on and how you might use the available criteria.

Filtering by Data Set Names

Using **INCLUDE** or **EXCLUDE**, you can filter on fully- or partially-qualified data set names. A fully-qualified data set name is one in which all qualifiers are completely spelled out. For example:

```
(INCLUDE(SYS1.UTIL3.LOAD))
```

A partially-qualified data set name is one in which the qualifiers are not completely spelled out. Using asterisks (*) and percent signs (%), you can select data sets without specifying their fully-qualified names.

The single asterisk (*) is used in place of one qualifier. For example:

```
(INCLUDE(ABC.*.LOAD))
```

This partially-qualified name matches **ABC.DEF.LOAD** and **ABC.XYZ.LOAD**. The single * is also used to indicate to **DFDSS** that only part of a qualifier has been specified. For example, if you want to filter using only the first three characters of the first qualifier of a name, specify it as follows:

```
(INCLUDE(SYS*.**))
```

This partially-qualified name matches data sets whose first qualifier was **SYS1** and **SYS1A**. The other qualifiers in the data set name are ignored.

When used with other qualifiers, the double asterisk (**) indicates that one or more leading, trailing, or middle qualifiers do not exist or they do not play a role in the selection process. For example:

```
(INCLUDE(**.LOAD))
```

This partially-qualified name selects any data set with **LOAD** as its last qualifier (such as data sets named **LOAD**, **ABC.LOAD**, and **ABC.DEF.LOAD**).

The percent sign (%) is used as an ignore character. One or more %s can be specified in any qualifier. Each specification represents one character in the name being filtered and any character in that position in the name is ignored. For example:

```
(INCLUDE(SYS1.A%B))
```

This partially-qualified name matches **SYS1.AZZB** and **SYS1.AXYB**, but not **SYS1.AXXXB**.

Filtering by Data Set Characteristics

In addition to filtering on data set names, you can also filter on the following data set characteristics:

- Allocation type (cylinder, track, block, absolute track, or movable)
- Creation date (absolute or relative)
- Expiration date (absolute or relative)
- Last-referenced date³ (absolute or relative)
- Data set organization (SAM, PAM, PDS, PDSE, BDAM, EXCP, ISAM, or VSAM)
- Data set size (number of allocated or used tracks)
- Number of extents
- Whether the data set is single- or multivolume
- Whether the data-set-changed flag is on or off³
- SMS class names (data, management, and storage class)
- Whether a data set is cataloged or not (using the standard catalog search order).

You can use any of the following operators with the **BY** keyword:

- EQ or =** Equal to
- LT or <** Less than
- LE or <=** Less than or equal to
- GT or >** Greater than
- GE or >=** Greater than or equal to
- NE or !=** Not equal to.

When you specify multiple arguments for an NE operation, DFDSS selects only those data sets not matching any of the arguments. When you specify multiple arguments for an EQ operation, DFDSS selects those data sets matching any of the arguments.

³ The data-set-changed flag (DSCHA) and the last-referenced date (REFDT) are supported only on systems that have data-set-changed flag support installed. For VSAM data sets, DSCHA and REFDT are maintained by Data Facility Hierarchical Storage Manager (DFHSM). Therefore, if you do not have DFHSM on your system, module IDATMSTP (in load module IDA0192A) should be modified to pass a return code of 4 to its caller.

Some Examples of Filtering by Data Set Characteristics

If you use the following specification of the BY keyword, DFDSS selects all data sets allocated in cylinders:

```
BY( ALLOC,EQ,CYL )
```

You can specify more than one criterion with the BY keyword. The following example selects all data sets allocated in cylinders and whose management class is MCNAME1:

```
BY(( ALLOC,EQ,CYL ) ( MGMTCLAS,EQ,MCNAME1 ))
```

You can specify multiple arguments for any of the filtering criteria. The following example selects all data sets that have a data class of DCNAME1 or DCNAME2:

```
BY( DATACLAS,EQ,(DCNAME1,DCNAME2) )
```

Some Uses of Filtering

To make good use of filtering by data set names, you should have meaningful naming conventions. Your naming conventions should allow you to identify large groups of data sets that can be treated similarly. If you have such conventions, you can use DFDSS data set name filtering to select large groups of data sets against which you can run DFDSS functions.

For more information on setting up data set naming conventions, see *SML: Managing Data Sets*.

Suppose you are a storage administrator and every day you want to back up all payroll data sets that have changed since they were last backed up. If the data sets you need to back up have some identifying qualifier(s), perhaps PAYROLL.FEDTAX, you can select them by coding:

```
//VRPAY    JOB  Accounting Information,MORGAN
//STEP1    EXEC PGM=ADRDSU,REGION=4000K
//SYSPRINT DD  SYSOUT=*
//DROUT    DD  DSN=PAYROLL.DAY1,DISP=(NEW,CATLG),UNIT=3480
//SYSIN    DD  *
           DUMP DATASET(INCLUDE(PAYROLL.FEDTAX.**)) -
           BY((DSCHA,EQ,YES) (MGMTCLAS,EQ,DAILY))) -
           OUTDD(DROUT)
/*
```


Filtering by data set characteristics also allows you to process large groups of data sets. You can use BY criteria to:

- Filter on the data-set-changed-flag to back up only those data sets that have not been backed up since they were last updated.
- Filter to select uncataloged data sets for deletion as a means of enforcing cataloging.
- Filter to select data sets whose expiration date passed for deletion.
- Filter on the last referenced date and archive or delete data sets that have gone unreferenced for a long period of time (for example, 18 months).
- Filter on data set size to ensure that when you use COMPRESS and RELEASE you only compress and release space in data sets where the savings may be significant.
- Filter on management class to perform space management.

It is possible to pass DFDSS filtering criteria in a data set. If you do this, the data set should have the following characteristics:

RECFM = F or FB

LRECL = 80

BLKSIZE = 80 for F or multiple of 80 for FB.

Chapter 4. Changing Command Specifications for DFDSS Version 2 Release 5

Some new keywords have been added for DFDSS Version 2 Release 5. The following sections briefly explain what these new keywords are and how they affect DFDSS processing.

Note: Refer to Appendix B, "Changing Command Specifications for DFDSS Version 2 Release 4" on page 113 for new keywords for that release.

Converting to and from a PDSE

You can use the CONVERT keyword in conjunction with the PDSE subkeyword or the PDS subkeyword when you perform a DFDSS COPY operation. If you use the PDSE subkeyword, the resulting target data set will be converted from the PDS format to the PDSE format. If you use the PDS subkeyword, the resulting target data set will be converted from the PDSE format to the PDS format. MVS/DFP V3 R2 and MVS/SP V3 R1.0E are required for this operation.

Restrictions for COPY, DUMP, and RESTORE

A PDSE cannot be copied, dumped, or restored on a system using releases prior to DFDSS Version 2 Release 5 nor can they be copied, dumped, or restored on a system using releases prior to MVS/DFP V3 R2 and MVS/SP V3 R1.0E

Processing SYS1 Data Sets

You can use the new PROCESS keyword in conjunction with the SYS1 subkeyword to:

- Copy SYS1 data sets to preallocated targets
- Move SYS1 data sets
- Allow DELETE when logically or physically dumping SYS1 data sets
- Release unused space in SYS1 data sets.

Processing Undefined DSORG Data Sets

You can use the new PROCESS keyword in conjunction with the UNDEFINEDSORG subkeyword to:

- Copy a data set with an undefined DSORG to an unlike device of larger track capacity
- Logically restore a data set with an undefined DSORG to an unlike device of larger track capacity.

Omitting the TGTALLOC Keyword with COPY or RESTORE Commands

If the TGTALLOC keyword is omitted when you use the COPY or RESTORE commands, the target allocation default has changed from tracks to source.

Specifying Both the OPTIMIZE and COMPRESS Keywords for Dumps

You no longer must choose between performance (by specifying OPTIMIZE) or saving space (by specifying COMPRESS). The two keywords can be specified together; they are no longer mutually exclusive.

Selecting Arguments for Filtering by Data Set Characteristics

PDS or PDSE are new arguments that you can use to qualify the selection characteristics of a data set.

Specifying the Status of the Target Generation Data Set

With the TGTGDS keyword, you can perform a data set COPY or RESTORE to specify the final status of a SMS-managed GDG data set.

Chapter 5. Invoking DFDSS

You can invoke DFDSS in any one of three ways:

- With the Interactive Storage Management Facility (ISMF)
- With JCL
- With the Application Interface.

General programming interfaces are contained in this chapter.

Invoking DFDSS with ISMF

You can use the menu-driven panels of ISMF, a component of the MVS/XA Data Facility Product, to build job streams for many DFDSS space management and backup functions of DFDSS. ISMF supports the DFDSS commands COMPRESS, CONVERTV, COPY, DEFRAG, DUMP, RELEASE, and RESTORE.

The information you supply on ISMF panels is used to build and submit job streams like those you generate using JCL and DFDSS commands. Using ISMF panels, you no longer have to remember DFDSS keywords and syntax. You merely fill in the values you want on the panels and ISMF generates the job stream. You can then either submit the job or save the job stream for later use.

Using ISMF panels, you can construct a list of data sets or volumes. The list provides you with a variety of information about each volume or data set (for example, allocated space and percent of unused space). The list-building process is similar to filtering in DFDSS—you supply the selection criteria to ISMF, and ISMF builds a list of data sets or volumes that fits your criteria. Because the list provides so much different information in one place, you can use it to analyze and manage your data and storage more efficiently.

How to Invoke ISMF

You invoke ISMF by logging on to TSO. If ISMF is installed as an option on the ISPF Master Application Menu or as an option on the ISPF/PDF Primary Option Menu, specify the selection option that corresponds to ISMF. To select ISMF in the ISPF Master Application Menu (see Figure 4 on page 22), enter I following the arrow on the command line.

```

----- ISPF MASTER APPLICATION MENU -----
SELECT APPLICATION ==> I

  1 SPF      - SPF PROGRAM DEVELOPMENT FACILITY      USERID  - IBMUSER
  . . .      - . . .                               TIME    - 07:30
  . . .      - . . .                               TERMINAL - 3279
  . . .      - . . .                               PF KEYS - 12
  . . .      - . . .
  I ISMF     - INTERACTIVE STORAGE MANAGEMENT FACILITY
  . . .      - . . .
  . . .      - . . .
  . . .      - . . .
  P PARMS    - SPECIFY TERMINATION PARAMETERS AND LIST/LOG DEFAULTS
  X EXIT     - TERMINATE USING LIST/LOG DEFAULTS

PRESS THE END KEY TO TERMINATE

```

Figure 4. Starting an ISMF Session from the ISPF Master Application Menu

Or if you want to invoke ISMF directly from TSO, use this command:

ISPSTART PGM(DGTFMD01) NEWAPPL(DGT)

In either case, the ISMF Primary Option Menu appears (Figure 5), and you can begin an ISMF session.

```

                                ISMF PRIMARY OPTION MENU
ENTER SELECTION OR COMMAND ==> 1

SELECT ONE OF THE FOLLOWING:

  0 ISMF PROFILE      - Change ISMF user profile
  1 DATA SET         - Perform Functions Against Data Sets
  2 VOLUME            - Perform Functions Against Volumes
  3 MANAGEMENT CLASS - Specify Data Set Backup and Migration Criteria
  4 DATA CLASS       - Specify Data Set Allocation Parameters
  5 STORAGE CLASS     - Specify Data Set Performance and Availability
  X EXIT              - Terminate ISMF

USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 5. ISMF Primary Option Menu

ISMF Data Set Application

You can use the data set application of ISMF to perform DFDSS functions against one or more data sets on a list you create. See *ISMF User's Guide* for more information on generating data set and volume lists.

Building a List of Data Sets

Complete the Data Set Selection Entry Panel with the values shown to the right of the arrows in Figure 6. Entering ** in the "Data Set Name" field selects all data sets that have your user ID as their high-level qualifier.

When each panel first appears, it is primed with the last-used values. If you want the default for any field instead, enter blanks over the last-used value.

```

                                DATA SET SELECTION ENTRY PANEL                Page 1 of 4
COMMAND ==>>

FOR A DATA SET LIST, SELECT SOURCE OF GENERATED LIST ==>> 2 (1 OR 2)

  1 GENERATE FROM A SAVED LIST
    LIST NAME ==>>

  2 GENERATE A NEW LIST FROM CRITERIA BELOW
    DATA SET NAME ==>> **

    SPECIFY SOURCE OF THE NEW LIST ==>> 1      (1 - VTOC, 2 - Catalog)

  1 GENERATE LIST FROM VTOC
    VOLUME SERIAL NUMBER ==>> 338002      (fully or partially specified)

  2 GENERATE LIST FROM CATALOG
    CATALOG NAME ==>>
    CATALOG PASSWORD ==>>                (if password protected)
    VOLUME SERIAL NUMBER ==>>            (fully or partially specified)
    ACQUIRE DATA FROM VOLUME ==>> Y    (Y or N)
    ACQUIRE DATA IF DFHSM MIGRATED ==>> N (Y or N)

USE ENTER TO PERFORM SELECTION; USE DOWN COMMAND TO VIEW NEXT SELECTION PANEL;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 6. Complete Page 1 of the Data Set Selection Entry Panel

Press Enter to generate the Data Set List. A list that conforms to your selection criteria appears, as shown in Figure 7 on page 24.

```

COMMAND ==> copy
DATA SET LIST
SCROLL ==> HALF
Entries 1-5 of 5
Data Columns 3-7 of 31
ENTER LINE OPERATORS BELOW:

LINE OPERATOR      DATA SET NAME      ALLOC SPACE  ALLOC USED  % NOT USED  NUM EXT  ALLOC UNIT
----- (1) ----- (2) ----- (3) ----- (4) ----- (5) ----- (6) ----- (7) -----
IBMUSER.CNTL.CNTL      2436  2436    0    1  TRK
IBMUSER.ISMF.COPY.D87035. 19    19     0    1  BLK
T111457
IBMUSER.ISMF.COPY.D87035. 19    19     0    1  BLK
T120522
IBMUSER.ISPPROF      141   141    0    1  TRK
IBMUSER.TONYS.STUFF  141   47     66   1  TRK
----- (1) ----- (2) ----- (3) ----- (4) ----- (5) ----- (6) ----- (7) -----
                                BOTTOM OF DATA

USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 7. The Resulting Data Set List

The data sets selected have your user ID as the high-level qualifier because you entered ** in the “Data Set Name” field of the Data Set Selection Entry Panel.

Copying the Data Sets to a 3380

Next, you can copy all of the listed data sets to a 3380. Enter COPY on the command line of the Data Set List Panel that you generated as shown in Figure 7. ISMF then displays the first page of the COPY Command Entry Panel, Figure 8. (Alternatively, to copy an individual data set, enter COPY in the “Line Operator” column next to that data set’s name.)

```

COMMAND ==>
COPY COMMAND ENTRY PANEL
Page 1 of 13

SPECIFY ONE OR MORE FOR THE 5 DATA SETS:

TARGET VOLUMES      ==> 338001 ==>      ==>      ==>
PERCENT UTILIZATION ==> 99      ==> 99      ==> 99      ==> 99
UNIT                ==> 3380      ==>      ==>      ==>
TARGET DATA SETS NEW FIRST LEVEL QUALIFIER ==>

SELECT WHERE TO CATALOG TARGET DATA SETS ==> 3
1 DO NOT CATALOG  2 SAME CATALOG  3 DEFAULT CATALOG  4 NEW CATALOG
SPECIFY NEW CATALOG NAME ==>

OPTIONALLY SPECIFY ONE OR MORE:
REPLACE DUPLICATES      ==> N      (Y or N)
REBLOCK TARGET DATA SETS ==> N      (Y or N)
RACF MODEL DATA SET NAME ==>
RACF MODEL VOLUME SERIAL ==>

VIEW OR CHANGE ADDITIONAL COPY OPTIONS ==> Y (Y or N)

USE ENTER TO CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 8. Completing Page 1 of the COPY Command Entry Panel

Complete the first page of the COPY Command Entry Panel as shown in Figure 8. Press Enter to display page 2 of the COPY Command Entry Panel, Figure 9.

```
COMMAND ==>>                                COPY COMMAND ENTRY PANEL                                Page 2 of 13

OPTIONALLY SPECIFY ONE OR MORE FOR THE 5 DATA SETS:

IF SOURCE DATA SETS ARE PASSWORD PROTECTED, SPECIFY:
  PASSWORD DSN ==>>

SELECT DISPOSITION OF SOURCE DATA SETS ==>> 1
  1 KEEP
  2 UNCATALOG
  3 UNCATALOG AND SCRATCH ALL
  4 UNCATALOG AND SCRATCH IF EXPIRED
  5 UNCATALOG ALL, SCRATCH IF EXPIRED

USE ENTER TO CONTINUE; USE DOWN TO VIEW MORE COPY COMMAND OPTIONS;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 9. Completing Page 2 of the COPY Command Entry Panel

Scroll forward if you want to view more entry panels, or press Enter if the field defaults are acceptable. If Enter is pressed, the COPY Job Submission Entry Panel, Figure 10, appears.

```
COMMAND ==>>                                COPY JOB SUBMISSION ENTRY PANEL

SELECT ONE OF THE FOLLOWING ==>> 1
  1 SUBMIT JOB FOR BACKGROUND PROCESSING
  2 SAVE GENERATED JOB IN A DATA SET

IF OPTION "2" IS SELECTED, SPECIFY:
  DATA SET NAME ==>>
  REPLACE CONTENTS ==>> N                    (Y or N)

JOB STATEMENT INFORMATION:                    (verify before proceeding)
==>> //IBMUSERF JOB (account),'name'
==>> /*
==>> /*
==>> /*
==>> /*
==>> /*
==>> /*

VIEW OR CHANGE EXECUTE STATEMENTS FROM PROFILE ==>> Y    (Y or N)

USE ENTER TO CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 10. Verifying the COPY Job Statement Information

Note: If, instead, you want to save the generated JCL and DFDSS command for later use, you can do so.

Verify the JOB statement on this panel. Then press Enter to view the DFDSS Execute Statement Entry Panel.

```

                                DFDSS EXECUTE STATEMENT ENTRY PANEL
COMMAND ==>

SPECIFY DFDSS EXECUTE STATEMENT INFORMATION:

==> /*
==> /*
==> /*
==> //STEP1 EXEC PGM=ADRSSU,REGION=nnnnK
==> //SYSPRINT DD SYSOUT=*
==> /*
==> /*

USE ENTER TO CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 11. The DFDSS Execute Statement Entry Panel

Press Enter to submit the job. ISMF then displays the Data Set List, Figure 12. The asterisks before each COPY command in the Line Operator column indicate that the job was successfully submitted.

```

                                DATA SET LIST
                                COPY JOB CREATED
COMMAND ==>                                SCROLL ==> HALF
                                           Entries 1-5 of 5
                                           Data Columns 3-7 of 31
ENTER LINE OPERATORS BELOW:

LINE OPERATOR          DATA SET NAME          ALLOC SPACE  ALLOC USED  % NOT USED  NUM EXT  ALLOC UNIT
---(1)---  -----(2)-----  --(3)--  --(4)--  -(5)-  (6)  -(7)-
*COPY      IBMUSER.CNTL.CNTL          2436    2436     0     1     TRK
*COPY      IBMUSER.ISMF.COPY.D87035.  19      19      0     1     BLK
           T111457
*COPY      IBMUSER.ISMF.COPY.D87035.  19      19      0     1     BLK
           T120522
*COPY      IBMUSER.ISPPROF          141     141     0     1     TRK
*COPY      IBMUSER.TONYS.STUFF       141     47     66     1     TRK
-----  -----  BOTTOM OF DATA  -----

USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 12. The List with Successful Submission of the COPY Command

ISMF Volume Application

Building a Volume List

Building a volume list is similar to building a data set list. You simply complete the Volume Selection Entry Panel as shown in Figure 13 by specifying your volume serial number. Verify that pages 2 and 3 of the Volume Selection Entry Panel are blank. If they are not blank, you receive the short informational message: OTHER VALUES PRESENT.

```

                                VOLUME SELECTION ENTRY PANEL
                                Page 1 of 3
COMMAND ==>>

SELECT SOURCE TO GENERATE VOLUME LIST ==>> 2 (1 - saved list, 2 - new list)

  1 GENERATE FROM A SAVED LIST
    LIST NAME ==>>

  2 GENERATE A NEW LIST FROM CRITERIA BELOW
    SPECIFY SOURCE OF THE NEW LIST ==>> 1 (1 - Physical, 2 - SMS)

    OPTIONALLY SPECIFY ONE OR MORE:
      TYPE OF VOLUME LIST ==>> 1 (1-Online, 2-Not Online, 3-Either)
      VOLUME SERIAL NUMBER ==>> LIB* (fully or partially specified)
      DEVICE TYPE ==>> (fully or partially specified)
      DEVICE NUMBER ==>> (fully specified)
      TO DEVICE NUMBER ==>> (for range of devices)
      ACQUIRE PHYSICAL DATA ==>> Y (Y or N)
      ACQUIRE SPACE DATA ==>> Y (Y or N)
      STORAGE GROUP NAME ==>> (fully or partially specified)
      CDS NAME ==>> (fully specified or 'ACTIVE')

USE ENTER TO PERFORM SELECTION; USE DOWN COMMAND TO VIEW NEXT SELECTION PANEL;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 13. Complete Page 1 of the Volume Selection Entry Panel

Press Enter to generate the volume list, as shown in Figure 14 on page 28.

```

VOLUME LIST
COMMAND ==>
SCROLL ==> HALF
Entries 1-1 of 1
Data Columns 3-8 of 39
ENTER LINE OPERATORS BELOW:

LINE   VOLUME  FREE  %   ALLOC  FRAG  LARGEST  FREE
OPERATOR SERIAL SPACE FREE SPACE INDEX EXTENT EXTENTS
---(1)---(2)---(3)---(4)---(5)---(6)---(7)---(8)---
dump   LIBTST  115845 19  506006  461  25296   58
       LIB001  100154 16  521697  300  40145   40
-----
                          BOTTOM OF DATA
-----

USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 14. The Volume List

Note: DFDSS list commands will not work for volume lists.

Dumping a Volume

Enter the DUMP line operator for the volume in the list, as shown in Figure 14, to dump the volume. ISMF displays the DUMP Entry Panel as shown in Figure 15. DFDSS commands can be issued against one volume at a time as a line operator.

```

DUMP ENTRY PANEL
COMMAND ==>
Page 1 OF 8
SPECIFY ONE OR MORE FOR VOLUME: LIBTST

OUTPUT DSN ==> tape.dump
IF OUTPUT DATA SET IS NEW OR NOT CATALOGED, SPECIFY:
VOLUME COUNT ==>
OUTPUT VOLUMES ==>
OUTPUT VOLUMES ==>
OUTPUT VOLUMES ==>
UNIT ==>

NUMBER OF COPIES ==> 1 (1 to 5 copies)
OUTPUT MEDIA TYPE ==> TAPE (TAPE or DASD)
TYPE OF DUMP ==> 2 (1 LOGICAL, 2 PHYSICAL)

IF SOURCE DATA SETS ARE PASSWORD PROTECTED, SPECIFY:
PASSWORD DSN ==>

VIEW OR CHANGE CURRENT ALLOCATION VALUES ==> Y (Y or N)
VIEW OR CHANGE ADDITIONAL DUMP OPTIONS ==> Y (Y or N)

USE ENTER TO CONTINUE; USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 15. Page 1 of the DUMP Entry Panel

Complete the DUMP Entry Panel as shown in Figure 15 and press Enter to display page 2.

```

                                DUMP ENTRY PANEL                                Page 2 OF 8
COMMAND ==>>>

SPECIFY ONE OR MORE TAPE ALLOCATION VALUES FOR VOLUME: LIBTST

LABEL TYPE          ==>> SL          (SL, SUL, NL, NSL, BLP, LTM)
DENSITY             ==>> 1600        (800, 1600, 6250)
FILE POSITION        ==>> 1          (1 to 9999)
STATUS              ==>> OLD         (NEW, OLD)
NORMAL DISPOSITION ==>> CATLG       (KEEP, DELETE, CATLG, UNCATLG)
ABNORMAL DISPOSITION ==>> DELETE    (KEEP, DELETE, CATLG, UNCATLG)
BLOCKSIZE           ==>> 32760      (7892 to 32760)
RETENTION PERIOD    ==>> 9999      (1 to 9999 days)

MODEL DSCB DSN ==>>

VIEW OR CHANGE ADDITIONAL DUMP OPTIONS ==>> Y (Y or N)

USE ENTER TO CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 16. Page 2 of the DUMP Entry Panel

Verify that the default values as shown in Figure 16 are acceptable and press Enter to display page 3.

Note: A density is not required for a 3480 device type. The default density field is set to 1600.

```

                                DUMP ENTRY PANEL                                Page 3 of 8
COMMAND ==>>>

SPECIFY ONE OR MORE DUMP CONTROL OPTIONS FOR VOLUME: LIBTST

MAXIMUM NUMBER OF RETRIES          ==>> 2          (0 to 99)
NUMBER OF SECONDS BETWEEN RETRIES ==>> 2          (0 to 255)
DUMP IN COMPRESSED FORM            ==>> N          (Y or N)
AMOUNT OF I/O BUFFERING            ==>> 1TRK      (1TRK, 2TRK, 5TRK, 1CYL)
RESET SOURCE CHANGED INDICATORS    ==>> Y          (Y or N)

FOR I/O ERROR HANDLING, SELECT ==>> 1
  1 TERMINATE IF ANY ERROR  2 TERMINATE AFTER 100 ERRORS  3 ALLOW ALL ERRORS

FOR DATA SETS THAT INDICATE USED SPACE, SELECT ==>> 1
  1 DUMP ONLY USED SPACE  2 DUMP ALL ALLOCATED SPACE  3 LIST DATA SETS

FOR DATA SETS THAT INDICATE NO USED SPACE, SELECT ==>> 2
  1 DUMP ALL ALLOCATED SPACE  2 DO NOT DUMP

USE ENTER TO CONTINUE; USE DOWN COMMAND TO VIEW MORE DUMP CONTROL OPTIONS;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 17. Completing the DUMP Entry Panel

Verify that the default values as shown in Figure 17 are also acceptable. Scroll forward if you want to view more entry panels, or press Enter if the field defaults are acceptable. If Enter is pressed, the DUMP Job Submission Entry Panel as shown in Figure 18 on page 30 is displayed.

```

                                DUMP JOB SUBMISSION ENTRY PANEL
COMMAND ==>

SELECT ONE OF THE FOLLOWING ==> 2
  1 SUBMIT JOB FOR BACKGROUND PROCESSING
  2 SAVE GENERATED JOB IN A DATA SET

IF OPTION "2" IS SELECTED, SPECIFY:
DATA SET NAME   ==> tape.dump
REPLACE CONTENTS ==> y           (Y or N)

JOB STATEMENT INFORMATION:           (verify before proceeding)
==> //IBMUSERS JOB (account),'name'
==> /**
==> /**
==> /**
==> /**
==> /**
==> /**

VIEW OR CHANGE EXECUTE STATEMENTS FROM PROFILE ==> Y   (Y or N)

USE ENTER TO CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 18. DUMP Job Submission Entry Panel

You can change the default selection to save the DFDSS job stream, as shown in Figure 18. Press Enter to display the DFDSS EXECUTE Statement Entry Panel, Figure 19.

```

                                DFDSS EXECUTE STATEMENT ENTRY PANEL
COMMAND ==>

SPECIFY DFDSS EXECUTE STATEMENT INFORMATION:

==> /**
==> /**
==> /**
==> //STEP1 EXEC PGM=ADRDUSSU,REGION=nnnnK
==> //SYSPRINT DD SYSOUT=*

==> /**
==> /**

USE ENTER TO CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 19. DFDSS EXECUTE Statement Entry Panel

Press Enter to save the job and redisplay the Volume List. The asterisk next to DUMP in Figure 20 on page 31 indicates that the job stream was successfully built.

```

VOLUME LIST
COMMAND ==>
ENTER LINE OPERATORS BELOW:
SCROLL ==> HALF
Entries 1-1 of 1
Data Columns 3-8 of 39

LINE   VOLUME  FREE   %   ALLOC  FRAG  LARGEST  FREE
OPERATOR SERIAL SPACE %FREE SPACE INDEX EXTENT EXTENTS
---(1)---(2)---(3)---(4)---(5)---(6)---(7)---(8)---
*DUMP  LIBTST  115845  19  506006  461  25296  58
      LIB001  100154  16  521697  300  40145  40
-----
                        BOTTOM OF DATA
-----

USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 20. The DUMP Job Stream is Successfully Built

For more information on using ISMF, see either *ISMF User's Guide* or the ISMF online help panels.

Invoking DFDSS with JCL

DFDSS is controlled by JCL statements and DFDSS commands. The JCL statements are used to invoke DFDSS and define the data sets used and created by it, whereas the DFDSS commands specify and control tasks. This section describes the JCL method of invoking and controlling DFDSS. This knowledge is needed even if you invoke DFDSS through an application program (as described in "Invoking DFDSS with the Application Interface" on page 36).

The JCL You Need

The following JCL statements can be used to invoke DFDSS:

Statement	Usage
JOB (required)	Initiates your job.
EXEC (required)	Specifies the program name (PGM = ADRDSSU) or, if the job control statements reside in a procedure library, the procedure name. You can enter additional information in the PARM parameter of the EXEC statement; see "Controlling DFDSS with the EXEC Statement" on page 33 following this list.
SYSPRINT DD (required)	Defines a sequential message data set. The data set can be written to a system output device, a tape volume, or a direct-access device. If the DCB keyword LRECL is specified on the DD statement, it must be in the ranges of 84 to 137 inclusive. If BLKSIZE is specified, it must be at least four greater than the LRECL. If an LRECL less than 84 is chosen, the return code is 8 and an error

	message is issued. If the specified LRECL is greater than 137, then LRECL and BLKSIZE are set to 137 and 141, respectively.
SYSDS DD (required)	Defines a command data set, consisting of cards or card-image records, that contains your DFDSS commands. It usually resides in the input stream; however, it can be defined as a blocked or unblocked sequential data set or as a member of a partitioned data set. Records must be fixed format, LRECL = 80.
input DD (optional)	Defines the input (also called source). The ddname, <i>input</i> , is supplied by you and is referred to by your DFDSS commands. This DD statement is not required for some operations. Do not specify BUFNO.
output DD (optional)	Defines the output (also called target). The ddname, <i>output</i> , is supplied by you and is referred to by your DFDSS commands. This DD statement is not required for some DFDSS operations. Do not specify BUFNO. Code a volume count when a new data set will reside on six or more volumes. (Note that DISP=MOD is not supported for an output DD statement.)
filter DD (optional)	Defines a data set, consisting of cards or card-image records, that contains the filtering criteria (INCLUDE, EXCLUDE, and BY) to be used in a data set command. The ddname, <i>filter</i> , is supplied by you and is referred to by your DFDSS commands. Note: You cannot concatenate DD statements to the <i>filter</i> DD statement.
password DD (optional)	Defines a data set, consisting of card-image records, that contains data set names and their passwords. The ddname, <i>password</i> , is supplied by you and is referred to by your DFDSS commands.

For more information on coding JCL, see *JCL Reference*.

The following example shows an input DD statement that specifies a DASD volume:

```
//DASD DD UNIT=3380,VOLUME=(PRIVATE,SER=111111),DISP=OLD
```

The following example shows an output DD statement that specifies a tape volume:

```
//TAPE DD UNIT=3480,VOLUME=SER=TAPE01,LABEL=(1,SL),
// DISP=(NEW,CATLG),DSNAME=USER2.DUMP
```

A standard label tape is recommended to help prevent an operator error. If you use a label, you must include a corresponding DSNAME parameter.

The command data set, which is pointed to by the **SYSIN DD** statement, contains commands that specify operations to be performed (function commands) and commands that control the execution of these operations (auxiliary commands).

Controlling DFDSS with the EXEC Statement

The **EXEC** statement for DFDSS can contain **PARM** information that is used by the program. You can use the following keyword parameters:

TYPRUN = SCAN

Only control card syntax checking is to be done. No tasks are to be executed.

TYPRUN = NORUN

For **COPY**, **DUMP**, **RESTORE**, **COMPRESS**, **CONVERTV**, and **RELEASE** operations, only data set filtering is to be done, without actually processing data sets. Printed output for the run indicates the data sets selected. For a **DEFRAG** operation, the initial volume statistics are to be printed, without actually relocating any extents. For a **CONVERTV** operation, a full report is produced, but no volumes or data sets are actually converted.

PAGENO = nnnn

The **nnnn** is a 1- to 4-digit number indicating the starting page number to be used for the **SYSPRINT** data set. The default is 1.

LINECNT = nnnn

The **nnnn** is a 1- to 4-digit number indicating the number of lines to be printed per page for the **SYSPRINT** data set. The default is 60. Specify **LINECNT = 9999** to prevent a page ejection.

SIZE = nnnnK

The **nnnn** is a 1- to 4-digit number indicating the number of kilobytes (1 kilobyte equals 1024 bytes) of main storage to be used by DFDSS. This amount must be less than or equal to that specified by the **REGION** keyword. The default value is the region size.

ABEND = nnn

The **nnn** is a 3-digit decimal message number (**ADRnnnx**). If this is specified, and this message is to be issued, DFDSS executes a user 0001 **ABEND** dump after issuing the message, and the task is ended. To get a dump, include a **DD** statement for **SYSABEND** or **SYSUDUMP**. This keyword is provided for debugging purposes only.

AMSGCNT = nnnn

The abend message occurrence count that tells DFDSS to abend on the *n*th occurrence of the message specified in **ABEND = nnn**. The **nnnn** for **AMSGCNT** can be a number between 1 and 9999. The default is 1 (1st occurrence).

RACFLOG = YES

RACF (Resource Access Control Facility) logging is done for all function tasks. This parameter can be overridden or set by the user installation options exit. See *Data Facility Data Set Services: Reference* for a description of DFDSS **RACF** logging.

TRACE = YES

When used during a **DEFRAG** operation, this results in messages being printed that indicate the relocated extents. This is a debugging tool.

UTILMSG = YES

Messages from auxiliary programs, such as **IEBCOPY** and **ICKDSF**, are copied into the DFDSS **SYSPRINT** output.

XABUFF= ABOVE16|BELOW16

The I/O buffers used by DFDSS for COPY, COPYDUMP, DEFRAG, DUMP, PRINT, and RESTORE operations are to be above or below 16 megabytes virtual storage. The default, when running under MVS/XA, is "ABOVE16". This parameter is ignored when running under MVS/370.

WORKUNIT = workunit

You can supply an esoteric DASD unit name (for example, SYSDA), generic DASD unit name (for example, 3380), or a specific DASD address. DFDSS passes this unit to dynamic allocation when temporary data sets are allocated.

WORKVOL = volser

You can supply a volser on which DFDSS should allocate temporary data sets. DFDSS passes the volser to dynamic allocation.

Notes:

1. WORKUNIT, WORKVOL, or both parameters can be specified when you invoke DFDSS.
2. When DFDSS invokes the IDCAMS utility to copy an ICF user catalog, the export data set is allocated as a permanent data set. The permanent data set cannot be placed on the volume specified by WORKUNIT and/or WORKVOL.
3. An esoteric unit name that requests virtual I/O can be used in the WORKUNIT parameter, but when DFDSS invokes the IEHMOVE utility during data set copy, the default dynamic allocation unit name is used (SYSALLDA).

Examples of PARM information are:

```
// EXEC PGM=ADRDUSSU,  
//   PARM='PAGENO=8,LINECNT=57,SIZE=500K,UTILMSG=YES'  
// EXEC PGM=ADRDUSSU,PARM='TYPRUN=SCAN,XABUFF=ABOVE16'
```

For further information on the PARM parameter of an EXEC statement, see *Application Development Guide*.

Examples of Invoking DFDSS with JCL

Following are some examples of JCL job streams for invoking DFDSS.

Note: Devices Available: Throughout the examples in this manual, devices are presented as UNIT = 3350 or UNIT = 3380, and a tape device as UNIT = 3480. This is for illustration only; you can specify any supported DASD device and any supported tape device. Refer to *System Generation* for the device-type notation to be used in the UNIT parameter of a DD statement.

Moving a Data Set: The following example shows how to move a data set from one DASD volume to another using JCL and a DFDSS command. The source data set is deleted and the target data set is cataloged to reflect its new location.⁴

⁴ This JCL must be removed when working in an SMS-managed environment.

```

//MYJOB   JOB   accounting information,REGION=nnnnK
//STEP1   EXEC  PGM=ADRDSSU
//SYSPRINT DD   SYSOUT=A
//DASD1   DD    UNIT=3350,VOL=(PRIVATE,SER=111111),DISP=OLD
//DASD2   DD    UNIT=3380,VOL=(PRIVATE,SER=222222),DISP=OLD
//SYSIN   DD    *
COPY DATASET(INCLUDE(MYDATSET)) -
  LOGINDDNAME(DASD1) OUTDDNAME(DASD2) DELETE CATALOG
/*

```

Dumping a Data Set: This example shows how to back up a DASD data set⁵ to tape:

```

//MYJOB   JOB   accounting information,REGION=nnnnK
//STEP1   EXEC  PGM=ADRDSSU
//SYSPRINT DD   SYSOUT=A
//DASD    DD    UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//TAPE    DD    UNIT=3480,VOL=SER=TAPE01,
// LABEL=(1,SL),DISP=(NEW,CATLG),DSNAME=MYDATSET.BACKUP
//SYSIN   DD    *
DUMP DATASET(INCLUDE(MYDATSET)) -
  INDDNAME(DASD) OUTDDNAME(TAPE)
/*

```

Restoring a Data Set: In this example, the data set⁵ (MYDATSET) that was backed up on tape in the "Dumping a Data Set" example is restored to the original DASD volume on which it resided.

```

//MYJOB   JOB   accounting information,REGION=nnnnK
//STEP1   EXEC  PGM=ADRDSSU
//SYSPRINT DD   SYSOUT=A
//TAPE    DD    UNIT=3480,VOL=SER=TAPE01,
// LABEL=(1,SL),DISP=(OLD,KEEP),DSNAME=MYDATSET.BACKUP
//DASD    DD    UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//SYSIN   DD    *
RESTORE DATASET(INCLUDE(MYDATSET)) -
  INDDNAME(TAPE) OUTDDNAME(DASD) REPLACE
/*

```

⁵ Do not use a data set name that DFDSS will reference during execution. Otherwise, enqueue contention with the operating system initiator will occur.

Invoking DFDSS with the Application Interface

You can invoke DFDSS from an application program by using the application interface. This allows you, for example, to gather statistical or auditing information, and to specify control variables.

The application interface allows you to:

- Fully utilize the invocation capabilities of DFDSS when the ATTACH, LINK, or CALL system macro is specified in your application program.
- Optionally, specify a list of parameters to be used by DFDSS during the processing caused by that invocation.
- Optionally, interact with DFDSS during processing of user installation options after the installation options exit has been called.
- Optionally, interact with DFDSS during the processing at convenient points where input/output (I/O) operations are being performed.

Note: DFDSS runs as an authorized problem program (nonsupervisor state); any program invoking DFDSS must also be authorized and in nonsupervisor state.

For more information on the application interface, see *Data Facility Data Set Services: Reference*.

User Interaction Module Exit Functions

When DFDSS is invoked from an application program, you can use the user interaction module (UIM) to interact with DFDSS at points where I/O operations are being performed. UIM exit functions can be used to:

- Replace, insert, delete, or modify a SYSIN record after DFDSS reads it or a SYSPRINT record when DFDSS is ready to print it.
- Replace, insert, delete, or modify a write-to-operator message before DFDSS writes it.
- Insert a statistics record during a logical DUMP.
- Modify the installation options specified in the ADRUFO control block to override the specified options.
- Bypass password and expiration date checking, or reject the tape volume and request a scratch tape, when DFDSS is ready to open a tape.
- Request a specific volume serial when a nonspecific tape is passed to DFDSS.
- Get information about the data set being allocated.
- End a task or processing of individual data sets.
- Bypass authority checking for individual data sets. This includes both RACF and password authorization.
- Bypass serialization checking of individual data sets.

For more information and examples of UIM exit functions, see *Data Facility Data Set Services: Reference*.

Chapter 6. Protecting DFDSS Functions

This chapter discusses the functions of DFDSS to which you can control access by using RACF. You can protect DFDSS/ISMF functions and some DFDSS keywords.

Protecting DFDSS/ISMF Functions with RACF

Using the program control feature of RACF Version 1 Release 7 or later releases, you can set authorization levels for ISMF functions: ISMF itself, each of the ISMF applications, and the individual line operators and commands. In addition, each ISMF function you identify to RACF for authorization checking is included in the RACF report and logging process. You can also use standard RACF authorization checking to limit access to individual data sets, volumes, or catalogs. Used in conjunction with program control, authorization checking ensures that the appropriate ISMF data and functions are available to users when they need them.

ISMF Functions You Might Want to Protect

Program control allows you to determine the ISMF functions to which users have access. The authorization scheme you set up can apply to both individual users and user groups. The DFDSS functions you can protect fall into two general categories: line operators and commands.

With program control, you can set up authorization levels for each category. You can also vary the level within a category to suit the needs of your site. Before you set up an authorization structure, you might want to consider the following:

Do you want all users at your site to have access to ISMF?

Do you want all users to have access to the data set, volume, or profile applications?

Are there line operators or commands to which you want to limit access?

Setting Up the Authorization Structure

RACF program control checks authorization before allowing access to an ISMF function. Protection for each function is based on the authorization level of the load module that contains the function. A user is allowed to execute an ISMF function (the RESTORE list command, for example) when one of the following is true:

The user is authorized to execute the load module corresponding to the function requested. Authorization is defined as READ level access or greater.

The user's RACF profile has the OPERATIONS attribute.

The user's group is authorized to execute the load module.

The universal access authority (UACC) for the load module is READ or greater. This makes the load module available to anyone who can access ISMF.

Finding the DFDSS/ISMF Module Names

The names of the load modules for DFDSS/ISMF are stored in command tables in both the panel library, DGTPLIB, and the load library, DGTL LIB. The load module names are listed in Figure 21 and Figure 22. The module names are found in the DGTSMMMD1 member of the panel library.

Figure 21 lists the names for the corresponding line operators. The module names for line operators are found in the DGTTLPD3 member of the load library. Figure 22 lists the names for commands. These names are in the DGTTCTD2 member of the load library.

Line Operator	Data Set Application Module Name	Volume Application Module Name
COMPRESS	DGTFM01	DGTFCS01
CONVERTV	—	DGTFM01
COPY	DGTFM01	DGTFM01
DEFMAG	—	DGTFM01
DUMP	DGTFM01	DGTFM01
RELEASE	DGTFM01	DGTFM01
RESTORE	DGTFM01	DGTFM01

Figure 21. Module Names for DFDSS/ISMF Line Operators

Command	Module Name
COMPRESS	DGTFM01
COPY	DGTFM01
DUMP	DGTFM01
RELEASE	DGTFM01
RESTORE	DGTFM01

Figure 22. Module Names for DFDSS/ISMF Data Set Application Commands

To view the command table, you need to know the data set names your site uses for the panel library and the load library. The installation of DFDSS/ISMF puts the panel library in SYS1.DGTPLIB and the load library in SYS1.DGTL LIB. However, your site's post-installation procedures might involve moving the DFDSS/ISMF libraries. If they were moved, you can determine the data set name by issuing the TSO LISTALC command and scanning the low-level qualifiers for DGTPLIB and DGTL LIB.

Protecting DFDSS/ISMF Modules

The two steps to protect DFDSS/ISMF modules are:

1. To define the modules you want to protect, use the **REDEFINE** command or the ISPF RACF entry panels. When you define the modules to RACF, supply the name of the load module you want to protect, the name of the data set that contains the module, and the volume serial number of the volume containing the data set. Each module you identify is added to the profile for the **PROGRAM** general resource class. You have several options when you define modules:
 - If you want to define several modules at the same time, you can use asterisk notation. For example, **DGT*** means all the modules beginning with the letters **DGT**.
 - You can add an access list with user IDs and/or group names with their associated access authority to the profile.
 - You can define the **UACC** to give default access to all users or to none.
 - You can use the **AUDIT** parameter to set up RACF logging or to bypass it.
2. To allow users to execute an application, line operator, or command, use the **PERMIT** command.

For more information on how to perform these steps and the options you have using program control, see *RACF Security Administrator's Guide*.

DFDSS Keywords You Can Protect with RACF

In addition to protecting DFDSS/ISMF functions, you can also protect certain DFDSS keywords. If a profile is not set up for these keywords, then anyone can use the keyword. Figure 23 lists these keywords and their associated RACF facility class profiles:

Keyword	Profile Name
BYPASSACS with COPY	STGADMIN.ADR.COPY.BYPASSACS
BYPASSACS with RESTORE	STGADMIN.ADR.RESTORE.BYPASSACS
CONVERTV	STGADMIN.ADR.CONVERTV
INCAT(catname) with DUMP	STGADMIN.ADR.DUMP.INCAT
INCAT(catname) with COPY	STGADMIN.ADR.COPY.INCAT
PROCESS(SYS1) with COPY	STGADMIN.ADR.COPY.PROCESS.SYS
PROCESS(SYS1) with DUMP	STGADMIN.ADR.DUMP.PROCESS.SYS
PROCESS(SYS1) with RELEASE	STGADMIN.ADR.RELEASE.PROCESS.SYS

Figure 23. RACF Facility Class Profile Names for DFDSS Keywords

Refer to *RACF Security Administrator's Guide* for additional information on RACF facility class profiles.

Chapter 7. Managing Availability with DFDSS

One of the major functions of DFDSS is the backup and recovery of data. Using the DUMP and RESTORE commands, you can backup and recover data sets and volumes.⁶

This chapter is organized as follows:

- “**Planning an Availability Strategy**” discusses the various considerations for an overall availability strategy. This section also describes disaster recovery requirements, vital records storage, and archived data sets.
- “**Backing Up Data Sets**” on page 45 describes how to use DFDSS to back up data sets to meet your general backup requirements.
- “**Backing Up SMS-Managed Data Sets**” on page 51 describes how to use DFDSS to back up data sets in a SMS environment.
- “**Backing Up Volumes**” on page 51 describes how to use DFDSS to back up volumes to meet your general backup requirements.
- “**Restoring Data Sets**” on page 53 describes how to use DFDSS to recover data sets and volumes to meet your general recovery requirements.

Planning an Availability Strategy

In planning your overall availability strategy, you should consider the following types of backup:

Backup of volumes and data sets—The general type of backup to guard against users accidentally losing or incorrectly changing their data sets and against losing volumes due to hardware failures.

Disaster recovery backup—Backup to protect against the loss of all your data in a major disaster at your site. These backups are stored off site and, in the event of a major disaster, are recovered at another site.

Vital records backup—Backup copies of data sets kept to meet externally imposed retention requirements, such as tax records.

Archival—Backup of data that is unused for a long period of time. You remove the data from DASD and retain it on tape in case it is needed again.

DFDSS is a very flexible backup and recovery tool. You can use DFDSS by itself to perform all backups listed above or to complement other backup and recovery tools.

Note: If you have a 3480 tape drive with the Improved Data Recording Capability (IDRC) and you want to use hardware data compaction, you do not need to use the COMPRESS keyword with the DUMP command. If you want software compression, specify the COMPRESS keyword, but do not specify DCB=TRTCH=COMP in the JCL.

⁶ You can also use DUMP and RESTORE on ranges of tracks. However, this is usually done as a means of diagnosing I/O errors rather than as a means of backing up and recovering data.

Backup and Recovery

General backup should be done at both the data set and volume level. To protect against users accidentally deleting or changing their data sets, it is usually more efficient to do logical data set backup of those data sets that changed since they were last backed up (incremental backup). Logical backup allows you to restore data sets to unlike devices. Incremental backups minimize processing time because you are not backing up every data set.

For data set backup, you need to consider the frequency of backup and the number of versions you want to keep. A number of factors can influence the frequency of backup and the number of versions to be kept, such as:

- The rate at which the data changes.
- The ease or difficulty of rebuilding the data (for example, it is easier to rebuild an object library than a source library).
- The importance of the data. For data that is extremely important to your business, you might want to keep extra backup versions.

These are only a few of the things you must consider when determining frequency of backup and number of versions. A more complete discussion of these considerations can be found in *SML: Managing Data Sets*.

Volume backup is necessary to guard against losing a volume, but it need not be done as often if you are doing incremental backup on a regular basis. If you lose a volume, you can recover from the latest volume backup, and then recover data sets from incremental backups to return the volume to its status before the failure. This form of recovery is sometimes referred to as forward recovery. In order to perform it, though, you must have a record of all your backups. Data Facility Hierarchical Storage Management (DFHSM) keeps its own inventory of the data sets it backs up and can perform forward recovery using that inventory. DFDSS prints the names of the data sets it dumps and the serial and data set sequence numbers of the tape volumes on which the dump begins and ends. You must use this printed record to perform forward recovery with DFDSS.

Backup and Recovery in an SMS Environment

Two kinds of data exist in an SMS environment: SMS-managed and non-SMS-managed data. DFDSS can help you fulfill your availability requirements for both kinds of data.

SMS-Managed Data: If you have Data Facility Hierarchical Storage Management (DFHSM) installed, it can perform automatic volume backup (by invoking DFDSS) and incremental backup on SMS-managed data. Each data set is assigned a management class that indicates how often DFHSM should back it up and how many versions of the backup to keep. Using DFHSM this way allows you to manage availability at the data set level.

If you do not have DFHSM installed, you can use DFDSS to back up and recover data sets and volumes. By filtering on management class name and the data-set-changed flag, you can perform incremental backup on all the data sets belonging to a particular management class. To automate this backup procedure, you can set up a DFDSS job to run periodically.

For more details on planning for backup of SMS-managed data, see *SML: Managing Data Sets*.

Non-SMS-Managed Data: Typically, non-SMS-managed data is data that SMS does not support or data that is in transition from non-SMS to SMS management. If it is data that SMS does not support, you can probably still use DFDSS to back it up and recover it, because DFDSS supports many kinds of data that SMS does not. If it is data in transition to SMS management, you can use DFHSM or DFDSS to back up and recover it until it is placed under SMS management.

Backup and Recovery in a Non-SMS Environment

If you do not have MVS/DFP Version 3 Release 1 or if you do have it but SMS is not active, then you are in a non-SMS environment. For availability management, the data in this environment can be treated much the same as the non-SMS-managed data in a SMS environment. DFDSS can be used to back up and recover it at the data set and volume level.

Disaster Recovery

Disaster recovery backups are made specifically for recovering data and applications following a disaster. Never rely on your regular backup data sets (for instance, DFHSM or DFDSS incremental backups) for disaster recovery. Disaster recovery backups require some very special considerations that normally do not apply to other types of backups.

Store at a Remote Site: A basic difference between regular backups and disaster recovery backups is that disaster recovery backups must be transported to a different site. How remote the recovery site is depends upon the type of disaster you are preparing for (in the case of a fire, the recovery site can be around the corner; in the case of an earthquake or flood, it should be many miles away). The fact that the backups must be taken to another site means that they must be on a portable media: tape.

Note: It is also possible to transmit backups to another site.

Use Logical Data Set DUMP: Because the environment at the remote site might differ from your environment, you should ensure that your disaster recovery backups can be restored in a different environment. In general, it is recommended that you use logical data set DUMP and filter on data set name to make disaster recovery backups. Logical data set DUMP allows you to back up only your critical data sets and restore to unlike devices.

Making logical data set dumps for disaster recovery backup requires a naming convention or some other method to identify your critical data sets. If, for example, you establish the convention of having the letters CRIT as the first four characters in the first qualifier of critical data sets, you can back them up for disaster recovery as follows:

```
DUMP -  
  DATASET(INCLUDE(CRIT*.**) -  
          BY(MGMTCLAS,EQ,MCNAME)) -  
  OUTDDNAME(TAPE) -  
  COMPRESS
```

If for some reason you must do volume dumps for disaster recovery, you should at least do logical volume dumps instead of physical volume dumps. This allows you to restore the backups to unlike devices. You can perform logical volume dumps by

using the **LOGINDDNAME** or **LOGINDYNAM** keyword with the **DUMP** command.

Only Back Up Critical Data Sets: You should back up only data sets that are critical to your operation. For example:

- Critical application data sets
- DFHSM control data sets
- RACF inventory data sets
- System data sets
- Catalogs.

Because you normally back up only critical data sets for disaster recovery, the amount of data you have to back up is only a small percentage of all your data. To identify those data sets you want backed up for disaster recovery, you should create a unique naming convention.

To maintain versions of your disaster recovery backups, you can use generation data group (GDG) dump data sets.

For more information on disaster recovery, see *SML: Managing Data Sets*.

Vital Records

Vital records are maintained to meet external retention requirements (such as legal requirements).

Like disaster recovery backups, vital records are kept at a remote site and therefore should reside on tape. Vital records are usually an even smaller percentage of all data than disaster recovery backups. Unlike disaster recovery backups, vital records are rarely necessary for normal processing.

Vital records are usually kept for long periods of time. As a result, the device they originally resided on may no longer be in use at the time of recovery. In that case, you will need to restore them to unlike devices. For this reason, vital records should be dumped logically so they can be restored to unlike devices. As with disaster recovery, using logical data set **DUMP** requires a naming convention or some other method to identify data sets for dumping.

If, for example, you establish the convention of having the letters **VR** as the first two characters in the first qualifier of data sets to be backed up for vital records purposes, you can dump them as follows:

```
DUMP -  
  DATASET(INCLUDE(VR**.**) -  
          BY(MGMTCLAS, EQ, MCNAME)) -  
  OUTDDNAME(TAPE) -  
  COMPRESS
```

For more details on vital records, see *SML: Managing Data Sets*.

Archiving

Archived data sets are data sets created to remove data from active status. This data is placed on alternate storage media because it is not currently being used but may be used in the future. Archived data sets are usually used for long-term retention.

You can use DFDSS to archive data sets by periodically filtering on last referenced date and then dumping and deleting data sets that have gone unreferenced for long periods of time. This frees up space for data that is being accessed more frequently and requires the faster access time of DASD. Because archived data sets might not be recovered for a long period of time, they should be dumped logically so they can be restored to unlike devices.

For example, the following logical DUMP command results in the archiving of all data sets in management class MCNAME1 that have not been referred to since April 10, 1986:

```
DUMP -
DATASET(BY((REFDT LT 86100)(MGMTCLAS EQ MCNAME1))) -
OUTDDNAME(TAPE1) -
DELETE -
COMPRESS -
PURGE
```

For more details on archiving, see *SML: Managing Data Sets*.

Backing Up Data Sets

With the DUMP command, you can dump DASD data to a sequential data set. The storage medium for the sequential data set can be tape, DASD, or mass storage virtual volume.

DFDSS can dump data sets both logically and physically. Data sets are located by searching either the catalog or the VTOC.

You can select data sets for DUMP and RESTORE processing based on data set names and numerous data attributes, as discussed in “Choosing Data Sets for Processing—Filtering” on page 13. If you want to perform incremental backups with DFDSS, you can filter with BY(DSCHA.EQ.1) to dump only data sets that have changed since the last dump was taken. If you also code the RESET keyword, DFDSS changes the data-set-changed (DSCHA) bit after successfully dumping the data set.

Notes:

1. If you are using DFDSS on data sets that DFHSM is also backing up, you should not use the RESET keyword because it might cause confusion as to which backup is the most current.
2. DFDSS does not preserve candidate volumes during DUMP processing.

The data-set-changed bit and the last-referenced date (REFDT) are supported for:

- Non-VSAM data sets.
- VSAM data sets in the DFP environment if you:
 - Have DFHSM, or
 - Replace the installation exit IDATMSTP with one that sets RC = 4.

Temporary data sets that are created and deleted by other jobs running at the same time DFDSS is running might be included in the data set list at the beginning of a DFDSS job, and might disappear at a later time before DFDSS finishes. DFDSS can issue a message informing the user what happened only at the time DFDSS tries to access the data set. If the user wants to hold all the data sets in a volume for the entire DFDSS execution then the user must write an enqueue installation exit to enqueue the volume for the entire job.

When you create backups of data sets with DUMP, you can make multiple (up to 255) dump copies with a single DUMP command. This is done by specifying multiple ddnames on the OUTDDNAME parameter. To specify multiple ddnames on the OUTDDNAME parameter, you might code:

```
DUMP -  
  DATASET(INCLUDE(**) -  
          BY(MGMTCLAS, EQ, MCNAME1)) -  
  OUTDDNAME(TAPE1, TAPE2, TAPE3) -  
  COMPRESS
```

This technique can be helpful if you want to create several backup copies to be used for different purposes.

Unless overridden by the installation options exit routine, DFDSS continues dumping while at least one output copy does not have an output error.

Logical Data Set DUMP

If you specify DATASET with the DUMP command and do not specify input volumes, DFDSS performs a logical data set DUMP using information in the catalog(s) to select data sets. For instance, the following DUMP command results in a logical data set dump:

```
DUMP -  
  DATASET(INCLUDE(**) -  
          BY(DSCHA, EQ, YES)) -  
  OUTDDNAME(TAPE1) -  
  COMPRESS
```

If you specify DATASET and either LOGINDDNAME or LOGINDDYNAM, DFDSS performs a logical data set DUMP using information in the VTOC(s) to

select data sets. For example, the following DUMP command results in a logical data set dump of all the data sets on volume 338001:

```
DUMP -  
  DATASET(INCLUDE(**)) -  
  LOGINDYNAM(338001) -  
  OUTDDNAME(TAPE) -  
  COMPRESS
```

DFDSS logical DUMP processing cannot be used to process PAM data sets containing location-dependent information that does not reside in note lists or the directory. As an example, PDS load libraries that contain overlay modules are not supported during logical DUMP processing.

As already discussed, logical processing offers the advantage of device independence. A logical data set backup can be restored to both like and unlike devices.

Physical Data Set DUMP

If you specify DATASET and INDDNAME or INDYNAM, DFDSS performs a physical data set dump. For instance, the following DUMP command results in a physical data set dump:

```
DUMP -  
  INDDNAME(DASD1) OUTDDNAME(TAPE) -  
  DATASET(INCLUDE(**)) -  
  COMPRESS -  
  OPTIMIZE(4)
```

When multiple input volumes are specified for a physical data set DUMP, multiple logical files (logical volumes) are created for each physical DASD source volume.

DFDSS facilitates backup and recovery procedures for physical data set dumps by printing the names of data sets dumped, and the serial and data set sequence numbers of the backup tape volumes on which the dump of a DASD volume begins and ends.

Note: DFDSS does not support any physical dump processing of a KSDS with key ranges. Logical processing should be used for this type of data set.

A Backup Scenario

As discussed under “Backup and Recovery” on page 42, you should consider using a combination of incremental and volume backup to fulfill your general availability requirements. One possible way of implementing this strategy is:

1. Dump a full volume at a given interval—perhaps once a week. You should use the RESET keyword to reset the data-set-changed flag. To do full volume

dumps of two volumes at once (in parallel, which is most effective if tapes are on separate channels), code the following:

```
PARALLEL
DUMP INDYNAM(111111) OUTDD(TAPE1) RESET OPTIMIZE(1)
DUMP INDYNAM(222222) OUTDD(TAPE2) RESET OPTIMIZE(2)
```

2. Dump only changed data sets at a shorter interval—perhaps every day.

```
DUMP LOGINDY((111111),(222222)) OUTDD(TAPE3) RESET -
OPTIMIZE(3) DATASET(INCLUDE(**) -
BY((DSCHA,EQ,YES)))
```

3. Use data set naming conventions to set up a dumping scheme that takes account of the relative importance of the data. For example, include **CRIT** in the first-level qualifier of all your critical data sets. With this convention in place, you can back up your critical data sets as follows:

```
DUMP LOGINDY((111111),(222222)) OUTDD(TAPE4) RESET -
OPTIMIZE(4) DATASET(INCLUDE(CRIT*.**) -
BY(DSCHA,EQ,YES))
```

Other naming conventions can also be used to identify groups of data sets. For instance, you can use department numbers, charge numbers, user initials, or project codes to identify data sets you want to dump together. For more information on naming conventions, see *SML: Managing Data Sets*.

SYSPRINT contains the names of all the data sets that were dumped for each run. You will have to keep them for reference if you have to restore a data set and you want it to be at the latest level. If you misplace the previous dump SYSPRINT, you can discover what data sets are on a particular dump tape by executing a RESTORE operation with the NORUN parameter on the EXEC statement and DATASET(INCLUDE(**)).

This prints a listing of all the data set names on the restore tape, and you can now find the latest dumped version of a particular data set.

Backing Up Data Sets with Special Requirements

Some data sets require special processing when they are backed up. The sections below discuss backing up multivolume data sets and integrated catalog facility user catalogs.

Dumping Multivolume Data Sets

An important advantage of DFDSS as a backup tool is that it can back up multivolume data sets without having to specify any or all of the input volumes. The catalogs are scanned to select an entire data set; that is, the data set is processed in its entirety from all the volumes it resides on. Logical processing consolidates the extents of the data set in one dump data set for you.

If you specify the **ALLMULTI** keyword and input volumes, DFDSS dumps the entire multivolume data set even if you do not specify all the volumes on which the data set resides. To use **ALLMULTI**, the data set must be cataloged and at least one primary volume must be specified on the **LOGINDDNAME** or **LOGINDYNAM** volume list.

Following is an example of the **DUMP** command with **ALLMULTI** specified:

```
DUMP -  
  DATASET(INCLUDE(**)) -  
  ALLMULTI -  
  LOGINDYNAM(338001)  
  OUTDDNAME(TAPE) -  
  COMPRESS
```

Note that **ALLMULTI** works only for logical data set dumps. If you dump a multivolume data set physically, you must make sure the segments from all the volumes are dumped together. If you dump a multivolume data set physically, it is dumped from all the volumes that are passed. The output dumped data contains a logical file for each selected volume.

A multivolume data set standard user label is not supported.

Dumping Integrated Catalog Facility User Catalogs

Another important use of DFDSS as a backup tool is the backing up of integrated catalog facility user catalogs and their aliases. The user catalog name must be fully-qualified with the **INCLUDE** keyword on the **DUMP** command. With **MVS/XA DFP Version 2 Release 3** and later releases, the **LOCK** attribute of an integrated catalog facility user catalog is dumped. The **LOCK** status is preserved if the catalog does not exist at restore time. Otherwise, the **LOCK** status of the existing catalog is used. See *Catalog Administration Guide* for a description of the **LOCK** attribute.

The following example shows the **JCL** used to dump an integrated catalog facility user catalog. The master password is required to dump the catalog. Passwords are not required if you have **RACF DASDVOL** update access or if the installation authorization exit bypasses authorization checking.

```

//SYSPRINT DD SYSOUT=*
//TAPE DD VOL=SER=TAPE01,UNIT=3480,LABEL=(,NL),DISP=(NEW,CATLG),
// DSN=BACKUP.CATALOG
//SYSIN DD *
DUMP OUTDD(TAPE) -
DS(INCL(TEST.CAT.USERCAT)) -
PSWD(USER.CAT/USERMPW1)
/*

```

Figure 24 shows printed output produced by the dump

```

PAGE 0001 DFDSS 5665-327 DATA FACILITY DATA SET SERVICES V2R5.0 89.069 18:58
DUMP OUTDD(TAPE) -
DS(INCL(TEST.CAT.USERCAT)) -
PSWD(TEST.CAT.USERCAT/)
ADR101I RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP '
ADR109I RI01 (01), 89046 10:30:38 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR006I (001)-SETUP(01), 89046 10:30:38 EXECUTION BEGINS
ADR454I (001)-DTDSC(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED.
TEST.CAT.USERCAT
ADR013I (001)-CLTSK(01), 89046 10:32:24 TASK COMPLETED WITH RETURN CODE 0000
ADR012I DSSU (01), 89046 10:32:25 DFDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000

```

Figure 24. Output From a Dumped Integrated Catalog Facility User Catalog

Note:

- A logical data set dump cannot be used to dump the master catalog unless the master catalog is first redefined as a user catalog.
- Data set passwords are suppressed and do not appear in the output.

Dumping VSAM Spheres

With DFDSS Version 2 Release 4, it is possible to dump an entire VSAM sphere (base cluster and all associated alternate index clusters and paths). The SPHERE keyword causes DFDSS to dump the entire VSAM sphere. To dump the base cluster and the other components, all you need to specify is the base cluster name.

An example of the DUMP command with the SPHERE keyword is:

```

DUMP -
OUTDDNAME(TAPE) -
DATASET(INCLUDE(PARTS.VSAM1)) -
SPHERE -
PSWD(PARTS.VSAM1/MASTUPW1) -
COMPRESS

```

Note: You should be aware that you cannot restore a sphere unless it is dumped as a sphere with the SPHERE keyword.

The PDSE cannot be dumped on a system using releases prior to DFDSS Version 2 Release 5, nor can they be dumped on a system using releases prior to MVS/DFP Version 3 Release 2 and MVS/SP Version 3 Release 1.0E.

Dumping SYS1 System Data Sets

With the PROCESS keyword in conjunction with the SYS1 subkeyword, you can delete datasets, except for a VVDS or a VTOCIX, when performing a logical or a physical data set dump.

To limit the use of PROCESS, you need to set up a RACF facility class profile.

Backing Up SMS-Managed Data Sets

When backing up data sets in an SMS environment, you need to think about some special conditions in addition to those discussed under "Backing Up Data Sets" on page 45. The following sections discuss how you can back up SMS-managed data sets in an SMS environment.

In most cases, you should let DFHSM back up SMS-managed data sets for you. However, if you do not have DFHSM or you prefer not to rely on it for all your backup requirements, you can use DFDSS to back up your SMS-managed data sets.

Filter on Class Names: DFDSS can select data sets for DUMP processing based on their storage, management, and data class names. Because management class is the construct that contains a data set's availability attributes, you might want to filter on it when selecting data sets for DUMP processing.

If you wish to back up data sets in a particular management class, you can filter on the management class name. For example, if you want to perform incremental backup on data sets in management classes MCNAME1 and MCNAME2, specify the DUMP command as follows:

```
DUMP -  
  DATASET(INCLUDE(**) -  
          BY((MGMTCLAS, EQ, (MCNAME1, MCNAME2)) (DSCHA, EQ, YES))) -  
  OUTDDNAME(OUTVOL1)
```

Class Names Saved: DFDSS saves the class names of the data sets it dumps. These names are then used as input to ACS routines when the data set is restored.

Backing Up Volumes

With DFDSS, you can back up volumes either logically or physically. If the volume is to be restored to an unlike device, you must dump it logically.

Logical Volume DUMP

To perform a logical volume dump, you specify DATASET(INCLUDE(**)) and either LOGINDDNAME or LOGINDYNAM. LOGINDDNAME identifies the input volume that contains the data set(s) to be dumped. LOGINDYNAM specifies that the volume(s) containing data sets to be dumped be dynamically allocated.

Here is an example of how you specify the **DUMP** command to perform a logical volume dump:

```
DUMP DATASET(INCLUDE(**)) -  
  LOGINDDNAME(DASD1) -  
  OUTDDNAME(TAPE)
```

Note: Certain data sets can be restored only to like devices even though they were dumped logically.

Physical Volume DUMP

To perform a physical volume dump, specify the **DUMP** command with **INDDNAME** or **INDYNAM** and **OUTDDNAME**. Because **FULL** is the default for **DUMP**, you need not specify it. Unallocated tracks are not dumped. The following example shows how you can specify the **DUMP** command to physically back up a volume:

```
DUMP INDDNAME(DASD1) OUTDDNAME(TAPE)
```

If you plan to use stand-alone **DFDSS** restore to restore a volume (with no operating system), the volume must be dumped physically. In addition, when doing a full physical volume dump to back up a system residence volume, you must use **JCL** to invoke **DFDSS**.

Efficiency Considerations for Dumping Data

When backing up data, you can specify both the **OPTIMIZE** and **COMPRESS** keywords to improve performance and save dump space. The two keywords can be used together; they are no longer mutually exclusive.

Doing a selective data set **DUMP** saves space, but performance is slower than a full volume **DUMP**. The same applies to the **COMPRESS** parameter. It saves dump space, but involves some processing overhead. In general, if you are dumping to tape, saving space is probably less of a concern than performance. Space savings is usually important only when it results in using fewer tapes to store the data because that, in turn, reduces the number of tape mounts necessary to recover the data.

Space Considerations

Using larger block sizes saves dump space and improves performance by minimizing the number of I/O operations performed during a **DUMP** operation.

Note: If you have a 3480 tape drive with the Improved Data Recording Capability (**IDRC**) and you want to use hardware data compaction, you do not need to use the **COMPRESS** keyword with the **DUMP** command. If you want software compression, specify the **COMPRESS** keyword, but do not specify **DCB=TRTCH=COMP** in the **JCL**.

The default block size for output records written to tape is 65 520 bytes (65 520 is also a maximum). You can change this default to 32 760 bytes by turning on the **UFOBK32K** bit in the **ADRUF0** macro when the installation exit is called. For

output records written to DASD, the block size is the track length of the output volume for devices whose track length is less than 32K bytes, or one half the track length for devices whose track length is greater than 32K bytes. You can select a different block size by coding *DCB=BLKSIZE=block size* in the corresponding DD statement. The minimum block size you can select is 7892 bytes; the maximum is 32760 bytes.

Note: To have the block size specification included in the tape label, you must specify the BLKSIZE parameter in the DD statement for the tape.

You can also save dump space by doing the following:

- Dump only the used space (the default), instead of all allocated space, in SAM and PAM data sets or in data sets with a null DSORG field.
- Use the COMPRESS parameter.
- Perform incremental data set backup instead of volume backup. This reduces the amount of data dumped and decreases processing time.

Performance Considerations

You can improve performance by doing the following:

- Dump to tape, where the larger block size reduces the number of I/O operations.
- Use OPTIMIZE(2), (3), or (4) to read a number of tracks per read operation, equal to two tracks, five tracks, or a full cylinder, respectively. The default, OPTIMIZE(1), reads one track at a time. OPTIMIZE(2), (3), or (4) results in less elapsed time and fewer EXCPs on the DASD device, provided:
 - The names of track overflow data sets not flagged as such in the VTOC's Format 1 DSCB are specified with the TRACKOVERFLOW keyword and
 - The load on the tape channel is low enough and the tape speed high enough to keep pace with the data being read from the DASD volume.

Shared DASD Considerations

Shared DASD presents volume and data set serialization problems not encountered in nonshared DASD environments. Care should be taken when you do data set operations if programs operating in another processor might be accessing the data sets at the same time.

A data set can be dumped from one processor while being processed from another. The dumped version may be partially updated on JES2 systems. This is the same exposure that is present on a full DUMP.

Restoring Data Sets

With the RESTORE command, you can restore data to DASD volumes from DFDSS-produced dump volumes. The dump volume to be restored is identified with the INDDNAME keyword.

The RESTORE is logical or physical depending upon the dump tape. If the dump tape was made physically, a physical RESTORE is done. If it was made logically, a logical RESTORE is done. If the data was compressed when it was dumped, it is automatically expanded to its original form during the RESTORE operation.

As with a data set DUMP command, you can use filtering to select data sets for RESTORE processing. See "Choosing Data Sets for Processing—Filtering" on page 13, for more information on filtering.

Note: Fully-qualified names are required to restore the following data sets:

VVDS
VTOCIX
SYS1.STGINDEX
Integrated catalog facility catalogs
OS catalog
VSAM read-only data sets (temporarily exported with the INHIBITSOURCE parameter).

Logical Data Set Restore

A logical data set RESTORE is done if you are restoring from a logical dump tape (a tape created with logical DUMP) and you specify the DATASET keyword. For instance, the following RESTORE command generates a logical data set RESTORE if the tape was created by a logical DUMP.

```
RESTORE INDDNAME(TAPE) -  
        DATASET(INCLUDE(USER1.OLDDS)) -  
        REPLACE
```

DFDSS logical RESTORE processing cannot be used to process PAM data sets containing location-dependent information that does not reside in note lists or the directory. As an example, PDS load libraries that contain overlay modules are not supported during logical RESTORE processing.

You can restore to unlike devices with logical RESTORE.

Output Volume Selection

In most cases, specifying output volumes is optional for logical data set RESTORE. Output volume specification is required only if the data set:

- Exists and is to be restored to a volume that is different from the current location or
- Does not exist and is to be restored to a volume that is different from the source volume.

Specify output volumes with the OUTDDNAME or OUTDYNAM keywords. An example of a logical data set RESTORE with OUTDDNAME is:

```
RESTORE -  
        INDDNAME(TAPE) OUTDDNAME(DASD1) -  
        DATASET(INCLUDE(**))
```

When not specified, the volume on which the data set currently resides is found from the catalog and dynamically allocated. This is particularly useful on a data set RESTORE from a data-set-selection-by-catalog dump because you need not know where the data sets resided at dump time.

You can specify multiple output DASD volumes on a logical data set RESTORE. This is required when all the data sets to be restored cannot fit on a single volume. An example of a logical data set RESTORE with a spill volume specified is:

```
RESTORE -  
INDDNAME(TAPE) OUTDYNAM((338001),(338002)) -  
DATASET(INCLUDE(PARTS.**)) -  
PCTU(80)
```

Note the use of PERCENTUTILIZED (PCTU) in the above example. With PERCENTUTILIZED, you can set a limit on the amount of space DFDSS can fill on the volume. When this limit is reached, subsequent data sets are allocated to other volumes. In the above example, PERCENTUTILIZED is used to specify that only 80% of the first target volume be filled. This leaves 20% free space for the data sets to extend, if necessary.

PERCENTUTILIZED is ignored for SMS volumes.

Note: User data-set labels on DASD volumes are supported during a data set RESTORE operation. However, either the data set on both the source and target volume must have these labels or neither must have them.

Restoring to Preallocated Target Data Sets

In some instances, you might want to control the placement of a data set on a volume when you restore it. Some data sets (such as data sets allocated by absolute track) have location-dependent data and must be preallocated. Others (such as catalogs) should be placed for performance reasons. For details on restoring such data sets, see "Restoring ISAM, Unmovable, DAM, and Absolute Track Data Sets" on page 61 and "Restoring Integrated Catalog Facility Catalogs" on page 60.

If a target data set is preallocated on either a like or unlike device, it is scratched and reallocated if it is not large enough to contain the dumped data set. VSAM preallocated target data sets are also scratched and reallocated when:

- Any of the following source and target data set attributes do not match:
 - CI size
 - Record length
 - IMBED (KSDS and key range data sets only)
 - Key length (KSDS and key range data sets only)
 - REPLICATE (KSDS and key range data sets only)
 - SPANNED.
- The preallocated target is multivolume and the space of the target data set on the first volume is not large enough to contain all of the dumped data.
- The data set was not defined as reusable and the high-used relative byte address (RBA) of a target VSAM KSDS is not 0.

In logical restore, partitioned data sets are restored to both like and unlike devices in a compressed format. If the partitioned data set is being restored to an unlike device, the device-dependent information (such as TTR pointers and note lists) are in an usable form after the restore. DFDSS is unable to resolve device-dependent information for all other data set types being restored to unlike devices.

The NOPACKING keyword is effective only for partitioned data sets. If NOPACKING is specified for preallocated partitioned data sets, the preallocated target must reside on a same or like device. Processing is stopped for the data set if the target resides on an unlike device. The target is not deleted and reallocated.

Cataloging Data Sets During Logical RESTORE

When you restore a data set, you might need to catalog it in the standard order of search or recatalog it in its original catalog. The CATALOG keyword catalogs the data set in the standard order of search. The RECATALOG(*) keyword catalogs it in the same catalog that points to the source data set.

When a data set is restored as an SMS data set, it is cataloged using the standard order of search. The RECATALOG keyword is ignored.

Examples of the CATALOG and RECATALOG keywords in a logical data set RESTORE follow.

```
RESTORE -  
  INDDNAME(TAPE) -  
  DATASET(INCLUDE(USER1.**)) -  
  CATALOG
```

```
RESTORE -  
  INDDNAME(TAPE) -  
  DATASET(INCLUDE(USER1.**)) -  
  RECATALOG(*)
```

When a VSAM KSDS or key range data set is being restored to an unlike device, the data set must be cataloged in the standard order of search. You should also note that a VSAM KSDS or key range data set that has had CA splits and is being restored to an unlike device, will take significantly longer to restore.

Renaming Data Sets During Logical RESTORE

In addition to cataloging data sets when they are restored, you can also rename restored data sets by using the RENAME keyword. For instance, you can code the following to rename a data set you are restoring:

```
RESTORE -  
  INDDNAME(TAPE) -  
  DATASET(INCLUDE(USER2.OLDDS)) -  
  RENAME(*.OLDDS,*.NEWDS)
```

Note: RENAME will work only if the data set exists on the output DASD with the old name. If you really want to unconditionally rename a data set, use RENAMEU. Both VSAM and non-VSAM data sets can be renamed. The rules for renaming VSAM clusters are the same as for non-VSAM data sets. You can only rename clusters. DFDSS assigns a new name for the components of VSAM clusters. If the old component names were assigned by the system, the new component names are

assigned by using a similar mechanism. If the component names were assigned by using the cluster name (such as clustername.data or clustername.index), the new component names are similarly assigned.

Logical RESTORE of RACF-Protected Data Sets

For logical data set RESTORE of data sets that were RACF-protected during the DUMP, you must have update access to the target data set. If the data set is being renamed, you must have read access to the source data set as well as update access to the target data set. If a profile no longer exists for non-VSAM data sets that were protected under a discrete profile at DUMP time, you will need to predefine a new discrete profile or preallocate the data set with a discrete profile. These conditions do not apply if authorization checking is bypassed in any way.

DFDSS tries to define renamed data sets to RACF during a data set RESTORE if they were RACF-protected during the DUMP. If the high-level qualifier of the data set is replaced with the user-supplied RENAME parameter, this new group of first-level qualifiers of newname must have been defined to RACF. If you restore a RACF-protected data set to a volume other than the one from which the dump was taken and do not use the RENAME parameter, the restored data set becomes a duplicate-named data set. DFDSS does not create password entries for renamed password-protected data sets.

The following table shows how DFDSS treats RACF-protected data sets for a logical RESTORE when the user has sufficient data set access authority.

	No RENAME— Preallocated	No RENAME— DFDSS-allocated	RENAME— Preallocated	RENAME— DFDSS-allocated
Unprotected source at DUMP time	No profile to define.	No profile to define.	No profile to define.	No profile to define.
Generically protected source at DUMP time	If already generically or discretely protected, no profile to define; else, define a discrete profile.	If already generically protected, no profile to define; else, define a discrete profile.	If already generically or discretely protected, no profile to define; else, define a discrete profile.	If already generically protected, no profile to define; else, define a discrete profile.
Discretely protected source at DUMP time	If already discretely protected, no profile to define; else, define a discrete profile.	Define a discrete profile.	If already discretely protected, no profile to define; else, define a discrete profile.	Define a discrete profile.

Figure 25. RACF-Protected Data Sets in a Logical Data Set RESTORE

Notes to Figure 25:

- Data sets protected by both generic and discrete profiles at DUMP time are considered to be discretely protected.
- If MENTIVITY is not specified when DFDSS defines discrete profiles, default discrete profiles are defined. The default discrete profiles give ALTER authority to the user ID specified on the JOB statement.

Physical Data Set Restore

A physical data set RESTORE is done if you are restoring from a tape created by physical DUMP and you specify the DATASET keyword. If the dump volumes resulted from a physical data set DUMP operation, you must do a physical data set RESTORE or a tracks RESTORE. A tracks RESTORE can consist of a subset of the dump data.

Notes:

- When you perform a physical restore of many data sets, there is an initial delay while DFDSS allocates the target data sets.
- To prevent index components from being restored inadvertently, you must specify the fully-qualified name of the cluster.

On a physical data set RESTORE, data sets from one or more logical volumes can be restored to a single DASD volume. If you want to restore data sets from specific source DASD volumes, use the LOGICALVOLUME keyword to specify the volume serial numbers of the source DASD volumes you want to restore. For example:

```
RESTORE -  
      INDDNAME(TAPE) OUTDDNAME(DASD1) -  
      DATASET(INCLUDE(**)) LOGICALVOLUME(111111) -  
      REPLACE
```

Note: Data from a specific volume can be restored only to a DASD volume of like device type.

Output Volume Selection

For physical data set RESTORE, you must specify an output volume with the OUTDDNAME or OUTDYNAM keyword. A physical data set RESTORE restores only the first volume in a list passed in the OUTDDNAME or OUTDYNAM parameter. Selection of target volumes in an SMS environment is considerably different. See "Restoring SMS-Managed Data Sets Physically" on page 67 for more details.

Cataloging Data Sets During Physical RESTORE

If you specify CATALOG on a physical data set RESTORE, DFDSS creates catalog entries for single volume non-VSAM data sets that were allocated by DFDSS. The cataloging is done immediately after successful allocation of a data set. Failure in cataloging, however, does not prevent the data set from being restored. Any data set that was allocated and cataloged but encounters errors during the RESTORE operation is neither uncataloged nor scratched by DFDSS. You must not specify the RECATALOG keyword for physical RESTORE.

The catalog that DFDSS uses to catalog a data set is determined as follows:

- The catalog specified by a STEPCAT or JOBCAT (but not both) DD statement.
- If the STEPCAT or JOBCAT DD statement specifies concatenated catalogs, only the first catalog in the concatenation is used.

- If no STEPCAT or JOBCAT DD statement is provided and the first qualifier of the data set name is an alias for a user catalog or for a CVOL, the catalog pointed to is used for that data set.
- Otherwise, the master catalog is used.

You should use the IDCAMS DEFINE RECATALOG command to catalog VSAM data sets that were allocated by DFDSS (not preallocated).

Physical RESTORE of RACF-Protected Data Sets

For physical data set RESTORE of RACF-protected data sets, you must have update access to the target data set. To rename non-VSAM data sets during RESTORE, you must have read access to the source data set and update access to the target data set. If a non-VSAM data set was protected under a discrete profile and the profile no longer exists, a new discrete profile must be defined before the RESTORE, or the data set must be preallocated with a discrete profile. These conditions do not apply if authorization checking is bypassed in any way.

The protection status of data sets that are restored by way of physical data set RESTORE is unpredictable if:

- RACF profiles (generic or discrete) of the data sets have been changed between DUMP and RESTORE
- The dump was produced on a system (that supports RACF generic profiles) other than the one used for the restore.

Restoring Data Sets with Special Requirements

Some data sets have special requirements for being restored. The sections that follow describe some of the special cases you might encounter when you restore data sets.

Restoring Multivolume Data Sets

Multivolume data sets from a logical data set dump tape can be restored either to a single volume or to multiple volumes. When they are not preallocated and the specified output volumes are different from the input volumes, multivolume data sets are restored to a single volume, space permitting. When multiple target volumes are specified, DFDSS selects target volumes as follows:

- If a target volume that has the same volume serial as the source volume is available and has adequate space, it is chosen.
- If a volume of the *same* device type is available, and if it has adequate space, it is selected.
- A volume of a *like* device type is selected if it has adequate space.
- A volume of an *unlike* device type is selected if it has adequate space.

Note: To catalog multivolume non-VSAM data sets, use IEHPROGM.

If you are restoring a multivolume data set from a physical dump, be sure the segments from all volumes are restored with successive RESTORE commands. Restoring a portion of a multivolume non-VSAM data set is allowed only if the volume sequence numbers of the source and target volumes are the same. A preallocated data set is required on the target volume.

Restoring Integrated Catalog Facility Catalogs

Integrated catalog facility user catalogs can be restored only to the same volumes from which they were dumped. The component names of the source and target user catalog must be the same. In addition, you must specify the fully-qualified name to restore a catalog.

Logical RESTORE is generally the best way to restore a catalog because it restores user catalog aliases if they are present in the logical dump data set. For logical RESTORE, user catalog aliases are restored as follows:

- If DFDSS allocated the user catalog, aliases are restored if the catalog is successfully restored.
- If the target catalog was preallocated and is not empty, aliases are not restored.
- If the target catalog was preallocated and empty, aliases are restored.

Physical RESTORE does not restore aliases, and physically dumped catalogs cannot be restored if they are open. In addition, if the entries in the catalog during DUMP do not match the entries during the physical RESTORE, some of the data sets may become inaccessible.

With MVS/XA DFP Version 2 Release 3 or later release, an integrated catalog facility user catalog can be restored dynamically. Catalog recovery jobs should be modified to include the IDCAMS ALTER LOCK command to lock the existing catalog before the DFDSS RESTORE. After the recovery is complete, unlock the catalog using IDCAMS ALTER UNLOCK. The LOCK attribute on the dump tape is used if the catalog does not exist. For information on the LOCK attribute and the access authority, see *Catalog Administration Guide*.

The following example shows the JCL used to restore an integrated catalog facility user catalog. The master password of the master catalog is required. Passwords are not required if you have DASDVOL update access or if the installation authorization exit bypasses authorization checking.

```
//SYSPRINT DD  SYSOUT=A
//TAPE      DD  UNIT=3480,VOL=SER=TAPE01,
// LABEL=(,NL), DISP=(OLD,KEEP),DSNAME=USER2.BACKUP
//DASD      DD  UNIT=3380,VOL=SER=338001,DISP=OLD
//SYSIN     DD  *
RESTORE OUTDD(DASD) INDD(TAPE) -
          DS(INCL(TEST.CAT.USERCAT)) REPLACE -
          PSWD(TEST.CAT.USERCAT/USERMPW1)
/*
```

Figure 26 on page 61 shows the printed output produced by the RESTORE.

Note: Data set passwords are suppressed and do not appear in the printed output.

```

RESTORE OUTDD(DASD) INDD(TAPE) -
  DS(INCL(TEST.CAT.USERCAT)) REPLACE -
  PSWD(TEST.CAT.USERCAT/)
ADR101I      RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'RESTORE '
ADR109I      RI01 (01), 88014 10:36:34 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR006I (001)-SETUP(01), 88014 10:36:36 EXECUTION BEGINS
ADR780I (001)-TDDS (01), THE INPUT DUMP DATA SET BEING PROCESSED IS IN LOGICAL DATA SET FORMAT
ADR396I (001)-NEWDS(01), DATA SET TEST.CAT.USERCAT ALLOCATED, IN CATALOG ICFMAST1, ON VOLUME(S): D8D020
ADR489I (001)-TDLOG(02), CLUSTER TEST.CAT.USERCAT WAS RESTORED
                CATALOG      ICFMAST1
                COMPONENT    TEST.CAT.USERCAT
                COMPONENT    CATINDEX.T71FCDB4.VID88014.T9DE8880
ADR393I (001)-TDNVS(01), ALIAS USER1 RESTORED FOR USER CATALOG TEST.CAT.USERCAT
ADR393I (001)-TDNVS(01), ALIAS USER2 RESTORED FOR USER CATALOG TEST.CAT.USERCAT
ADR393I (001)-TDNVS(01), ALIAS USER3 RESTORED FOR USER CATALOG TEST.CAT.USERCAT
ADR393I (001)-TDNVS(01), ALIAS USER4 RESTORED FOR USER CATALOG TEST.CAT.USERCAT
ADR393I (001)-TDNVS(01), ALIAS USER5 RESTORED FOR USER CATALOG TEST.CAT.USERCAT
ADR393I (001)-TDNVS(01), ALIAS USER6 RESTORED FOR USER CATALOG TEST.CAT.USERCAT
ADR393I (001)-TDNVS(01), ALIAS USER7 RESTORED FOR USER CATALOG TEST.CAT.USERCAT
ADR393I (001)-TDNVS(01), ALIAS USER8 RESTORED FOR USER CATALOG TEST.CAT.USERCAT
ADR393I (001)-TDNVS(01), ALIAS USER9 RESTORED FOR USER CATALOG TEST.CAT.USERCAT
ADR393I (001)-TDNVS(01), ALIAS USER0 RESTORED FOR USER CATALOG TEST.CAT.USERCAT
ADR013I (001)-CLTSK(01), 88014 10:37:13 TASK COMPLETED WITH RETURN CODE 0000
ADR012I      DSSU (01), 88014 10:37:15 DFDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000

```

Figure 26. Output from Restored Integrated Catalog Facility User Catalog

Restoring ISAM, Unmovable, DAM, and Absolute Track Data Sets

One important use of DFDSS is restoring data sets that contain device-dependent information. In some cases, such data sets can be restored without preallocating the target data sets. In other cases, however, you must preallocate the target in order to restore the data set.

Restoring without Preallocated Targets: If you have MVS/XA DFP Version 2 Release 1 (or any later release), you can use logical data set RESTORE to restore an ISAM data set without preallocating it if the tracks on which it is to be restored are not occupied and the volume has an indexed VTOC. If an ISAM or unmovable data set is not preallocated and the system is running in the MVS/XA DFP Version 2 Release 1 environment, DADSM is called to allocate the data set. If DADSM fails or if the system does not run in the MVS/XA DFP Version 2 Release 1 environment, the data sets are allocated to any available location if FORCE is specified.

When you specify FORCE and some of the data sets have truly location-dependent data, you should prevent DFDSS from restoring them by specifying their names in the EXCLUDE parameter. Subsequently, you must either free up the area of the DASD where these data sets were located on the source volume and rerun the RESTORE operation or restore the data set onto a scratch volume.

If you do not have MVS/XA DFP Version 2 Release 1 (or later release), you can restore an unmovable or an ISAM data set without preallocating it by doing the following:

1. Restore track 0 and the VTOC from the dump tape to a scratch volume.
2. Restore the desired unmovable data set to the scratch volume from the dump tape.
3. Either scratch all unwanted data sets from the scratch volume and use the unmovable data set that is left, or use an appropriate utility such as IEBISAM, IEBCOPY, or IEHMOVE to move this data set to the desired volume.

Restoring to Preallocated Targets: If you are restoring any of these types of data sets by preallocating them, the size and location of the extents on the dump volume and on the restore volume should match.

For unmovable data sets (those allocated as ABSTR, PSU, POU, or DAU), a preallocated data set is restored if the extents match and REPLACE is specified. Even if the extents do not match, the data set is restored if you specify both REPLACE and FORCE. When an unmovable data set cannot be restored, the extents of the data set on the source volume are listed so that you can take action to restore it.

Note: ISAM data sets and CVOL data sets cannot be restored to unlike devices.

Restoring DAM Data Sets: When DFDSS restores Direct Access Method (DAM) data sets, the following apply:

- If the receiving volume track capacity is equal to or greater than the source volume track capacity, the DAM data set is allocated on the receiving volume with the same allocation as the source contained. The restore yields an usable data set; however, some unused space might remain on each track of the target data set.
- If the receiving volume track capacity is smaller than the source, FORCE must be specified. This forces the DAM data set to be restored on any available space by track packing. DFDSS updates the relative byte address (RBA) information contained in fixed-length data sets so these will still be usable after the restore; variable- and undefined-length TTR-organized BDAM data sets might be unusable.

Restoring an Undefined DSORG Data Set

The PROCESS(UNDEFINEDSORG) keyword permits logical data set RESTORE of an undefined DSORG data set to an unlike device of larger track capacity. The restore yields an usable data set; however, some unused space might remain on each track of the target data set.

Restoring a VSAM Sphere

With DFDSS Version 2 Release 4, you can restore an entire VSAM sphere (base cluster and all associated alternate index clusters and paths). The SPHERE keyword causes DFDSS to restore the entire VSAM sphere. Provided the dump was also taken with the SPHERE keyword specified, all you need to specify is the base cluster name to restore the base cluster and the other components.

When you restore a SPHERE to a preallocated target, all components (base clusters, alternate indexes, and paths) of the sphere must be preallocated. DFDSS does not restore a sphere if only some parts of the sphere are preallocated.

An example of the RESTORE command with the SPHERE keyword is:

```
RESTORE -  
      INDDNAME(TAPE) -  
      DATASET(INCLUDE(PARTS.VSAM1)) -  
      SPHERE -  
      REPLACE -  
      PSWD(PARTS.VSAM1/MASTUPW1)
```

“Restrictions for RESTORE” on page 115 lists considerations for restoring VSAM spheres.

Restoring a VSAM Cluster: Restoring a VSAM cluster is allowed if the following are the same on the source and destination volumes:

- The number of components on the volume
- The beginning relative byte address (RBA)
- The component names
- Catalog names.

If a cluster is preallocated, its size must be equal to or greater than that on the source volume. (Only the tracks that were dumped are restored.)

If you preallocate a VSAM data set before restoring the data set via DFDSS, you should ensure that the control interval size, allocation unit, and secondary allocation quantity are the same as in the initial definition.

Note: VSAM data sets residing on DFDSS data set dump tapes created in an MVS/XA DFP Version 2 Release 2 (or later release) environment cannot be restored in an MVS/XA DFP Version 2 Release 1 or an MVS/370 DFP Version 1 environment without certain VSAM N*512 toleration PTFs.

Restoring the VVDS and the VTOCIX

To restore a VVDS or VTOCIX data set, you must specify the fully-qualified data set name. The VVDS and the VTOCIX data set cannot be restored with other data sets in the same RESTORE command. VVDS and VTOCIX data sets should not be restored by data set as a normal recovery procedure. The VTOCIX data set is an extension of the VTOC and can be rebuilt using IBM's Device Support Facilities program (ICKDSF).

The VVDS is an extension of the VTOC and of the catalogs for the VSAM data sets on the volume. If it is restored by a data set RESTORE, it is possible that some of these data sets can become unusable because of a mismatch between the catalog, VVDS, and the VTOC. If this occurs, run the diagnose function of Access Method Services to determine the extent of the problem and take appropriate corrective action.

Restoring a PDSE

With DFDSS Version 2 Release 5 it is possible to restore a PDSE. The PDSE cannot be restored by using releases prior to DFDSS Version 2 Release 5, nor can they be restored on a system using releases prior to MVS/DFP Version 3 Release 2 and MVS SP Version 3 Release 1.0E. The target volume must be SMS-managed.

Restoring a Damaged PDS

A PDS is monitored by DFDSS during track packing for conditions that are not normal. The following conditions are detected and reported:

- Missing high key entry in the PDS directory.
- Missing directory EOF.
- Invalid member start TTR
 - TTR points before directory EOF
 - TTR points after end of data set.

- **Missing member EOF.** (Each member of a partitioned data set is normally terminated by an EOF record.)
- **Invalid note or notelist TTR.**
 - Note pointing before the start of member data
 - Note pointing after the member EOF
 - Note pointing past the last valid record on a track
 - Note pointing to record 0 of a track.

All of these conditions are noted by DFDSS with a message.

During track packing, DFDSS repairs missing high key directory entry, missing directory EOF, and missing member EOFs.

Invalid start TTRs prevent DFDSS from track packing data for that member. DFDSS translates all valid note and notelist TTRs during track packing.

Damaged partitioned data sets can be restored to same or like device target volumes by using the **NOPACKING** keyword. This results in an **exact track-for-track image** of the source data set. Obviously, no compression is performed in this case.

Restoring Data Sets in an SMS Environment

Use the **RESTORE** command to recover data sets in an SMS environment. If the data set was dumped logically, it is recovered logically. If it was dumped physically, it is recovered physically.

As discussed earlier, an SMS environment can contain both SMS-managed and non-SMS-managed data. The following sections discuss how you can use the **RESTORE** command to recover these data sets.

Restoring SMS-Managed Data Sets

When you use the **RESTORE** command in an SMS environment, Automatic Class Selection (ACS) routines are invoked.

When you use **RESTORE**, you are in the ACS RECOVER environment (&ACSENVIR = 'RECOVER'). The storage class ACS routine is executed first. If the storage class assigned is not null, the management class ACS routine and then the storage group ACS routine are executed. (See "ACS Variables Available during RESTORE and CONVERTV" on page 110 for a list of variables available to ACS routines during RESTORE processing.)

DFDSS passes a data set's classes to ACS as input, and the ACS routines can assign or override these input classes.

If the source data set is non-SMS managed and has no class names, DFDSS passes null classes to ACS. If the source data set is SMS managed and you do not specify otherwise, DFDSS passes ACS the source data set's classes. If you specify what you want passed to ACS with the **STORCLAS**, **MGMTCLAS**, **NULLSTORCLAS**, or **NULLMGMTCLAS** keywords, DFDSS passes ACS what you specify. In all cases, the ACS routines ultimately decide the classes assigned to the data set.

You can, however, force the storage and/or management class you specify to be assigned to a data set by using the **BYPASSACS** keyword with the **RESTORE** command.

The following RESTORE command results in ACS routines determining the target classes, using the source classes as input:

```
RESTORE -  
    INDDNAME(TAPE) -  
    DATASET(INCLUDE(USER12.**))
```

For details about using RESTORE to convert data to and from SMS management, see Chapter 9, "Converting Data to and from SMS Management" on page 91.

Changing Storage Class with RESTORE

In some cases, you might want to pass ACS a storage class that is different from that of the source data set. You can specify RESTORE with STORCLAS to pass ACS a storage class name as follows:

```
RESTORE -  
    INDDNAME(TAPE) -  
    DATASET(INCLUDE(USER12.**)) -  
    STORCLAS(SCNAME1) -  
    REPLACE
```

However, using STORCLAS does not guarantee that the data set is assigned the storage class you specify. It means only that the storage class you specified is passed to the ACS routines. Depending on how the ACS routines are written, the storage class you specify can be ignored, assigned to the data set, or used in combination with other input variables to determine a new storage class for the data set.

RACF checks if the RESOWNER of a given data set is authorized to define the data set with the given STORCLAS. Ensure that the RESOWNER of the data set has the proper authority to use the indicated storage class.

To make certain that the storage class you specify is assigned to the data set, you can use BYPASSACS as follows:

```
RESTORE -  
    INDDNAME(TAPE) -  
    DATASET(INCLUDE(USER12.**)) -  
    STORCLAS(SCNAME1) -  
    BYPASSACS(**) -  
    REPLACE
```

In this case, ACS is not invoked, and therefore the data set is assigned whatever you have specified with STORCLAS or, if you do not use STORCLAS, the source data set's storage class. However, when you use BYPASSACS, you should make certain that the storage class you specify is valid. If you use BYPASSACS and assign an invalid storage class, you get an error.

To limit the use of **BYPASSACS**, you need to set up a RACF facility class profile.

You can use the **NULLSTORCLAS** keyword in conjunction with **BYPASSACS** to make a data set non-SMS managed. For example, the following specification of **RESTORE** causes the specified data sets not to be SMS managed:

```
RESTORE -
  INDDNAME(TAPE) -
  DATASET(INCLUDE(USER12.**)) -
  NULLSTORCLAS -
  BYPASSACS(**)
```

Changing Management Class with **RESTORE**

In addition to influencing a data set's storage class when you restore it, you can also give ACS input for assigning or overriding the data set's management class. By specifying **MGMTCLAS**, you can pass a management class name to ACS and, as with **STORCLAS**, ACS may ignore it, assign it to the data set, or use it in combination with other input variables to determine the data set's management class. By specifying **NULLMGMTCLAS**, you can pass a null management class to ACS, which may or may not assign a management class.

An example of **RESTORE** with the **MGMTCLAS** keyword is:

```
RESTORE -
  INDDNAME(TAPE) -
  DATASET(INCLUDE(USER12.**)) -
  MGMTCLAS(MCNAME1) -
  REPLACE
```

Like **STORCLAS**, RACF checks if the **RESOWNER** of a given data set is authorized to define the data set with the given **MGMTCLAS**. Ensure that the **RESOWNER** of the data set has the proper authority to use the indicated management class.

Also just as you can with **STORCLAS**, you can use **MGMTCLAS** with **BYPASSACS** to make certain that the data set is assigned the management class you specify. For instance:

```
RESTORE -
  INDDNAME(TAPE) -
  DATASET(INCLUDE(USER12.**)) -
  MGMTCLAS(MCNAME1) -
  BYPASSACS(**) -
  REPLACE
```

If you use **BYPASSACS** with **MGMTCLAS**, you should be certain that you are assigning a valid management class to the data set. If the management class is not

valid, the data set is not restored. Remember that **BYPASSACS** skips both the **STORCLAS** and **MGMTCLAS ACS** routines.

To limit the use of **BYPASSACS**, you need to set up a **RACF** facility class profile.

When you influence or assign the management class of a data set, you also need to be careful that the data set resides in a storage group capable of providing for the management class attributes associated with the management class you specify. For instance, if a data set has a management class that makes it eligible for migration, it needs to reside in a storage group on which **DFHSM** does migration. Otherwise, the data set will never migrate. For this reason, you might have to change the storage class along with the management class to ensure the data set resides on volumes that can accommodate its management class.

Restoring SMS-Managed Data Sets Physically

In general, it is recommended that you use logical data set **RESTORE** in a **SMS** environment. If you use physical data set **RESTORE**, you should be aware of the special rules for volume and construct selection.

When restoring a non-**SMS** user catalog on an **SMS**-managed volume or an **SMS**-managed user catalog on a non-**SMS** volume, physical restore does *not* convert the catalog. Instead, **DFDSS** physical restore ensures that the user catalog looks exactly like the source catalog (**SMS** or non-**SMS**) and then places the output volume in **INITIAL** status.

DFDSS physical **RESTORE** is sensitive to the number of logical volumes in a dump data set. A **DFDSS** physical dump tape can contain multiple logical volumes. Because a physical dump operates at the track-image level, every volume from which data was dumped is on the tape in the form of a logical volume. The following example shows how a dump tape can contain more than one logical volume:

```
DUMP -  
  DATASET(INCLUDE(**)) -  
  INDYNAM({338001},{338002}) -  
  OUTDD(TAPE) -  
  COMPRESS
```

If data sets are dumped from both volumes, two logical volumes are on the dump tape.

During physical **RESTORE**, the process is as follows for each logical volume:

- Determine the **SMS** constructs of each data set to be restored from that logical volume.
- Determine the target volume for the **RESTORE**. In a physical **RESTORE**, you can have only one target volume per logical volume.

The following process is used to select the target data set's SMS constructs:

1. If a data set is preallocated and **REPLACE** is specified, the constructs of the preallocated data set are used.
2. For data sets that are preallocated and have **RENAME** or **RENAMEU** specified and data sets that are not preallocated:
 - a. If **BYPASSACS** is specified for the data set, the constructs from the data set on the dump tape are used (unless overridden by **STORCLAS**, **MGMTCLAS**, **NULLSTORCLAS**, and/or **NULLMGMTCLAS**).
 - b. If **BYPASSACS** is not specified for the data set, the ACS routines are invoked to determine the data set's constructs. The constructs from the data set on the dump tape are passed to ACS as input (unless overridden by **STORCLAS**, **MGMTCLAS**, **NULLSTORCLAS**, and/or **NULLMGMTCLAS**).

The following process is used to select a target volume. The process continues until a volume is selected.

1. If any input data set is preallocated on the user-specified volume and **REPLACE** or **RENAMEU** is specified, the user-specified volume is the target volume for the entire logical volume.
2. If any input data set is non-VSAM and multivolume with a sequence number greater than one, the user-specified volume is the target volume for the entire logical volume.
3. Repeat the following steps until a data set is successfully allocated. Once a data set is allocated, all other data sets from the logical volume are restored on the same volume:
 - a. If the data set is to be SMS managed, call VTOC Data Set Services to select a volume to restore the entire logical volume to.
 - b. If the data set is to be non-SMS managed and the user-specified output volume is also not SMS managed, select the user-specified volume for the entire logical volume.
 - c. If the data set is non-SMS managed and the user-specified volume is SMS managed, continue with the next data set.

Once a target volume is selected, all data sets from the logical volume are restored to it with the following exceptions:

- If the **TGTGDS** keyword is specified, the appropriate status is assigned to the data set.
- The requested target status does not violate rules of the generation data group.
- The default status of logical and physical data set **RESTORE** is **DEFERRED**.

Restoring GDG Data Sets

For generation data group (GDG) data sets, filtering on generations is supported. Generation names in relative generation number, **dsn(n)**, can be specified in the **INCLUDE** and **EXCLUDE** parameters.

Restoring SMS-Managed GDG Data Sets

SMS-managed GDG data sets can be in any one of the following states:

DEFERRED
ACTIVE
ROLLED-OFF.

When restoring a GDG data set to SMS-managed storage, DFDSS does one of the following:

- Preallocated RESTORE retains the status of the preallocated generation data set (GDS).
- Physical data set RESTORE ensures that the non-VSAM volume records (NVRs) it builds for GDG data sets contain the correct status.
- Logical RESTORE places the GDS in DEFERRED status, if the TGTGDS keyword is not specified. DFDSS leaves the GDS in DEFERRED status to enable you to (1) roll it back as an ACTIVE generation or (2) leave it as DEFERRED.

Restoring Non-SMS-Managed Data Sets

If you want to restore a data set to a non-SMS-managed target volume, you can use the **NULLSTORCLAS** and **BYPASSACS** keywords on the **RESTORE** command. If you use these keywords, the data set is placed on a non-SMS-managed volume, regardless of whether or not the source data set was SMS managed.

Restoring Volumes

You can recover a volume or ranges of tracks from a full-volume DUMP. If the dump volumes resulted from a full DUMP operation, you can do a full or a tracks RESTORE (that is, ranges of tracks) or a data set RESTORE. If the dump volumes resulted from a tracks DUMP operation (that is, ranges of tracks), you must do a tracks RESTORE, which can consist of a subset of the dump data.

An example of a full-volume RESTORE is:

```
RESTORE -  
  INDDNAME(TAPE) -  
  OUTDDNAME(DASD1) -  
  PURGE
```

With the RESTORE operation, you can copy the volume serial number to the output DASD with the COPYVOLID keyword. For example:

```
RESTORE -  
  INDDNAME(TAPE) -  
  OUTDDNAME(DASD1) -  
  COPYVOLID -  
  PURGE
```

COPYVOLID is required if you are restoring an SMS volume, unless the source and target volume serial numbers match.

You must consider several factors when restoring volumes in an SMS environment. Before you start to restore a full volume, you must ensure that the status of the target volume is synchronized with its environment. For example, if the target volume is a non-SMS volume, the volume must not be defined in a storage group. Conversely, if the target volume is an SMS volume, the volume must be defined in a storage group. Finally, if the target volume is SMS-managed, then SMS must be active for the full-volume restore operation.

Output Volume Specification

For a full or tracks RESTORE, you must specify an output volume by using the **OUTDDNAME** or **OUTDYNAM** keywords.

The device type of the source volume used in the DUMP and the device type of the target volume used in the RESTORE must be the same. However, the following exceptions are possible:

- Data from an IBM 3330 Model 1 can be restored to an IBM 3330 Model 11.
- Data from an IBM 3340 with 35-megabyte capacity can be restored to an IBM 3340 with 70-megabyte capacity or to an IBM 3344.
- Data from a smaller capacity IBM 3380 model can be restored to a larger capacity IBM 3380 model.
- Data from a minivolume or a virtual volume can be restored to a real volume of like device type, and vice versa, device capacity permitting.
- Data from an IBM 3330 Model 11 can be restored to an IBM 3330 Model 1, from a 70-megabyte 3340 to a 35-megabyte 3340, or from a larger capacity IBM 3380 model to a smaller capacity IBM 3380 model, provided the range of data to be processed falls within the capacity of the output device.

Note: When you perform a full-volume restore to a DASD that is shared between multiple systems, the DASD should be offline to all systems except the one performing the restore.

When doing a full-volume restore to a larger-capacity DASD from a tape dumped from a smaller-capacity DASD with equivalent track lengths, you used to have to allocate a temporary data set in order to correct the free space chain. With DFDSS Version 2 Release 3 and later releases, DFDSS updates the free-space DSCB and calls ICKDSF to rebuild indexed VTOCs. ICKDSF Release 8 and prior releases prompt the operator for permission to do this.

As a result, you might want to determine if the output volume is allocated before performing the full-volume restore. This can be done with the following operator command:

```
D U,DASD,ALLOC,cuu,1
```

This command displays a list of currently allocated volumes and the names of the jobs enqueued on them.

If CATALOG has enqueued on the output volume, you can take the following steps:

1. Use the following catalog modify command to display a list of all open catalogs:

```
MODIFY CATALOG,LIST
```

2. Use the following catalog modify command to make CAS unallocate the catalog:

```
F CATALOG,UNALLOCATE(catname)
```

If there are no other allocated catalogs on the volume and the volume is not allocated by any other users, you can proceed with your full-volume restore.

For more information on CAS allocation, see *Catalog Administration Guide*.

Processing RACF-Protected Data Sets

On a physical RESTORE, DFDSS does not delete the profiles of RACF-protected data sets on the volume before a full RESTORE. After a full RESTORE, RACF profiles are not built for RACF-indicated data sets on the restored volume. If RACF data set profiles do not exist for these data sets, these data sets are inaccessible until RACF profiles are built for them.

When you use the COPYVOLID keyword to change the volume serial number or if the volume serial for the dump volume and the restored volume are different, profiles are not built for the RACF-protected data sets on the restored volume or for the RACF DASDVOL for RACF-protected DASD volume.

The protection status of data sets that are restored through a full RESTORE is unpredictable if:

- RACF profiles (generic or discrete) of the data sets were changed between DUMP and RESTORE
- The dump was produced on a system (that supports RACF generic profiles) other than the one used for the restore.

Recovering System Volumes

You can use stand-alone DFDSS restore to perform either a full or a tracks restore from the first data set of DFDSS-produced dump tape(s), without the use of a host system environment. Using stand-alone DFDSS restore, you can recover system volumes in order to bring up the host environment.

You can also use stand-alone DFDSS restore in a VM environment. Stand-alone DFDSS restore operates in Enterprise Systems Architecture (ESA)/370 mode, System/370 extended-architecture mode or System/370 mode. Refer to *Data Facility Data Set Services: Reference* for instructions on how to do the restore.

Chapter 8. Managing Data Movement with DFDSS

Data movement is necessary when you are doing the following tasks:

Replacing devices	When you remove devices to be replaced with other ones, you must move the data off the devices you are removing.
Adding devices	If you add new devices at your site, you must move data onto them to take advantage of the added capacity.
Maintaining devices	When you are servicing a volume, you might need to move data off the volume so users can continue to access the data.
Tuning performance	If a volume is performing poorly, it might be because a data set(s) on the volume is being frequently accessed and causing an I/O bottleneck. In this case, you might move that data set(s) to another volume(s) that is better able to handle it (either because it is less full or because it is cached).

You can use the DFDSS COPY command to move data between volumes.

This chapter is organized as follows:

- **“Preparing for Data Movement”** discusses some things you should do prior to moving data.
- **“Moving Data Sets” on page 74** describes how to use DFDSS to move data sets.
- **“Moving to Preallocated Data Sets” on page 76** describes how to use DFDSS to move to preallocated target data sets.
- **“Defining RACF Profiles” on page 78** explains how profiles should be defined.
- **“Moving Data Sets with Special Requirements” on page 79** describes how to use DFDSS to move data sets with special requirements.
- **“Moving Volumes” on page 84** describes how to use DFDSS to move volumes.
- **“Considerations for Moving Data Sets in an SMS Environment” on page 86** explains some of the special considerations for moving data sets in an SMS environment.
- **“Considerations for Moving Volumes in an SMS Environment” on page 89** explains some of the special considerations for moving volumes in an SMS environment.

Preparing for Data Movement

In addition to the actual movement of data, you can do a number of things to effectively move your data.

Before moving your data, you should determine the amount of space the data requires. You can determine this by building a data set or volume list with ISMF. A data set list indicates how much space is allocated for each data set and how much space it actually uses. A volume list indicates how much free space there is on each volume in the list. You can use this information to calculate how much space the data to be moved requires and to ensure that enough free space exists on the target volume(s). This calculation is especially important when combining multiple devices onto one larger capacity device.

Note: However, in an SMS environment, this calculation is unnecessary because the system finds the necessary free space and places the data for you.

In addition to ensuring that enough free space exists to contain the data, you should back up the data before moving it to guard against its loss during the movement. You can use the DFDSS DUMP command to back up volumes or data sets. See Chapter 7, “Managing Availability with DFDSS” and *Data Facility Data Set Services: Reference* for more information on backing up data and the DUMP command.

Evaluating the Use of Logical and Physical COPY

As previously stated, you can use the COPY command to perform the actual data movement. However, you must determine whether to use logical or physical COPY to move the data. Physical COPY gives you better performance, but logical COPY allows you to move data to unlike devices.

Because physical volume COPY enqueues on the volume, you should do it only at a time of low activity. Logical data set COPY enqueues at the data set level, and, as a result, it is less critical that you perform the COPY at a time of low activity. You should, however, seriously consider doing so because DFDSS cannot move data sets that are in use. In addition, because logical processing takes more time than physical processing, you should do it at a time of low activity, especially if you are moving a large number of data sets.

Note: It is best not to specify TOLERATE(ENQFAILURE) when moving data with COPY. Moving data while it is being updated can result in the loss of the updates.

After DFDSS has finished processing, you can verify that the data has moved by looking at the ISMF data set or volume list.

What DFDSS Copies

DFDSS copies only used space for SAM or PAM data sets and data sets with null DSORG fields (X'0000'), unless overridden by ALLDATA or ALLEXCP. Use the ALLDATA(*) and ALLEXCP parameters to process allocated space if:

- You are not sure of the data set organization (DSORG) of a data set on a volume when doing a full-volume COPY.
- There are SAM, PAM, or data sets with null DSORG fields (X'0000') that are not accessed using SAM or PAM.

Note: The COPY command requires temporary work space. Be sure public or storage volumes are available. Some temporary data sets are allocated to nonspecific devices by referring to SYSDA and/or SYSALLDA generic groups. If DFDSS is to function, these allocations must be allowed by the installation. Allocation validation exits to SVC 99 must not restrict DFDSS allocations.

Moving Data Sets

Using the COPY command with the DATASET keyword, you can copy one or more data sets from one DASD volume to another of like or unlike device type. If you specify the DELETE keyword with COPY, the data set on the source volume is deleted when it has been successfully copied to the target volume. In this way, you can perform a data set move.

Specifying Input Volumes

For DATASET COPY, input volume specification is not required. If you do not specify input volumes, data sets are selected from all the data sets cataloged in the standard search order.

If you do specify input volumes (but not ALLMULTI), catalogs are used to select only data sets that reside in their entirety on the volumes you specify with INDDNAME, INDYNAM, LOGINDDNAME, or LOGINDYNAM.

If you specify input volumes, a data set is selected if any of the following apply:

- Its volume sequence number is 1, its last volume flag is off, and it is cataloged in the standard catalog search order.
- Its volume sequence number is 1, and its last volume flag is on.
- It is a multivolume non-VSAM data set, all the volumes on which it resides are specified in the INDDNAME, INDYNAM, LOGINDDNAME, or LOGINDYNAM parameter, and it is cataloged in the standard catalog search order.
- It is a VSAM cluster and all volumes for all components are specified.

If a data set is found on more than one specified input volume and the volume sequence numbers match, DFDSS cannot determine which data set is to be selected for processing.

Output Volume Selection

Specifying output volumes is required for DATASET COPY in a non-SMS environment. (For a discussion of SMS considerations for moving data, see "Considerations for Moving Data Sets in an SMS Environment" on page 86.)

With the OUTDDNAME or OUTDYNAM keywords, you can specify multiple target volumes. This allows you to specify *spill* volumes. These spill volumes are used if the data sets you are moving require more space than is available on your first choice of volume.

If the output volume has unexpired data sets, you can either not process the data sets or write over them.

Renaming Data Sets

You can rename data sets by using the RENAMEU keyword of the COPY command. For VSAM data sets, you can rename only clusters. The new name for the components of VSAM clusters is derived by DFDSS. If the old component names were derived by the system, the new component names are derived by using a similar mechanism. If the component names were derived by using the cluster name (such as clustname.data or clustname.index), the new component names are similarly derived; however, if the new component name exceeds 44 characters, then the new component names are derived by using the naming convention of the system.

Moving to Preallocated Data Sets

In some cases, you might want to COPY data sets to preallocated targets. Note, however, that CVOLs, integrated catalog facility catalogs and system data sets that are named SYS1.* cannot be copied to preallocated data sets unless the PROCESS(SYS1) keyword is specified.

Rules for Moving to Preallocated Target Data Sets

The rules for moving VSAM and non-VSAM data sets to preallocated data sets follow.

VSAM Preallocation

An existing data set qualifies as a preallocated target for data set COPY if the cluster name matches and the complete cluster is available on target volumes.

The preallocated data set is usable if:

- The user is authorized to update the target data set.
- The cluster types match.
- The number of components match.
- The key length and offset match.
- The KEYRANGES match.
- None of the components are multivolume.
- Sufficient space is available for each component.
- KSDS data sets are reusable or empty.
- Key range data sets are empty.
- The data set is in the catalog structure, if required for the COPY.
- The data set has no alternate indexes (AIXs) or paths defined over it (except for a single path defined directly over the base cluster).

Non-VSAM Preallocation

An existing data set qualifies as a preallocated target for data set COPY if the Format 1 DSCB names match, the complete data set is available on target volumes, and:

- For single volume target qualification, the DSORG is PAM or the DS1VOLSQ is 1 and the last volume flag is on.
- For multivolume target or single volume target with the last volume flag off, the data set is cataloged in the standard order of search. All volume serial numbers returned by a LOCATE on the data set are in the output volume list. (Candidate volumes for which no Format 1 DSCBs exist are acceptable.)

The preallocated data set is usable if:

- The user is authorized to update the target data set.
- The DSORG matches.
- For BDAM data sets, the target does not exist if COPY is done using the IEHMOVE utility. If REBLOCKADDRESS is specified for the data set, preallocated targets are allowed.

- For unmovable data sets, extents match exactly when you copy to a like device without specifying the FORCE keyword.
- For movable data sets or unmovable data sets with FORCE, sufficient space is present in the target data set.
- For PAM data sets, the target directory can contain all source members and aliases.
- For preallocated standard user label data sets, the target has more than one extent when the source data set has more than one extent.

If a VSAM or non-VSAM preallocated data set is determined to be unusable, message ADR439E is issued, and the COPY is stopped for that data set only. No attempt is made to clear or alter the target data set if:

- The source data set is empty.
- The DSORG is not supported.
- The target is preallocated but not empty.

Message ADR363E is issued to inform the user.

Specifying Multiple Target Volumes

When multiple target volumes and REPLACE are specified, more than one existing data set may qualify as a preallocated target. The first existing data set that qualifies as a preallocated target when you use the OUTDDNAME/OUTDYNAM list order is used as the target data set. For non-VSAM data sets that require catalog verification, the catalog standard order of search determines the data set used as the preallocated target.

The device selection criteria used for data set COPY (like, same, then unlike device preference) is not observed if a preallocated data set target is used.

How Keywords Work with Preallocated Targets

When you use preallocated data sets with COPY, some keywords have a different effect and others have no effect at all.

ALLEXCP and ALLDATA: If ALLEXCP or ALLDATA is specified and the target is a like device, the data in the source data set is moved to the target.

CATALOG and RECATALOG: Data set COPY cannot change the catalog or the catalog status (cataloged or uncataloged) of the preallocated target data set. As a result, the CATALOG and RECATALOG keywords have no effect on preallocated target data sets. (Similarly, passwords and expiration dates of preallocated data sets cannot be changed.)

NOPACKING: The NOPACKING keyword is effective only for partitioned data sets. If NOPACKING is specified for preallocated partitioned data sets, the preallocated target must reside on the same or like device. Processing is stopped for the data set if the target resides on an unlike device. The target is not deleted and reallocated.

PROCESS(SYS1): Data set COPY permits moving SYS1 data sets to a preallocated target.

PERCENTUTILIZED: The PERCENTUTILIZED keyword also has no effect when the target data set is preallocated.

REBLOCK: If a data set qualifies for reblocking when REBLOCK is specified (SAM and PAM only) and a preallocated target is used, the target block size is overwritten with one of the following values:

- The source data set block size
- A DFDSS-selected block size
- A user-selected block size passed by the installation reblock exit
- A system-determined block size.

The block size used is determined by the installation reblock exit return code and the reblockable indicator in the Format 1 DSCB.

If REBLOCK is not specified, the target BLKSIZE of a non-VSAM data set is overwritten with the source BLKSIZE.

If a partitioned data set is specified with both NOPACKING and REBLOCK keywords, the data set is not reblocked.

RENAMEUNCONDITIONAL: RENAMEUNCONDITIONAL has no effect for preallocated target data sets.

Note: Data set COPY does not scratch and reallocate preallocated data sets.

Defining RACF Profiles

Figure 27 shows how profiles should be defined, when you move data without specifying MENTITY with COPY and have RACF Release 1.5 or later installed.

	No RENAME No DELETE	No RENAME DELETE	RENAME No DELETE	RENAME DELETE
Unprotected source	No profile to define.	No profile to define.	No profile to define.	No profile to define.
Generic profile for source	Leave source profile. No target profile to define.	Leave source profile. No target profile to define.	Leave source profile. ⁷	Leave source profile. ⁷
Discrete profile for source	Leave source profile. Define discrete target profile. ⁸	Delete source profile. Define discrete target profile. ⁸	Leave source profile. Define a discrete target profile. ⁸	Delete source profile. Define a discrete target profile. ⁸

Figure 27. Defining Profiles—with RACF 1.5 or Later Installed

For a preallocated target data set, the RACF status is checked. If the source is protected but the target is not, DFDSS protects the target data set with a discrete RACF profile.

⁷ Issue RACHECK to determine if a generic or discrete profile already exists for the renamed target. If so, there is no profile to define. If not, a discrete profile is defined.

⁸ When a discrete profile exists for a preallocated target data set, that profile is not deleted and no new profile is defined.

Moving Data Sets with Special Requirements

Some data sets require special treatment when they are moved. The following sections discuss some considerations for moving these special data sets.

Moving Undefined DSORG and Empty Data Sets

To copy a data set with undefined DSORG, ensure that the PROCESS(UNDEFINEDSORG) keyword is specified. In addition, the selected target volume must be of like-device type or of an unlike-device type with equal or greater track capacity.

To copy an empty data set (DS1LSTAR=0, DS1EXT1=0, DSVLSEQ=1), an EOF record must exist in the first track of the source data set. To copy an empty data set, use ALLDATA or ALLEXCP to force a copy of all allocated space in the data set. Ensure that the selected target volume is of like-device type, and the data is not being reblocked.

Moving System Data Sets

Some system data sets do not require movement, either because they are allocated during system generation or because they are built at IPL time. Other system data sets, however, can be moved by DFDSS for various reasons.

Unless excluded, system data sets are copied. However, they generally remain open while the system is running and cannot be scratched or uncataloged because DELETE and UNCATALOG apply only to data sets not in use.

Frequently system data sets are prefixed with a high-level qualifier of SYS1. Previously, DFDSS placed restrictions on the processing of SYS1 data sets, however, DFDSS Version 2 Release 5.0 removes some of these restrictions by using the PROCESS(SYS1) keyword. The PROCESS(SYS1) keyword can be used for a data set COPY of a SYS1 data set to move it to a preallocated target or to copy it with the DELETE option. PROCESS(SYS1) does not apply to VTOCIX or VVDS.

To limit the use of PROCESS, you need to set up a RACF facility class profile. Refer to *Resource Access Control Facility (RACF) Security Administrator's Guide* for more information on RACF facility class profiles.

Note: The PROCESS(SYS1) option does not lift the restrictions on the processing of volume VVDSs or VTOC indexes.

When the PROCESS(SYS1) keyword is not specified, you cannot move system data sets the way you normally move data sets with DFDSS. In order for DFDSS to move system data sets, you must do one of the following:

- Dump the data sets, and then restore them to a different volume.
- Copy the data sets to a different volume and then catalog them in a different catalog.

Note: When a data set COPY is used to copy the following data sets, space is defined for the target data set but no data is copied:

Model DSCBs
Page and swap data sets
SYS1.STGINDEX.

Moving User Catalogs

When you copy an integrated catalog facility user catalog, **DELETE** must be specified, and an input volume and the **RENAMEU** keyword must **not** be specified. You must specify the fully-qualified name of the user catalog in the **INCLUDE** parameter. In any processor in the complex, there should be no other jobs executing that access the user catalog being moved; otherwise, the **COPY** operation might fail or the copied catalog might contain errors.

The operator is prompted to enter the password for the system's master catalog. If the master catalog was defined with **ATTEMPTS=0**, you must **RACF**-protect the master catalog so that password prompting is bypassed.

For integrated catalog facility clusters that will be copied using the **IDCAMS** utility, a preallocated target data set will be renamed using a **DFDSS** generated temporary name. This allows dynamic allocation and **IDCAMS REPRO** to work, because both are currently undirected in catalog usage.

An integrated catalog facility cluster with **AIX** and path associations does not use a preallocated target if **DELETE** is specified. No search is made for existing data sets in this case. An integrated catalog facility **AIX** cannot use a preallocated target. No search is made for existing data sets when copying an **AIX**.

With **MVS/XA DFP Version 2 Release 3** or a later release, user catalog aliases are automatically redefined after the copy. The **LOCK** attribute of an integrated catalog facility user catalog is preserved during the copy. See *Catalog Administration Guide* for a description of the **LOCK** attribute and the proper access authority.

Moving CVOLs: When you move a **CVOL**, **DELETE** must be specified and **RENAMEU** must not be specified. The **CATALOG** parameter is used as a default. You must rebuild catalog aliases afterward.

Note: **DFDSS** cannot be used to move an active **VSAM** master catalog, the **VVDS**, or the **VTOCIX**.

Moving Multivolume Data Sets

When moving multivolume data sets, you should use the **ALLMULTI** keyword on the **COPY** command. **ALLMULTI** allows you to move multivolume data sets in their entirety, even if you do not specify all the volumes on which the data set resides. To use **ALLMULTI**, however, you must specify at least one primary input volume on the **LOGINDDNAME** or **LOGINDYNAM** volume lists. The data set must also be cataloged in order to qualify for **ALLMULTI** processing.

A multivolume data set can be copied to a single volume or to multiple volumes. For a multivolume data set with standard user label, only the standard user label on the first volume is copied to the target volumes.

Moving VSAM Data Sets

When you move a **VSAM** data set, **DELETE**, **RENAMEU**, or **RECATALOG** (to a different catalog than the source catalog) must be specified if **REPLACE** is not specified. If **REPLACE** is specified and a preallocated target is not found, then **DELETE**, **RENAMEU**, or **RECATALOG** (to a different catalog than the source catalog) must be specified for the data set to be processed.

You can move the base cluster, all associated alternate index clusters, and paths by using the SPHERE keyword with the COPY command. "Restrictions for COPY" on page 114 lists considerations for moving VSAM spheres.

If you do not use the SPHERE keyword and the base cluster has associated alternate index clusters, only the base cluster is moved as follows:

- If you specify DELETE, only the base cluster is moved, but the alternate index cluster continues to be related to the base cluster.
- If DELETE is not specified, a second copy of the base cluster is created, and the alternate index cluster continues to be related to the original base cluster.

To move an alternate index cluster, DELETE must be specified on the COPY command. Only the alternate index cluster is moved and it continues to relate to its base cluster. An AIX cannot be moved by itself to or from an SMS-managed volume.

For an empty VSAM data set (zero data relative block address or zero record count), the data set is defined on the target volume but is not copied. Message ADR474W is issued for the data set.

During COPY, you can specify the password of the catalog instead of the password of each data set. If you specify the passwords of both the data set and the catalog, the data set password is used to check authorization.

Note: DFDSS does not preserve candidate volumes during COPY processing.

Moving a PDSE

In DFDSS Version 2 Release 5, the COPY command can be used to move a PDSE. The new CONVERT keyword along with the new PDSE and PDS subkeywords can be used with COPY to convert a PDS to a PDSE and vice versa.

The PDSE cannot be moved by using releases prior to DFDSS Version 2 Release 5, nor can they be moved on a system using releases prior to MVS/DFP Version 3 Release 2 and MVS SP Version 3 Release 1.0E.

Moving a Damaged PDS

Partitioned data sets are monitored by DFDSS during track packing for conditions that are not normal. The following conditions are detected and reported:

- Missing high key entry in the PDS directory
- Missing directory EOF
- Invalid member start TTR
 - TTR points before directory EOF
 - TTR points after end of data set.
- Missing member EOF (each member of a partitioned data set is normally ended by an EOF record)
- Invalid note or notelist TTR
 - Note pointing before the start of member data
 - Note pointing after the member EOF
 - Note pointing past the last valid record on a track
 - Note pointing to record 0 of a track.

DFDSS notes all these conditions with a message.

During track packing, DFDSS repairs missing high key directory entry, missing directory EOF, and missing member EOFs.

Invalid start TTRs prevent DFDSS from track packing data for that member. DFDSS translates all valid note and notelist TTRs during track packing.

Damaged partitioned data sets can be moved to same or like device target volumes by using the NOPACKING keyword. This results in an exact track-for-track image of the source data set. Obviously, no compression is performed in this case.

Moving Unmovable Data Sets

When copying unmovable data sets to like devices, DFDSS places them at the same track locations on the target volume under the following conditions:

- The operating system is MVS/XA DFP Version 2 Release 1 or later release.
- The target volume has an indexed VTOC.
- The space where the unmovable data would be placed is available.

If any of these conditions do not exist, you must specify the FORCE keyword to move the data set. FORCE enables DFDSS to treat the unmovable data set as movable and also move it to an unlike device. Because DFDSS places the data set in any available location when FORCE is specified, use FORCE with caution.

If some data sets have CCHHR (cylinder, cylinder, head, head, record) location-dependent data and you are using FORCE, you should exclude these data sets with the EXCLUDE keyword to prevent DFDSS from moving location-dependent data sets.

Another way to position data sets in a specific location on a volume is to allocate all space on the target volume except where you plan to place the unmovable data sets. Then move the unmovable data sets with FORCE and afterwards scratch the dummy space allocation.

Moving ISAM Data Sets

When an ISAM data set is being moved, the target data set has the prime, index, and overflow areas combined into one allocated area, except when the target data set can be allocated to the same location as the source data set. You must move an ISAM data set to a volume with an indexed VTOC. To guarantee a successful allocation, you should make certain that there is no other activity on this volume.

For a single-volume ISAM data set to be selected, either the last-volume indicator must be on or the data set must be cataloged.

Moving BDAM Data Sets

When TTR-organized BDAM data sets are copied to unlike devices, the target device must have a larger track capacity than the source, and the target data sets must not be preallocated.

DFDSS updates BDAM data sets organized by relative byte address (RBA) when copying them. The only other kind of data set with location-dependent data that DFDSS can copy to an unlike device is a PAM data set with NOTE lists. Also, DFDSS cannot be used to copy any data sets containing location-dependent data to like devices unless they are defined as unmovable and FORCE keyword criteria are

met. DFDSS updates the block reference count of dummy records contained within RBA-organized BDAM data sets when copying them to unlike devices of smaller track capacity.

Moving GDG Data Sets

For generation data group (GDG) data sets, filtering on generations is supported. You can specify generation names in relative generation number, `dsn(n)`, with the INCLUDE and EXCLUDE keywords. During COPY, if you catalog the GDGs in a different catalog or you rename them, you must define ahead of time the target GDG base name because the source GDG base name is unusable.

Moving Generation Data Sets to SMS Volumes

An SMS-managed generation data set (GDS) can be in one of three states:

- DEFERRED
- ACTIVE
- ROLLED-OFF.

When copying a GDS to an SMS-managed volume and the data set is not preallocated, DFDSS allocates the target GDS as follows:

- If DELETE is specified and RENAMEU is not specified, the target GDS is allocated with the same state as the source GDS.
- If the TGTGDS keyword is specified, the appropriate status is assigned to the data set. The requested target status must not violate rules of the generation data group.

The following chart describes the default situation for DFDSS to allocate the SMS-managed GDG data set (MOVE refers to COPY with the DELETE keyword specified):

Target Environment	Source Environment	Source Status	DFDSS Function	TGTGDS Default
SMS	Non-SMS	Cataloged	COPY	DEFERRED
			MOVE	ACTIVE
		Not cataloged	COPY	DEFERRED
			MOVE	DEFERRED
	SMS	ACTIVE	COPY	DEFERRED
			MOVE	ACTIVE
		DEFERRED	COPY	DEFERRED
			MOVE	DEFERRED
ROLLEDOFF	COPY	DEFERRED		
	MOVE	ROLLEDOFF		

If the data set is preallocated, the state of the target GDS is not altered. During a move operation, you can use the TGTGDS keyword to alter the target status only if the data set is also being renamed.

Moving Generation Data Sets to Non-SMS Volumes

A non-SMS-managed generation data set (GDS) can be in one of two states:

- Cataloged
- Uncataloged.

When you copy a GDS to a non-SMS volume, the state of the GDS is determined only by the CATALOG or RECATALOG keywords.

Moving Volumes

You can move volumes logically or physically with DFDSS.

As with moving data sets, if the output volume has unexpired data sets, you can stop the COPY operation or write over the unexpired data sets.

Logical Volume COPY

To move a volume logically, use the DATASET keyword, specify input volumes with LOGINDDNAME, LOGINDYNAM, INDDNAME, or INDYNAM, and use INCLUDE(**). This method of moving volumes allows you to move data between unlike devices (for example from a 3350 to a 3380).

Some data sets require special processing when you move them (see also “Moving Data Sets with Special Requirements” on page 79). For example:

- Unmovable data sets
- Multivolume data sets
- Integrated catalog facility catalogs
- Data sets beginning with SYS1
- Data sets used by device-dependent application programs.

If you use DATASET COPY to move a volume and the volume contains such data sets, you must move them in the proper sequence to achieve the expected results.

You may want to process unmovable data sets first so you can place them at the same track location on the target device. User catalogs should be moved only when quiesced. In addition, you should not move catalogs together with the data sets cataloged in them.

Note: Some data sets are not eligible for movement by DFDSS (for example, VSAM data sets not cataloged in integrated catalog facility catalogs). Others might require special parameters (for example, unmovable data sets).

Physical Volume COPY

If you do not specify DATASET or TRACKS on the COPY command, the COPY defaults to FULL and moves the volume physically. You must also specify INDDNAME or INDYNAM to indicate the source volume and OUTDDNAME or OUTDYNAM to indicate the target volume. FULL volume COPY can move data only between like devices of equal or greater capacity (for example, from a double capacity 3380 model to a double or triple capacity 3380 model).

Notes:

While it is true that in most cases you can physically move volumes only between like devices, you can move:

1. From an IBM 3330 Model 1 to an IBM 3330 Model 11
2. From an IBM 3340 with 35-megabyte capacity to an IBM 3340 with 70-megabyte capacity or to an IBM 3344
3. From a smaller capacity IBM 3380 model to a larger capacity IBM 3380 model
4. From a minivolume or virtual volume to a real volume of like device type, and vice versa, device capacity permitting.

With tracks COPY you can move data:

5. From an IBM 3330 Model 11 to an IBM 3330 Model 1
6. From a 70-megabyte 3340 to a 35-megabyte 3340
7. From a larger capacity IBM 3380 model to a smaller capacity IBM 3380 model, provided the range of data to be processed falls within the capacity of the output device.

When you perform a full-volume copy to a DASD that is shared between multiple systems, the DASD should be offline to all systems except the one performing the copy.

When you use physical volume COPY, you can specify COPYVOLID (see "Considerations for Moving Volumes in an SMS Environment" on page 89 for SMS considerations). If you specify COPYVOLID, the volume serial number of the source volume is copied to the target volume. This ensures that RACF profiles and catalog entries for the data sets on the volume have the correct volume serial number.

Note: However, changing the volume serial number of a volume causes the operating system to demount the target volume at the end of the COPY operation. To use the target volume, you must demount the source volume and mount the target volume.

VTOC Considerations

When moving volumes, you need to make certain that the VTOC on the source volume is large enough to hold entries for all the data sets to be placed on it. A DFDS physical volume COPY copies the VTOC from the source to the target volume. If you determine that this VTOC is too small for the target volume (for example, if you are moving from a standard capacity 3380 model to a double capacity 3380 model), use ICKDSF to reinitialize the target volume with a larger VTOC. Then use DATASET COPY to move the data sets to the target volume.

If you use DFDS Version 2 Release 3 or a later release to perform a physical volume COPY from a smaller capacity to a larger capacity volume on a like device, the Format 5 DSCB is updated to reflect the size and number of the allocated/free DSCBs of the target volume. DFDS sets a bit on in the Format 4 DSCB of the target volume that indicates that the associated Format 5 DSCB needs an update. This update occurs the next time a data set is allocated to the volume. If the source volume has an indexed VTOC, the target volume's corresponding Format 4 indexed-VTOC bit is turned off. The actual VTOC index data set is copied. DFDS Version 2 Release 3.0 causes the Format 5 DSCB to be updated and calls ICKDSF to rebuild the indexed VTOC.

Moving Volumes to Like Devices of Equal Capacity

When you move volumes to like devices of equal capacity, you can use logical or physical COPY.

Moving Volumes to Like Devices of Greater Capacity

If you allocate a VTOC on a larger capacity volume, it is preserved when moving from a smaller capacity volume of a like device. DFDSS uses this VTOC when copying the source data sets. When the full-volume COPY is complete, additional data sets can be placed on the target volume.

Moving Volumes to Unlike Devices

When moving data between unlike devices, you must use logical processing. If you specify DATASET with COPY, DFDSS does a logical COPY. To copy all the data on a volume logically, you also need to specify input volumes with LOGINDDNAME, LOGINDYNAM, INDDNAME, or INDYNAM. (LOGINDDNAME or LOGINDYNAM is required if you are also using ALLMULTI.)

Considerations for Moving Data Sets in an SMS Environment

When moving data sets and volumes in an SMS environment, there are some special considerations.

As discussed earlier, volumes within an SMS environment may or may not be managed by SMS. The following sections discuss how you can use the COPY command to move these data sets.

Moving SMS-Managed Data Sets

As with the RESTORE command, COPY invokes the Automatic Class Selection (ACS) routines, which in turn assign or override a data set's classes.

When you use the COPY command, you are in the ACS ALLOC environment (&ACSENVIR='ALLOC'). The storage class ACS routine is executed first. If the storage class assigned is not null, the management class ACS routine and then the storage group ACS routine are executed. (See "ACS Variables Available during COPY" on page 109 for a list of variables available to ACS routines during COPY processing.)

If you do not specify otherwise, DFDSS passes the source data set's class names as input to ACS. If you want to specify storage and management class names to be passed to ACS, you can use the STORCLAS and MGMTCLAS keywords. You can use the NULLSTORCLAS and NULLMGMTCLAS keywords to pass null storage and management classes to the ACS routines.

If you do not want a data set to be SMS managed, you should specify BYPASSACS and NULLSTORCLAS.

All of these keywords work the same for COPY as they do for RESTORE (see "Changing Storage Class with RESTORE" on page 65 and "Changing Management Class with RESTORE" on page 66).

Target Volume Selection

In some cases, you might want to move data sets off some volumes and onto others (for example, because of performance problems or because you want to put data sets on some new, empty volumes you have just added to a storage group). However, in an SMS environment you generally allow the system to place data sets for you. If for some reason you want to control the placement of the data sets, you must take special steps.

If you use `OUTDDNAME` or `OUTDYNAM` to specify a volume list, the volume serial numbers are passed as input to the ACS routines. Depending on how your ACS routines are written, this input might or might not be used in determining where to place the data set.

One way to guarantee that data sets go to particular volumes is to write your storage group ACS routine such that data sets are moved to the volumes you select.

Alternatively, if a data set's storage class has the guaranteed space attribute, the data set is placed on the user-specified volumes if the volumes reside in the same storage group and ACS selects that storage group for the data set. By using `BYPASSACS` and `STORCLAS`, you can ensure that the storage group selected contains the volumes you specify with `OUTDDNAME` or `OUTDYNAM`. However, for this procedure to work, your storage group ACS routine must use storage class to determine the storage group for a data set. This allows you to determine which storage class to specify with `STORCLAS` to ensure that the storage group containing the volumes specified with `OUTDDNAME` or `OUTDYNAM` is selected.

Changing Storage Class with COPY

You can use the `STORCLAS` keyword to specify a storage class name for DFDSS to pass to ACS. You can specify `NULLSTORCLAS` if you want DFDSS to pass a null storage class to ACS.

Note: RACF checks if the `RESOWNER` of a given data set is authorized to define the data set with the specified `STORCLAS`. Ensure that the `RESOWNER` of the data set has the proper authority to use the indicated storage class.

Using `STORCLAS` does not guarantee that the data set is assigned the storage class you specify. To be certain that the storage class you specify is assigned to the data set, you must specify `BYPASSACS`. In this case, using `BYPASSACS` causes the storage class and management class ACS routines to be bypassed, so the data set is assigned whatever you have specified with `STORCLAS` or, if you don't use `STORCLAS`, whatever the source data set's storage class is. However, when you use `BYPASSACS`, you should make certain that the storage class you specify is valid or you will get an error.

You can also use **STORCLAS** and **BYPASSACS** to move data sets into a newly defined storage class. For example, suppose you want to combine all your storage classes except two into one new, large storage class. You can code the following:

```
COPY -  
  DATASET(INCLUDE(**) -  
          BY(STORCLAS,NE,(SCNAME1,SCNAME2))) -  
  STORCLAS(SCNAME3) -  
  BYPASSACS(**) -  
  DELETE
```

If you specify **NULLSTORCLAS** and **BYPASSACS** together, the target data set is non-SMS managed.

Changing Management Class with COPY

In addition to influencing a data set's storage class with **COPY**, you can also give **ACS** input for assigning or overriding the data set's management class. By specifying **MGMTCLAS**, you can pass a management class name to **ACS** and, as with **STORCLAS**, **ACS** ignores it, assigns it to the data set, or uses it in combination with other things to determine the data set's management class. By specifying **NULLMGMTCLAS**, you can pass null management class to **ACS**, which might or might not assign a management class to the data set.

Note: **RACF** checks if the **RESOWNER** of a given data set is authorized to define the data set with the specified **MGMTCLAS**. Ensure that the **RESOWNER** of the data set has the proper authority to use the indicated management class.

Also just as with **STORCLAS**, you can use **MGMTCLAS** with **BYPASSACS** to make certain that the data set is assigned the management class you specify. If you use **BYPASSACS**, you should be certain that the management class you are giving the data set is valid. If it is not, the data set is not moved. You must be authorized to use **BYPASSACS** and the management class you specify with **MGMTCLAS**.

When you use **COPY** to influence or assign a management class, you also need to be careful that the data set resides in a storage group capable of providing for the management class attributes associated with the management class you specify. For instance, if a data set has a management class that makes it eligible for migration, it needs to reside in a storage group on which **DFHSM** does migration. Otherwise, the data set will never be migrated. For this reason, you might have to change the storage class along with the management class to ensure that the data set resides on volumes that can accommodate its management class.

Moving Non-SMS-Managed Data Sets

If the data set you are moving is to be non-SMS managed, you should use the **NULLSTORCLAS** and **BYPASSACS** keywords on the **COPY** command. By using these keywords, you can make an SMS-managed data set non-SMS managed. Using **NULLSTORCLAS** and **BYPASSACS** also prevents a non-SMS-managed data set from becoming SMS managed.

Considerations for Moving Volumes in an SMS Environment

When you move volumes physically, you must specify COPYVOLID if the source is an SMS volume.

You must consider several factors when copying volumes in an SMS environment. Before you start to copy a full volume, you must ensure that the status of the target volume is synchronized with its environment. For example, if the target volume is a non-SMS volume, then the volume must not be defined in a storage group. Conversely, if the target volume is an SMS volume, then the volume must be defined in a storage group. Finally, if the target volume is SMS, then SMS must be active for the full-volume copy operation.

Chapter 9. Converting Data to and from SMS Management

DFDSS is the primary tool for converting data to and from SMS management. Conversion can be done with or without data movement.

This chapter is organized as follows:

- **“Evaluating Conversion to SMS Management”** discusses the advantages and disadvantages of the two types of conversion.
- **“Conversion by Data Movement”** on page 92 describes how to use DFDSS to convert data sets to SMS management with the COPY and DUMP/RESTORE commands.
- **“Conversion Without Data Movement”** on page 93 describes how to use DFDSS to convert volumes to SMS management with the CONVERTV command.
- **“Special Data Set Requirements for Conversion to SMS”** on page 96 describes some of the data sets that have special requirements for conversion to SMS management.
- **“Converting from SMS Management without Data Movement”** on page 98 describes how to use the CONVERTV command to convert volumes from SMS management.
- **“Special Data Set Requirements for Conversion from SMS”** on page 98 describes some of the data sets that have special requirements for conversion from SMS management.

Evaluating Conversion to SMS Management

One of the first things to consider when you convert data to SMS management is whether to convert data sets with or without data movement. If you have SMS volumes with sufficient free space, you can convert data sets by simply moving them from non-SMS volumes to SMS volumes. The same is also true if you are converting data from SMS-management. Converting data sets to SMS management by data movement is often preferable because it allows the system to place the data sets for you. This ensures that the data sets are placed on volumes in storage groups that can meet the availability and performance requirements of the data set.

If, however, you do not have sufficient free space on your SMS volumes to convert by data movement, you might have to convert data sets without data movement. The drawback to this method of conversion is that it does not allow the system to place data sets for you. You must ensure that the availability and performance requirements of the data sets can be met by the storage group in which you place the volume.

Regardless of how you convert to SMS management, you must determine the eligibility for conversion of your data sets and volumes prior to conversion.

Data Sets Ineligible for Conversion to SMS

The following data sets cannot be converted to SMS management:

- ISAM data sets
- Absolute track allocation data sets
- Unmovable data sets
- Model DSCBs
- CVOLs
- Non-integrated catalog facility VSAM catalogs
- VSAM data sets not cataloged in an integrated catalog facility catalog
- Uncataloged, multivolume data sets
- SYS1 storage index data sets (SYS1.STGINDEX)
- Page and swap data sets
- BDAM with OPTCD=A
- VVDS/VTOCIX⁹.

(Using CONVERTV with the SMS and TEST keywords identifies ineligible data sets without actually converting any data.)

Volume Eligibility for Conversion to SMS

A volume is eligible for conversion if it meets the following conditions:

- It is a DASD volume.
- It is permanently mounted and accessible online.
- It has an indexed VTOC.
- It is defined in an SMS storage group in an active configuration.

Conversion by Data Movement

By using logical data set COPY or DUMP/RESTORE, you can move data sets between non-SMS and SMS volumes. When moving data sets to SMS volumes, COPY and RESTORE invoke ACS to assign classes to the data sets. This type of conversion to SMS allows the data sets to be placed on the most appropriate SMS volume.

Converting to SMS Management by Data Movement

When moving data sets to SMS-managed volumes, you can use the COPY or RESTORE commands. You can specify storage and management class names with the STORCLAS and MGMTCLAS keywords. You can also specify output volumes with OUTDDNAME and OUTDYNAM. DFDSS passes the class names and volume serial numbers to ACS, which might use them in determining the classes and placement of the data set.

This method of converting data sets to SMS management is similar to moving data sets in a SMS environment as described in "Considerations for Moving Data Sets in an SMS Environment" on page 86.

If you use COPY or RESTORE on a data set that is ineligible for SMS management, DFDSS puts it on a non-SMS volume. However, if you specify STORCLAS and BYPASSACS with COPY or RESTORE for a data set that is ineligible for SMS management, the COPY or RESTORE fails.

⁹ VVDS/VTOCIX data sets can be SMS managed, but DFDSS cannot be used to convert them.

JOB CAT and STEPCAT are not allowed when you process SMS-managed data sets. For data sets cataloged outside the standard order of search, use the INCAT keyword on COPY or DUMP to identify what catalog to search. Use the ALLMULTI keyword on COPY or RESTORE to convert multivolume data sets. This allows you to specify only the volume with the primary component on the LOGINDD or LOGINDY parameter. You can use the SPHERE keyword on COPY or RESTORE to convert entire VSAM spheres.

Converting from SMS Management by Data Movement

To take a data set out of SMS management with COPY or DUMP/RESTORE, you should specify the BYPASSACS and NULLSTORCLAS keywords. This forces DFDSS to make the data set non-SMS managed.

Conversion Without Data Movement

Conversion without data movement is divided into two phases: conversion of data sets and conversion of volumes. Convert data sets and the volumes they reside on without moving data by using the DFDSS CONVERTV command. You need to set up RACF facility class authorization to limit the people who can use the CONVERTV command. When you use the CONVERTV command to perform conversion, it attempts to convert all the data sets on the volume. After all the data sets are processed, the volume is placed in one of the following three states:

- **CONVERTED**—the volume and its data sets are converted to SMS management. A volume can be placed in this state with the CONVERTV command and the SMS keyword.
- **INITIAL**—new allocations cannot be made to the volume and, although users can access their data sets, the data sets cannot be extended to other volumes. A volume may be placed in this state because you have used the CONVERTV command with the PREPARE keyword to reduce activity to the volume prior to conversion. A volume may also be placed in this state if you are attempting to convert it and it contains data sets that are not eligible for conversion.
- **NONSMS**—the volume and its data sets were taken out of the CONVERTED or the INITIAL state and are non-SMS managed. A volume can be placed in this state with the CONVERTV command and the NONSMS keyword.

Simulating Conversion

Before you convert a volume to SMS management, you should simulate the conversion to ensure that all the data sets on the volume are eligible for conversion to SMS. In addition, simulating conversion shows you the classes ACS would assign to the data sets eligible for conversion.

You can simulate conversion by using the CONVERTV command with the SMS and TEST keywords. If the volume is ineligible for conversion, the data sets on the volume are still examined to determine their eligibility for conversion (provided the volume is permanently mounted and online).

When you use CONVERTV SMS TEST, you are in the ACS CONVERT environment (&ACSENVIR = 'CONVERT'). Only the storage class ACS routine is executed. (See "ACS Variables Available during RESTORE and CONVERTV" on page 110 for a list of variables available to ACS routines during CONVERTV processing.)

Simulated conversion creates a report that identifies data sets ineligible for conversion. For a sample of this report, see “SMS Report” on page 95. Note that this report indicates the management, data, and storage class that would be assigned to each data set. Careful analysis of this report allows you to determine if your ACS routines will assign appropriate classes to the data sets before doing the actual conversion.

Data sets unsupported by SMS should be moved off the volume prior to actual conversion. Other data sets (for example, uncataloged data sets) can be made eligible for conversion by taking some action (for example, using the CATALOG keyword to catalog uncataloged data sets).

If you have ineligible data sets on a volume and you run CONVERTV with SMS, DFDSS still converts the eligible data sets on the volume. It then puts the volume in the INITIAL state. You must then take action to make the ineligible data sets eligible for conversion or move them off the volume. Once all the ineligible data sets are dealt with, you can run CONVERTV again to complete the conversion.

Preparing a Volume for Conversion

Before you convert a volume to SMS management, you should reduce the amount of activity to the volume being converted. CONVERTV with the SMS keyword automatically places the volume in a state of reduced activity before doing the actual conversion. You might, however, want to reduce activity without doing the actual conversion (for example, if you want to simulate conversion). This is done by specifying the PREPARE keyword on the CONVERTV command.

Specifying PREPARE prevents data sets from extending onto other volumes and new allocations from being made on the volume. However, users can still access the data on the volume from either the SMS system or a system sharing the volume.

When you use PREPARE, a report is generated that tells you the volumes that have been placed in the INITIAL state. If any of the volumes are ineligible to be placed in the INITIAL state, the report also lists them and the reason they were ineligible (for example, they did not have an indexed VTOC or were offline).

If you use the TEST keyword with PREPARE, you still get the report indicating which volumes would and would not be placed in the INITIAL state, but the PREPARE is not actually performed. You can then take some action to make those volumes eligible or simply not run PREPARE against those volumes.

CONVERTV with the NONSMS keyword reverses the effect of PREPARE and takes a volume out of the INITIAL state.

Converting to SMS Management without Data Movement

To convert data to SMS management, use the CONVERTV command with the SMS keyword. (Because SMS is the default for CONVERTV, you can simply specify CONVERTV.) Of course, the volume and all its data sets must be eligible for conversion to run CONVERTV with SMS successfully.

If the volume is eligible for conversion, the INITIAL indicator on the volume is set. This means the volume is in the same state as when you specify CONVERTV with the PREPARE keyword. Once a volume has its INITIAL indicator set on, DFDSS begins processing the data sets on the volume.

If a data set is eligible for conversion, ACS is called to assign SMS classes to the data set. When you use CONVERTV with SMS, you are in the ACS CONVERT environment (&ACSENVIR = 'CONVERT'). The storage class ACS routine is executed first. If the storage class assigned is not null, the management class ACS routine is executed. (See "ACS Variables Available during RESTORE and CONVERTV" on page 110 for a list of variables available to ACS routines during CONVERTV processing.)

RACF checks if the RESOWNER of a given data set is authorized to define the data set with the given STORCLAS and/or MGMTCLAS. Ensure that the RESOWNER of the data set has the proper authority to use the indicated storage class and/or the indicated management class.

If no errors occur, the catalog entry for the data set is updated to include the classes. For VSAM data sets, the VVDS entry is updated to indicate that it is SMS managed. For non-VSAM data sets, a VVDS entry is added that indicates the data set is SMS managed. After the catalog and VVDS updates and additions are successfully made, the data set's VTOC entry is updated to indicate it is SMS managed.

If a VSAM data set has the guaranteed space attribute, a check is done to verify the eligibility of its candidate volumes. If this check fails, the data set is not converted to SMS management. Non-VSAM data sets have candidate volumes in their catalog entries made non-specific.

When DFDSS encounters a data set that is not eligible for conversion, it does not process the data set and it continues to process other data sets on the volume. The only time conversion of data sets stops is when an error prevents ACS from returning class information for any data set.

DFDSS does not mark a volume as SMS managed until all the data sets on the volume are SMS managed. If a volume contains data sets that are ineligible for conversion, you must take some action to make them eligible or move them off the volume. You can then resubmit the CONVERTV command to convert any data sets not already converted and mark the volume as an SMS volume.

On subsequent invocations of CONVERTV, DFDSS processes only those data sets not yet converted unless you specify the REDETERMINE keyword. If REDETERMINE is specified, all the data sets, including those already converted, are reprocessed. You may wish to do this if your ACS routines changed since the last time you ran CONVERTV on the volume. REDETERMINE causes DFDSS to reprocess all the data sets on the volume, even the ones that were previously converted.

SMS Report

Figure 28 on page 96 shows a sample report generated by DFDSS during CONVERTV SMS processing.

```

1PAGE 0001 DFDSS 5665-327 DATA FACILITY DATA SET SERVICES V2R4.0 08.140 10:23
-
CONV SMS
ADR101I RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'CONV '
DDN(DASD52) -
CAT
ADR109I RI01 (01), 08140 10:24:07 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR006I (001)-SETUP(01), 08140 10:24:07 EXECUTION BEGINS
ADR860I (001)-KVSMS(01), PROCESSING BEGINS ON VOLUME D5B005
ADR873I (001)-KVSMS(01), VOLUME D5B005 IN STORAGE GROUP SG50 IS ELIGIBLE FOR CONVERSION TO SMS MANAGEMENT
ADR877I (001)-KVSMS(01), THE FOLLOWING DATA SETS ON VOLUME D5B005 WERE SUCCESSFULLY PROCESSED
0 DSCV.DDS.ESNVRS10 CATALOG: TEST.CAT.DSCV
STORCLAS: SC50 MGMTCLAS: MCCV
0 DSCV.DDS.KSNVRS10 CATALOG: TEST.CAT.DSCV
STORCLAS: SC50 MGMTCLAS: MCCV
AIX: DSCV.AIX.KSNVRS10
PATH: DSCV.PATH.KSNVRS10
0 TEST.CAT.DSCV CATALOG: TEST.CAT.DSCV
STORCLAS: SC50 MGMTCLAS: MCCV
3 DSCV.DDS.SSNRS10 CATALOG: TEST.CAT.DSCV
STORCLAS: SC50 MGMTCLAS: MCCV
0 DSCV.DDS.BDNRS10 CATALOG: TEST.CAT.DSCV
STORCLAS: SC50 MGMTCLAS: MCCV
ADR885I (001)-KVSMS(01), VOLUME D5B005 HAS BEEN SUCCESSFULLY CONVERTED TO SMS MANAGEMENT
1PAGE 0002 DFDSS 5665-327 DATA FACILITY DATA SET SERVICES V2R4.0 08.140 10:23
-ADR892I (001)-KVRPT(01), THE STATUS OF EACH VOLUME IS AS FOLLOWS
0 VOLUME FINAL STATUS REASON FOR FAILURE
-----
0 D5B005 - CONVERTED SMS
ADR013I (001)-CLTSK(01), 08140 10:25:07 TASK COMPLETED WITH RETURN CODE 0000
ADR012I DSSU (01), 08140 10:25:12 DFDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000

```

Figure 28. SMS Report

Special Data Set Requirements for Conversion to SMS

Some data sets have special requirements for conversion to SMS management. The sections below describe the special considerations for converting these data sets to SMS management.

VSAM Sphere Eligibility

A VSAM sphere is considered to be a single data set by the CONVERTV command. As a result, either all the data sets of the sphere are converted or none of them are.

If any of the following parts are ineligible for conversion, then all the components comprising the sphere are ineligible for conversion:

- Any part of the VSAM base cluster
- Components of a base cluster
- AIXs related to the base cluster
- AIX components
- Paths relating AIXs to the base cluster.

You must direct all parts of a VSAM sphere (the base cluster, base cluster components, AIXs, AIX components, and paths) to the same catalog by using an alias. If they are not directed to the same catalog, the sphere cannot be converted to SMS management. To correct this problem you can either rename the data sets in the sphere or add or delete catalog aliases and rerun the CONVERTV command.

Multivolume Data Sets

If you do not specify ALLMULTI, all volumes must be included in DDNAME or DYNAM volume lists.

Multivolume data sets are not eligible for conversion if any part of the data set resides on volumes that:

- Do not have indexed VTOCs
- Are not defined in a SMS storage group
- Are defined to a different storage group
- Are not permanently mounted and online.

In addition, if the first volume on which a data set resides is not in the volume list, the data set is ineligible for conversion.

When DFDSS encounters an extent of a multivolume data set, it processes the entire data set at that time, even though the extent might not be on the first volume of the data set.

If you specify ALLMULTI, you need to specify only the volume containing the first extent of a non-VSAM data set. For VSAM data sets with ALLMULTI specified, the first extent of each data component, index component, key range, AIX data component, AIX index component, and AIX key range must be on a volume being converted. Otherwise, no part of the sphere can be converted.

If these requirements are satisfied, DFDSS verifies that all the volumes on which the data set resides:

- Are permanently mounted and online
- Have indexed VTOCs
- Are defined to the same storage group.

If all these criteria are met, the data set is converted to SMS management.

Note: Volumes not specified in the DDNAME or DYNAM volume lists will not have the INITIAL indicator set on.

If you do not specify ALLMULTI, DFDSS determines if all the extents of the data set reside on volumes in the DDNAME or DYNAM volume list. If an extent resides on a volume not in the list, the data set is not converted and DFDSS begins processing the next data set. If all the extents reside on volumes in the volume list, DFDSS determines the eligibility of the volumes. If the volumes are all eligible, DFDSS converts the data set to SMS management. Note in this case, the INITIAL indicator is set on for each volume containing an extent of the data set.

GDG Data Sets

Generation data groups (GDGs) require special consideration with cataloging and uncataloging during SMS conversion. Uncataloged GDGs are converted to SMS management, but are left uncataloged. Messages ADR877I and ADR879I indicate NOT CATALOGED for the catalog name in the data set name lists for SMS processing.

Temporary Data Sets

Format 1 DSCBs of temporary data sets are updated to indicate uncataloged SMS status.

VTOC and VVDS

Format 1 DSCBs of the VTOC and VVDS are updated to indicate SMS management.

Converting from SMS Management without Data Movement

If you want to take volumes out of SMS management, you can use CONVERTV with the NONSMS keyword. All data sets on all volumes are eligible for NONSMS processing. After you execute this command, the volume indicators that designate the volume as a SMS volume are turned off. From then on, only non-SMS-managed data sets can be allocated to the volume.

As with the SMS keyword, you can specify the TEST keyword with NONSMS. No conversion is actually done, but a report is generated that identifies the data sets that are and are not eligible for conversion from SMS management. The report also indicates whether the volume as a whole is eligible for conversion from SMS management.

To convert a data set from SMS management, the data set's classes are deleted from its catalog entry. Non-specific volumes also are deleted from the catalog entry. For a VSAM data set, the SMS-related items are deleted from the VVDS entry. For a non-VSAM data set, the VVDS entries are deleted. After the catalog and VVDS updates and deletions are made, the VTOC entry is updated to be non-SMS managed.

Notes:

1. You cannot specify the CATALOG and REDETERMINE keywords with NONSMS.
2. A PDSE is not supported in a non-SMS environment, therefore, a volume containing a PDSE cannot be converted to non-SMS.

Special Data Set Requirements for Conversion from SMS

When being converted from SMS management, some data sets require special consideration. The following sections discuss some of the special requirements for converting data sets from SMS management.

Multivolume Data Sets

All pieces of a multivolume data set must be converted from SMS management at the same time. You can do this by using the ALLMULTI keyword. For NONSMS processing, you do not have to specify the volume on which the first extent resides when you use ALLMULTI. You need only identify one of the volumes on which the data set resides in the DDNAME or DYNAM volume list and the entire data set is converted. Those volumes not included in the volume list have their CONVERTED bits turned off, but their INITIAL bits are left on. Having the INITIAL bit on locks all allocations to the volume until all data sets residing on it are converted.

If you do not specify ALLMULTI, then you must specify all the volumes on which the data set resides in the DDNAME or DYNAM volume list.

GDG Data Sets

When you convert from SMS management, generation data group (GDG) data sets require special consideration with regard to cataloging. Data sets marked as “deferred roll in and rolled out” are uncataloged.

Temporary Data Sets

Format 1 DSCBs of temporary data sets are updated to non-SMS status.

VTOC, VTOCIX, and VVDS

Format 1 DSCBs of the VTOC, VTOCIX, and VVDS are updated to non-SMS status.

Chapter 10. Managing Space with DFDSS

You can use DFDSS to help manage your DASD space. This chapter is organized as follows:

- “**Reclaiming DASD Space**” discusses how to use DFDSS to reclaim DASD space.
- “**Consolidating Free Space on Volumes**” on page 104 discusses how to use the DEFrag command to reduce fragmentation on volumes.

Reclaiming DASD Space

You can reclaim DASD space with DFDSS in the following ways:

Releasing unused space in data sets.

Compressing partitioned data sets to consolidate unused space at the end of the data sets and then releasing the unused space.

Deleting unwanted data sets.

Combining data set extents.

Releasing Unused Space in Data Sets

The RELEASE command releases, on specified volumes, allocated but unused space from all sequential and partitioned data sets that you select with INCLUDE, EXCLUDE, and or BY criteria. For an explanation of these criteria, see “Choosing Data Sets for Processing—Filtering” on page 13. DFDSS selects only data sets that have releasable space. You can also use ISMF to build a list of data sets based on the amount of unused space and invoke DFDSS to release the unused space in them.

The RELEASE function requires MVS/XA DFP Version 2 Release 1 or later release. (ISMF requires MVS/XA DFP Version 2 Release 2 or later release.)

You should exclude, by using the EXCLUDE keyword, data sets whose DS1LSTAR (last block pointer in the Format 1 DSCB) field is not maintained in the VTOC. This can occur if you use an access method other than BSAM, QSAM, or BPAM. DFDSS does not release space for data sets whose DS1LSTAR field is 0.

The following options can help you use RELEASE more effectively:

- MINSECQTY(n)** Allows you to specify that space not be released unless the user's secondary allocation is greater than or equal to *n*. In this way, you ensure that the user can still add to the data set after the RELEASE.
- MINTRACKSUNUSED(n)** Allows you to specify that space not be released unless the number of unused tracks is greater than or equal to *n*. Without MINTRACKSUNUSED, space is released if there is one or more unused tracks in the data set.

To protect the user, DFDSS does not release any space in a data set if:

- The data set has 16 allocated extents.
- The cylinder-allocated data set has unused tracks but not an entire unused cylinder.
- The data set's name begins with SYS1, unless the PROCESS(SYS1) keyword is specified. To limit the use of PROCESS, you need to set up a RACF facility class profile.

Compressing a PDS

The COMPRESS command compresses a PDS on a specified volume. Compression removes unused space between members in a partitioned data set. This recovered space is then available for reuse at the end of the data set. Depending on the filtering criteria you specify, you can compress all the partitioned data sets or only some of the data sets. This command is useful for compressing system partitioned data sets before applying maintenance (thus avoiding certain space-related abends). You must not compress the data sets that contain DFDSS or IEBCOPY executable code.

The actual PDS compression is done in place by using the IEBCOPY utility. To prevent loss of data if the system or IEBCOPY abnormally ends during processing, you should back up your volume or data sets that meet the filtering criteria before using this command.

COMPRESS does not support processing partitioned data sets that:

- Are unmovable
- Contain track overflow records
- Have no directory.

Deleting Unwanted Data Sets

When used with the DELETE and PURGE keywords and data set filtering, a physical data set DUMP is a useful tool to delete unwanted data sets from DASD. This does not apply to VSAM data sets or to multivolume non-VSAM data sets.

On a logical data set DUMP, when using the DELETE keyword, VSAM, non-VSAM, and multivolume data sets are deleted.

Note: For a *physical* data set DUMP, cataloged VSAM data sets are not deleted with the DELETE function.

The following steps show how to delete (scratch and uncatalog) all expired data sets and all data sets that have not been referred to in the last year. The data sets are not actually moved to a dump volume.

1. JCL requirement:

```
//NOTAPE DD DUMMY
```

The above prevents moving any data sets.

2. Control statements to delete (scratch and uncatalog) all data sets not referred to in the last year:

```
DUMP INDD(111111) OUTDD(NOTAPE) -  
DATASET(BY(REFDT,LE,*, -366)) -  
DELETE PURGE
```

3. Control statements to delete all expired data sets:

```
DUMP INDD(111111) OUTDD(NOTAPE) -  
DATASET(INCLUDE(**) -  
BY(EXPDT,LT,*)) -  
DELETE
```

Note: You can modify the above example to apply to VSAM and multivolume data sets by omitting the INDD statement or specifying LOGINDD. This results in a logical data set DUMP operation.

Combining Data Set Extents

The DUMP command used with the DELETE and PURGE keywords scratches and uncatalogs the data sets from DASD after they are dumped. If you restore those data sets to the same DASD, allocation attempts to get the space for the entire data set. If the DASD volume has sufficient contiguous unused space, the allocated space will most likely be in one contiguous extent. Because unmovable data sets are not deleted, however, the volume might be fragmented, preventing a complete RESTORE for all data sets.

If you COPY those data sets to the same DASD, allocation attempts to get the space for the entire data set. If the DASD volume has sufficient contiguous unused space, the allocated space will most likely be in one contiguous extent. If you do not specify ALLDATA and ALLEXCP for sequential and partitioned data sets, only used spaces are allocated.

Note: Do not use this technique for unmovable data sets such as ABSTR allocated or ISAM.

The following steps show how to dump and delete (scratch and uncatalog) all movable non-VSAM data sets, defragment VSAM data sets, and restore all movable non-VSAM data sets.

1. Control statements to dump and delete all movable single-volume non-VSAM data sets:

```
DUMP INDD(DASD1) OUTDD(TAPE1) OPTIMIZE(3) -  
DATASET(BY((DSORG,NE,VSAM),(ALLOC,EQ,MOV),(MULTI,EQ,NO))) -  
DELETE PURGE
```

2. Control statements to defragment the volume:

```
DEFRAG INDD(DASD1)
```

3. Control statements to restore all dumped data sets:

```
RESTORE INDD(TAPE1) OUTDD(DASD1) -  
DATASET(INCLUDE(**)) -  
CATALOG
```

Consolidating Free Space on Volumes

Because of the nature of allocation algorithms and the frequent creation, extension, and deletion of data sets, free space on DASD volumes becomes fragmented. This results in:

- Inefficient use of DASD storage space
- An increase in space-related abends (abnormal endings)
- Performance degradation caused by excessive DASD arm movement
- An increase in the time required for direct access device space management (DADSM)-related functions.

By using the DEFRAG command, you can consolidate the free space on volumes and avoid this problem. The DEFRAG command relocates data set extents on a DASD volume (including IBM 3330 virtual volumes) to reduce or eliminate free-space fragmentation, and prints a report about free space and other volume statistics. Also, you can specify which data sets, if any, are to be excluded from data-set-extent relocation.

When to Run DEFRAG

Although the DEFRAG function can be run at any time on a volume, it does lock the VTOC (via RESERVE) and VVDS if it exists on the volume. The DEFRAG function also serializes on data sets via ENQ or dynamic allocation. These activities might cause excessive wait time for other jobs to update the VTOC. Therefore, times of low system activity are best for DEFRAG runs.

Data Sets Excluded from DEFRAG Processing

DFDSS does not relocate the following types of data sets in a DEFRAG operation:

- User-specified data sets (EXCLUDE)
- Data sets that do not satisfy all BY criteria
- ISAM data sets
- VSAM data sets not cataloged in an integrated catalog facility catalog
- Key range VSAM data sets
- Catalogs (system, user, and CVOLs)
- The VTOC index data set
- Page, swap, and SYS1.STGINDEX data sets
- VSAM volume data sets (VVDS)

Unmovable data sets
Data sets allocated by absolute track
Data sets that it cannot serialize for exclusive access.

Because DEFrag does not relocate these data sets, the effectiveness of a DEFrag run is affected by their presence.

DEFrag Options

You can use the following keywords to make more efficient use of the DEFrag command.

- DYNALLOC** Use dynamic allocation, rather than enqueue, to serialize the use of data sets. This allows cross-system serialization in an MVS/JES3 environment.
- FRAGI(n)** Perform a DEFrag operation only if the fragmentation index is more than *n*.
- MAXMOVE(n)** Stop the defragmentation run when *n* contiguous free tracks are assembled. If *n* contiguous free tracks already exist, no defragmentation is performed.
- WAIT(s,r)** If the volume or data set is unavailable, wait *s* seconds before retrying to obtain control of it and retry only *r* times.

To determine the fragmentation index of a volume without actually performing the DEFrag operation, code the NORUN parameter on the EXEC statement in your JCL.

General Hints

1. If you want DEFrag to perform in the shortest period of time and/or create the largest single free space extent, you should perform the first pass only. This can be controlled by coding the MAXMOVE(*n*) parameter, using a very high value (9999) for *n*. When the value is higher than DEFrag can assemble, the DEFrag run stops at the end of the first pass. For example:

```
DEFrag DYNAM(388002) MAXMOVE(9999)
```

2. Experimenting with the DEFrag FRAGI and MAXMOVE parameters will allow you to compare results when you operate on DASD with different fragmentation characteristics. The fragmentation index represents a number between 0 and 1 and can be one to three digits long. (FRAGI(333) represents 0.333 and FRAGI(3) represents 0.3. The recommended basic DEFrag parameters are to let MAXMOVE default and use FRAGI(3). To DEFrag DASD volume 388001, the following could be used:

```
DEFrag DYNAM(388001) FRAGI(3)
```

Serialization

The DEFrag command serializes access to the VTOC by issuing a RESERVE on the VTOC. This serialization must be ended via the DEQ macro before generating the ending statistics provided by message ADR213I. Therefore, another job may allocate or delete data sets on the processed volume between the time the DEQ macro is issued and the ending statistics are obtained. See *Data Facility Data Set Services: Reference* for a description of the serialization scheme.

DEFrag also serializes access to each data set before relocating a data set's extent. You will notice that LINKLIST data sets and other system data sets might remain

open but not enqueued. When defragmenting a volume with LINKLIST data sets, put them in the EXCLUDE list of DEFRAG to prevent them from being relocated.

The enqueue scheme used by DEFRAG ensures integrity for data sets on a single processor but not for data sets on DASD shared between processors (due to the use of an ENQ scope of SYSTEM). To ensure the integrity of data sets on a shared DASD, you must do one of the following:

- Vary the volume offline from all processors except the one on which DEFRAG runs. After DEFRAG finishes, you can vary the volume back online for the other processors.

For systems requiring continuous availability, this solution is not viable. The following solutions offer alternatives to varying volumes offline.

- Use GRS or a GRS equivalent to convert the scope of all enqueues with a major name of SYSDSN from SYSTEM to SYSTEMS (place SYSDSN in the GRS SYSTEM Inclusion RNL). In addition, GRS must not be allowed to convert any of DEFRAG's SYSTEMS enqueues to SYSTEM (GRS SYSTEMS Exclusion RNL), nor may it convert any of DEFRAG's RESERVEs to simple enqueues (GRS RESERVE Conversion RNL).
- If you are running on a system using JES3 with MDS enabled, you can use the DEFRAG keyword DYNALLOC to guarantee integrity for data sets on shared DASD.

You can ensure successful DEFRAG processing of volumes having a significant number of free or allocated extents by specifying appropriate SIZE and REGION parameters in the EXEC statement. If you receive a message that the region size is not large enough, specify a larger region size in the EXEC or JOB statement and rerun your job.

Note: During DEFRAG processing, a Format 1 DSCB with the unique name "SYS1.DFDSS.DEFRAG.xxxxxxx.volser.DUMMY" is used by DEFRAG to keep track of an extent when it is being relocated. This DSCB is not cataloged and automatically deleted after a successful run. If a job is cancelled or abnormally ends, this DSCB remains on the volume. After the restart, DADSM functions might fail with message IEC602. To correct this problem or to delete the "SYS1.DFDSS.DEFRAG.xxxxxxx.volser.DUMMY" entry, rerun DEFRAG on the volume.

Maximizing Track Utilization by Reblocking Data Sets

DFDSS provides a REBLOCK keyword that allows users to maximize the track usage by data sets during COPY and RESTORE. When REBLOCK is specified on a fully- or partially-qualified name(s) of a SAM or PAM data set during COPY or RESTORE, DFDSS reblocks the specified data sets using a block size determined by:

- A combination of the return code supplied by the installation reblock exit Routine (if there is any) and
- The block size selected by DFDSS.

The following types of data sets are not eligible for reblocking:

- PAM data sets with NOTELISTS
- Load modules
- SAM and PAM data sets with a variable spanned record format or with RECFM = U.

DFDSS calculates the new block size to be used using either: (1) DASD Calculation Services if MVS/XA DFP Version 2 Release 2 or a later release is installed, or (2) by the reblocking scheme of DFDSS. This new block size is then passed to the installation reblock exit if the exit has not been disabled. DFDSS determines the target block size based on the return code set by the exit.

If MVS/DFP Version 3 Release 1 or a later release is installed, the reblockable indicator (DS1REBLK) in the Format 1 DSCB also determines whether a data set is to be reblocked or not. When the indicator is on, the data set is always reblocked to the system-determined, optimal block size. A new return code has been added in the installation reblock exit routine to turn this indicator on when requested.

See Chapter 5 of *Data Facility Data Set Services: Reference* guide for a more detailed description of the operations of the installation exit.

Security Considerations

For security purposes, the data set tracks used before the relocation are erased after relocation under these conditions:

- RACF Version 1 Release 6 (or earlier release) is installed and either:
 - The data set is RACF-protected,
 - The VSAM data set has the ERASE attribute.
 - The data set is password-protected.
- When RACF Version 1 Release 7 and MVS XA DFP Version 2 Release 1 (or a later release) are installed and either:
 - The data set was defined to RACF with the RACF ERASE option,
 - The VSAM data set has the ERASE attribute,
 - The data set is password-protected. (In this case, if the data set is also defined to RACF, the RACF ERASE option is taken. Figure 29 on page 108 provides more detail.

	Password Protected	Defined ERASE	RACF Protected	Erased on Scratch
User Install Exit = ERASE (default)	No	No	No	No
	Yes	No	No	Yes
		Yes	No	Yes
	No	No	Yes (= ERASE)	Yes
	No	No	Yes (= NOERASE)	No
	No	Yes	Yes (= ERASE)	Yes
	No	Yes	Yes (= NOERASE)	Yes
	Yes	Yes	Yes (= ERASE)	Yes
	Yes	Yes	Yes (= NOERASE)	Yes
User Install Exit = NOERASE	No	No	No	No

Figure 29. Data Set Erase Table for DEFRAG (with RACF Version 1 Release 7)

Note to Figure 29:

The catalog entry contains the ERASE attribute specified when the data set was defined (VSAM only).

You can prevent the tracks from being erased by using a user installation options exit routine. See the UFOERASE field in the ADRUFO data area in *Data Facility Data Set Services: Reference*.

DEFRAG does not relocate protected data sets unless:

- You have RACF DASDVOL update access to the volume.
- You have RACF DATASET read access to the data sets on the volume.
- You specify the read or update password for password-protected data sets, or the authorization exit routine supplied with DFDSS is changed to allow relocation of protected data sets.

When RACF DASDVOL class is active and a profile exists for the volume, a DASDVOL authorization failure causes the DEFRAG task to abend with a system code 913. This happens regardless of RACF data set access authority.

Appendix A. ACS Routine Variables

This appendix lists the variables available to Automatic Class Selection (ACS) routines during DFDSS COPY, RESTORE, and CONVERTV. For details on writing ACS routines, see *Storage Administration Reference*.

ACS Variables Available during COPY

When Automatic Class Selection (ACS) is invoked during DFDSS COPY, the following variables are passed to the ACS routines:

Variable Name	Description
&ACCT_JOB	Accounting information from the JOB statement.
&ACCT_STEP	Accounting information on the EXEC statement.
&ALLVOL	Output volume serial numbers. References the same volume list as &ANYVOL, but, when used in a comparison, returns true only if all volume serial numbers satisfy the condition.
&ANYVOL	Output volume serial numbers. References the same volume list as &ALLVOL, but, when used in a comparison, returns true if any volume serial numbers satisfy the condition.
&APPLIC	Application identifier associated with the data set (available only if RACF is installed).
&ACSENVIR	Environment in which ACS was invoked. Set to 'ALLOC.'
&DD	Ddname
&DEF_DATACLAS	Default data class name (available only if RACF is installed).
&DEF_MGMTCLAS	Default management class name (available only if RACF is installed).
&DEF_STORCLAS	Default storage class name (available only if RACF is installed).
&DSN	Data set name.
&DSORG	Data set organization.
&DSOWNER	RACF owner or group that is considered the data set owner (available only if RACF is installed).
&DSTYPE	Data set type (for example, GDS, PERM, or TEMP).
&EXPDT	Expiration date.
&GROUP	Group identifier from the JOB statement.
&HLQ	High-level qualifier of the data set name.
&JOB	Job name, started task name, or TSO user ID from the JOB statement.
&LLQ	Low-level qualifier of the data set name.
&MAXSIZE	Maximum size of data set in kilobytes. For non-VSAM data sets, the primary value plus 15 extents. For VSAM data sets, the primary value plus 123 extents.
&MSVGP	Mass storage (MSS) volume names.
&NQUAL	Number of qualifiers in the data set name.

Variable Name	Description
&NVOL	Number of output volumes specified by the user.
&PGM	Program name from the EXEC card.
&RECO RG	Data set record organization.
&RETPD	Retention period.
&SIZE	Size of the data set in kilobytes.
&UNIT	Unit name.
&USER	User ID from the JOB statement or the user ID propagated from the environment.
&XMODE	Execution mode (for example, TSO, BATCH, or TASK).

Figure 30. Variables Passed to ACS Routines during DFDSS COPY.

Note: The following variables are not available to the storage group ACS routine: &ACCT_JOB, &ACCT_STEP, &DD, &JOB, &MSVGP, &PGM, and &XMODE.

ACS Variables Available during RESTORE and CONVERTV

When ACS is invoked during DFDSS RESTORE or CONVERTV, the following variables are passed to the ACS routines:

Variable Name	Description
&ALLVOL	For RESTORE, the output volume serial numbers. For CONVERTV, the volumes on which the data set resides. References the same volume list as &ANYVOL, but, when used in a comparison, returns true only if all volume serial numbers satisfy the condition.
&ANYVOL	For RESTORE, the output volume serial numbers. For CONVERTV, the volumes on which the data set resides. References the same volume list as &ALLVOL, but, when used in a comparison, returns true if any volume serial numbers satisfy the condition.
&APPLIC	Application identifier associated with the data set (available only if RACF is installed).
&ACSENVIR	Environment in which ACS was invoked. Set to 'RECOVER' for RESTORE. Set to 'CONVERTV' for CONVERTV.
&DEF_DATA CLAS	Default data class name (available only if RACF is installed).
&DEF_MGMT CLAS	Default management class name (available only if RACF is installed).
&DEF_STOR CLAS	Default storage class name (available only if RACF is installed).
&DSN	Data set name.
&DSORG	Data set organization.
&DSOWNER	RACF owner or group that is considered the data set owner (available only if RACF is installed).
&DSTYPE	Data set type (for example, GDS, PERM, or TEMP).
&EXPDT	Expiration date.

Variable Name	Description
&HLQ	High-level qualifier of the data set name.
&LLQ	Low-level qualifier of the data set name.
&MAXSIZE	Maximum size of data set in kilobytes. For non-VSAM data sets, the primary value plus 15 extents. For VSAM data sets, the primary value plus 123 extents.
&NQUAL	Number of qualifiers in the data set name.
&NVOL	For RESTORE, number of volumes specified by the user. For CONVERTV, the number of volumes (including candidate volumes) on which the data set resides.
&RECORD	Data set record organization.
&RETPD	Retention period.
&SIZE	Size of data set in kilobytes.
&UNIT	Unit name.

Figure 31. Variables Passed to ACS Routines during DFDSS RESTORE and CONVERTV.

Appendix B. Changing Command Specifications for DFDSS Version 2 Release 4

The following sections briefly explain what the new keywords were for Version 2 Release 4 and how they affect DFDSS processing.

Volume Specification

Because the system places data sets in an SMS environment, you need not know on what volume a particular data set resides. As a result, when you move data sets within an SMS environment, you should avoid specifying output volumes. You should specify commands without OUTDD and OUTDYNAM. For example:

```
COPY -  
  DATASET(INCLUDE(**)) -  
  LOGINDYNAM((338001),(338002)) -  
  DELETE
```

You can specify output volumes in an SMS environment, but, depending upon how your ACS routines are written, the volume serial numbers you specify might or might not be used to determine where the data is placed.

Only data sets with the guaranteed space attribute in their storage class are guaranteed to go on a volume you specify. However, the volumes you specify must all be SMS volumes in the same storage group. ACS must assign to the data set(s) the storage group that contains all the volumes. If these conditions are not met, the allocation of the data set fails.

Cataloging

In an SMS environment, all data sets must be cataloged. You cannot use JOBCAT and STEPCAT. To find data sets outside the standard order of search with DFDSS, you must use the INCAT keyword and specify the catalog name as in the following example:

```
DUMP -  
  DATASET(INCLUDE(**)) -  
  INCAT(SYS1.PAYROLL.ICFCAT, PAYROLL.ADMIN.ICFCAT) -  
  DELETE -  
  COMPRESS
```

In addition, because SMS-managed data sets must be cataloged in the standard order of search, the RECATALOG keyword is ignored for these data sets.

JCL Changes

The main change to the job control language (JCL) is that **JOB**CAT and **STEP**CAT are prohibited in an SMS environment. Therefore, to find data sets cataloged outside the standard order of search, you must use the **DFDSS INCAT** keyword.

Processing VSAM Spheres

If you want to process an entire VSAM sphere with **COPY**, **DUMP**, or **RESTORE**, you should use the **SPHERE** keyword. This keyword causes **DFDSS** to **COPY**, **DUMP**, or **RESTORE** the base cluster and all associated alternate index clusters and paths of a VSAM data set. For instance, if you want to restore a VSAM sphere, code the **RESTORE** command as shown in the following example.

Note: You can **RESTORE** an entire VSAM sphere only if the input volume was also dumped using the **SPHERE** keyword.

```
RESTORE -  
    DATASET(INCLUDE(T01729.ACCOUNTS.JAN88)) -  
    INDD(TAPE1) -  
    SPHERE
```

All parts of a target sphere must be directed through the standard order of search to the same catalog. This includes the base cluster, all of its related AIXs, and all of its paths.

DFDSS normally processes data sets on an individual basis. Use of the **SPHERE** keyword, however, causes **DFDSS** to consider a VSAM **SPHERE** as a collection of individual data sets and associated catalog information.

Although **DFDSS** will consider the sphere as one entity for movement purposes, other keywords will still operate on the individual components of the sphere. The most important example of this is the **RENAME** and **RENAMEU** keywords.

To rename all of a **SPHERE**, you must either rename each of the individual sphere components, or if all of your sphere components start with the same high level qualifier, you can rename by specifying a new high level prefix for the sphere.

Considerations for Processing VSAM Spheres

There are certain restrictions to bear in mind when processing VSAM spheres with **COPY**, **DUMP**, or **RESTORE**.

Restrictions for **COPY**

- The copying of a sphere is permitted only if all parts of the sphere resolve to the same catalog.
- To logically **COPY** a sphere without the **DELETE** or **RECAT** keyword, you must rename every data set in the sphere. This includes all paths, all AIXs, and the base cluster. If the target sphere is to be SMS managed, the data sets must be renamed even if **RECATALOG** is specified because **RECATALOG** is ignored for SMS-managed data sets.

- When copying with **DELETE** between **SMS** and non-**SMS** environments or between different storage groups in **SMS**, you must include all parts of the sphere. You can do this by specifying **SPHERE** and naming the base cluster.

Restrictions for **RESTORE**

- You can restore a sphere only if all parts of the sphere resolve to the same catalog.
- Unlike the **COPY** function, you may not need to rename all of the parts in the **VSAM** sphere as you would with **COPY**.
- When restoring a sphere that has one or more **AIXs** missing from the dump tape, **DFDSS** issues a message to indicate that the sphere was incompletely restored.

Restrictions for **DUMP**

You should be aware that you cannot restore a sphere unless it is dumped as a sphere with the **SPHERE** keyword.

Reblocking Data Sets with **RESTORE**

You can reblock one or more of the data sets you are restoring by specifying the **REBLOCK** keyword with the **RESTORE** command. **REBLOCK** processing for **RESTORE** operates in the same way as it does for the **COPY** command. Reblocking is done only if the new block size is more efficient than the existing one.

The following **RESTORE** job stream reblocks data set **T01729.ACCOUNTS.JAN88** (if the new block size is more efficient than the existing one):

```
RESTORE -  
  DATASET(INCLUDE(*.ACCOUNTS.JAN88)) -  
  INDD(TAPE1) -  
  REBLOCK(T01729.ACCOUNTS.JAN88)
```

Note: If **PAM** data sets are specified with both **REBLOCK** and **NOPACKING** keywords, the data set will not be reblocked.

Appendix C. Stand-Alone Restore Program

This chapter describes the hardware and software requirements for DFDSS and stand-alone DFDSS restore.

Stand-alone DFDSS restore must operate in engineering change (EC) mode.

Storage Devices

You can use stand-alone DFDSS restore with the following IBM direct access storage devices:

Direct access storage devices:	2305 Fixed Head Storage Model 2
	3330 Disk Storage Models 1 and 11
	3340 Direct Access Storage Facility
	3344 Direct Access Storage
	3350 Direct Access Storage (in native mode and in 3330 Models 1 and 11 compatibility mode)
	3375 Direct Access Storage
	3380 Direct Access Storage (all models)
	3330 Virtual volumes residing in a 3850 Mass Storage System

Tape devices:	2400 series Magnetic Tape Units
	3420 series Magnetic Tape Units
	3430 Magnetic Tape Unit
	3480 Magnetic Tape Subsystem (all models)

Other Devices

You can use stand-alone DFDSS restore with the following IBM devices:

System consoles:	1052 Printer-Keyboard Model 7
	3036 Console
	3056 Remote System Console
	3138 Processing Unit Console
	3148 Processing Unit Console
	3158 Processing Unit Display Console Keyboard
	3210 Console Printer-Keyboard Models 1 and 2
	3215 Console Printer-Keyboard Model 1
	3270 Information Display System
	3277 Display Station Model 2
	3278 Display Station Model 2A

Printers:	1403 Printer
	1443 Printer
	3203 Printer
	3211 Printer
	3262 Line Printer Model 5
	3800 Printing Subsystem
	4245 Printer
	4248 Printer

Card readers:	2501 Card Reader
	2520 Card Read Punch
	2540 Card Read Punch
	3505 Card Reader

You can use the appropriate generic name (CNSL, PRNT, CARD, or TAPE) to access these devices.

Restrictions

The following restrictions apply to stand-alone DFDSS restore:

- Stand-alone DFDSS restore can restore from tape volumes created by full or tracks DUMP, or it can restore a track or tracks from the first logical volume of a physical data set DUMP. It cannot restore from tapes created by a logical DUMP operation.
- DASD volumes that are to be restored must be initialized.
- Stand-alone DFDSS restore can only restore from the first physical tape file of tape volumes.
- There must be a match between the device type of the source volume used in a (system DFDSS) DUMP and the device type of the receiving volume used in a stand-alone DFDSS restore. However, you can do the following:
 - Dump data from an IBM 3330 Model 1 can be restore to an IBM 3330 Mode
 - Dump data from an IBM 3340 with 35-megabyte capacity can be restored an IBM 3340 with 70-megabyte capacity or to an IBM 3344.
 - Dump data from a smaller capacity IBM 3380 model can be restored to a larger capacity IBM 3380 model.

Note: When you restore data from a smaller device to a larger device, the freespace information becomes invalid. The freespace information in the VTOC is rebuilt when you allocate a data set on the volume.
- The following system consoles are not supported by stand-alone DFDSS restore:
 - System/370 Model 115/125 Operator Console
 - IBM 3066 System Console (System/370 Model 168)
- Stand-alone DFDSS restore has no function other than restore.
- Stand-alone restore requires exclusive control of all input output devices.
- Stand-alone restore in System/370 extended-architecture mode requires your source tape on a different tape drive than the IPL device.
- Neither mass storage virtual volumes nor minivolumes can be restored.
- Stand-alone DFDSS restore cannot restore from dump tapes produced by other utilities.
- Stand-alone DFDSS restore can only restore from the first logical volume of a dump tape produced from multiple source volumes.
- The devices used to do a stand-alone restore in System/370 mode must be attached to the processor you are restoring (where there are two or more processors).

Compatibility with DFDSS

Stand-alone DFDSS restore can restore any physical full-volume dump made by DFDSS as long as the physical dump is the first data set on the tape volume.

Glossary of Terms and Abbreviations

This glossary includes definitions of some terms found in this document. Some of the terms defined below are from:

- *The American National Dictionary for Information Processing Systems*, copyright 1982 by the Computer and Business Equipment Manufacturers Association. Copies may be purchased from the American National Standards Institute at 1430 Broadway, New York, New York 10018. These definitions are identified by an asterisk (*).
- *The ISO Vocabulary - Information Processing*, and *The ISO Vocabulary - Office Machines*, developed by the International Standards Organization, Technical Committee 97, Subcommittee 1. Definitions from published sections of this vocabulary are identified by the symbol "(ISO)" preceding the definition. Definitions from draft proposals and working papers under development by the ISO/TC97 vocabulary subcommittee are identified by the symbol "(TC97)," indicating that final agreement has not yet been reached among its participating members.

A

ABEND. Abnormal end of task. End of a task, a job, or a subsystem because of an error condition that cannot be resolved by recovery facilities while the task is performed.

ABENDxxx. The keyword that identifies the abnormal end of DFDSS because of a system-detected error.

ABSTR. A subparameter of the SPACE parameter in a DD statement. It indicates that specified tracks be assigned to a data set.

ACCEPT processing. An SMP/E process necessary for installing the FMIDs. SMP/E ACCEPT processing uses JCL to accept the modules and macros necessary to run the FMIDs. The FMIDs are accepted into the DLIBs from the temporary data sets.

Access method services. A multifunction service program that is used to manage both VSAM and non-VSAM data sets and integrated catalog facility or VSAM catalogs. It is used to define data sets and allocate space for them, convert indexed-sequential data sets to key-sequenced data sets, modify data set attributes in the catalog, reorganize data sets, facilitate data portability between operating systems, create backup copies of data sets, data set records and catalog

entries, help make inaccessible data sets accessible, list the records of data sets and catalogs, define and build alternate indexes, and convert OS CVOLs and VSAM catalogs to integrated catalog facility catalogs.

ACDS. Active control data set.

ACS. Automatic class selection.

AIX. Alternate index.

alias. An alternate name for a member of a partitioned data set.

ALLOC. A space allocation parameter that indicates type, such as cylinders or tracks.

alternate index (AIX). In systems with VSAM, a key-sequenced data set containing index entries organized by the alternate keys of its associated base data records. It provides an alternate means of locating records in the data component of a cluster on which the alternate index is based.

alternate index cluster. In VSAM, the data and index components of an alternate index.

APAR. Authorized Program Analysis Report.

Authorized Program Analysis Report (APAR). A request for correction of a problem caused by a suspected defect in a current unaltered release of a program.

APF. Authorized program facility.

Application Interface. An interface used to invoke DFDSS from another program.

Apply processing. In SMP and SMP/E, the process, initiated by the APPLY command, that places system modifications (SYSMODS) into the target system libraries.

attach. In programming, to create a task that can be performed asynchronously with the performance of the mainline code.

authorization. (1) (TC97) The right granted to a user to communicate with or make use of a computer system. (2) The process of giving a user either complete or restricted access to an object, resource, or function.

automatic class selection (ACS). A mechanism for assigning SMS classes and storage groups.

B

base cluster. In systems with VSAM, a key-sequenced or entry-sequenced data set over which one or more alternate indexes are built.

basic catalog structure (BCS). The name of the catalog structure in the integrated catalog facility environment. An integrated catalog facility catalog consists of a BCS and its related VSAM volume data sets (VVDSSs).

basic direct access method (BDAM). An access method used to directly retrieve or update particular blocks of a data set on a direct access device.

basic partitioned access method (BPAM). An access method that can be applied to create program libraries in direct access storage for convenient storage and retrieval of programs.

basic sequential access method (BSAM). An access method for storing or retrieving data blocks in a continuous sequence, using either a sequential access or a direct access device.

BCS. Basic catalog structure.

BDAM. Basic direct access method.

BLK. A subparameter of the SPACE parameter in a DD statement. It specifies that space is allocated by blocks.

block length. Synonym for block size.

block size. (1) (TC97) The number of data elements in a block. (2) * A measure of the size of a block, usually specified in units such as records, words, computer words, or characters. (3) Synonymous with block length.

BPAM. Basic partitioned access method.

bpi. bits per inch.

BSAM. Basic sequential access method.

C

call. (ISO) * The action of bringing a computer program, a routine, or a subroutine into effect, usually by specifying the entry conditions and jumping to an entry point.

card image. * A one-to-one representation of the hole patterns of a punched card; for example, a matrix in which a one represents a punch and a zero represents the absence of a punch.

CCHHR. Cylinder, cylinder, head, head, record.

CCW. Channel command word.

CDE. Contents directory entry.

CDS. Control data set.

channel command word (CCW). A doubleword at the location in main storage specified by the channel address word. One or more CCWs make up the channel program that directs data channel operations.

CI. Control interval.

CLIST. Command list.

component identification keyword. The first keyword, represented as a number, in a set of keywords used to describe a DFDSS program failure.

COMPRESS command. The DFDSS function that reduces partitioned data sets by taking unused space and consolidating it at the end of the data set.

concatenation. An operation that joins two characters or strings in the order specified, forming one string whose length is equal to the sum of the lengths of the two characters or strings.

control interval (CI). A fixed-length area of auxiliary storage space in which VSAM stores records. It is the unit of information transmitted to or from auxiliary file storage by VSAM.

control volume (CVOL). A volume that contains one or more indexes of the catalog.

constructs. A collective name for data class, storage class, management class, and storage group.

CONVERTV command. The DFDSS function that converts volumes to and from Storage Management Subsystem management without data movement.

COPY command. The DFDSS function that performs data sets, volumes, and tracks movement.

CREDIT. Creation date.

CSW. Channel status word.

CVAF. Common VTOC access facility.

CVOL. Control volume.

CVT. Communication vector table.

CYL. A subparameter of the SPACE parameter in a DD statement. It specifies that space is allocated by cylinders.

D

DADSM. The direct access space management program that maintains the VTOC, VTOCIX, and space on a volume.

DAM. Direct access method.

DASD. Direct access storage device.

DASD ERP. DASD error recovery procedure.

data class. A list of data set allocation parameters and their values that are used when allocating a new SMS-managed data set.

data compression (run-length). A method of encoding repetitive series of identical characters so that they occupy less space on a dump tape. Data compression is supported by both physical dump and logical dump processing.

Data Facility Data Set Services (DFDSS). A DASD data and space management tool for backup and recovery, space management functions, and copying data between like and unlike devices. DFDSS is the primary data mover of the Data Facility family.

Data Facility family. A family of products, consisting of DFP, DFDSS, DFHSM, and DFSORT, that performs key data and storage management functions.

Data Facility Hierarchical Storage Manager (DFHSM). A facility that automates space and availability management by backing up and migrating data within a hierarchy of storage.

Data Facility Product (DFP). The key architectural component of the Storage Management and non-Storage Management Subsystem environments. DFP manages programs, data, and devices in the MVS environment.

Data Facility Sort (DFSORT). A data and storage management tool that handles sorting, merging, and copying applications.

Data Facility Storage Management Subsystem (DFSMS). The complementary functions of MVS/DFP and other individual products of the Data Facility family, which, together with RACF provide a system-managed, administrator-controlled storage environment.

data set backup. Backup to protect against the loss of individual data sets.

data set change indicator. Bit 6 of the DS1DSIND field in the Format 1 DSCB. It is set by OPEN when the data set is opened for processing other than input. This

flag is supported on MVS systems that have data-set-changed flag support installed.

DAU. Direct access unmovable.

DB2. IBM DATABASE 2.

DCB. Data control block.

DEFRAG command. The DFDSS function that consolidates the free space on a volume to help prevent out-of-space abends on new allocations.

DEQ. An Assembler language macro instruction used to remove control of one or more serially reusable resources from the active task.

DFDSS. Data Facility Data Set Services.

DFHSM. Data Facility Hierarchical Storage Manager.

DFP. Data Facility Product.

DFSMS. Data Facility Storage Management Subsystem

DFSORT. Data Facility Sort.

DGTHIX50. The panel identification number for the DFDSS Help Index. It is used to verify that DFDSS ISMF is installed correctly.

DIAGNOSE. An access method services command that scans an integrated catalog facility basic catalog structure (BCS) or a VVDS to validate the data structure.

DIRF. DADSM interrupt recording facility. If a system fails, or a permanent I/O error occurs during allocation of space or during performance of a routine that updates the VTOC, the VTOC may be in error. To ensure that an error is recorded, the DADSM routines turn on a bit in the VTOC upon entry to a DADSM function, and, if no errors occur during processing, turn off that bit upon exiting from that function. This DIRF bit is bit 5 of the DS4VTOCI field of the Format 4 DSCB.

distribution libraries. IBM-supplied partitioned data sets on tape containing one or more components that the user restores to disk for subsequent inclusion in a new system.

DLIB. Distribution library.

DOC. In diagnosing program failures, the keyword that identifies an error in the documentation of a program.

DOS. Disk Operating System.

DOS bit. On a volume without an indexed VTOC a bit that indicates that the free space map (Format 5 DSCB) is invalid.

DOS/VSE. DOS/Virtual Storage Extended.

DSCB. Data set control block.

DSCHA. Data-set-changed flag

dsn. Data set name

DSORG. Data set organization. It is specified in the JCL as "DSORG=." It is indicated in a field in the Format 1 DSCB control block in the VTOC.

DS1LSTAR. A field in the Format 1 DSCB in the VTOC that indicates the last-used track and last block on that track, in the form of a relative track and record number (TTR).

DUMP command. The DFDSS function used to back up data sets, tracks, and volumes.

dynamic allocation. Assignment of system resources to a program when the program is performed rather than when it is loaded main storage.

E

Early Warning System (EWS). A microfiche copy of the information contained in the Software Support Facility (SSF), organized by component identification number and indexed by APAR symptom code. EWS is published monthly and available to customers of IBM licensed programs.

ECB. Event control block.

EC mode. Engineering change mode.

ENQ. An Assembler language macro instruction that requests the control program to assign control of one or more serially reusable resources to the active task. It is also used to determine the status of a resource; that is, whether it is immediately available or in use, and whether control has been previously requested for the active task in another ENQ macro instruction.

entry-sequenced data set (ESDS). In VSAM, a data set whose records are loaded without respect to their contents and whose RBAs cannot change. Records are retrieved and stored by addressed access, and new records are added at the end of the data set.

EOF. End-of-file.

EOJ. End of job.

ESDS. Entry-sequenced data set.

ESTAE. Extended specify task abnormal exit.

EQ. Equal to.

EWS. Early Warning System.

EXCP. Execute channel program.

Execute channel program (EXCP). A macro used to access a data set without specifying the organization.

EXPDT. Expiration date.

Extended specify task abnormal exit (ESTAE). A task recovery routine that provides recovery for those programs that run enabled, unlocked, and in task mode.

extent. A continuous space on a DASD volume occupied by a data set or portion of a data set. An extent of a data set contains a whole number of control areas.

F

FC. CVAF function code.

FCEC. CVAF function-error code.

filtering. The process of selecting data sets based on specified criteria. These criteria consist of fully- or partially-qualified data set names and/or of certain data set characteristics.

FMID. Function modification identifier.

Format 1 DSCB (F1 DSCB). An identifier DSCB that describes a data set or VSAM data space and its first three extents. See the appropriate system data areas manual.

Format 2 DSCB (F2 DSCB). An index DSCB that describes the indexes of an ISAM data set. See the appropriate system data areas manual.

Format 3 DSCB (F3 DSCB). An extension DSCB that describes the 4th through 16th extents of a data set or VSAM data space. See the appropriate system data areas manual.

Format 4 DSCB (F4 DSCB). A VTOC DSCB that describes the extent and contents of the VTOC and volume and device characteristics. See the appropriate system data areas manual.

Format 5 DSCB (F5 DSCB). A free space DSCB that describes the space on a volume that has not been allocated to a data set or to a VSAM data space (available space). See the appropriate system data areas manual.

fully-qualified data set name. A data set in which all the qualifiers are completely spelled out.

function modification identifier (FMID). A code that identifies the release levels of a program product.

FVL. function vector list.

G

GDG. Generation data group.

GDS. Generation data set.

Generation data group (GDG). A collection of historically related non-VSAM data sets that are arranged in chronological order; each data set is a generation data set.

Generation data set. One generation of a generation data group.

GT. Greater than.

I

ICKDSF. Device Support Facilities.

IDCAMS. Access Method Services.

IMS/VS. Information Management System/Virtual Storage.

INCORROUT. In diagnosing program failures, the keyword that identifies incorrect or missing program output.

incremental backup. A process in which data sets are backed up only if they have changed since their last backup.

integrated catalog facility. A facility by which VSAM data set volume-related fields are separated from the catalog and maintained in the VVDS on the volume on which the data set resides.

Interactive Storage Management Facility (ISMF). A DFP component that provides an interactive interface to DFDSS and DFHSM. ISMF uses menu-driven panels to build jobs streams for most DFDSS functions. DFDSS ISMF performs the following functions: COMPRESS, CONVERTV, COPY, DEFRAG, DUMP, RELEASE, and RESTORE.

Interactive System Productivity Facility (ISPF). An IBM licensed program used to develop, test, and run application programs interactively. ISPF is the interactive interface for all storage management functions.

I/O. Input/output.

IPL. Initial program load.

ISMF. Interactive Storage Management Facility.

ISAM. Indexed sequential access method.

ISMF. Interactive Storage Management Facility.

ISPF. Interactive Systems Productivity Facility.

ISPF/PDF. Interactive Systems Productivity Facility/Program Development Facility.

J

JCL. Job control language.

JES. Job entry subsystem.

JES2. An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for operation, processes their output, and purges them from the system. In an installation site with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing.

JES3. An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for operation, processes their output, and purges them from the system. In complexes that have several loosely-coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them via a common job queue.

JFCB. Job file control block.

job control language (JCL). A problem-oriented language used to identify the job or describe its requirements to an operating system.

job entry subsystem (JES). A system facility for spooling, job queuing, and managing I O.

JSCB. Job step control block.

K

key-sequenced data set. A VSAM file or data set whose records are loaded in ascending key sequence and controlled by an index. Records are retrieved and stored by keyed access or by addressed access, and new records are inserted in key sequence by means of distributed free space. Relative byte addresses can change because of control interval or control area splits.

keyword. A symptom that describes one aspect of a program failure.

KSDS. Key-sequenced data set.

L

LASTCC. Last condition code.

LDS. Linear data set.

like devices. Devices that have the same track capacity and number of tracks per cylinder (for example, 3380 Standard and 3380 Model J or K).

LINK. An Assembler language macro instruction that causes control to be passed to a specified entry point. The linkage relationship established is the same as that created by a BAL instruction.

link-pack area (LPA). An area of virtual storage that contains re-enterable routines that are loaded at IPL (initial program load) time and can be used concurrently by all tasks in the system.

load module. (TC97) A computer program in a form suitable for loading into main storage for operation.

load module library. A partitioned data set used to store and retrieve load modules.

logical (data set) DUMP operation. A DUMP operation in which logical processing is performed.

logical (data set) processing. Processing that treats each data set and its associated information as a logical entity. As an example, DFDSS processes an entire data set before beginning with the next one.

logical (data set) RESTORE operation. A RESTORE operation that uses as input a data set produced by a logical DUMP operation.

logical volume. The output produced from a physical DUMP operation, for which all data is derived from a single DASD volume.

LOOP. In diagnosing program failures, the keyword that identifies a program failure in which some part of the program repeats endlessly.

LPA. link-pack area.

LT. Less than

LRECL. Logical record length.

LVOL. logical volume.

M

maintenance-level keyword. In diagnosing program failures, a keyword that identifies the maintenance level of DFDSS.

management class. A list of the migration, backup and retention parameters and their values for an SMS-managed data set.

map record. The record that maps the tracks that were dumped by DFDSS.

MAXCC. Maximum condition code.

MCS. Multiple Console Support.

MENTITY. Model entity.

minivolume. In an MVS system running on VM/370, an OS/VS-formatted VM/370 minidisk whose size is equal to or less than that of the real volume. DFDSS uses the device size specified in the Format 4 DSCB in their VTOC. Minivolumes are supported only by the system version of DFDSS.

MSGADRnnt. In diagnosing program failures, the DFDSS message keyword that tells of an error, or seems itself to be in error.

MSS. Mass storage system.

MVS. Multiple virtual storage.

N

NVR. Non-VSAM volume record.

O

Operating System (OS). Software that controls the execution of programs; an operating system may provide services such resource allocation, scheduling, input/output control, and data management.

OS. Operating System.

OS/VS2. Operating System/Virtual Storage 2.

P

pageable link-pack area (PLPA). Link-pack area.

PAM. Partitioned access method.

partially-qualified data set name. A data set name in which the qualifiers are not spelled out. Asterisks and

percent signs are used in place of the undefined qualifier(s).

partitioned data set (PDS). A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

partitioned data set extended (PDSE). A SMS-managed, page-formatted data set on direct access storage. A PDSE contains an indexed directory and members similar to the directory and members of partitioned data sets. A PDSE can be used instead of a partitioned data set.

PDS. Partitioned data set.

PDSE. Partitioned data set extended.

PERFM. In diagnosing program failures, the keyword that identifies degradation in program performance.

physical (data set) DUMP operation. A DUMP operation in which physical processing is performed.

physical (data set) processing. Processing that moves data at the track-image level and can operate against volumes, tracks, and data sets. As an example, DFDSS may only process one volume of a multivolume data set.

PLPA. Pageable link-pack area.

POU. Partitioned organization unmovable.

PRB. Program request block.

private library. A user-owned library that is separate and distinct from the system library.

PSU. Physical sequential unmovable.

PSW. Program status word.

PTF. Program temporary fix.

Q

QSAM. Queued sequential access method.

qualified name. A data set name consisting of a string of names separated by periods; for example, "TREE.FRUIT.APPLE" is a qualified name.

qualifier. Each component name in a qualified name other than the rightmost name. For example, "TREE" and "FRUIT" are qualifiers in "TREE.FRUIT.APPLE."

Queued sequential access method (QSAM). An extended version of the basic sequential access method

(BSAM). Input data blocks awaiting processing or output data blocks awaiting transfer to auxiliary storage are queued on the system to minimize delays in I/O operations.

R

RACF. Resource Access Control Facility.

RB. Request block.

RBA. Relative byte address.

RDJFCB. Read job file control block.

RECEIVE processing. An SMP/E process necessary to install new product libraries. During this process, the code, organized as unloaded partition data sets, is loaded into temporary SMPTLIB data sets. SMP/E RECEIVE processing automatically allocates the temporary partitioned data sets that correspond to the files on the tape, and loads them from the tape.

RECFM. Record format.

REFDT. Last-referenced date.

relative byte address (RBA). The displacement (expressed as a fullword binary integer) of a data record or a control interval from the beginning of the data set to which it belongs, independent of the manner in which the data set is stored.

Relative record data set (RRDS). A VSAM data set whose records are loaded into fixed-length slots.

RELEASE command. The DFDSS function that releases the unused space in sequential and partitioned data sets for use by other data sets.

RESERVE. A method of serializing DADSM update accesses to the VTOC. It is also a method of serializing processor accesses to a shared DASD volume.

Resource Access Control Facility (RACF). An IBM program product that provides for access control by identifying and verifying users to the system, authorizing access to DASD data sets, logging the detected unauthorized attempts to enter the system, and logging the detected accesses to protected data sets.

RESTORE command. The DFDSS function used to recover data sets, tracks, and volumes.

RMID. Replacement module identifier.

RRDS. Relative record data set.

run-length data compression. Data compression (run-length).

S

SAF. System Authorization Facility.

SAM. Sequential access method.

scheduler task. A DFDSS subtask that interprets and schedules commands.

SCP. System control program.

SEQ. Sequential or sequential processing

SEREP. System environmental recording, editing, and printing

SMF. System management facilities.

SMP. System Modification Program.

SMP/E. System Modification Program/Extended.

SMPE. A cataloged procedure that includes the required DD statements for running SMP/E and is used in the RECEIVE, APPLY, and ACCEPT steps of SMP/E processing.

SMS. Storage Management Subsystem.

Software Support Facility (SSF). An IBM online data base that allows for storage and retrieval of information about all current APARs and PTFs.

SP. System Product.

sphere. A VSAM cluster with one or more associated alternate indexes and paths. The VSAM cluster (sometimes called the base cluster), alternate indexes (AIXs), and paths are sometimes referred to as sphere components.

SSF. Software Support Facility.

stand-alone DFDSS restore program. One of two DFDSS programs. Stand-alone DFDSS restore runs independently of the MVS system environment and is limited to one function—a full or partial (tracks) RESTORE from a dump tape.

storage class. A named list of data set storage attributes that SMS uses to identify performance and availability requirements.

storage constructs. The group of predefined models (data class, management class, storage class, and storage group) that are used to classify storage management needs and procedures for data sets under the Storage Management Subsystem. Each data set has construct names associated with it by explicit specification or defaulting.

storage group. A named collection of DASD volumes that have been grouped to meet a defined service strategy.

storage management. The task of managing auxiliary storage resources for an installation.

Storage Management Subsystem (SMS). An MVS subsystem that helps automate and centralize the management of storage. To manage storage, the storage management subsystem provides the storage administrator with control over data class, storage class, management class, storage group, and automatic class selection routine definitions.

subtask. A task initiated and ended by a higher order task.

SVC. Supervisor call instruction.

SVRB. Supervisor request block.

SYSRES. System residence disk.

system library. A collection of data sets or files in which the parts of an operating system are stored.

system link library. System library.

System Modification Program (SMP). A program used to install software and software changes on the MVS system.

System Modification Program Extended (SMP/E). An IBM licensed program used to install software and software changes on the MVS system. In addition to providing the services of SMP, SMP/E consolidates installation data, allows more flexibility in selecting changes to be installed, provides a dialog interface, and supports dynamic allocation of data sets.

T

TCB. Task control block.

Time sharing option (TSO). An option on the operating system for a System 370 that provides interactive time sharing from remote terminals.

TIOT. Task input output table.

TLIBs. Target data sets.

track packing. A technique used by DFDSS that builds target tracks for any DASD device using input physical record information.

TRK. A subparameter of the SPACE parameter in a DD statement. It specifies that space is to be allocated by tracks.

TSO. Time-sharing option.

TSO/E. TSO/Extensions.

TTR. track-track-record.

type-of-failure keyword. In diagnosing program failures, a keyword that identifies the type of program failure that has occurred in DFDSS.

U

UACC. Universal access authority.

UCB. Unit control block.

UIM. User interaction module.

unlike devices. Devices that have different track capacities, for example, 3350 and 3380.

used tracks. Tracks from the beginning of data sets to the last-used track (as indicated in the DS1LSTAR field of the Format 1 DSCB).

V

VDRL. Volume restore limits.

VDSS. VTOC Data Set Services.

virtual storage access method (VSAM). An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by the relative-record number.

VM. Virtual machine.

VOLID. Volume ID.

VOLSER. Volume serial number.

volume backup. Backup of an entire volume to protect against the loss of the volume.

volume header record. The record in the DFDSS dump tape that identifies and contains data pertinent to the whole volume, and identifies the type of operation that created a dump.

volume trailer record. The record in the DFDSS dump tape that identifies the end of the data for a DASD volume.

VSAM. Virtual storage access method.

VSAM volume data set (VVDS). A data set that describes the VSAM and SMS-managed non-VSAM data sets on a volume. The name of the data set is SYS1.VVDS.Vvolser.

VSE. Virtual storage extended.

VTOC. Volume table of contents.

VTOCIX. The data set on which the location of the Format 1 DSCBs of the VTOC are kept in an index for quick access by DADSM. The name of the data set is SYS1.VTOCIX.Vvolser.

VVDS. VSAM volume data set.

VVR. VSAM volume record.

W

WAIT. In diagnosing program failures, the keyword that identifies DFDSS suspended activity, while waiting for some condition to be satisfied. DFDSS does not issue a message to tell why it is waiting.

WTO. Write to operator

X

XA. Extended Architecture.

Index

A

ABEND=nnn in the EXEC statement 33
ABOVE16 in the EXEC statement 34
ACS variables
 name/description 109–111
 passed in COPY command 109–110
 passed in RESTORE/CONVERTV 110–111
ALTER LOCK, IDCAMS command 60
AMSGCNT=nnnn in the EXEC statement 33
application interface
 function 36
 invoking 36
 module names 38
auditing information 36

B

backing up an integrated catalog facility user
 catalog 49
BDAM data sets, supported 10
BELOW16 in the EXEC statement 34
block size
 default when dumping to tape or DASD 53
 minimum 53
books, DFDSS iv
BY criteria 13

C

card readers supported 9
 stand-alone DFDSS restore 118
catalog recovery 60
cataloging non-VSAM data sets during RESTORE 58
catalog, locking a user 60
changes in DFDSS, summary of xiv
changes, command specifications for V2 R5.0
 arguments for filtering data set characteristics 20
 PDSE, converting to 19
 PDS, converting to 19
 SYS1 data set, processing 19
 TGTALLOC keyword, omitting 20
 undefined DSORG, processing 19
changes, command specifications V2 R4.0
 data sets, reblocking with RESTORE 115
 JCL changes 114
 volume specification 113
 cataloging data sets 113
 VSAM spheres, processing 114
changes, summary of previous xiv
characteristics of data sets 13
combining data set extents 103
command data set, following SYSIN DD 33

compaction feature 41
compatibility
 stand-alone DFDSS restore and DFDSS 119
COMPRESS command
 module name 38
consoles supported 9
 stand-alone DFDSS restore 117
controlling DFDSS
 using ISMF 21
 using JCL 31
 using PARM information in the EXEC
 statement 33
converting to a PDS 19
converting to a PDSE 19
converting to and from SMS 91
CONVERTV operation
 variables passes to ACS routines 110–111
COPY command
 entry panel example 24, 25
 example of 34
 ISAM data sets 82
 job statement
 verifying information 25
 list example 26
 module name 38
 restrictions, sphere 114
 spheres, processing 114
 TGTALLOC keyword, omitting 20
 user catalog 80
 variables passed to ACS routines 109–110
 VSAM data sets 80
criteria for filtering 13
CVOL, moving 80

D

DASD devices supported 9
DASD initialized 10
DASD space fragmentation 104
DASD space reclamation 102
DASD space utilization 104
DASD space, reclaiming 101
DASD (direct access storage device)
 supported
 stand-alone DFDSS restore 117
data compaction
 hardware exception 52
 software exception 52
Data Facility Data Set Services
 See DFDSS
data integrity
 shared DASD considerations 53
data security 107

- data set
 - characteristics (BY criteria) 13
 - erase table
 - DEFRAG 108
 - extents, combining 103
 - filtering 13
 - line operators
 - module names 38
 - list panel example 24
 - logical RESTORE
 - RACF-protected data sets 57
 - restoring 53
 - selection entry panel example 23
- default block size when dumping to tape or DASD 53
- DEFRAG command
 - data set erase table 108
 - when to run 104
- detailed notes on using DFDSS
 - storage requirements 7
- Device Support Facilities utility 10
- devices
 - stand-alone DFDSS restore
 - card readers 118
 - consoles 117
 - printers 117
 - storage 117
 - system consoles 117
 - tape drives 117
- devices supported
 - DFDSS 9
- DFDSS
 - control 31
 - devices supported 9
 - filtering 13
 - input 32
 - interactive 21
 - invoking 21
 - invoking with Application Interface 36
 - JCL requirements 31
 - line operators 38
 - output 32
 - overview 1
 - PARM information in the EXEC statement 33
 - storage requirements 7
 - system requirements 7
 - volume formats supported 10
- DFDSS JCL 33
- DFDSS publications iv
- DFDSS (Data Facility Data Set Services)
 - compatibility with stand-alone DFDSS restore 119
- diagnosis guide, DFDSS iv
- disaster recovery, backup 41
- disaster recovery, backups 43
- disk drives
 - See DASD devices supported
- display panels, ISMF 21
- DUMP command
 - entry panel example 28, 29

- DUMP command (*continued*)
 - example of data set 35
 - exceptions to
 - hardware data compaction 52
 - software data compaction 52
 - job stream example 31
 - job submission entry panel 30
 - module name 38
 - printed output
 - produced by integrated catalog facility user catalog 50
 - restrictions, sphere 115
 - spheres, processing 114

E

- environment, system 7
- erase on scratch for DEFRAG 107
- ESDS data sets, supported 10
- examples of DFDSS jobs
 - COPY 34
 - data set RESTORE 35
 - dumping a data set 35
 - ISMF used 21–31
- EXCLUDE criteria 13
- EXCP data sets, supported 10
- EXEC statement (JCL) 31–34
- EXECUTE statement
 - entry panel example 26, 30

F

- filtering
 - general description 13
- free-space fragmentation 104

G

- GDG data sets, conversion consideration 97, 99
- generic device names
 - stand-alone DFDSS restore 118
- glossary 121–129

I

- ICKDSF, initialize DASD volumes with 10
- INCLUDE criteria 13
- incremental backup 45
- indexed VTOCs 10
- initialize all DASD volumes 10
- input to DFDSS (in DD statement) 32
- integrated catalog facility
 - backing up a user catalog, example of 49
 - printed output for a dumping a user catalog 50
 - printed output for restoring a user catalog 61
- Interactive Storage Management Facility 21–31
- invoking DFDSS
 - using ISMF 21
 - using JCL 31

- invoking DFDSS (*continued*)
 - with Application Interface 36
- invoking DFDSS via an application program 36
- invoking ISMF 21
- ISAM data sets, copying 82
- ISAM data sets, supported 10
- ISMF
 - line operators 38
 - primary option menu 22
 - starting a session 22
- ISMF panels
 - COPY command entry panel 25
 - data set selection entry panel 23
 - DFDSS execute statement entry panel 26, 30
 - DUMP entry panel 28, 29
 - DUMP job submission entry panel 30
 - ISMF primary option menu 22
 - starting an ISMF session 22
 - successful DUMP job stream 31
 - volume selection 28
 - volume selection entry panel 27
- ISMF use and examples 21–31

J

- JCL requirements for DFDSS 31
- JCL, using to invoke DFDSS 31
- JOB statement
 - DFDSS.JCL 31

K

- key range data sets, supported 10
- keyword
 - module protection 39
 - profile names 39
- KSDS data sets, supported 10

L

- LDS data sets, supported 10
- line operators
 - DFDSS/ISMF 38
- LINECNT = nnnn in the EXEC statement 33
- LINKLIST data sets in a DEFrag operation 106
- location-dependent data 61, 82
- locking a user catalog 60
- logging on to ISMF 21
- logging, RACF 33
- logical data set processing 11
- logical data set RESTORE
 - RACF-protected data sets 57

M

- magnetic tape devices supported 9
 - stand-alone DFDSS restore 117

- main storage requirements
 - DFDSS 7
- manuals, DFDSS iv
- menu-driven panels, ISMF 21
- minivolumes 10
- module name
 - data set application commands 38
- moving a user catalog 80
- moving volumes 73, 84
- multivolume data sets, conversion 97, 98
- MVS environments supported by DFDSS 7
- MVS/Extended Architecture Interactive Storage Management Facility 21

N

- nonindexed VTOCs 10
- NORUN in the EXEC statement 33

O

- online panels, ISMF 21–31
- operating environment 7
- operating system requirements 7
- output from DFDSS (out DD statement) 32
- overview of DFDSS 1

P

- page ejection, suppressing 33
- PAGENO = nnnn in the EXEC statement 33
- PAM data sets
 - storage requirements 9
- PAM data sets, supported 10
- PARM
 - information in the EXEC statement used to control DFDSS 33
- partitioned data set 10
- partitioned data sets extended 10
- PDSE, converting to 19
- PDS, converting to 19
- physical data set processing 11
- preallocated data set COPY usage 76
- printers supported 9
 - stand-alone DFDSS restore 117
- problem program, invoking DFDSS via
 - See invoking DFDSS via an application program
- processing undefined DSORG data sets 19
- profiles
 - defining with RACF installed 78
- publications, DFDSS iv

R

RACF

- data set erase table for DEFrag 108
- data set logical RESTORE 57
- defining profiles 78
- keyword profiles 39

RACF (*continued*)

- protecting keyword modules 39
- RACFLOG= YES in the EXEC statement 33
- readers, card, supported 9
- REBLOCK processing
 - determining block size 106
 - example of JCL 115
 - ineligible data sets 106
 - track usage 106
- recovering system volumes 71
- recovery of a user catalog 60
- related publications vi
- RELEASE command
 - module name 38
- releasing unused space 101
- renaming data sets 75
- requirements
 - stand-alone restore 117
- RESTORE command
 - example of data set 35
 - module name 38
 - printed output
 - produced by integrated catalog facility user catalog 61
 - restrictions, sphere 115
 - spheres, VSAM 114
 - TGTALLOC keyword, omitting 20
- RESTORE operation
 - RACF-protected data sets 57
 - variables passed to ACS routines 110–111
- restoring data sets 53
- restriction
 - converting to and from a PDSE 19
 - COPY processing 81
 - DUMP processing 45
 - moving a PDSE 81
 - physical dump processing 47
 - RESET keyword 45
 - restoring a PDSE 63
 - stand-alone restore program 118
- RRDS data sets, supported 10

S

- SAM data sets, supported 10
- SCAN in the EXEC statement 33
- SIZE = nnnnK in the EXEC statement 33
- SMS Report
 - example 95–96
- source, definition 32
- space fragmentation on DASD 104
- specified operating environment 7
- sphere processing
 - VSAM eligibility 96
- spheres, VSAM
 - processing with COPY, DUMP, RESTORE commands 114

stand-alone DFDSS restore

- compatibility with DFDSS 119
 - hardware and software requirements 117
 - physical processing 13
 - requirements 117
 - restrictions 118
 - system volumes, backing up 13
 - system volumes, recovering 71
 - VM, running stand-alone DFDSS under 71
 - volume DUMP, physical 52
 - what you can restore 1
 - standard label tape recommended 32
 - statements
 - EXEC 31
 - input DD 32
 - JOB 31
 - output DD 32
 - SYSIN DD 32
 - SYSPRINT DD 31
 - statistical information 36
 - storage devices
 - stand-alone DFDSS restore 117
 - Storage Management Subsystem
 - converting to and from 91
 - storage requirements
 - DFDSS 7
 - PAM data sets 9
 - per command
 - above 16-MB 8
 - below 16-MB 7
 - VSAM data sets 9
 - summary of changes
 - Release 2 Version 3 xiv
 - this release xiii
 - Version 2 Release 4 xiv
 - SYSALLDA 74
 - SYSDA 74
 - SYSIN DD statement 32
 - SYSPRINT DD statement
 - DFDSS, JCL 31
 - system consoles supported 9
 - stand-alone DFDSS restore 117
 - system environment 7
 - system requirements 7
 - stand-alone DFDSS restore 117
 - SYS1 data set, processing 19
- ## T
- tape devices supported 9
 - stand-alone DFDSS restore 117
 - tape, standard label, recommended 32
 - target generation data set, specifying status 20
 - target, definition 32
 - temporary work space 74
 - TRACE = YES in the EXEC statement 33
 - TYPRUN = NORUN in the EXEC statement 33

TYPRUN=SCAN in the EXEC statement 33

U

undefined DSORG. processing 19

unused space. releasing 101

user catalog, moving 80

using DFDSS

JCL requirements 31

UTILMSG=YES in the EXEC statement 33

V

vital records, backup 41, 44

VOLUME

entry panel example 27

line operators

module names 38

list 28

volume count subparameter, JCL 32

volume formats supported by DFDSS 10

VSAM

data sets. notes on copying 80

sphere eligibility 96

spheres, considerations for processing 114

VSAM data sets

storage requirements 9

W

when to run DEFrag 104

work space 74

WORKUNIT=workunit in the EXEC statement 34

WORKVOL=volser in the EXEC statement 34

X

XABUFF=ABOVE16;BELOW16 in the EXEC statement 34

Numerics

16 megabyte virtual storage

buffers above 34

storage requirements per command 7, 8

2305 DASD device 9

33xx DASD devices supported 9

Special Characters

* (single asterisk) used in partially qualified data set names 14

** (double asterisk) used in partially qualified data set names 14

Order No. SC26-4388-1

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

If we require further information, may we contact you by phone? ___ Tel. No. _____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail them directly to the address in the Edition Notice on the back of the front cover or title page.)

Reader's Comment Form

Fold and Tape

Please Do Not Staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, NEW YORK



POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department 61C
P.O. Box 27155
Tucson, AZ, USA 85775-4401

Fold and Tape

Please Do Not Staple

Fold and Tape



Order No. SC26-4388-1

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

If we require further information, may we contact you by phone?

_____ Tel. No. _____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail them directly to the address in the Edition Notice on the back of the front cover or title page.)

Reader's Comment Form

Fold and Tape

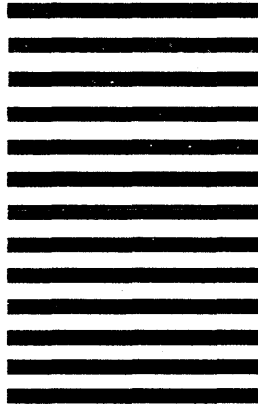
Please Do Not Staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, NEW YORK



POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department 61C
P.O. Box 27155
Tucson, AZ, USA 85775-4401

Fold and Tape

Please Do Not Staple

Fold and Tape



Order No. SC26-4388-1

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

If we require further information, may we contact you by phone? ___ Tel. No. _____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

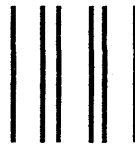
Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail them directly to the address in the Edition Notice on the back of the front cover or title page.)

Reader's Comment Form

Fold and Tape

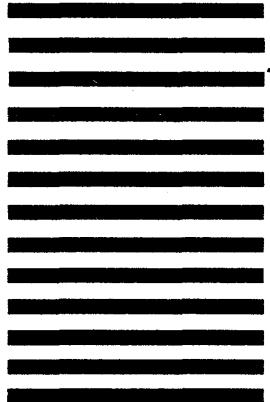
Please Do Not Staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, NEW YORK



POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department 61C
P.O. Box 27155
Tucson, AZ, USA 85775-4401

Fold and Tape

Please Do Not Staple

Fold and Tape



Order No. SC26-4388-1

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

If we require further information, may we contact you by phone? _____ Tel. No. _____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail them directly to the address in the Edition Notice on the back of the front cover or title page.)

Reader's Comment Form

Fold and Tape

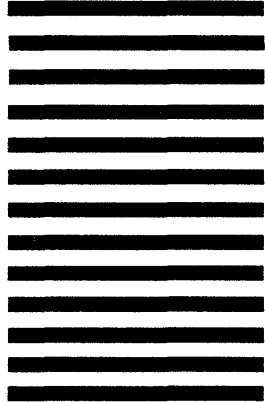
Please Do Not Staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, NEW YORK



POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department 61C
P.O. Box 27155
Tucson, AZ, USA 85775-4401

Fold and Tape

Please Do Not Staple

Fold and Tape

