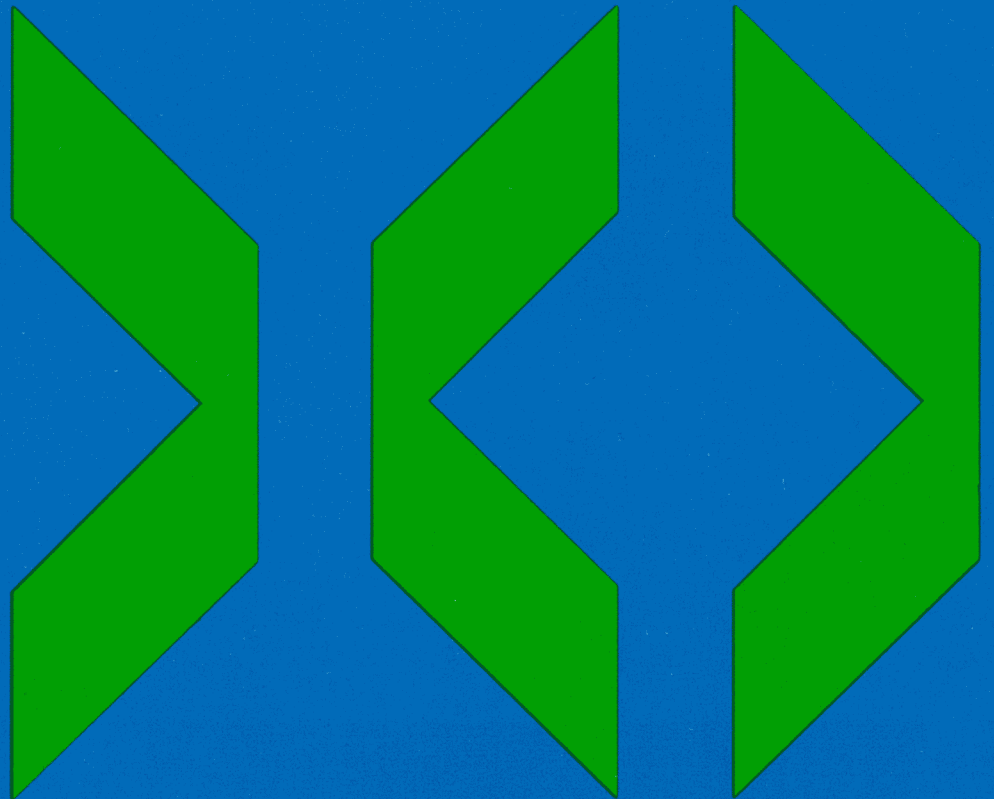




**Data Facility  
Data Set Services:  
User's Guide and Reference**

Licensed  
Program





**Data Facility  
Data Set Services:  
User's Guide and Reference**

Licensed  
Program

#### **Fourth Edition (July 1987)**

This edition replaces and makes obsolete the previous edition, SC26-4125-2.

This edition applies to Version 2 Release 3 of Data Facility Data Set Services, Licensed Program 5665-327, and to any subsequent releases until otherwise indicated in new editions or technical newsletters.

The changes for this edition are summarized under "Summary of Changes" following the preface. Specific changes are indicated by a vertical bar to the left of the change. These bars will be deleted at any republication of the page affected. Editorial changes that have no technical significance are not noted.

Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370, 30xx and 4300 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

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 form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, P.O. Box 50020, Programming Publishing, San Jose, California, U.S.A. 95150. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

## Preface

This book describes how to use Data Facility Data Set Services (DFDSS), and is a guide and reference for system programmers. Don't let the size of this book deter you. There is duplication, to give you information where you need it, rather than referring you to a common section. Messages and other reference material in the appendixes also increase its size.

Just as a picture is worth a thousand words, an example can illustrate what may take pages to explain. For this reason, numerous examples and their explanations are included in Chapter 3; Chapter 5 consists **entirely** of examples.

To find needed information, see:

- The index, which includes entries pointing you to examples that you can use as models for your DFDSS jobs
- The table of contents and figure list

In addition, you will find Appendix E, "Quick Reference to DFDSS Function Commands" on page 309, useful.

This book is organized as follows:

- Chapters 1 through 7 and Appendixes A, B, D, and E apply to DFDSS running under the MVS system environment.
  - Chapter 1 is an overview of DFDSS. It describes what DFDSS is, what you can do with it, logical and physical processing, and the devices and volume formats that can be used.
  - Chapter 2 tells you how to run DFDSS using either ISMF or JCL. It also introduces you to filtering, and describes compatibility with DFDSS Version 1.
  - Chapter 3 describes the syntax of the function commands. It includes DFDSS job examples to illustrate the use of individual commands.
  - Chapter 4 describes the syntax of the auxiliary commands, which allow you to control DFDSS's processing of the function commands in the SYSIN data set.
  - Chapter 5 contains DFDSS job examples, many of which illustrate the use of two or more function commands within one job. It includes a table of contents to help you select the example you want.
  - Chapter 6 contains notes and hints on using DFDSS.
  - Chapter 7 explains how to invoke DFDSS from an application program.
  - Appendix A explains how to write various user exit routines.

- Appendix B lists and explains the DFDSS messages.
- Appendix D contains two data areas, corresponding to control blocks ADRTAPB and ADRBMB.
- Appendix E is a quick reference to the command syntax.
- Chapter 8 and Appendix C apply to DFDSS running in a stand-alone environment.
  - Chapter 8 tells how to use stand-alone DFDSS Restore. It includes examples of job streams to execute it, and storage requirements.
  - Appendix C lists and explains the stand-alone DFDSS Restore messages.

## Data Facility Data Set Services Publications

- *Data Facility Data Set Services: General Information*, GC26-4123, introduces you to DFDSS and helps in evaluating and planning for its use. It is primarily directed to data processing management and to system programmers.
- *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.

## Required Publications

You should be familiar with the information in the *Linkage Editor and Loader User's Guide*, the *JCL* manuals, and the *System Generation* manual for your system. These are cited below under "MVS/370 Publications" and "MVS/Extended Architecture (MVS/XA) Publications."

## MVS/370 Publications

- *MVS/370 Installation: System Generation*, GC26-4063, (**Short Title: System Generation**), describes the device-type notation to be used in the UNIT parameter of a DD statement, and in the unit subparameter of DFDSS commands.
- *MVS/370 JCL User's Guide*, GC28-1349, and *MVS/370 JCL Reference*, GC28-1350, contain the information necessary to code JCL, job entry subsystem 2 (JES2) control statements, and job entry subsystem 3 (JES3) control statements.
- *MVS/370 Linkage Editor and Loader User's Guide*, GC26-4061, describes how to link-edit modules into libraries in the MVS/370 environment.
- *MVS/370 Catalog Administration Guide*, GC26-4053 (**Short Title: Catalog Administration Guide**), describes how to use the integrated catalog facility, and also VSAM catalogs and OS CVOLs.

- *MVS/370 Common VTOC Access Facility Diagnosis Reference*, LY27-9508, (**Short Title:** *Common VTOC Access Facility Diagnosis Reference*), explains the CVAF function codes and error codes.
- *MVS/370 Data Administration: Utilities*, GC26-4065, (**Short Title:** *Utilities*), describes how to use utilities in the MVS/370 environment.
- *MVS/370 DADSM Diagnosis Reference*, LY27-9510, describes the Direct Access Device Space Management (DADSM) return codes.
- *MVS/370 Integrated Catalog Administration: Access Method Services Reference*, GC26-4051, (**Short Title:** *Access Method Services Reference*), describes the access method services commands for the integrated catalog facility catalog.
- *MVS/370 Message Library: System Messages*, Volumes 1 and 2, GC28-1374 and GC28-1375, (**Short Title:** *System Messages*), describe the MVS/370 messages — in particular, the DADSM message IEC602I and VSAM catalog message IDC3009I.
- *MVS/370 System-Data Administration*, GC26-4056, (**Short Title:** *System-Data Administration*), contains information on how the system programmer can modify and extend the data management capabilities of the control program. It also describes the data set control block (DSCB), non-VSAM password protection, and processing DASD volumes with indexed VTOC.
- *MVS/370 VSAM Administration Guide*, GC26-4066, tells you how to use VSAM macro instructions, access method service commands, or JCL to process data.
- *MVS/370 VSAM Administration: Macro Instruction Reference*, GC26-4074, describes VSAM return codes and reason codes.
- *OS/VS Message Library: VS2 System Codes*, GC38-1008 (**Short Title:** *System Codes*).
- *OS/VS2 MVS Supervisor Services and Macro Instructions*, GC28-0683 (**Short Title:** *Supervisor Services and Macro Instructions*), contains information on the PARM parameter of an EXEC statement.
- *OS/VS2 MVS System Programming Library: Job Management*, GC28-0627, describes dynamic allocation.
- *OS/VS2 MVS System Programming Library: Service Aids*, GC28-0674, describes the AMASPZAP, AMDSADMP, and AMBLIST service aids.
- *OS/VS2 MVS System Programming Library: Supervisor*, GC28-0628, describes how to authorize a program, using the Authorized Program Facility (APF) program. DFDSS must be run as an authorized program.

## **MVS/Extended Architecture (MVS/XA) Publications**

- *MVS/Extended Architecture JCL User's Guide*, GC28-1351, and *MVS/Extended Architecture JCL Reference*, GC28-1352, contain the information necessary to code JCL, job entry subsystem 2 (JES2) control statements, and job entry subsystem 3 (JES3) control statements.
- *MVS/Extended Architecture Message Library: System Messages*, Volumes 1 and 2, GC28-1376 and GC28-1377 (**Short Title:** *System Messages*), describe the MVS/XA messages in general and the DADSM message IEC602I and VSAM catalog message IDC3009I in particular.

- *MVS/Extended Architecture Message Library: System Codes*, GC28-1157 (Short Title: *System Codes*).
- *MVS/Extended Architecture System Programming Library: Service Aids*, GC28-1159, describes the AMASPZAP, AMDSADMP, and AMBLIST service aids for MVS/XA.
- *MVS/Extended Architecture System Programming Library: Supervisor Services and Macro Instructions*, GC28-1154 (Short Title: *Supervisor Services and Macro Instructions*), contains information on the PARM parameter of an EXEC statement.
- *MVS/Extended Architecture System Programming Library: System Macros and Facilities*, Volume 1, GC28-1150, describes dynamic allocation. It also describes how to authorize a program, using the Authorized Program Facility (APF) program; DFDSS must be run as an authorized program.

## The Storage Management Library

The MVS/XA Storage Management Library outlines efficient storage management strategies and techniques for MVS/XA systems and explains how you can use today's hardware and software to take advantage of current storage management products and prepare for the transition toward system-managed storage. The concepts of efficient storage management can also be applied to MVS/370 operating systems. To order the complete Storage Management Library, including a binder, order GBOF-1755. The library consists of the following books:

- *MVS/Extended Architecture Storage Management Library: Focus on Storage Management*, GC26-4260, provides management with an overview of efficient storage management concepts, and explains the advantages of both storage management and a storage administration group.
- *MVS/Extended Architecture Storage Management Library: Leading an Effective Storage Administration Group*, GC26-4261, provides advice for the data processing manager on setting up a storage administration department, providing service to user groups, and planning for storage management in the future.
- *MVS/Extended Architecture Storage Management Library: Managing Data Sets*, GC26-4263, provides the storage administrator with information on communicating with user groups, setting up service level agreements, managing user data sets, establishing and enforcing data set policies, and managing catalogs and control data sets.
- *MVS/Extended Architecture Storage Management Library: Managing Storage Pools*, GC26-4264, describes, for the storage administrator, how to plan for, design, implement, and maintain storage pools. It also outlines the tasks that users must perform in order to take advantage of storage pools.
- *MVS/Extended Architecture Storage Management Library: Configuring Storage Subsystems*, GC26-4262, describes, for the storage administrator, how to configure storage hardware to balance performance, space utilization, and availability to meet the goals detailed in a service level agreement. It gives examples of different storage hardware configurations and shows how to avoid various performance and availability problems.
- *MVS/Extended Architecture Storage Management Library: Storage Management Reader's Guide*, GC26-4265, provides the storage administrator with a list of

books in the various libraries for the products that perform essential storage management tasks.

## **MVS/XA Data Facility Product Version 1 Publications**

- *MVS/Extended Architecture Installation: System Generation*, GC26-4009 (**Short Title: System Generation**), describes the device-type notation to be used in the UNIT parameter of a DD statement, and in the unit subparameter of DFDSS commands.
- *MVS/Extended Architecture Linkage Editor and Loader User's Guide*, GC26-4011, describes how to link-edit modules into libraries in the MVS/XA environment.
- *MVS/Extended Architecture Catalog Administration Guide*, GC26-4041 (**Short Title: Catalog Administration Guide**), describes how to use the integrated catalog facility, and also VSAM catalogs and OS CVOLs.
- *MVS/Extended Architecture Common VTOC Access Facility Diagnosis Reference*, SY26-3929 (**Short Title: Common VTOC Access Facility Diagnosis Reference**), explains the CVAF function codes and error codes.
- *MVS/Extended Architecture DADSM Diagnosis Reference*, SY26-3904, describes the Direct Access Device Space Management (DADSM) return codes.
- *MVS/Extended Architecture Data Administration: Utilities*, GC26-4018 (**Short Title: Utilities**), describes how to use utilities in the MVS/XA environment.
- *MVS/Extended Architecture Integrated Catalog Administration: Access Method Services Reference*, GC26-4019 (**Short Title: Access Method Services Reference**), describes the access method services commands for the integrated catalog facility catalog.
- *MVS/Extended Architecture System-Data Administration*, GC26-4010 (**Short Title: System-Data Administration**), contains information on how the system programmer can modify and extend the data management capabilities of the control program. It also describes the data set control block (DSCB), non-VSAM password protection, and processing DASD volumes with indexed VTOC.
- *MVS/Extended Architecture VSAM Administration Guide*, GC26-4015, tells you how to use VSAM macro instructions, access method service commands, or JCL to process data.
- *MVS/Extended Architecture VSAM Administration: Macro Instruction Reference*, GC26-4016, describes VSAM return codes and reason codes.

## **MVS/XA Data Facility Product Version 2 Publications**

- *MVS/Extended Architecture Installation: System Generation*, GC26-4148 (**Short Title: System Generation**), describes the device-type notation to be used in the UNIT parameter of a DD statement, and in the unit subparameter of DFDSS commands.
- *MVS/Extended Architecture Interactive Storage Management Facility User's Guide*, GC26-4266, tells you how to perform many of the DFDSS space management and backup functions using the interactive storage management facility (ISMF).

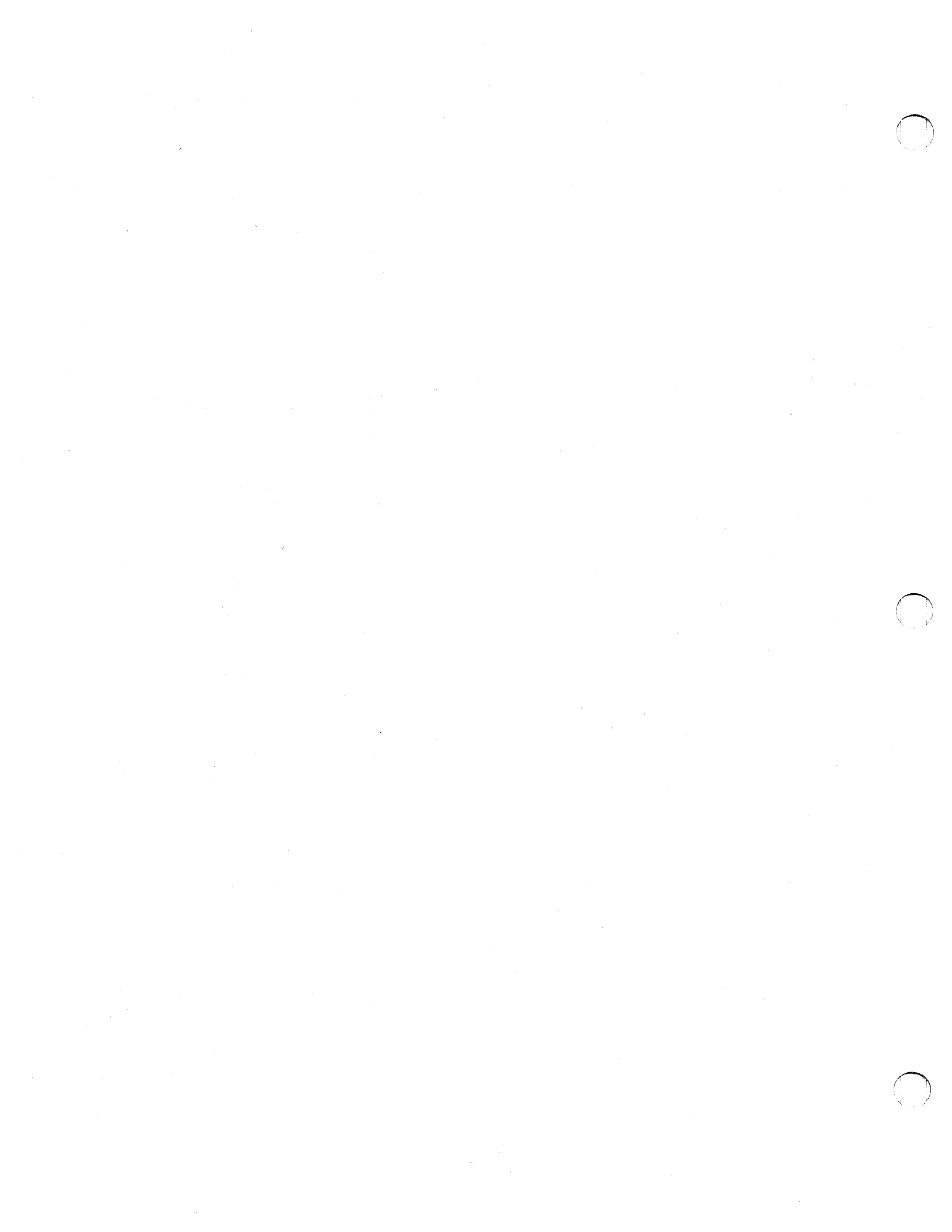


- *MVS/Extended Architecture Linkage Editor and Loader User's Guide*, GC26-4143, describes how to link-edit modules into libraries in the MVS/XA environment.
- *MVS/Extended Architecture Catalog Administration Guide*, GC26-4138 (**Short Title: *Catalog Administration Guide***), describes how to use the integrated catalog facility, and also VSAM catalogs and OS CVOLs.
- *MVS/Extended Architecture Common VTOC Access Facility Diagnosis Reference*, LY26-3958 (**Short Title: *Common VTOC Access Facility Diagnosis Reference***), explains the CVAF function codes and error codes.
- *MVS/Extended Architecture Data Facility Product Version 2: Diagnosis Reference*, LY27-9530, describes the Direct Access Device Space Management (DADSM) return codes.
- *MVS/Extended Architecture Data Administration: Utilities*, GC26-4150 (**Short Title: *Utilities***), describes how to use utilities in the MVS/XA environment.
- *MVS/Extended Architecture Integrated Catalog Administration: Access Method Services Reference*, GC26-4135 (**Short Title: *Access Method Services Reference***), describes the access method services commands for the integrated catalog facility catalog.
- *MVS/Extended Architecture System-Data Administration*, GC26-4149 (**Short Title: *System-Data Administration***), contains information on how the system programmer can modify and extend the data management capabilities of the control program. It also describes the data set control block (DSCB), non-VSAM password protection, and processing DASD volumes with indexed VTOC.
- *MVS/Extended Architecture VSAM Administration Guide*, GC26-4151, tells you how to use VSAM macro instructions, access method service commands, or JCL to process data.
- *MVS/Extended Architecture VSAM Administration: Macro Instruction Reference*, GC26-4152, describes VSAM return codes and reason codes.

## Other Publications

- *IBM 3380 Direct Access Storage: Migration*, GC26-4197, describes how data can be moved from an IBM 3350 Direct Access Storage to an IBM 3380 Direct Access Storage.
- *Device Support Facilities User's Guide and Reference*, GC35-0033, describes how to initialize and maintain direct access storage device (DASD) volumes.
- *Data Facility Hierarchical Storage Manager: General Information*, GH35-0092, provides an overview of the Data Facility Hierarchical Storage Manager (DFHSM).
- *Data Facility Hierarchical Storage Manager: System Programmer's Command Reference*, SH35-0083, describes DFHSM functions and commands and how they are used by the system programmer.
- *OS/VS2 MVS Planning: Global Resource Serialization*, GC28-1062, contains information on how to serialize access to data sets on shared DASD volumes.

- *Resource Access Control Facility (RACF) General Information Manual, GC28-0722*, provides an overview and planning information for the RACF program.



## Summary of Changes

### Version 2 Release 3, July 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, DEFrag and full-volume DUMP.
  - DFDSS supports Enhanced ICF 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 keyword parameters 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 parameter 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.

## Version 2 Release 2

Version 2 Release 2 of DFDSS is upward-compatible with, and contains all the functions of, DFDSS Version 2 Release 1 and earlier releases (of Version 1). In addition, it includes:

- Data movement enhancements
  - You can restore data sets to unlike DASD from logical dump tapes. These include VSAM (ESDS, RRDS, KSDS, and IBM DATABASE 2 [DB2]), SAM, PAM, and selected DAM data sets.
  - You can copy to preallocated data sets, thus controlling data set placement on the target volume.
  - You can restore to a data set that is equal to or larger than the source data set without first scratching and reallocating the target space. You must specify REPLACE on your RESTORE command.
  - You can copy and restore DB2 data sets to unlike devices.
  - You can copy VSAM ESDS and RRDS data sets to unlike devices faster.
  - Partitioned data sets are compressed during data set RESTORE.
- Data set passwords residing in the DFDSS input command stream are suppressed, and will not appear in the SYSPRINT output.
- You can place I/O buffers above or below 16 megabytes virtual storage by specifying the XABUFF parameter on your EXEC statement. The default, when running under MVS/XA, is that DFDSS uses buffers above 16 Mb.
- Specification of the model volser (MVOLSER) keyword is now optional when the model entity (MENTITY) is cataloged.
- Additional user exit points are provided for security verification and for I/O errors on standard labeled tapes.

In addition, if DFDSS runs under MVS/Extended Architecture Data Facility Product Version 2 Release 2:

- You can use the interactive storage management facility (ISMF) to perform many of the DFDSS space management and backup functions by using menu-driven panels.
- DFDSS supports the VSAM N\*512 block size scheme during data set COPY and RESTORE.

You have RACF protection of tape data sets, if RACF Version 1 Release 7 is installed.

DEFRAG takes advantage of the RACF erase-on-scratch support, which overwrites the old version of the data set with binary zeros, if RACF Version 1 Release 7 is installed. If you also have MVS/XA Data Facility Product Version 2 Release 1 (or later release) installed, DFDSS overwrites the old data set with binary zeros during a DUMP with DELETE, a data set COPY with DELETE, or a RELEASE.

# Stand-Alone DFDSS Support for System/370 Extended-Architecture Mode

Stand-alone DFDSS now operates on IBM extended architecture processors (for example, IBM 308x series) in either System/370 extended-architecture mode or System/370 mode.

## Version 2 Release 1

Version 2 of Data Facility Data Set Services (DFDSS) is a new version of DFDSS. It is upward-compatible with, and contains all the functions of, DFDSS Version 1 Release 2.1 and earlier releases. In addition, it includes:

- Data movement enhancements
  - You can copy data sets between DASD devices of like or unlike type; for example, you can move data sets between existing DASD types and newer types.
  - In a data set DUMP, if you do **not** specify input volume(s), data set information obtained from catalog(s) is used for data set selection. The result is a **logical data set DUMP**, in which data sets are treated as logical entities. Thus, DFDSS processes an entire data set before it begins processing the next data set.

In a data set RESTORE if the input was created by a logical data set DUMP, a logical data set RESTORE results.

In a data set COPY, a logical data set COPY is always performed, whether or not input volume(s) are specified.
  - You can rename data sets during a COPY operation.
  - You can reblock SAM or PAM data sets during a COPY operation.
  - You can uncatalog and/or catalog single- and multivolume data sets during COPY, DUMP, and RESTORE operations.
  - You can dump or restore multivolume data sets with a *logical* data set DUMP or data set RESTORE.
- You can compress partitioned data sets
- You can filter data sets in more ways than before
- DEFrag enhancements
  - You no longer need the WORKSIZE parameter
  - You can do data set filtering using BY criteria
- You can invoke DFDSS from a problem program, using the application interface.
- You have enhanced readability of DFDSS output messages. Messages are now grouped by task.

In addition, if DFDSS runs under MVS/XA Data Facility Product Version 2 Release 1:

- You can release allocated but unused space from SAM or PAM data sets
- You can copy or restore unmovable data sets (PSU, POU, DAU, or ABSTR) to an identical location on a target of like device type, if the target has an indexed VTOC and the location is available.

# Contents

<b>Chapter 1. Overview</b> .....	<b>1</b>
Things You Can Do With DFDSS .....	1
Sysgen Support .....	2
Volumes, Tracks, and Data Sets .....	2
Logical and Physical Processing .....	3
Advantages of Logical Processing .....	3
Selecting Data Sets via the Catalog or the VTOC .....	4
Advantages of Physical Processing .....	4
Data Set Organizations You Can Use .....	5
Devices You Can Use .....	5
Volume Formats You Can Use .....	5
Specified Operating Environment .....	6
<b>Chapter 2. Using Data Facility Data Set Services</b> .....	<b>7</b>
Using DFDSS via the Interactive Storage Management Facility (ISMF) .....	7
How to Invoke ISMF .....	8
Examples of Using DFDSS via ISMF .....	9
Using the Data Set Application .....	9
Copying the Data Sets to a 3380 .....	11
Using the Volume Application .....	13
Dumping the Volume .....	14
How You Invoke DFDSS via Job Control Language .....	17
The JCL You Need .....	17
How You Control DFDSS via PARM Information in the EXEC Statement .....	19
Example 1: Moving a Data Set .....	20
Example 2: Dumping a Data Set .....	20
Example 3: Restoring a Data Set .....	20
Choosing the Data Sets You Want Processed—Filtering .....	21
Using Qualified Data Set Names .....	21
The Use of Quotation Marks .....	23
The Filtering Process .....	23
Standard Catalog Search Order .....	24
Compatibility Between Version 1 and Version 2 .....	24
<b>Chapter 3. Specifying Tasks—Function Commands</b> .....	<b>29</b>
Specifying DFDSS Commands—Command Syntax .....	29
Understanding the Command Syntax .....	30
How Many Subparameters Are Allowed? .....	31
About the Examples in This Manual .....	32
COMPRESS Command .....	33
Explanation of COMPRESS Command Parameters .....	33
Example of a COMPRESS .....	37
COMPRESS Command Usage .....	37



COPY Command .....	38
Explanation of COPY Command Parameters .....	42
Examples of Full and Tracks COPY Operations .....	54
Examples of Data Set COPY Operations .....	55
COPY Command Usage .....	58
Preallocated Data Set COPY Usage .....	62
COPYDUMP Command .....	65
Explanation of COPYDUMP Command Parameters .....	65
Examples of COPYDUMP Operations .....	66
COPYDUMP Command Usage .....	67
DEFRAG Command .....	68
Explanation of DEFRAG Command Parameters .....	69
Examples of DEFRAG Operations .....	74
Results of a Successful DEFRAG Operation .....	75
DEFRAG Command Usage .....	75
DUMP Command .....	79
Logical Volumes of a Physical Dump .....	79
Error Termination .....	80
Authorization Checking .....	80
Input Serialization .....	80
Explanation of DUMP Command Parameters .....	83
Examples of Full and Tracks DUMP Operations .....	91
Examples of Data Set DUMP Operations .....	91
DUMP Command Usage .....	94
PRINT Command .....	97
Explanation of PRINT Command Parameters .....	99
Examples of PRINT Operations .....	102
RELEASE Command .....	104
Explanation of RELEASE Command Parameters .....	104
Example of a RELEASE Operation .....	108
RELEASE Command Usage .....	108
RESTORE Command .....	109
Explanation of RESTORE Command Parameters .....	112
Examples of Full and Tracks RESTORE Operations .....	120
Examples of Data Set RESTORE Operations .....	121
Data Set RESTORE—RENAME, RENAMEUNCONDITIONAL, and REPLACE Parameters .....	126
RESTORE Command Usage .....	126
<b>Chapter 4. Controlling Tasks—Auxiliary Commands .....</b>	<b>135</b>
Writing to the Operator .....	135
Operating in Serial or Parallel Mode .....	135
Controlling Task Execution .....	136
Condition Codes .....	136
SET Command .....	136
IF-THEN-ELSE Command Sequence .....	137
DO-END Command Sequence .....	138
Null Command .....	138
Terminating Your DFDSS Job .....	139
Examples of Controlling Task Execution .....	139
Common Continuation Errors .....	140
Permissible Truncations of Auxiliary Commands .....	141
<b>Chapter 5. Examples of DFDSS Jobs .....</b>	<b>143</b>

Selecting an Example to Use as a Model .....	143
Example 1: Device Conversion—A Data Set COPY to an Unlike Device .	144
Example 2: A Full DUMP, Followed by a DEFRAG .....	145
Example 3: A Physical Data Set DUMP .....	146
Example 4: A Physical Data Set DUMP with Various Parameters .....	146
Example 5: A Data Set RESTORE .....	148
Example 6: A Full DUMP, Followed by a COPYDUMP .....	148
Example 7: A Physical Data Set DUMP with the DELETE Parameter ...	149
Example 8: A Full Volume COPY .....	150
Example 9: Full DUMPs, Followed by DEFRAGs .....	151
Example 10A: Backing Up an Integrated Catalog Facility User Catalog and the VSAM Data Sets Cataloged in It .....	152
Example 10B: Restoring an Integrated Catalog Facility User Catalog ...	152
Example 10C: Restoring VSAM Data Sets Cataloged in an Integrated Catalog Facility User Catalog .....	153
Example 10D: Restoring a Single VSAM Data Set Cataloged in an Integrated Catalog Facility User Catalog .....	154
Example 11: A Data Set RESTORE With the RENAMEUNCONDITIONAL Parameter .....	155
Example 12: Executing DUMP, COMPRESS, and RELEASE on a Volume .....	156
<b>Chapter 6. Notes on Using DFDSS .....</b>	<b>159</b>
Restrictions .....	159
ADRMCLVL Macro—Dynamically Determining the DFDSS Version, Release, and Modification Level .....	159
Allocating Volumes Dynamically .....	160
Block Size .....	161
DASD Space Utilization .....	161
DEFRAG Function .....	161
General Hints .....	162
Space Management DUMP Techniques .....	163
DASD Space Reclamation .....	163
Combining Data Set Extents .....	163
Data Integrity .....	164
WAIT(numsecs,numretrys) Option for Serialization .....	164
RESERVE .....	165
ENQ .....	165
An Example of RESERVE-ENQUEUE Processing .....	166
DYNALLOC .....	167
How DFDSS Checks Authorization for You .....	167
DASD Volume Authorization Checking .....	168
Data Set Authorization Checking .....	169
RACF Logging During Authorization Checking .....	171
Using RACF for ISMF Authorization .....	171
ISMF Function You May Want to Protect .....	171
Setting Up the Authorization Structure .....	172
Finding the DFDSS/ISMF Module Names .....	172
Protecting DFDSS/ISMF Modules .....	173
Data Security .....	174
Installation Authorization Exit .....	174
Example of Security Processing .....	175
Data Set Passwords—Abend Dumps .....	175
DUMP Operation—Dumping to a DASD Sequential Data Set .....	175
DUMP Operation—Format of the Output Data Set .....	176

DUMP Operation—OPTIMIZE Parameter .....	176
Dump/Restore Techniques and Considerations .....	177
Enqueuing versus Dynamic Allocation of Data Sets .....	178
RESERVE-ENQUE-DYNALLOC Matrix .....	178
Full or Tracks DUMP/RESTORE Integrity Considerations .....	179
Shared DASD Considerations .....	180
Filtering Criteria and/or Passwords—Passing via a Data Set .....	180
Lockouts during DFDSS Runs .....	180
Storage Requirements—DFDSS .....	181
Multivolume Data Sets—Processing .....	181
PARM Statement—“SIZE=” .....	183
Read/Write Access Serialization Scheme .....	183
RESTORE Operation—Full Volume Restore and RACF-Protected Data Sets .....	183
RESTORE Operation—Full Volume or Data Set Restore and RACF Generic Profiles .....	185
RESTORE Operation—Logical Data Set RESTORE and RACF-Protected Data Sets .....	185
RESTORE Operation—Renaming RACF-Protected Data Sets During Data Set Restore .....	186
RESTORE Operation—Restore Unmovable (ISAM and Absolute Track) Data Sets .....	186
RESTORE Operation—Tracks RESTORE .....	186
SYSPRINT DD Statement .....	187
<b>Chapter 7. How You Invoke DFDSS via an Application Program .....</b>	<b>189</b>
Invocation .....	189
Calling Block Structure .....	192
User Interactions .....	193
System Programming Information .....	193
Examples of Invoking DFDSS .....	197
Invoke DFDSS via Job Control Language .....	197
Invoke DFDSS via an Application Program .....	198
Invoke DFDSS, Supplying Options .....	199
Invoke DFDSS, with ddname Overrides .....	199
Invoke DFDSS, with Starting Page Number .....	200
Sample User Interaction Module (UIM) .....	201
Invoke DFDSS, Supplying a New UIM .....	202
Invoke DFDSS Supplying a User Area Pointer .....	204
Disable UIM Usage .....	205
Disable the UIM From the User Installation Options Exit .....	205
Disable UIM Usage via the UIM .....	205
Prevent Command Execution via the UIM .....	206
Alter an Installation Option via the UIM .....	207
Supply In-Storage SYSIN Data and Delete SYSPRINT Records .....	208
Modify SYSIN Data .....	210
Recover from an Incorrect Exit Action .....	210
Disable Functional Exit via the UIM .....	212
Create a Function-Taskid Table via SYSPRINT Data .....	213
Request Only User Statistics Records .....	215
Process WTO/WTOR Records via the UIM .....	216
Termination Exit Processing in the UIM .....	216
Link-editing Your UIM .....	218
ADREID0 Data Area .....	219

<b>Chapter 8. Using Stand-Alone DFDSS Restore</b> .....	<b>221</b>
Devices You Can Use .....	221
Restrictions .....	222
Compatibility with System DFDSS .....	223
Input and Output .....	223
Control .....	223
Control Statements .....	223
Continuing Control Statements .....	224
Generic Device Names .....	224
JOB Statement .....	224
MSG Statement .....	224
RESTORE Statement .....	225
VDRL Statement .....	226
END Statement .....	227
Copying Stand-Alone DFDSS Restore to Tape .....	227
Executing Stand-Alone DFDSS Restore .....	227
WAIT State Codes .....	229
Examples of Stand-Alone DFDSS Restore Jobs .....	229
Example 1: A Full RESTORE Operation .....	229
Example 2: A Tracks RESTORE Operation .....	229
Example 3: Same as Example 2, Using Generic Device Names .....	230
Storage Requirement .....	230
<b>Appendix A. User Exit Routines</b> .....	<b>231</b>
How to Write a User Authorization Exit Routine .....	232
How to Write a User Enqueue Exit Routine .....	235
How to Write a User Installation Options Exit Routine .....	237
ADRUFO Data Area .....	239
How to Write a User Reblock Exit Routine .....	243
<b>Appendix B. DFDSS Messages</b> .....	<b>245</b>
Problem Determination .....	291
Table I .....	291
<b>Appendix C. Stand-Alone DFDSS Restore Messages</b> .....	<b>293</b>
Problem Determination .....	299
Table I .....	299
<b>Appendix D. Data Area Descriptions</b> .....	<b>301</b>
ADRBMB Data Area .....	301
ADRTAPB Data Area .....	302
<b>Appendix E. Quick Reference to DFDSS Function Commands</b> .....	<b>309</b>
<b>Glossary</b> .....	<b>321</b>
<b>Index</b> .....	<b>323</b>



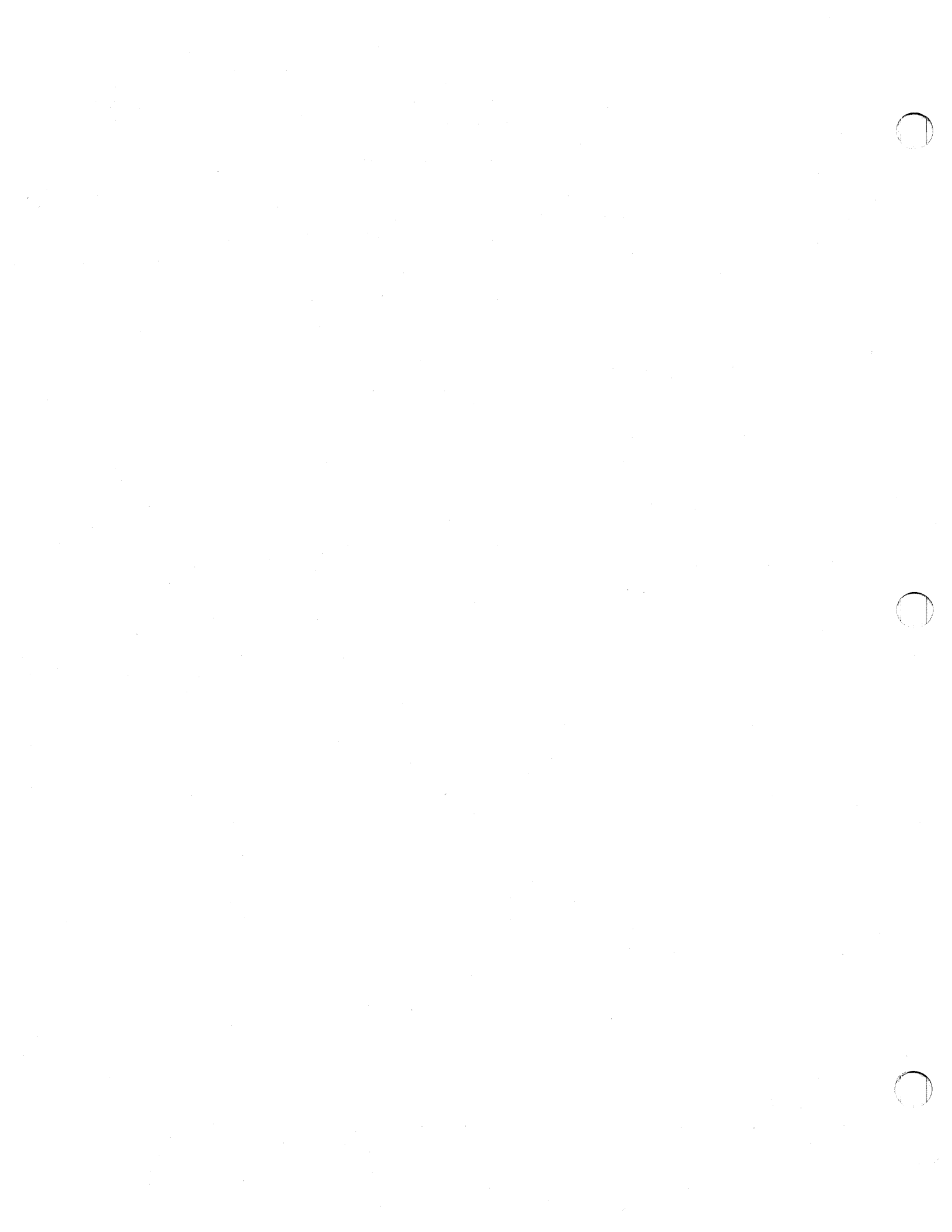
## Figures

1.	Starting an ISMF Session from the ISPF Master Application Menu	8
2.	ISMF Primary Option Menu	9
3.	Complete Page 1 of the Data Set Selection Entry Panel	10
4.	The Resulting Data Set List	10
5.	Completing Page 1 of the COPY Command Entry Panel	11
6.	Completing Page 2 of the COPY Command Entry Panel	11
7.	Verifying the COPY Job Statement Information	12
8.	The DFDSS Execute Statement Entry Panel	12
9.	The List with Successful Submission of the COPY Command	13
10.	Complete Page 1 of the Volume Selection Entry Panel	13
11.	The Volume List	14
12.	Page 1 of the DUMP Entry Panel	14
13.	Page 2 of the DUMP Entry Panel	15
14.	Completing the DUMP Entry Panel	15
15.	DUMP Job Submission Entry Panel	16
16.	DFDSS EXECUTE Statement Entry Panel	16
17.	The DUMP Job Stream is Successfully Built	17
18.	DFDSS JCL	18
19.	System DFDSS RESTORE Compatibility among Releases	26
20.	Stand-Alone DFDSS RESTORE Compatibility among Releases	27
21.	COMPRESS Command Syntax	33
22.	BY Parameters—COMPRESS Command	35
23.	COPY Command Syntax—FULL or TRACKS	40
24.	COPY Command Syntax—DATASET	41
25.	BY Parameters—COPY Command	45
26.	Data Set COPY ALLDATA and ALLEXCP Results (When REBLOCK Is Not Specified)	47
27.	Defining Profiles—With RACF 1.5 or Later Installed	60
28.	Data Mover Selection Matrix for Data Set COPY	61
29.	COPYDUMP Command Syntax	65
30.	DEFRAG Command Syntax	69
31.	BY Parameters—DEFRAG Command	72
32.	Printed Output Resulting from a Successful DEFRAG Run	76
33.	Data Set Erase Table for DEFRAG (with RACF Version 1 Release 7)	77
34.	DUMP Command Syntax—FULL or TRACKS	81
35.	DUMP Command Syntax—DATASET	82
36.	BY Parameters—DUMP Command	85
37.	PRINT Command Syntax	98
38.	Output Resulting from a PRINT Command	103
39.	RELEASE Command Syntax	104
40.	BY Parameters—RELEASE Command	106
41.	RESTORE Command Syntax—FULL or TRACKS	110
42.	RESTORE Command Syntax—DATASET	111
43.	BY Parameters—RESTORE Command	114

44.	RESTORE Actions on Movable Data Sets	126
45.	Restore Actions on Non-VSAM Movable Data Sets	127
46.	Example Table of Contents	144
47.	Example 10A Printed Output	153
48.	Example 10B Printed Output	153
49.	Example 10C Printed Output	154
50.	Example 10D Printed Output	155
51.	Expansion of Macro ADRMCLVL	160
52.	Data Set Enqueue Options for Non-VSAM Data Sets	166
53.	Access Authority Requirement—RACF-Protected Data Sets and DAS DVOL	170
54.	Module Names for DFDSS/ISMF Applications	173
55.	Module Names for DFDSS/ISMF Line Operators	173
56.	Module Names for DFDSS/ISMF Data Set Application Commands	173
57.	Resource Serialization	178
58.	Minimum Storage Requirements for Each DFDSS Operation	182
59.	Minimum Storage Requirements for a Restore to an Unlike Device	183
60.	Read/Write Access Serialization Scheme	184
61.	RACF-Protected Data Sets in a Logical Data Set RESTORE	185
62.	Invoking DFDSS via Its Application Interface	189
63.	DFDSS Application Interface Structure	192
64.	DFDSS Exit Interface Structure	194
65.	DFDSS Exit Identification Block	194
66.	Invoke DFDSS via JCL	198
67.	Invoke DFDSS from a User Program	198
68.	Invoke DFDSS via a LINK	199
69.	Invoke DFDSS with Options	199
70.	Alternate SYSIN for DFDSS	200
71.	Override SYSIN ddname	200
72.	Dynamic Page Numbering on SYSPRINT	201
73.	Sample User Interaction Module (UIM)	203
74.	Invoke DFDSS and Use a UIM Module	204
75.	Invoke DFDSS Supplying a User Area Pointer	204
76.	Disable UIM from ADRUIXIT	205
77.	Disable UIM from a UIM User Installation Options Exit	206
78.	Disable a Function from a UIM User Installation Options Exit	207
79.	Alter UFO from a UIM Exit	208
80.	Invoke DFDSS for In-Storage I/O	208
81.	In-Storage SYSIN and SYSPRINT	209
82.	Modify the SYSIN Data Stream	211
83.	Recognize a Disallowed Option	212
84.	Disconnect a Single UIM Exit	213
85.	Table Function-Task ID Information	214
86.	Request Statistics Records Only	215
87.	Handle WTOR Messages in a UIM	216
88.	Function Termination Exit Processing	217
89.	Link-editing Your UIM	218
90.	ADREID0 Data Area	219
91.	User Authorization Exit Routine Return Codes	234
92.	ADRUF0 Data Area	239
93.	Example of Link-Editing a User Reblock Routine	244
94.	ADRBMB Data Area	301
95.	ADRTAPB Data Area	302
96.	COMPRESS Command Syntax	309
97.	COPY Command Syntax—FULL or TRACKS	310

98.	COPY Command Syntax—DATASET .....	311
99.	COPYDUMP Command Syntax .....	312
100.	DEFRAG Command Syntax .....	313
101.	DUMP Command Syntax—FULL or TRACKS .....	314
102.	DUMP Command Syntax—DATASET .....	315
103.	PRINT Command Syntax .....	316
104.	RELEASE Command Syntax .....	317
105.	RESTORE Command Syntax—FULL or TRACKS .....	318
106.	RESTORE Command Syntax—DATASET .....	319





## Chapter 1. Overview

Data Facility Data Set Services (DFDSS) is a direct access storage device (DASD) data and space management tool. Its functions include copying or moving data sets between volumes of like or unlike device types,<sup>1</sup> dumping and restoring data sets as well as entire volumes or specific tracks, and reducing or eliminating DASD free-space fragmentation.

DFDSS also includes a stand-alone DFDSS Restore program (ADRDMPRS), which executes without a host environment. Alternatively, it runs in a VM/370 virtual machine. The stand-alone DFDSS Restore program is limited to one function—a full or tracks RESTORE from a physical dump tape. It is described in Chapter 8.

### Things You Can Do With DFDSS

Some of the significant features of DFDSS are:

- The **interactive storage management facility (ISMF)**, which uses interactive menu-driven panels to build job streams for most of the DFDSS functions.
- A **COPY** command to copy data from one DASD volume to another. You can copy data sets (in which case DASD may be of unlike device type), or a full volume or ranges of tracks (in which case DASD must be of like device type). You can move most types of single- and multivolume data sets between old and new DASD devices.
- A **DUMP** command and a **RESTORE** command to dump DASD data to, and restore data from, a sequential data set. The data can be a full volume, ranges of tracks, or single- or multivolume data sets. The storage medium for the sequential data set can be a tape, DASD, or a mass storage virtual volume. Data sets can be restored to unlike DASD from logical dump tapes.
- A **DEFRAG** command to reduce or eliminate DASD free-space fragmentation.
- A **PRINT** command to print DASD data. You can print ranges of tracks, a non-VSAM data set or a component of a VSAM data set, or all or part of the VTOC.
- A **COMPRESS** command to compress partitioned data sets on a specified volume(s).

---

<sup>1</sup> Like devices have the same track capacity and number of tracks per cylinder (for example, 3380 Standard to 3380 Model D or E). Unlike DASD devices have different track capacities, a different number of tracks per cylinder (for example, 3350 to 3380), or both.

- A **RELEASE** command to release, on a specified volume(s), allocated but unused space from sequential and partitioned data sets.
- A **COPYDUMP** command to create additional copies of DFDSS-produced dump data. The data to be copied can be on tape or DASD, and copies can be written to tape or DASD.
- You can specify **filtering** in the data set **COPY**, data set **DUMP**, data set **RESTORE**, **DEFRAG**, **COMPRESS**, and **RELEASE** commands. Filtering is 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.
- With the use of the above **function** commands and the decision-making facilities of DFDSS's **auxiliary** commands, you can control the flow of tasks to be executed within a job, based on task return codes and specified criteria. You can also execute two or more tasks concurrently.
- **User exit routines** supplied with DFDSS do:
  - Data set authorization checking
  - Enqueuing on the VTOC
  - Installation options
  - Data set reblocking during a data set **COPY**

If you have special installation requirements, you can replace these exit routines with your own routines.

- You can invoke DFDSS from another program, using the "Application Interface." The Application Interface allows you to interact with DFDSS at various points during processing.
- Passwords are checked for password-protected data sets and the following Resource Access Control Facility (RACF) functions are supported:
  - RACF authorization at the volume and at the data set level
  - Generic profiles
  - System Authorization Facility functions
- DFDSS permits you to retrieve readable data from a volume even if that volume has severe errors, providing the volume is online.

## System Support

If you do a system generation, you need not reinstall DFDSS nor reapply DFDSS maintenance after the system generation.

## Volumes, Tracks, and Data Sets

When you use the **COPY**, **DUMP**, and **RESTORE** commands, you specify that you want the operation performed on either the entire DASD volume (by specifying **FULL**), a range of tracks (by specifying **TRACKS**), or the data sets that you select (by specifying **DATASET**). Other commands, by their nature, process either volumes or data sets. For instance, **DEFRAG** defragments volumes. **COMPRESS** does a rather similar thing to partitioned data sets. For more detail, see Chapter 3.

## Logical and Physical Processing

DFDSS offers two kinds of processing when executing COPY, DUMP, and RESTORE commands: logical and physical. Logical processing is data set-oriented and operates against data sets and volumes independently of physical device format. Physical processing can operate against volumes, tracks, and data sets, but is oriented towards moving data at the track-image level. Each type of processing offers different capabilities and advantages, described below.

The processing method is determined by the parameters specified on the COPY or DUMP command, and in the case of RESTORE, it is chosen by DFDSS. Physical processing is selected when:

- The FULL or TRACKS parameter is used, resulting in a physical volume or physical tracks operation.
- The DATASET parameter is used on the DUMP command and input volumes are specified 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.

Logical processing is selected when:

- You specify DATASET with COPY. A data set COPY is always a logical operation, regardless of how or whether you specify input volumes.
- You specify DATASET with DUMP, and either no input volume is specified, or LOGINDDNAME or LOGINDDYNAM is used to specify an input volume(s).
- The RESTORE command is executed and the input volume was created by a logical DUMP.

Physical and logical dump tapes have different formats.

## Advantages of 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. Both catalogs and VTOCs are used to select data sets. Each data set is moved as a set of data records, allowing data movement between devices with different track and cylinder configurations.

Logical processing can also be specified at the volume level (using LOGINDDNAME or LOGINDDYNAM). If an input volume(s) is specified for a logical COPY or DUMP, some or all of the data sets on that volume could be processed, allowing data movement for entire volumes. For example, you can copy an entire volume to another volume or back up and restore an entire volume.

Logical processing must be used for device conversion and backup/recovery processing if any of the following requirements exists at your installation:

- Data will be copied to an unlike device type, or data will be dumped from one type of DASD device and restored to an unlike device.

- Aliases of VSAM user catalogs are to be preserved during copy and restore.
- Unmovable data sets or data sets with absolute track allocation will be moved to different locations.
- Multivolume data sets are to be moved.
- 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.
- For copy and restore, the percentage of space allocated on each of the output volumes is to be controlled.

### Selecting Data Sets via the Catalog or the VTOC

When you do logical data set processing, DFDSS uses:

- The catalog(s) for selecting data sets for COPY if you do not specify input volume(s)
- The VTOC(s) for selecting data sets for COPY if you do specify input volume(s)
- The catalog(s) for selecting data sets for DUMP if you do not specify input volume(s)
- The VTOC(s) for selecting data sets for DUMP if you specify LOGINDDNAME or LOGINDYNAM

If you wish to copy or dump entire multivolume data sets, you need not specify all the volumes in the LOGINDDNAME or LOGINDYNAM volume list if you also specify the ALLMULTI keyword.

### Advantages of Physical Processing

Physical processing moves data based on physical track images. Since data movement is carried out at the track level, only target devices with track sizes equal to the source device are supported. Physical processing operates on volume(s), a range or 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 will process only that part of a data set residing on the specified input volume(s).

Physical processing has these advantages:

- The stand-alone DFDSS Restore program supports only a physical dump tape.
- Generally, the fastest way (measured by elapsed time) to copy or dump an entire volume is with a physical full volume command. This is largely due to the absence of catalog searching during physical processing.
- 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. This may be useful when substituting one physical DASD volume for another, or recovering an entire volume.

- Physical processing provides the capability to COPY, DUMP, and RESTORE a specific track or range of tracks.

## Data Set Organizations You Can Use

DFDSS can copy, dump, and restore data sets of the following organization types:

- SAM (sequential access method)
- PAM (partitioned access method)
- BDAM (basic direct access method)
- ISAM (indexed sequential access method)
- VSAM (virtual storage access method), including:
  - ESDS (entry sequenced data set)
  - RRDS (relative record data set)
  - KSDS (key sequenced data set)
  - key range data set
  - LDS (linear data set)
- DATABASE 2 (DB2)
- EXCP (execute channel program)

*Note:* In this book, unless otherwise indicated, *VSAM data set(s)* refers to VSAM data set(s) cataloged in an integrated catalog facility catalog.

## Devices You Can Use

DFDSS can be used with the following IBM **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 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, DFDSS can be used with all **magnetic tape devices, system consoles, printers, and card readers** supported by Data Facility Product (DFP).

## Volume Formats You Can Use

DFDSS can be used with the following DASD volume formats:

Volumes with indexed VTOCs  
 Volumes with nonindexed VTOCs  
 OS/VS minivolumes in a VM/370 environment

All DASD volumes used by DFDSS must be initialized by Device Support Facilities (ICKDSF).

## | Specified Operating Environment

| For the MVS environments under which DFDSS runs, see *Data Facility Data Set Services: General Information*. It also lists the DFDSS functions that require a specific version or release of the Data Facility Product (DFP) or another licensed program.

| *Note:* VSAM data sets residing on DFDSS data set Dump tapes created in an MVS/XA DFP V2.2 (or later release) environment cannot be restored in an MVS/XA DFP V2.1 environment without certain VSAM N\*512 toleration PTFs.

## Chapter 2. Using Data Facility Data Set Services

This chapter contains information you will need as background for the rest of the book. It describes such topics as:

- Using DFDSS via the interactive storage management facility
- Invoking DFDSS via JCL—a description of the JCL requirements, including examples
- Choosing the data sets you want processed—a description of filtering
- Compatibility between the versions of DFDSS

You can also invoke DFDSS from an application program. For information on this, see “Chapter 7. How You Invoke DFDSS via an Application Program” on page 189.

### Using DFDSS via the Interactive Storage Management Facility (ISMF)

Using the interactive storage management facility (ISMF), a component of the MVS/XA Data Facility Product, you can build job streams for many of the space management and backup functions of DFDSS using menu-driven panels. ISMF supports the COMPRESS, COPY, DEFRAG, DUMP, RELEASE, and RESTORE commands.

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 or 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 JCL for later use.

Using ISMF panels, you can also construct a list of attributes of specific data sets or volumes. 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 a variety of information all in one place, you can use it to analyze and manage your data and storage more efficiently.



## How to Invoke ISMF

How you invoke ISMF depends on your installation. You begin 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. The ISMF Primary Option Menu will appear, and you can begin an ISMF session. For example, in the ISPF Master Application Menu, Figure 1, to select ISMF, you enter **I** following the arrow on the command line.

```
----- ISPF MASTER APPLICATION MENU -----
SELECT APPLICATION ==> I_
1 SPF      - SPF PROGRAM DEVELOPMENT FACILITY   USERID  - USER20
. . .      - . . .                             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 1. Starting an ISMF Session from the ISPF Master Application Menu**

If you want to invoke ISMF directly from TSO, use this command:

```
ISPSTART PGM(DGTFMD01) NEWAPPL(DGT)
```

The ISMF Primary Option Menu will appear, as in Figure 2, 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 APPLICATION       - Perform data set functions
2 VOLUME APPLICATION         - Perform volume functions
X EXIT                        - Terminate ISMF

USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 2. ISMF Primary Option Menu

### Examples of Using DFDSS via ISMF

Two examples of using ISMF follow. In the first example, we select the Data Set Application to copy a group of data sets. In the second, we select the Volume Application to dump an entire volume. For more information on using ISMF, see either *MVS/Extended Architecture Interactive Storage Management Facility User's Guide* or the ISMF online help panels.

#### Using the Data Set Application

You begin by building a list of the data sets to be copied. Complete the Data Set Selection Entry Panel with the values shown to the right of the arrows in Figure 3. Entering \*\* in the "Data Set Name" field selects those data sets that have your userid 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 3
COMMAND ==>>

TO GENERATE A DATA SET LIST, SPECIFY:

DATA SET NAME   ==>> **                                (fully or partially qualified)

SELECT SOURCE OF GENERATED LIST ==>> 2                (1 or 2)

  1 GENERATE LIST FROM VTOC
    VOLUME SERIAL NUMBER ==>>                          (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 3. 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 will appear, as shown in Figure 4.

```

                                DATA SET LIST                                REDISPLAYED LIST
                                COMMAND ==>> copy                                SCROLL ==>> HALF
                                ENTER LINE OPERATORS BELOW:                    Entries 1-5 of 5
                                                                                   Data Columns 3-7 of 21

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
                ----- (8)----- (9)----- (10)----- (11)----- (12)-----
                BOTTOM OF DATA

USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

**Figure 4. The Resulting Data Set List**

The data sets selected have your userid 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, copy the listed data sets to a 3380. Enter COPY on the command line of the data set list that you have generated. ISMF then displays the first page of the COPY Command Entry Panel, Figure 5. Alternatively, to move a data set separately, enter COPY in the "Line Operator" column next to that data set's name. Complete the first page as shown in Figure 5. Press Enter to display page 2 of the COPY Command Entry Panel, Figure 6.

```

                                COPY COMMAND ENTRY PANEL                                Page 1 of 6
COMMAND ==>>

SPECIFY ONE OR MORE FOR THE 5 DATA SETS:

TARGET VOLUMES      ==>> D8C040  ==>>          ==>>          ==>>
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 5. Completing Page 1 of the COPY Command Entry Panel

```

                                COPY COMMAND ENTRY PANEL                                Page 2 of 6
COMMAND ==>>

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                4 UNCATALOG AND SCRATCH IF EXPIRED
2 UNCATALOG           5 UNCATALOG ALL, SCRATCH IF EXPIRED
3 UNCATALOG AND SCRATCH ALL

MAXIMUM NUMBER OF RETRIES      ==>> 2  (0 to 99)
NUMBER OF SECONDS BETWEEN RETRIES ==>> 2  (0 to 255)
COPY EVEN IF DATA SETS IN USE  ==>> N  (Y or N)
SERIALIZE WITH DYNAMIC ALLOCATION ==>> N  (Y or N)
ACCESS DATA SETS IN SHARED MODE ==>> N  (Y or N)
COPY UNMOVABLE DATA SETS AS MOVABLE ==>> N  (Y or N)
VERIFY DATA WRITTEN           ==>> N  (Y or N)

USE ENTER TO CONTINUE; USE DOWN TO VIEW MORE COPY COMMAND OPTIONS;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 6. Completing Page 2 of the COPY Command Entry Panel

Press Enter if the defaults shown on page 2 are acceptable. The COPY Job Submission Entry Panel, Figure 7 on page 12, will appear. Verify the JOB

statement on this panel. Then press Enter to submit the job. If, instead, you want to save the generated JCL and DFDSS command for later use, you can do so.

```
                                COPY JOB SUBMISSION ENTRY PANEL
COMMAND ==>

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   ==> TONYS.STUFF
REPLACE CONTENTS ==> Y                (Y or N)

JOB STATEMENT INFORMATION:                (verify before proceeding)
==> //IBMUSERE JOB MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=IBMUSER
==> /*
==> /*
==> /*
==> /*
==> /*
==> /*

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 7. Verifying the COPY Job Statement Information**

The DFDSS Execute Statement Entry Panel, Figure 8, appears.

```
                                DFDSS EXECUTE STATEMENT ENTRY PANEL
COMMAND ==>

SPECIFY DFDSS EXECUTE STATEMENT INFORMATION:

==> /*
==> /*
==> /*
==> //STEP1 EXEC PGM=ADDRSSU,PARM='UTILMSG=YES',REGION=2048K
==> //SYSPPRINT DD SYSOUT=*
==> /*
==> /*

USE ENTER TO CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

**Figure 8. The DFDSS Execute Statement Entry Panel**

Press Enter. ISMF then displays the Data Set List, Figure 9. The asterisks before each COPY command in the Line Operator column indicate that the job has been successfully submitted.

```

COMMAND ==>
DATA SET LIST
COPY JOB CREATED
SCROLL ==> HALF
Entries 1-5 of 5
Data Columns 3-7 of 21

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 9. The List with Successful Submission of the COPY Command

### Using the Volume Application

Building a volume list is similar to building a data set list. You simply complete the Volume Selection Entry Panel as shown in Figure 10 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 will receive the short informational message: OTHER VALUES PRESENT.

```

COMMAND ==>
VOLUME SELECTION ENTRY PANEL
Page 1 of 3

TO GENERATE A VOLUME LIST, SPECIFY ONE OR MORE:
SELECT SOURCE FOR LIST GENERATION ==> 2 (1 or 2)

1 GENERATE FROM A SAVED LIST
LIST NAME ==>

2 GENERATE A LIST FROM ONE OR MORE OF THE FOLLOWING

VOLUME SERIAL NUMBER ==> libtst (fully or partially specified)
DEVICE TYPE ==> (fully or partially specified)
DEVICE NUMBER ==> (fully specified)
TO DEVICE NUMBER ==> (for range of devices)

USE ENTER TO PERFORM SELECTION; USE DOWN COMMAND TO VIEW NEXT SELECTION PANEL;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 10. Complete Page 1 of the Volume Selection Entry Panel

Press Enter to generate the volume list, as shown in Figure 11 on page 14.

```

COMMAND ==>>                                VOLUME LIST                                SCROLL ==>> HALF
                                             Entries 1-1 of 1
ENTER LINE OPERATORS BELOW:                Data Columns 3-8 of 15

  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
-----
                                BOTTOM OF DATA

```

USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

Figure 11. The Volume List

### Dumping the Volume

Enter the DUMP line operator for the volume in the list, as shown in Figure 11, to dump the volume. ISMF will display the DUMP Entry Panel as shown in Figure 12.

```

COMMAND ==>>                                DUMP ENTRY PANEL                                PAGE 1 OF 7

SPECIFY ONE OR MORE FOR VOLUME: LIBTST

OUTPUT DSN ==>> tape.dsn
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 ==>> 1 (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 12. Page 1 of the DUMP Entry Panel

Complete the DUMP Entry Panel as shown in Figure 12, and press Enter to display page 2.

```

                                DUMP ENTRY PANEL                                PAGE 2 OF 7
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 13. Page 2 of the DUMP Entry Panel**

Verify that the default values as shown in Figure 13 are acceptable, and press Enter to display page 3.

```

                                DUMP ENTRY PANEL                                Page 3 of 7
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)

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 14. Completing the DUMP Entry Panel**

Verify that the default values as shown in Figure 14 are also acceptable. Press Enter to display the DUMP Job Submission Entry Panel, Figure 15 on page 16.



```

                                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   ==> tonys.stuff
REPLACE CONTENTS ==> y           (Y or N)

JOB STATEMENT INFORMATION:           (verify before proceeding)
==> //IBMUSERS  JOB MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=IBMUSER
==> // *
==> // *
==> // *
==> // *
==> // *
==> // *

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 15. DUMP Job Submission Entry Panel**

You may change the default selection to save the DFDSS job stream, as shown in Figure 15. Press Enter to display the DFDSS EXECUTE Statement Entry Panel, Figure 16.

```

                                DFDSS EXECUTE STATEMENT ENTRY PANEL
COMMAND ==>>

SPECIFY DFDSS EXECUTE STATEMENT INFORMATION:

==> // *
==> // *
==> // *
==> //STEP1   EXEC PGM=ADRDUSSU,PARM='UTILMSG=YES',REGION=2048K
==> //SYSPRINT DD  SYSOUT=*
==> // *
==> // *

USE ENTER TO CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

**Figure 16. DFDSS EXECUTE Statement Entry Panel**

Press Enter to save the job and redisplay the Volume List. The asterisk next to DUMP in Figure 17 indicates that the job stream has been successfully built.

```

COMMAND ==>>
VOLUME LIST
SCROLL ==>> HALF
Entries 1-1 of 1
Data Columns 3-8 of 15
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
-----  -----  -----  -----  -----  -----  -----  -----
                                BOTTOM OF DATA

USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 17. The DUMP Job Stream is Successfully Built

## How You Invoke DFDSS via Job Control Language

DFDSS is controlled by Job Control Language (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 via an application program (as described in “Chapter 7. How You Invoke DFDSS via an Application Program” on page 189).

### The JCL You Need

Figure 18 on page 18 shows the JCL needed for using DFDSS. For more information on coding JCL, see the *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,KEEP),DSNAME=USER2.DUMP
```

A standard label tape is recommended to help prevent an operator error. If used, the DSNAME must be included.

The command data set, which follows your SYSIN DD \* statement, contains your DFDSS function commands and auxiliary commands. Function commands specify

tasks, which are operations to be performed; auxiliary commands control these tasks.

Statement	Usage
<b>JOB</b>	Initiates your job.
<b>EXEC</b>	Specifies the program name (PGM=ADRDSSU) or, if the job control statements reside in a procedure library, the procedure name. Additional information can be entered in the PARM parameter of the EXEC statement; see "How You Control DFDSS via PARM Information in the EXEC Statement," following this figure.
<b>STEPCAT DD</b>	Required (1) if in a full or tracks operation a volume contains a non-integrated catalog facility VSAM data set that is cataloged in a user catalog or (2) for copying or logical dumping of VSAM data sets whose catalog is outside the standard order of search. This DD statement must point to the user catalog. It is not required for other DFDSS operations.
<b>SYSPRINT DD</b>	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 you specify the LRECL and BLKSIZE parameters, see "Block Size" on page 161 and "SYSPRINT DD Statement" on page 187 for recommendations.
<i>input</i> DD	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. See also "Block Size" on page 161. Do not specify BUFNO.
<i>output</i> DD	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. See also "Block Size" on page 161. Do not specify BUFNO. Code a volume count when a new data set will reside on six or more volumes.
<i>filter</i> DD	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.
<i>password</i> DD	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.
<b>SYSIN DD</b>	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.

*Note:* DFDSS may fail (because of enqueueing techniques) if your DD DSNAME is the same as a data set name being processed. This is because of initiator contention.

Figure 18. DFDSS JCL

## How You Control DFDSS via PARM Information in the EXEC Statement

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

### **TYPRUN=SCAN**

Only control card syntax checking is to be done. No tasks are to be executed.

### **TYPRUN=NORUN**

For COPY, DUMP, RESTORE, COMPRESS, 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.

### **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 have no page ejection occur.

### **SIZE=nnnnK**

The "nnnn" is a 1- to 4-digit number indicating the multiple of 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 terminated. To get a dump, include a DD statement for SYSABEND or SYSUDUMP. This keyword is provided for debugging purposes only.

### **RACFLOG=YES**

RACF (Resource Access Control Facility) logging will be 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 171 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 an attached utility are to be printed in the SYSPRINT listing during a COPY or COMPRESS operation.

### **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.

Examples of PARM information are:

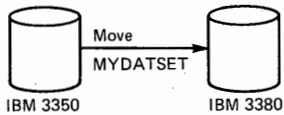
```
// EXEC PGM=ADDRSSU,
//   PARM='PAGENO=8,LINECNT=57,SIZE=500K,UTILMSG=YES'

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

For further information on the PARM parameter of an EXEC statement, see *Supervisor Services and Macro Instructions*.

### Example 1: Moving a Data Set

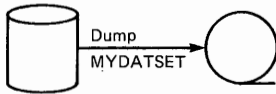
The following example shows how to move a non-VSAM data set from one DASD volume to another, using JCL and a DFDSS command. The source data set is deleted, and the target data set is cataloged to reflect its new location.



```
//MYJOB   JOB   accounting information,REGION=nnnnK2
//STEP1   EXEC  PGM=ADDRSSU
//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)) -
      INDDNAME(DASD1) OUTDDNAME(DASD2) DELETE CATALOG
/*
```

### Example 2: Dumping a Data Set

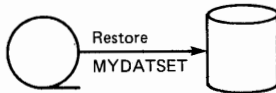
This example shows how to back up a DASD data set to tape:



```
//MYJOB   JOB   accounting information,REGION=nnnnK
//STEP1   EXEC  PGM=ADDRSSU
//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,KEEP),DSNAME=MYDATSET.BACKUP
//SYSIN   DD    *
DUMP DATASET(INCLUDE(MYDATSET)) -
      INDDNAME(DASD) OUTDDNAME(TAPE)
/*
```

### Example 3: Restoring a Data Set

In this example, the data set (MYDATSET) that was backed up on tape in Example 2 is restored to the original DASD volume on which it resided.



```
//MYJOB   JOB   accounting information,REGION=nnnnK
//STEP1   EXEC  PGM=ADDRSSU
//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
/*
```

<sup>2</sup> See "Storage Requirements—DFDSS" on page 181 for information on how to estimate *nnnn*.

## Choosing the Data Sets You Want Processed—Filtering

You can use filtering to select data sets, based on specified criteria. Filtering can be done on fully or partially qualified data set names (using the INCLUDE or EXCLUDE keyword) and it can be done (using the BY keyword) on any of 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 date, absolute or relative<sup>3</sup>
- Data set organization (SAM, PAM, 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 off<sup>3</sup>
- Whether a data set is cataloged or not (using the standard catalog search order)

At least one of the INCLUDE, EXCLUDE, or BY parameters must be specified in the following DFDSS commands:

Data set COPY  
Data set DUMP  
Data set RESTORE  
DEFRAG  
COMPRESS  
RELEASE

### Using Qualified Data Set Names

You can use fully or partially qualified data set names in the following commands:

COMPRESS  
Data set COPY  
DEFRAG  
Data set DUMP  
RELEASE  
Data set RESTORE

The data set name used in the PRINT command must be fully qualified.

Examples of fully qualified data set names are:

- USER2LD (the first and only qualifier is "USER2LD")
- SYS1.UTIL3.LOAD (the first qualifier is "SYS1", etc.)
- USER2.PROGRAM1.LIST (the first qualifier is "USER2", etc.)

---

<sup>3</sup> The data-set-changed flag (DSCHA) and the last-referenced date (REFDT) are only supported on systems that have data-set-changed flag support installed. In addition, if you do not have Data Facility Hierarchical Storage Manager (DFHSM) on your system, for VSAM data sets, module IDATMSTP (in load module IDA0192A) should be modified to pass a return code of 4 to its caller.

Examples of partially qualified data set names are:

1. \*\*.LOAD (All data sets whose last, or only, qualifier is "LOAD")
2. SYS1.\*\* (All data sets whose first, or only, qualifier is "SYS1")
3. USER2.\*\*.LIST (All data sets whose first and last qualifiers are "USER2" and "LIST", respectively, including "USER2.LIST")
4. \*.LOAD (All data sets with two qualifiers, whose last qualifier is "LOAD")
5. SYS1.\* (All data sets with two qualifiers, whose first qualifier is "SYS1")
6. SYS1\*.LOAD (All data sets with three qualifiers and whose first and last qualifiers are "SYS1" and "LOAD", respectively)
7. SYS1.UT\*.LIST (All data sets with three qualifiers, whose first and last qualifiers are "SYS1" and "LIST", respectively, and whose second qualifier begins with "UT")
8. \*\*.LIB (All data sets whose last, or only, qualifier ends with "LIB")
9. \*\*.LIB\* (All data sets whose last, or only, qualifier has "LIB" in it)
10. \*.PLI\*. (All data sets with three qualifiers, whose second qualifier has "PLI" in it)
11. \*.P\*M (All data sets with three qualifiers, whose last qualifier begins with "P" and ends with "M")
12. \*\* (All data sets)

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 **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.

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

1. Two asterisks are the maximum permissible in a qualifier.
2. If there are two asterisks in a qualifier, then the asterisks should 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

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

### The Use of Quotation Marks

The use of single quotation marks around a data set name (fully or partially qualified) specifies that the name is a fully qualified name. Do not use single quotation marks to enclose a data set name that will appear in a DFDSS control statement.

### The Filtering Process

The optional DFDSS filtering process works as follows:

1. If inclusion (INCLUDE) criteria (fully or partially qualified data set names) are specified, DFDSS tentatively selects a data set if it satisfies any of the inclusion criteria.  
  
If inclusion (INCLUDE) criteria are **not** specified, then EXCLUDE or BY must be. DFDSS tentatively selects **all** data sets (equivalent to "INCLUDE(\*\*)"). For the DEFRAG command, "INCLUDE(\*\*)" is **always** implicit.
2. If exclusion (EXCLUDE) criteria (fully or partially qualified data set names) are specified, DFDSS rejects a data set, tentatively selected in Step 1, if it satisfies any of the exclusion criteria.
3. If data set characteristics (BY) criteria are specified, and if a tentatively selected data set from Step 1 is not rejected in Step 2, DFDSS selects that data set only if it satisfies **all** the data set characteristics criteria.

DFDSS permits you to find out which data sets are selected by the filtering process without actually performing the requested operation(s). See TYPRUN=NORUN on page 19.

When you are filtering, VSAM data sets are handled as follows:

- 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 these 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. When a VSAM data set is defined with IDCAMS, the expiration date is in the catalog instead of the VTOC. Thus this data set will not be selected.
- A fully qualified name is required to dump a VSAM data set that is in open-for-update status.
- Data set RESTORE: To prevent index components from being restored inadvertently, you must specify the fully qualified name of the cluster.
- 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.

When doing logical processing, selecting data sets using catalog(s), and a CVOL is found, INCLUDE/EXCLUDE processing is applied against the qualifier name "SYSCTLG.Vvolser", where volser is the serial number of the volume on which the CVOL resides. For all other operations, INCLUDE/EXCLUDE processing is applied against the "SYSCTLG" qualifier name.

## Standard Catalog Search Order

For a complete description of the standard catalog search order, see *Access Method Services Reference*.

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

1. User catalog(s) specified in the current job step (STEPCAT) or, if none is specified for the job step, user catalog(s) specified for the current job (JOB CAT) are searched. If more than one catalog is specified for the job step or job, the catalogs are searched in order of concatenation. The search continues as follows:
2. If the data set name is a qualified name and is the name of a user catalog, or the first qualifier is the same as the alias of a user catalog or the alias of a control volume, then that user catalog or control volume is searched. The search continues as follows:
3. The master catalog is searched.

*Note:* If data sets with the same name are cataloged in more than one catalog within the standard search order, only the first one encountered will be selected.

## Compatibility Between Version 1 and Version 2

**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 will be scratched from all the volumes on which it

resides, and will be uncataloged. On a data set DUMP, this is true only for a logical dump. Otherwise, it will process 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, not specifying the input volumes results in a logical data set DUMP. The data sets and the volumes to be processed are determined from the catalog(s).

**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:

- The information kept on the dump tape if the data set does **not** exist in the catalog, or
- 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. But 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 parameter is ignored when doing 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 parameter 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 parameter 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 using a Different DFDSS Release:** Tapes dumped using an older version or release of DFDSS can be restored using a newer version or release. IBM does not recommend **and does not guarantee** that tapes dumped using a newer version or release of DFDSS can be restored 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 V2.2 (or later release) environment cannot be restored in an MVS/XA DFP V2.1 environment without certain VSAM N\*512 toleration PTFs.

Figure 19 shows whether a tape dumped using one DFDSS release can be restored using a different release. Figure 20 on page 27 shows similar information for stand-alone DFDSS Restore.

RESTORE under—>	Version 1 Release 2.1	Version 2 All releases
Full (F), tracks (T), or data set (D)—>	F T D	F T D
<b>Full DUMP from:</b>		
Version 1 Release 2.1	Y Y Y	Y Y Y
Version 2	Y Y Y	Y Y Y
<b>Tracks DUMP from:</b>		
Version 1 Release 2.1	N Y N	N Y N
Version 2	N Y N	N Y N
<b>Physical data set DUMP from:</b>		
Version 1 Release 2.1	N Y Y	N Y Y
Version 2	N Y Y	N Y Y
<b>Logical data set DUMP from:</b>		
Version 2	N N N	N N Y <sup>1</sup>

Y=Yes

N=No

<sup>1</sup>Tapes dumped using an older version or release of DFDSS can be restored using a newer version or release. IBM does not recommend and does not guarantee that tapes dumped using a newer version or release of DFDSS can be restored 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.

**Figure 19. System DFDSS RESTORE Compatibility among Releases.** This matrix shows which dump tapes can be restored by the various releases of DFDSS.

---

<b>Stand-alone RESTORE under—&gt;</b>	Version 1 Release 2.1	Version 2 All releases
---	--------------------------	---------------------------

---

<b>Full (F) or tracks (T)—&gt;</b>	F T	F T
------------------------------------	-----	-----

---

<b>Full DUMP from:</b>		
Version 1 Release 2.1	Y Y	Y Y
Version 2	Y Y	Y Y

---

<b>Tracks DUMP from:</b>		
Version 1 Release 2.1	Y <sup>1</sup> Y	Y <sup>1</sup> Y
Version 2	Y <sup>1</sup> Y	Y <sup>1</sup> Y

---

<b>Physical data set DUMP from:</b>		
Version 1 Release 2.1	N Y	N Y
Version 2	N Y	N Y

---

<b>Logical data set DUMP from:</b>		
Version 2	N N	N N

---

Y=Yes  
N=No

<sup>1</sup>All the tracks that were dumped can be restored.

**Figure 20. Stand-Alone DFDSS RESTORE Compatibility among Releases.** This matrix shows which dump tapes can be restored by the various releases of stand-alone DFDSS Restore.

---



## Chapter 3. Specifying Tasks—Function Commands

The **function** commands (described in this chapter in alphabetic order) specify tasks, which are operations to be performed. Most command descriptions end with a subsection "... Command Usage," which contains detailed information on using that command. "Chapter 6. Notes on Using DFDSS" on page 159 contains other usage notes, including DFDSS restrictions. Refer to these notes as needed.

Before using DFDSS, you should understand:

- "Specifying DFDSS Commands—Command Syntax," shown below
- "Understanding the Command Syntax" on page 30

The following sections, formerly in this chapter, are now in Chapter 6:

- "Read/Write Access Serialization Scheme" on page 183
- "How DFDSS Checks Authorization for You" on page 167

### Specifying DFDSS Commands—Command Syntax

DFDSS permits commands to be written in free form in card columns 2 through 72, inclusive. Syntax requirements are as follows:

- A *command* must appear first, followed by its parameters.

A command is separated from its parameters by one or more blanks, a comment, or both. For example:

```
COPY DATASET ALLDATA(*) CATALOG REBLOCK ...
```

- 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 (`,`), one or more blanks, or a comment.
- *Parameters* are separated from one another by one or more separators.
- *Subparameters* are separated from one another by one or more separators. Subparameters follow their associated parameters 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:

```
PASSWORD ( DATASET1/PASSWRD1 )  
or  
PASSWORD(DATASET1/PASSWRD1)
```

If two or more subparameters are permissible for a single parameter, these are separated from one another by at least one blank. Subparameters 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:

```
PASSWORD(DATASET1/PASSWRD1 , DATASET2/PASSWRD2)  
or  
PASSWORD(DATASET1/PASSWRD1 ,DATASET2/PASSWRD2)  
or  
PASSWORD(DATASET1/PASSWRD1 DATASET2/PASSWRD2)
```

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

```
COPY DATASET ALLDATA(*) CATALOG -  
REBLOCK ...
```

The absence of such a hyphen specifies the termination of the command. If a parameter or subparameter 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.

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

## Understanding the Command Syntax

A uniform system of notation (and spacing) is used to describe the syntax of DFDSS commands in this chapter. This notation is not part of the language but provides a means of describing the structure of these commands.

- *Required* parameters appear in the upper portion of each command syntax figure, separated from one another by blank lines. *Optional* parameters appear in the lower portion, in alphabetic order. They too are separated from one another by blank lines.

Below each command syntax figure is the explanation of the parameters shown in that figure. *Required* parameters are presented in the order of their occurrence. These are followed by *optional* parameters, which are presented in alphabetic order.

- The minimum truncation allowed for parameter keywords is shown in **BOLDFACE** uppercase type. For example:

```
ALLData (Enter ALLD, ALLDA, ALLDAT, or ALLDATA)  
DYNALloc (Enter DYNAL, DYNALL, DYNALLO, or DYNALLOC)  
KEYlength (Enter KEY, KEYL, KEYLE, KEYLEN, etc.)
```

- Alternative abbreviations are sometimes shown with a parameter keyword. The alternative abbreviation immediately follows the "OR" (|) bar. For example:

DATASet | DS  
TRACKS | TRKS  
INDDname | IDD

- Subparameters shown in lowercase *italics* are those that you are to supply—for example, the data set name (*dsn*) in ALLDATA(*dsn*).
- When only a *single* parameter or subparameter can be selected from a group, the parameters in the group are separated by “OR” (|) bars. If needed for clarity, braces, { }, delimit the group of parameters. The default parameter, if any, is underlined.
- When *one or more* parameters or subparameters must be selected from a group, the members of the group are shown on consecutive lines, each surrounded by brackets, [ ], with the left brackets lining up vertically.
- Brackets, [ ], surrounding ungrouped parameters or subparameters indicate that they are optional.
- A comma followed by an ellipsis and enclosed by brackets, [...], indicates that additional subparameters of the type immediately preceding the ellipsis are optional.
- Other punctuation (parentheses, commas, slashes, etc.) must be entered as shown. However, a **comma may be replaced by a blank**.

## How Many Subparameters Are Allowed?

In the following parameters, up to 255 subparameters may be specified in the input (command) stream:

```
ALLDATA(dsn [...])  
BY((schar,op,arg [...]) (255 of “schar,op,arg”)  
BY(LIST((schar,op,arg [...])) (255 of “schar,op,arg”)  
DDNAME(ddn [...])  
DYNAM((volser [,unit]) [...])  
EXCLUDE(dsn [...])  
EXCLUDE(LIST(dsn [...]))  
INCLUDE(dsn [...])  
INDDNAME(ddn [...])  
INDYNAM((volser [,unit]) [...])  
LOGICALVOLUME(volser[...])  
LOGINDDNAME(ddn [...])  
LOGINDYNAM((volser [,unit]) [...])  
OUTDDNAME(ddn [...])  
OUTDYNAM((volser [,unit]) [...])  
OUTTRACKS((c1,h1) [...]) (255 of “c1,h1”)  
PASSWORD(dsn/pswd [...]) (255 of “dsn/pswd” and/or “catname/pswd”)  
      (catname/pswd [...])  
PERCENTUTILIZED(n [...])  
REBLOCK(dsn [...])  
REBLOCKADDRESS(dsn[...])
```



RENAMEUNCONDITIONAL {(pfx) |  
((on,nn)[...]) | (255 of "on,nn")  
((pfx),(on,nn) [...])}  
TRACKOVERFLOW(dsn [...])  
TRACKS((c1,h1,c2,h2) [...]) (255 of "c1,h1,c2,h2")

If you have more than 255 subparameters, you can point to a sequential data set with a ddname as follows:

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

## About the Examples in This Manual

Throughout the examples in this manual, a DASD device is 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. See "Devices You Can Use" in Chapter 1 for a list of supported devices.

# COMPRESS Command

The COMPRESS command compresses (or “degases”) partitioned data sets (PDS) 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 all of 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 using the IEBCOPY utility. To prevent loss of data if the system or IEBCOPY abnormally terminates during processing, you should back up your volume or data sets that meet the filtering criteria before using this command.

Here’s the syntax of the COMPRESS command:

Command	Parameters
COMPRESS	<pre> {{FILterdd   FDD} (ddn) }   { [ INClude (dsn [...]) ]   [ EXClude (dsn [...]) ]   [ BY((schar,op,arg) [...]) ] }  { DDName (ddn [...])     DYNAM ((volser [,unit] ) [...]) }  Optional parameters that can be used with COMPRESS:  DYNALloc  PASsword   PSWD ( [ddn                   [dsn/pswd [...]] )  WAIT(numsecs,numretrys) </pre>

Figure 21. COMPRESS Command Syntax

## Explanation of COMPRESS Command Parameters

### FILterdd | FDD (ddn)

*ddn*

specifies the name of your DD statement that identifies the sequential data set, or member of a partitioned data set, containing 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 parameters described below.

### INClude(dsn[...])

*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 “Using

## COMPRESS Command

Qualified Data Set Names” on page 21. 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.

### EXCLUDE(*dsn*[,...])

*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.

### BY((*schar,op,arg*) [...])

specifies that the data sets selected up to this point, by the processing of the INCLUDE and EXCLUDE parameters, are to be further filtered. To select the data set, all BY criteria must be met.

*schar*

specifies one of the following selection characteristics:

- ALLOC (allocation type)
- CATLG (data set cataloged)
- CREDIT (creation date)
- DSCHA (data-set-changed flag)
- DSORG (data set organization)
- EXPDT (expiration date)
- EXTNT (number of extents)
- FSIZE (number of allocated tracks)
- MULTI (multivolume data set)
- REFDT (last-referenced date)

*op*

specifies one of the following operators:

- 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*

specifies an argument meaningful for the selection characteristic used.

Figure 22 shows the permissible combinations of *schar*, *op*, and *arg*.

Here are some examples of the BY parameter:

```
BY((ALLOC EQ CYL) (CATLG EQ YES))  
or  
BY(FSIZE GE 100)
```

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)	If MOV is picked, the data sets to be compressed cannot be allocated as: <ul style="list-style-type: none"> <li>POU (partitioned organization unmovable)</li> </ul>
CATLG	EQ	YES (or 1) NO (or 0)	If YES is specified, only cataloged data sets are processed. If NO is specified, only uncataloged data sets are processed.
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 one-hundredth 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.	If you specify a yy of 00 through 49, DFDSS assumes you want years 2000 through 2049; if you specify a yy of 50 through 99, DFDSS assumes 1950 through 1999.
DSCHA	EQ NE	YES (or 1) NO (or 0)	
DSORG	EQ	PAM	This need not be specified. DFDSS assumes this.
EXTNT FSIZE	LT GT EQ NE GE LE	A 1- to 5-digit number (0 to 99999)	This is the number of extents (EXTNT) or allocated tracks (FSIZE) for a data set on the volume being processed.
MULTI	EQ	NO	This need not be specified. DFDSS assumes this.

Figure 22. BY Parameters—COMPRESS Command

### DDNAME (*ddn* [...])

*ddn*

specifies the name of your 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 *volser*.

### DYNAM ((*volser* [,*unit*]) [...])

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.

*volser*

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

## COMPRESS Command

*unit*

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

### DYNALloc

specifies that dynamic allocation, instead of enqueue, is to be used to serialize the use of selected partitioned data sets. Selected data sets are serialized for the duration of the COMPRESS operation. This allows cross-system serialization in a JES3/MVS environment. This is only of value if the dynamic allocation/JES3 interface is not disabled. For more information, see "DYNALLOC" on page 167.

### PASsword | PSWD ( [*ddn*] [*dsn/pswd* [...]] )

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

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

*ddn*

specifies the name of your DD statement that identifies the sequential data set, or member of a partitioned data set, containing 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 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:* Because the IEBCOPY utility is used, the operator must also supply the same password information supplied above for each password-protected data set that you select. However, if you specify DYNALLOC, the prompt message issued to the operator contains a system-generated ddname and does not contain a data set name, which makes it difficult for the operator to supply the password for the correct data set. When compressing password-protected data sets, DYNALLOC should be used in a controlled manner that allows the operator to specify the correct password.

### WAIT(*numsecs,numretrys*)

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.

*numretrys*

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.

## COMPRESS Command

The default 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 *numretrys*.

### Example of a COMPRESS

#### *Compressing Selected Partitioned Data Sets*

```
//JOB1      JOB  accounting information,REGION=nnnnK4
//STEP1     EXEC PGM=ADDRSSU
//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 in case of system failures when COMPRESS is running (DSCHA EQ 0).

### COMPRESS Command Usage

COMPRESS does not support processing PAM data sets that:

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

---

<sup>4</sup> See "Storage Requirements—DFDSS" on page 181 for information on how to estimate *nnnn*.

# COPY Command

With the COPY command, you can copy data from one DASD volume to another.

**The DATASET COPY command allows you to:**

- Copy one or more data sets from one DASD volume to another DASD volume of like or unlike device type.
- Copy multivolume data sets, user catalogs, or CVOLs.
- Copy VSAM data sets or catalogs at the cluster (not component) level.
- Copy data sets (except for CVOLs, integrated catalog facility catalogs, system data sets [those named SYS1.xxx] and VVDSs) to preallocated targets.
- Copy BDAM data sets, which are assumed to be organized in such a way that their records are located using TTR (TT is the relative track number from the beginning of the data set and R is the record number on the track). 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.
- Specify multiple target volumes to be used as spill volumes.

Whether you specify input volume(s) or not, a DATASET COPY always uses *logical* processing (see “Logical and Physical Processing” on page 3).

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

**The TRACKS and FULL COPY commands allow you to:**

- Copy data from one DASD volume to another, of the same device type, in track format.
- Copy either a full volume or ranges of tracks. For ranges of tracks, the source and target volumes can be the same or different. Also, the relative location of the source and target tracks can be the same or different. However, when the tracks are to be copied to the same volume, there cannot be any overlap between the source and target track ranges.
- Copy the volume serial number.
- Terminate the COPY operation if there are unexpired data sets on the output volume or write over the unexpired data sets on the output volume.

In addition to copying to another DASD volume of the same type, you can:

- Copy from an IBM 3330 Model 1 to an IBM 3330 Model 11.
- Copy from an IBM 3340 with 35-megabyte capacity to an IBM 3340 with 70-megabyte capacity, or to an IBM 3344.

## COPY Command

- Copy from a smaller capacity IBM 3380 model to a larger capacity IBM 3380 model.
- Copy from a minivolume or virtual volume to a real volume of like device type, and vice versa, device capacity permitting.
- Using the tracks COPY, copy data from an IBM 3330 Model 11 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.

In prior releases, when doing a full-volume copy from a smaller-capacity DASD to a larger-capacity DASD, with equal track lengths, you had to allocate a temporary data set and then rebuild the indexed VTOC. Starting with DFDSS Version 2 Release 3, 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.

Figure 23 on page 40 shows the syntax of the FULL or TRACKS COPY command. Figure 24 shows the syntax of the DATASET COPY command. Before using the command, you should review “COPY Command Usage” on page 58.

If you do not specify FULL, TRACKS, or DATASET, the default is FULL.



# COPY Command

Command	Parameters
COPY	<p> <b>{{FULL   FUL}  </b>  <b>{TRACKS   TRKS}((c1,h1,c2,h2) [...]) }</b> </p> <p> <b>{{INDDname   IDD}(ddn)  </b>  <b>{INDYnam   IDY}(volser [,unit] ) }</b> </p> <p> <b>{{OUTDDname   ODD}(ddn)  </b>  <b>{OUTDYnam   ODY}(volser [,unit] ) }</b> </p> <p>Optional parameters that can be used with FULL:</p> <p> <b>ALLData { (dsn [...])   (*) }</b> </p> <p> <b>ALLExcp   ALLX</b> </p> <p> <b>CANcelerror</b> </p> <p> <b>COPYVolid   CPYV</b> </p> <p> <b>PASsword   PSWD ( [ddn]</b>  <b>[dsn/pswd [...]]</b>  <b>[catname/pswd [...]])</b> </p> <p> <b>PURge   PRG</b> </p> <p> <b>TOLerate (IOERror)</b> </p> <p> <b>WAIT (numsecs,numretrys)</b> </p> <p> <b>WRIttecheck   WRC</b> </p> <p>Optional parameters that can be used with TRACKS:</p> <p> <b>CANcelerror</b> </p> <p> <b>COPYVolid   CPYV</b> </p> <p> <b>OUTTRacks   OUTTRKS (( c1,h1) [...])</b> </p> <p> <b>PASsword   PSWD ( [ddn [...]]</b>  <b>[dsn/pswd [...]]</b>  <b>[catname/pswd [...]])</b> </p> <p> <b>PURge   PRG</b> </p> <p> <b>TOLerate (IOERror)</b> </p> <p> <b>WAIT (numsecs,numretrys)</b> </p> <p> <b>WRIttecheck   WRC</b> </p>

Figure 23. COPY Command Syntax—FULL or TRACKS

Command	Parameters
COPY	<p> <b>{DATASet   DS} { {FILTerdd   FDD} (ddn) or {DATASet   DS} ( [ INClude(dsn[,...]) [ EXClude(dsn[,...]) [ BY((schar,op,arg) [...]) ] )</b> </p> <p> <b>{{OUTDDname   ODD}(ddn [,...])   {OUTDYnam   ODY}((volser [,unit]) [...]) }</b> </p> <p>Optional parameters that can be used with DATASET:</p> <p> <b>ALLData { (dsn [,...])   (*) }</b>  <b>ALLExcp   ALLX</b>  <b>ALLMulti</b>  <b>CANcelerror</b>  <b>CATalog   RECATalog(newcatname)   RECATalog(*)</b>  <b>DELeTe</b>  <b>DYNALloc</b>  <b>FORce</b>  <b>{{INDDname   IDD} (ddn [,...])   {INDYnam   IDY} ((volser [,unit]) [,...])   {LOGINDDname   LIDD} (ddn [,...])   {LOGINDYnam   LIDY} ((volser [,unit]) [,...]) }</b>  <b>MENTity (modeldsn) [MVOlser (volser)]</b>  <b>PASsword   PSWD ( [ddn] [dsn/pswd [,...]] [catname/pswd [,...]] )</b>  <b>PERCENTUtilized   PCTU (n [,...])</b>  <b>PURge   PRG</b>  <b>REBLock (dsn [,...])</b>  <b>RELBlockaddress   RELBLKA (dsn [,...])</b>  <b>RENAMEUnconditional   RENUNC { (pfx)   ((on,nm)[,...])   ((pfx),(on,nm) [...]) }</b>  <b>REPlace</b> </p>

Figure 24 (Part 1 of 2). COPY Command Syntax—DATASET

# COPY Command

Command	Parameters
<b>COPY</b>	Optional parameters that can be used with DATASET (continued):  <b>SHAre   SHR</b>  <b>TGTAlloc(BLK   CYL   TRK   SOURCE   SRC)</b>  <b>TOLerate([IOERror] [ENQFailure])</b>  <b>TRACKOverflow   TRKO (dsn[...])</b>  <b>UNCATalog</b>  <b>WAIT (numsecs,numretrys)</b>  <b>WRIttecheck   WRC</b>

Figure 24 (Part 2 of 2). COPY Command Syntax—DATASET

## Explanation of COPY Command Parameters

### **FULL | FUL**

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 (null DSORG field, for example in JES2/JES3 data sets). 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).

### **TRACKS | TRKS((c1,h1,c2,h2) [...])**

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

#### *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 may be omitted. For example:

Specified	Results
None	Syntax error
c1	c1,0,c1,maximum head number
c1,h1	c1,h1,c1,maximum head number

c1,h1,c2	c1,h1,c2,maximum head number
c1,,c2	Syntax error
,h1	Syntax error

**DATASet | DS** ({**FILterdd** | **FDD**} (*ddn*))

or

**DATASet | DS** ( [ **INClude**(*dsn*[...]) ]  
 [ **EXClude**(*dsn*[...]) ]  
 [ **BY**((*schar,op,arg*) [...]) ] )

specifies a data set COPY operation, using filtering. See “Choosing the Data Sets You Want Processed—Filtering” on page 21 for an explanation of the filtering process used. Unless specified via 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.

**FILterdd | FDD** (*ddn*)

*ddn*

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

**INClude**(*dsn*[...])

*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 “Using Qualified Data Set Names” on page 21. If INCLUDE is omitted (but EXCLUDE or BY is specified) or “INCLUDE(\*\*)” is specified, *all* data sets are eligible to be selected for copying.

**EXClude**(*dsn*[...])

*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.

**BY**((*schar,op,arg*) [...])

specifies that the data sets selected up to this point, by the processing of the INCLUDE and EXCLUDE parameters, are to be further filtered. To select the data set, **all** BY criteria must be met.

*schar*

specifies the applicable selection characteristic, consisting of one of the following:

- ALLOC (allocation type)
- CATLG (data set cataloged)
- CREDIT (creation date)
- DSCHA (data-set-changed flag)
- DSORG (data set organization)
- EXPDT (expiration date)
- EXTNT (number of allocated or used extents for the entire data set on all the volumes on which it resides)

## COPY Command

- FSIZE (number of allocated or used tracks for the entire data set on all the volumes on which it resides)
- MULTI (multivolume data set)
- REFDT (last-referenced date)

*op*

specifies the operator, consisting of one of the following:

- 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*

specifies an argument meaningful for the selection characteristic used.

Figure 25 shows the permissible combinations of *schar*, *op*, and *arg*.

Here are some examples of the BY parameters:

```
BY((ALLOC EQ CYL) (CATLG EQ YES))  
or  
BY(FSIZE GE 100)
```

**INDDname | IDD (*ddn* [...])**

*ddn*

specifies the name of your 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 *volser*. For multivolume data set requirements, see “Multivolume Data Sets—Processing” on page 181.

See the note under INDYNAM.

**INDYnam | IDY ((*volser* [,*unit*] ) [...])**

specifies that volume(s) to be copied are 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 a DASD volume to be copied.

*unit*

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

*Note:* If no input volume(s) are specified for a DATASET COPY, data set selection is from all data sets cataloged in the catalogs that are accessible through the standard search order. If either INDDNAME or INDYNAM is specified, though the same catalog search order is used, data set selection is limited to data sets residing entirely on the specified volume(s). For multivolume data sets, use LOGINDDNAME or LOGININDYNAM with ALLMULTI.

ALLOC	EQ NE	CYL (cylinder allocation) TRK (track allocation) BLK (block length allocation) ABSTR (absolute track allocation) MOV (movable data sets)	If MOV is picked, the data sets to be copied cannot be allocated as any of the following: <ul style="list-style-type: none"> <li>• PSU (physical sequential unmovable)</li> <li>• POU (partitioned organization unmovable)</li> <li>• DAU (BDAM unmovable)</li> <li>• ABSTR (absolute tracks)</li> <li>• ISAM (indexed sequential access method)</li> </ul>
CATLG	EQ	YES (or 1) NO (or 0)	If YES is specified, only data sets cataloged in the standard search order are processed. If NO is specified, only uncataloged data sets are processed. (Only valid for a COPY with INDD or LOGINDD specified.)
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 one-hundredth 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.	If you specify a yy of 00 through 49, DFDSS assumes you want years 2000 through 2049; if you specify a yy of 50 through 99, DFDSS assumes 1950 through 1999.  CREDIT and EXPDT: For a multivolume data set, the date from its first volume's VTOC is used for checking. REFDT: For a multivolume data set, the latest date from all its VTOCs is used for checking.
DSCHA	EQ NE	YES (or 1) NO (or 0)	For a multivolume data set, the value used for checking is 1 if any part of the data set has been changed. Otherwise, the value used is 0.
DSORG	EQ NE	SAM PAM BDAM ISAM VSAM EXCP	EXCP applies to data sets not allocated as any of the listed organizations and not accessed using any of the listed access methods.
EXTNT FSIZE	LT GT EQ NE GE LE	A 1- to 5-digit number (0 to 99999)	This is the number of used extents (EXTNT) or used tracks (FSIZE) for an entire data set on all the volume(s) on which it resides. 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 only processes multivolume data sets. If NO is specified, it only processes single volume data sets.

Figure 25. BY Parameters—COPY Command

# COPY Command

## OUTDDname | ODD (*ddn* [...])

*ddn*

specifies the name of your 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 *volser*. For a data set COPY, you can specify multiple names, separated by commas.

## OUTDYnam | ODY ((*volser* [,*unit*]) [...])

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 parameter is optional.

## ALLData {(*dsn*[...]) | (\*)}

*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 source and target are the same device type, all allocated space on the source is to be allocated and copied to the target.
- When source and target 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 footnote<sup>5</sup>, and Figure 26 on page 47.

\*

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

- When source and target are the same device type, all allocated space on the source is to be allocated and copied to the target.
- When source and target 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.

## ALLExcp | ALLX

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

---

<sup>5</sup> 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*) or ALLEXCP to copy all its data. Examples of data sets that may have these characteristics are JES2/JES3 data sets, and CICS journal and MSVCJRNL data sets.

- When source and target are the same device type, all allocated space on the source is to be allocated and copied to the target.
- When source and target are different device types, all allocated space on the source is to be allocated on the target, but no data is copied.

See footnote<sup>5</sup> and Figure 26.

DSORG/Device Geometry	Neither ALLDATA nor ALLEXCP Specified	Nonempty Data Set (DS1LSTAR ≠ 0 and BLKSIZE ≠ 0) with ALLDATA Specified	Empty Data Set (DS1LSTAR = 0 with ALLEXCP Specified) or (BLKSIZE = 0 with ALLDATA Specified)
SAM to Like	Allocates and copies only used space	Allocates and copies all allocated space	Allocates and copies all allocated space
SAM to Unlike	Allocates and copies only used space	Allocates all allocated space, and copies the used space	Allocates all allocated space but doesn't copy
PAM to Like	Allocates and copies only used space	Allocates and copies all allocated space	Allocates and copies all allocated space
PAM to Unlike	Allocates and copies only used space	Allocates all allocated space, and copies the used space	Allocates all allocated space but doesn't copy
Undefined DSORG to Like	Allocates and copies only used space	Allocates and copies all allocated space	Allocates and copies all allocated space
Undefined DSORG to Unlike	Allocates only used space but doesn't copy	Allocates all allocated space but doesn't copy	Allocates all allocated space but doesn't copy

Figure 26. Data Set COPY ALLDATA and ALLEXCP Results (When REBLOCK Is Not Specified)

### 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. ALLMULTI is valid only when LOGINDDNAME or LOGINDYNAM is specified.

### CANcelerror

specifies that the COPY is to be terminated if a permanent read error, such as a data check, occurs. If this parameter 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 parameter. CANCELERROR has no effect on the following types of errors on a DASD volume:



## COPY Command

Equipment check  
Command reject  
Intervention required  
Busout parity

### **CATalog | RECATalog(*newcatname*) | RECATalog(\*)**

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

#### **CATALOG**

catalogs the target data set in a catalog as determined by the standard catalog search order. This is the default for VSAM and multivolume 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.

CATALOG or RECATALOG fails if the target data set is already cataloged in the same catalog and DELETE, RENAMEU, or UNCATALOG is not specified.

An alternate index (AIX) is always cataloged in the same catalog as its associated base cluster. If the base cluster is recataloged, then the AIX will be recataloged. DFDSS messages on AIX allocations are snapshots and do not necessarily specify the catalog in which the AIX will be cataloged when processing has completed.

### **COPYVolid | CPYV**

specifies that the VOLID (the volume serial number) from the input DASD volume is to be copied to the output DASD volume. This applies to full and tracks COPY operations, but only if track 0 is copied. For more information on COPYVOLID, see "RESTORE Operation—Full Volume Restore and RACF-Protected Data Sets" on page 183.

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.

### **DELeTe**

deletes VSAM and non-VSAM data sets from the source volume after a successful copy. This is, in effect, a data set "move." Data sets are scratched and uncataloged. If you do not specify DELETE when copying cataloged data sets, the target data set must either be cataloged (RECATALOG) in a different catalog or renamed (RENAMEU). *Unexpired* source data sets will be deleted only if you also specify PURGE. A data set with a high-level qualifier of "SYS1." (unless it is a user catalog), a page data set, or a swap data set cannot be deleted. Do not specify SHARE if you specify DELETE.

### **DYNALloc**

specifies that dynamic allocation, instead of enqueue, is to be used to serialize the use of data sets. The data set whose extent is to be relocated is serialized for the duration of the COPY. This allows cross-system serialization in a JES3/MVS environment. This is only of value if the

dynamic allocation/JES3 interface is not disabled. For more information, see "DYNALLOC" on page 167.

## FORce

specifies that DFDSS copy one or more unmovable data sets to a like or unlike device type. Unmovable data sets are those that are allocated as ABSTR (absolute track) or as unmovable (PSU, POU, DAU or ISU). FORCE does not apply to ISAM data sets unless they are also allocated with the unmovable attribute. The allocation attribute, unmovable or ABSTR, is carried over to the output volume.

When copying to a like or same device type, DFDSS will copy the data sets to the same track locations on the target volume and 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 will 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, FORCE must be specified. The unmovable data sets are placed in any available location.

## LOGINDDname | LIDD (*ddn* [...])

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 your 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 volser. See the note under LOGINDDNAME.

## LOGINDYnam | LIDY ((*volser* [*unit*] ) [...])

specifies that volume(s) that contain the data set(s) to be copied are 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 copied.

*unit*

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

*Note:* If neither LOGINDDNAME nor LOGINDYNAM is specified, data set selection is from all data sets cataloged in the catalogs that are accessible through the standard search order. If either LOGINDDNAME or LOGINDYNAM is specified, though the same catalog search order is used, data set selection is limited to data sets residing entirely on the specified volume(s). A multivolume data set that has extents on volumes not specified with LOGINDDNAME or LOGINDYNAM will be copied only if you specify the ALLMULTI parameter.

## COPY Command

### **MENTity**(*modeldsn*) [**MVOLser**(*volser*)]

specifies, for RACF-protected data sets, an entity (model data set name) and, optionally, the serial number of the volume containing that entity (*volser*). These parameters are used to define the data sets to RACF. Specification of MVOLSER is optional when the model entity (MENTITY) is cataloged in an integrated catalog facility catalog, or it is a non-VSAM data set and that MENTITY is cataloged in the standard catalog search order. When MVOLSER is specified for a VSAM model entity, the *volser* specified should be the *volser* of the catalog in which the model entity is cataloged. If these parameters 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 27 on page 60). Target data sets (preallocated or nonpreallocated) are only RACF protected 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.

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.

### **OUTTRacks** | **OUTTRKS** ((*c1,h1*) [...])

specifies, for a tracks COPY, the cylinder and head number 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** | **PSWD** ( [*ddn*]

[*dsn/pswd* [...]]

[*catname/pswd* [...]] )

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 if:

- You do not have the required RACF DASDVOL or RACF DATASET access.
- The user 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 your DD statement that identifies the sequential data set, or member of a partitioned data set, containing 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 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 your input command stream is suppressed in the SYSPRINT output.

When a system utility is being used to perform the DFDSS COPY, the operator must also supply the same password information supplied above for each password-protected data set selected or have the proper RACF data set access authority. However, if you specify DYNALLOC, the prompt message issued to the operator contains a system-generated ddname and does not contain a data set name, which makes it difficult for the operator to supply the password for the correct data set. When copying password-protected data sets, DYNALLOC should be used in a controlled manner that allows the operator to specify the correct password.

## **PERCENTUtilized | PCTU (*n* [...])**

specifies that DFDSS should 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 parameter is used as a guide only and may not be precise for all situations. This parameter may be ignored by DFDSS for VSAM data sets allocated to multiple target volumes.

## **PURge | PRG**

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, that unexpired source data sets are to be deleted after they are successfully copied. This parameter is only valid when DELETE is also specified.

## **REBLock (*dsn* [...])**

specifies that DFDSS is to reblock one or more of the selected SAM or PAM data sets, unless 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.

When copying PAM load modules to an unlike device, IEBCOPY is used with COPYMOD specified. This may result in a reblocked data set. When copying other PAM or SAM data sets with RECFM=U, the REBLOCK parameter is ignored. Non-load module PAM data sets with NOTE lists are also not reblocked.

At installations with MVS/XA DFP Version 2 Release 2 or later release, DFDSS uses the DASDCALC macro to determine the optimal blocksize for the target. The reblocking method used, DFDSS or DASDCALC, is presented to the user reblock exit. Track overflow data sets can be reblocked by either method.

## COPY Command

### **RELBlockaddress | RELBLKA** (*dsn*[,...])

identifies the BDAM data sets, whose names match the fully or partially qualified names specified (*dsn*), that 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 parameter is ignored for TTR-organized BDAM data sets and for BDAM data sets with standard user labels.

### **RENAMEUnconditional | RENUNC** {(*px*) | ((*on,nn*) [...]) | ((*px*),(*on,nn*) [...]) }

specifies that the data set should 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 will occur, and the data set will not be copied, 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 sublist. 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*

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

#### *nn*

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

The syntax rules for prefix, old name, and new name are the same as in the RENAME parameter in a RESTORE operation.

### **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. A usable preallocated data set must meet the criteria specified in "Preallocated Data Set COPY Usage" on page 62.

### **SHAre | SHR**

specifies that DFDSS is to share for read access with other programs, the data sets to be copied. See "Read/Write Access Serialization Scheme" on page 183 for information about data set enqueues. Do not specify DELETE if you specify SHARE.

### **TGTAlloc(BLK | CYL | TRK | SOURCE | SRC)**

specifies how DFDSS will allocate the target data set.

#### **BLK**

specifies allocate by blocks.

#### **CYL**

specifies allocate by cylinders.

#### **TRK**

specifies allocate by tracks.

**SOURCE | SRC**

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

If the TGTALLOC parameter is omitted, the target allocation defaults to tracks. If BLK is specified for VSAM data sets, TRK is used instead. The final VSAM allocation may be different than the requested one because of VSAM allocation rules.

**TOLerate(ENQFailure)**

specifies that data sets are to be processed even though shared or exclusive access fails. For a data set COPY (except of a user catalog) in which a utility performs the copy (see Figure 28 on page 61), this parameter is ignored. For more information on TOL(ENQF), see "ENQ" on page 165.

**TOLerate(IOERror)**

specifies that, if the input volume can be opened, DFDSS is to continue copying even though permanent input errors, such as equipment checks, occur, but to terminate after 100 errors. The default is to terminate on permanent input errors. On a data set COPY in which a utility performs the copy, DFDSS ignores this parameter.

**TRACKOverflow | TRKO (dsn[...])**

*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 parameter is not needed for data sets indicated as having overflow records. When performing 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 (and not scratch) successfully copied non-VSAM data sets which 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.

**WAIT(numsecs,numretrys)**

For a *volume or tracks* COPY, 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.

*numretrys*

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, specifies to DFDSS the length of 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.

# COPY Command

## *numsecs*

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

## *numretrys*

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 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 *numretrys*. For more information, see "WAIT(numsecs,numretrys) Option for Serialization" on page 164.

## **WRitecheck | WRC**

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

## **Examples of Full and Tracks COPY Operations**

```
//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
//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(1,0,1,14) 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(\*)), if the DS1LSTAR field in format-1 DSCB is nonzero. For these data sets, DFDSS copies all allocated space (ALLEXCP) if the DS1LSTAR field in the format-1 DSCB in the VTOC is zero. The COPY operation is to be terminated 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       /* TERMINATE 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 is terminated 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 3: 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 4: A Data Set COPY to Move Data Sets to a Single Volume**

```
//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 (USER1.\*\*\*) data sets are to be copied to a single target volume labeled "338001" for consolidation. The data sets are scattered on multiple source volumes. The target volume can be of unlike device type. 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,



## COPY Command

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 5: A Data Set COPY of a Multivolume Data Set**

```
//JOB5      JOB  accounting information,REGION=nnnnK
//STEP1     EXEC PGM=ADDRSSU
//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 these data sets (PCTU).

Change this example as shown below to include ALLMULTI missing volume processing. The INCLUDE parameter 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 6: A Data Set COPY With DELETE and RENAMEU**

```
//JOB6      JOB  accounting information,REGION=nnnnK
//STEP1     EXEC PGM=ADDRSSU
//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 renaming the data sets.

### Example 7: A Data Set COPY With REBLOCK

```
//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 parameter are reblocked on the target volume. The block size is selected by DFDSS, unless modified by the user reblock exit routine.

### Example 8: 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 (USER2.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 parameter 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.

# COPY Command

## COPY Command Usage

### *General Notes:*

- The DFDSS 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.
- When utilities are not invoked, the COPY command allows you to:
  - Retrieve readable data from a volume even though that volume has severe errors.
  - Terminate on a permanent read error or continue processing even though you receive permanent I/O errors on the input volume.
  - Verify that the data is written successfully.
- For VSAM data sets, you can specify the password of the catalog instead of the password of each data set. If the password of the data set and the catalog are both specified, the data set password is used to check authorization.
- DFDSS copies only used space for sequential access method (SAM) or partitioned access method (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.

### *Data Set COPY Notes:*

When you copy:

- A user catalog, DELETE must be specified and an input volume and the RENAMEU parameters must **not** be specified. You must specify the fully qualified name of the user catalog in the INCLUDE parameter. There should be no other jobs executing in the processor, or any processor in the complex, that access the user catalog being moved; otherwise, the COPY operation may fail or the copied catalog may contain errors.
- A user catalog, the operator is prompted to enter the system's master catalog's password. If the master catalog was defined with ATTEMPTS=0, you must RACF-protect the master catalog so that password prompting is bypassed.

In an MVS/XA DFP Version 2 Release 3 environment, 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).

- A CVOL, DELETE must be specified and RENAMEU must **not** be specified. The CATALOG parameter is used as a default. You need to rebuild catalog aliases afterward.

- Data sets cataloged in a CVOL, ensure that the number of data sets to be copied is fewer than 1455.
- A data set and a utility is used to move a data set, the data set must be cataloged in the standard catalog search order.
- An ISAM data set, 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.
- An ISAM data set to a volume containing an indexed VTOC, be sure that there is no other activity on this volume. This is suggested in order to guarantee a successful allocation on the volume.
- A single-volume ISAM data set, the last-volume indicator (DS1IND80) must be on, or the data set must be cataloged to be selected.
- A VSAM data set, then 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.
- A VSAM base cluster which has associated alternate index clusters, only the base cluster is copied, 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.
- A VSAM alternate index cluster, DELETE must be specified, and only the alternate index cluster is moved. The moved alternate index cluster continues to relate to its base cluster.
- An empty VSAM data set (zero data relative block address or zero record count), the data set will be defined on the target volume but not copied. Message ADR474W is issued for the data set.
- A single volume data set, it is always copied to a single output volume. A multivolume data set may be copied to a single volume or to multiple volumes. See “Multivolume Data Sets—Processing” on page 181 for details.
- A multivolume data set with standard user label, only the standard user label on the first volume is copied to the target volumes(s). For more information, see “Multivolume Data Sets—Processing” on page 181.
- A generation data group (GDG) data set, filtering on generations is supported. Generation names in relative generation number, dsn(n), can be specified in the INCLUDE and EXCLUDE parameters. If, during COPY, you catalog the GDGs in a different catalog or rename them, you must define ahead of time the target GDG base name, because the source GDG base name is unusable.

Both VSAM and non-VSAM data sets can, optionally, be renamed. The rules for renaming VSAM clusters are the same as for non-VSAM data sets. You can only rename 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 using a similar mechanism. If the component names were derived using the cluster name (such as clustername.data or clustername.index), the new component names are similarly derived.

# COPY Command

During a data set COPY without ALLMULTI specified, catalogs are used to select only data sets that reside in their entirety on the volumes that you specify in INDDNAME, INDYNAM, LOGINDDNAME, or LOGINDYNAM. A data set is selected if:

- Its volume sequence number is 1, its last volume flag is off, and it is cataloged in the standard catalog search order, or
- Its volume sequence number is 1, and its last volume flag is on, or
- It is a partitioned data set, or
- 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, or
- It is a VSAM cluster and all volumes for all components are specified.

If you do not specify MENTITY and you have RACF Release 1.5 or later installed, Figure 27 shows how profiles should be defined. See also Figure 61 on page 185.

	<b>No RENAME Parameter No DELETE Parameter</b>	<b>No RENAME Parameter DELETE Parameter</b>	<b>RENAME Parameter No DELETE Parameter</b>	<b>RENAME Parameter DELETE Parameter</b>
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. See note 1, below.	Leave source profile. See note 1, below.
Discrete profile for source	Leave source profile. Define discrete target profile. See note 2, below.	Delete source profile. Define discrete target profile. See note 2, below.	Leave source profile. Define a discrete target profile. See note 2, below.	Delete source profile. Define a discrete target profile. See note 2, below.

**Figure 27. Defining Profiles—With RACF 1.5 or Later Installed**

**Notes to Figure 27:**

1. 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.
2. When a discrete profile exists for a preallocated target data set, that profile is not deleted and no new profile is defined.

DFDSS cannot be used to copy an active VSAM master catalog, the VSAM volume data set (VVDS), and the VTOC index data set.

For data sets that DFDSS copies by invoking MVS utilities, DFDSS will not copy those data sets that the utilities do not support.

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 abends, 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.

DFDSS updates PAM data sets that have NOTE lists, and RBA-organized BDAM data sets, when copying them. DFDSS cannot be used to copy any other type of data set containing location-dependent data to unlike devices. Also, DFDSS cannot be used to copy any data sets containing location-dependent data to like devices unless they are defined as “unmovable” and the FORCE parameter 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.

DFDSS cannot be used to copy data sets with a volume serial of MIGRAT.

DFDSS cannot be used to copy multivolume data sets that are not cataloged.

Figure 28 shows when DFDSS invokes a utility for a data set COPY.

Data Set Type	Like Devices	Unlike Devices
SAM	DFDSS	DFDSS
PAM	DFDSS <sup>2</sup>	DFDSS <sup>2</sup>
PAM load modules	DFDSS	IEBCOPY
DAM nonrelative block address mode	DFDSS	IEHMOVE <sup>3</sup>
DAM relative block address mode <sup>1</sup>	DFDSS	DFDSS
ISAM to same track. MVS/XA DFP V 2.1 or later installed. Target volume uses VTOC index. Target space available.	DFDSS	N/A
ISAM (all other cases)	IEBISAM	IEBISAM
ESDS	DFDSS	DFDSS
RRDS	DFDSS	DFDSS
KSDS	DFDSS <sup>4</sup>	IDCAMS (REPRO)

Figure 28 (Part 1 of 2). Data Mover Selection Matrix for Data Set COPY

# COPY Command

Data Set Type	Like Devices	Unlike Devices
Key range data set	DFDSS <sup>5</sup>	IDCAMS (REPRO)
CVOL	IEHMOVE	IEHMOVE
ICF user catalogs	IDCAMS (EXPORT/IMPORT)	IDCAMS (EXPORT/IMPORT)
Undefined DSORG	DFDSS	Allocated but not copied

Figure 28 (Part 2 of 2). Data Mover Selection Matrix for Data Set COPY

**Notes to Figure 28:**

- <sup>1</sup> DFDSS RELBLOCKADDRESS parameter must be specified.
- <sup>2</sup> 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.
- <sup>3</sup> Not copied if target data set is preallocated, or if the target device has a smaller track capacity than the source.
- <sup>4</sup> IDCAMS is used if target's CISIZE, CASIZE, physical record size, physical block size, or imbed or span attributes are different than the source's.
- <sup>5</sup> IDCAMS (REPRO) is used if any key range data component has extended, or if any of the conditions in Note 4 is true.

### Preallocated Data Set COPY Usage

- The following rules are used to determine if an existing data set qualifies as a preallocated target for data set COPY:
  - VSAM Preallocation:
    - Cluster name match
    - Complete cluster available on target volumes
  - Non-VSAM Preallocation:
    - Format-1 DSCB names match
    - Complete data set available on target volumes:
      - Single volume target qualification:
        - DS1DSORG must be PAM or
        - DS1VOLSQ must be 1 AND DS1IND80 must be on (Last volume flag).
      - Multivolume target or single volume with DS1IND80 off:
        - Data set must be cataloged in the standard order of search.
        - All volsers returned by a LOCATE on the data set must be in the output volume list.
        - Candidate volumes for which no format-1 DSCBs exist, are acceptable.
- The following rules are used to determine if a preallocated data set is usable for COPY:
  - VSAM Data Sets:
    - User must be authorized to update the target data set.
    - Cluster types match.
    - Number of components match.
    - Key length and offset match.
    - KEYRANGES match.
    - No multivolume components.
    - Sufficient space available for each component.
    - KSDS data sets must be reusable or empty.
    - Key range data sets must be empty.

- Data set in catalog structure if required for the COPY.
  - 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 Data Sets:
    - User must be authorized to update the target data set.
    - DSORG must match.
    - For unmovable data sets, extents must match exactly when copying to a like device without the FORCE parameter specified.
    - For movable data sets or unmovable data sets with FORCE, sufficient space must be present on the target data set.
    - For PAM data sets, target directory must be sufficient to contain all source members and aliases.
    - For BDAM data sets, target must not exist if COPY will be done using the IEHMOVE utility. If RELBLOCKADDRESS is specified for the data set, preallocated targets are allowed.
    - Preallocated unmovable ABSTR data sets on an unlike device will not be copied without specifying FORCE.
    - Preallocated standard user label data sets must have more than one extent if the source data set has more than one extent.
  - When multiple target volumes are specified, and REPLACE is specified, more than one existing data set may qualify as a preallocated target. The first existing data set that qualifies as a preallocated target, using 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 will determine the data set that will be used as the preallocated target.
  - Data set COPY will not change the following attributes for a preallocated target data set:
    - Catalog in which the target resides if cataloged, and the catalog status of the data set (cataloged or uncataloged)
    - Passwords
    - Expiration date
- Thus the following parameters are not effective for a data set COPY when the target data set is preallocated:
- PERCENTUTILIZED
  - CATALOG
  - RECATALOG
- The RACF status of a preallocated target data set will be checked. If the source is protected, but the target is not, the target data set will be protected with a discrete RACF profile.
  - Device selection criteria used in DFDSS Version 2 Release 1 data set COPY (like, same, then unlike device preference) are not observed if a preallocated data set target is used.
  - If a preallocated data set is determined to be unusable, message ADR439E is issued and the COPY terminated for that data set only.
  - Data set COPY does not scratch and reallocate any preallocated data sets.
  - 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



## COPY Command

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 will not use a preallocated target if DELETE is specified. No search will be made for existing data sets in this case.
- An integrated catalog facility AIX cannot use a preallocated target. No search will be made for existing data sets when copying an AIX.
- An integrated catalog facility user catalog, a VVDS, a page/swap data set, a CVOL, or a system data set (one named SYS1.xxx) cannot be copied to a preallocated data set.
- If a data set qualifies for reblocking when REBLOCK has been specified (SAM and PAM only), and a preallocated target is used, the target block size will be overwritten with one of the following values:
  - The source data set block size
  - A COPY-selected block size
  - A user-selected block size passed by the user reblock exit

The block size used is determined by the user reblock exit return code.

- If REBLOCK is not specified, the target BLKSIZE of a non-VSAM data set is overwritten with the source BLKSIZE.
- If the source data set is empty, or unsupported DSORG, and the target is preallocated but not empty, no special attempt is made to clear or alter the target data set. Message ADR363E is output to inform the user. If ALLEXCP is specified and the target is a like device, the data in the source data set is moved to the target, as in Version 2 Release 1 COPY.
- COPY will not support the use of RENAMEUNCONDITIONAL for preallocated target data sets.
- Non-VSAM data sets with standard user labels can use preallocated target data sets with or without standard user labels. The source label will be copied only if the target data set has standard user labels.

## 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 via 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).

Figure 29 shows the syntax of the COPYDUMP command.

Command	Parameters
COPYDump or CPYD	INDDname   IDD ( <i>ddn</i> )
	OUTDDname   ODD ( <i>ddn</i> [...])
	The following parameter is optional:  LOGicalvolume   LVOL ( <i>volser</i> [...])

Figure 29. COPYDUMP Command Syntax

### Explanation of COPYDUMP Command Parameters

#### INDDname | IDD (*ddn*)

*ddn*

specifies the name of your 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).

#### OUTDDname | ODD (*ddn*[...])

*ddn*

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

#### LOGicalvolume | LVOL (*volser*[...])

*volser*

specifies the source DASD volume serial number from which dumped data is to be copied. Omission of the LOGICALVOLUME parameter causes DFDSS to copy data from all logical volumes in the dump data set. This parameter 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.

# COPYDUMP Command

## 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,KEEP),DSNAME=V111111.BACKUP
//COPY2    DD    UNIT=3480,VOL=SER=TAPE07,
//          DISP=(NEW,KEEP),DSNAME=V111111.BACKUP
//SYSIN    DD    *
COPYDUMP   INDD(BACKUP) -
           OUTDD(COPY1,COPY2)
/*
```

Two copies are to be made from a DFDSS dump tape (OUTDD(COPY1,COPY2)).

### Example 2: Copying a Dump Created Via 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,KEEP),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 dump tape, volume TAPE20, was created via a *physical* data set DUMP. Also, that source DASD volumes 338000, 338001, etc. 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 Via 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,KEEP),DSN=USER.BACKUP.REL3B.A
//SYSIN     DD    *
COPYDUMP    -
  INDD(TAPE3)           /* DUMP TAPE TO BE COPIED      */ -
  OUTDD(OUTT3)          /* NEW DUMP TAPE           */ -
/*
```

Assume that a dump tape, volume TAPE30, was created via a *logical* data set DUMP. All dump data is copied.

## COPYDUMP Command Usage

COPYDUMP maintains the same block size as that of the input dump data set.

The dump data produced by the DUMP command should not be copied by programs other than DFDSS.

For a COPYDUMP, if a DASD volume is specified as the output device, the track size of the DASD volume must be equal to or greater than the maximum block size of the input data set.

During a COPYDUMP to multiple output devices, if one device has an I/O error, processing terminates for all devices and no valid copies of the dump data set are produced.

## 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.

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 terminations)
- Performance degradation caused by excessive DASD arm movement
- An increase in the time required for direct access device space management (DADSM)-related functions

DFDSS excludes the following types of data sets from a DEFRAG operation:

- User-specified data sets (EXCLUDE), if any
- Data sets that do not satisfy all the BY criteria, if BY is specified
- ISAM data sets
- Non-integrated catalog facility cataloged VSAM data sets
- 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

You can use the DEFRAG command to do the following:

- Serialize data sets on MVS/JES3 systems.
- Perform a DEFRAG operation only if the fragmentation is more than a specified value.
- Stop the defragmentation run when a user-specified, single free space extent is obtained.
- Have DFDSS wait if the volume or one or more data sets is momentarily unavailable.
- Check the VTOC for errors, such as overlapping extents.

Using the NORUN parameter on the EXEC statement, you can perform a DEFRAG operation to determine:

- The fragmentation index
- If the VTOC has errors, such as overlapping extents (more than one extent mapping the same area on the volume)
- If space on the volume is not mapped by the VTOC
- If extents in the VTOC fall outside the range of the volume

## DEFRAG Command

Figure 30 shows the syntax of the DEFRAG command. Before using the command, you should review “DEFRAG Command Usage” on page 75.

Command	Parameters
DEFrag	<pre>{ DDName (ddn)     DYNAM (volser [,unit] ) }</pre> <p>Optional parameters that can be used with DEFRAG:</p> <pre>BY { (LIST((schar,op,arg) [...]))     (DDName(ddn)) }</pre> <p>DYNALloc</p> <pre>EXClude { (LIST(dsn[,... ]))     (DDName(ddn)) }</pre> <p>FRAGmentationindex   FRAGI (n)</p> <p>MAXmove(n)</p> <pre>PASsword   PSWD ( [ddn]   [dsn/pswd [...]] )</pre> <p>WAIT(numsecs,numretrys)</p>

Figure 30. DEFRAG Command Syntax

### Explanation of DEFRAG Command Parameters

#### DDName(ddn)

*ddn*

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

#### DYNAM(volser[,unit])

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 parameter is optional.

#### BY(DDName(ddn))

specifies data set filtering criteria.

*ddn*

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

## DEFRAG Command

### **BY(LIST(*schar,op,arg*) [...]))**

specifies data set filtering. To select the data set for inclusion in the DEFRAG operation, all BY criteria must be met.

#### *schar*

specifies the applicable selection characteristic, consisting of one of the following:

- ALLOC (allocation type)
- CATLG (data set cataloged)
- CREDIT (creation date)
- DSCHA (data-set-changed flag)
- DSORG (data set organization)
- EXPDT (expiration date)
- EXTNT (number of allocated extents)
- FSIZE (number of allocated tracks)
- MULTI (multivolume data set)
- REFDT (last-referenced date)

#### *op*

specifies the operator, consisting of one of the following:

- 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*

specifies an argument meaningful for the selection characteristic used.

Figure 31 shows the permissible combinations of *schar*, *op*, and *arg*.

Here are some examples of the BY parameters:

```
BY(LIST((ALLOC EQ CYL),(CATLG EQ YES)))  
or  
BY(LIST(FSIZE GE 100))
```

### **DYNALloc**

specifies that dynamic allocation, instead of enqueue, is to be used to serialize the use of data sets. The data set whose extent is to be relocated is serialized for the duration of the relocation. This allows cross-system serialization in a JES3/MVS environment. This is only of value if the dynamic allocation/JES3 interface is not disabled. For more information, see "DYNALLOC" on page 167.

**EXCLUDE** { (LIST(*dsn*[...])) | (DDNAME(*ddn*)) }

*dsn*

specifies a fully or partially qualified data set 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 your DD statement that identifies a sequential, or member of a partitioned, data set. It contains card-image records, in DFDSS command syntax, of the fully or partially qualified data set names that are to be excluded from the DEFRAG operation.



## DEFRAG Command

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)	If MOV is picked, the data sets to be defragmented cannot be allocated as: <ul style="list-style-type: none"> <li>• PSU (physical sequential unmovable)</li> <li>• POU (partitioned organization unmovable)</li> <li>• DAU (BDAM unmovable)</li> <li>• ABSTR (absolute tracks)</li> <li>• ISAM (indexed sequential access method)</li> </ul>
CATLG	EQ	YES (or 1) NO (or 0)	If YES is specified, only cataloged data sets are processed. If NO is specified, only uncataloged data sets are processed.
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 one-hundredth 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.	If you specify a yy of 00 through 49, DFDSS assumes you want years 2000 through 2049; if you specify a yy of 50 through 99, DFDSS assumes 1950 through 1999.  For both single- and multivolume data sets, the date in the VTOC of the volume being processed is used for filtering.
DSCHA	EQ NE	YES (or 1) NO (or 0)	For a multivolume data set, the value obtained from the VTOC of the volume being processed is used for checking.
DSORG	EQ NE	SAM PAM BDAM ISAM VSAM EXCP	EXCP applies to data sets not allocated as any of the listed organizations and not accessed using any of the listed access methods.
EXTNT FSIZE	LT GT EQ NE GE LE	A 1- to 5-digit number (0 to 99999)	This is the number of allocated extents (EXTNT) or allocated tracks (FSIZE) for a data set on the volume being processed.
MULTI	EQ	YES (or 1) NO (or 0)	If YES is specified, DFDSS only processes multivolume data sets. If NO is specified, it only processes single volume data sets.  If a data set's volume sequence number is greater than 1 in the VTOC, it is assumed to be multivolume. If this sequence number is 1 it is assumed to be a single volume data set.

Figure 31. BY Parameters—DEFRAG Command

### FRAGmentationindex | FRAGI (*n*)

specifies that DEFRAG is to terminate 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 “DEFRAG Function” on page 161.

### MAXmove(*n*)

*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, message ADR228W is issued and the DEFRAG operation terminated. If MAXMOVE is not specified, DFDSS tries to assemble a contiguous free area of size equal to the total number of free tracks on the volume.

**Note:** The DEFRAG operation makes two passes on a 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 terminate after the first pass, specify MAXMOVE.

The second pass attempts to assemble multiple groups of large contiguous free space, and generally moves data around more than during the first pass. This technique usually refines the results of the first pass but may 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 | PSWD ( [*ddn*] [*dsn/pswd* [...]] )

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

- You do not have the required RACF DASDVOL or RACF DATASET access.
- The user 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 your DD statement that identifies the sequential data set, or member of a partitioned data set, containing 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 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.

## DEFRAG Command

### **WAIT**(*numsecs,numretrys*)

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.

#### *numretrys*

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 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 *numretrys*.

## Examples of DEFRAG Operations

### **A DEFRAG With Excluded Data Sets**

```
//JOB1 JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRDSSU
//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 an 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=ADRDSSU
//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 */
/*
```

## DEFRAG Command

Only data sets last referenced more than 1 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 32 on page 76 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=ADRSSU,PARM='TRACE=YES'
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//DASD1 DD UNIT=3380,VOL=SER=338001,DISP=OLD
//SYSIN DD *
DEFRAG DDNAME(DASD1)
/*
```

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 **prior** to 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.

### DEFRAG Command Usage

For security purposes, the data set tracks used before the relocation are erased after relocation when:

- RACF Version 1 Release 6 (or earlier release) is installed and (1) the data set is RACF-protected, or (2) the data set is password-protected, or (3) the VSAM data set has the ERASE attribute.
- If both RACF Version 1 Release 7 and MVS/XA DFP Version 2 Release 1 (or later release) are installed and (1) the data set was defined to RACF with the RACF ERASE option, or (2) the VSAM data set has the ERASE attribute, or (3) the data set is password-protected, and if it is also defined to RACF, the RACF ERASE option is taken. Figure 33 on page 77 gives you more detail.

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 Appendix A.

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.

# DEFRAG Command

```

PAGE 0001 DFDSS 5665-327 DATA FACILITY DATA SET SERVICES V2R3.0 86.324 14:28
ADR041I PRIME(02), I/O BUFFER BELOW 16MB VIRTUAL REQUESTED
DEFRAG DYNAM(SHARE1)
ADR101I RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DEFRAG '
ADR109I RI01 (01), 86324 14:28:09 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR006I (001)-SETUP(01), 86324 14:28:10 EXECUTION BEGINS
ADR317I (001)-OPNCL(01), THE DEVICE SIZE FIELD IN THE VTOC IS INCORRECT IF VOLUME SHARE1 IS NOT A MINIVOLUME. DEVICE SIZE OF 0040
CYLINDERS WILL BE USED
ADR317I (001)-OPNCL(01), THE DEVICE SIZE FIELD IN THE VTOC IS INCORRECT IF VOLUME SHARE1 IS NOT A MINIVOLUME. DEVICE SIZE OF 0040
CYLINDERS WILL BE USED
ADR208I (001)-DANAL(01), BEGINNING STATISTICS ON SHARE1:
FREE CYLINDERS 0031
FREE TRACKS 0009
FREE EXTENTS 0008
LARGEST FREE EXTENT (CYL,TRK) 0024,0000
FRAGMENTATION INDEX .198
PERCENT FREE SPACE 79
ADR220I (001)-DANAL(01), INTERVAL BEGINS AT CC:HH 0001:0006 AND ENDS AT CC:HH 0016:0000
ADR209I (001)-DFRAG(01), MOVED EXTENT 001 FROM 0003:0000-0003:0014 TO 0002:0000-0002:0014 FOR TEST.LDS.S01.DATA
ADR209I (001)-DFRAG(01), MOVED EXTENT 002 FROM 0005:0000-0005:0014 TO 0003:0000-0003:0014 FOR TEST.LDS.S01.DATA
ADR209I (001)-DFRAG(01), MOVED EXTENT 003 FROM 0007:0000-0007:0014 TO 0004:0000-0004:0014 FOR TEST.LDS.S01.DATA
ADR209I (001)-DFRAG(01), MOVED EXTENT 004 FROM 0009:0000-0009:0014 TO 0005:0000-0005:0014 FOR TEST.LDS.S01.DATA
ADR209I (001)-DFRAG(01), MOVED EXTENT 005 FROM 0011:0000-0011:0014 TO 0006:0000-0006:0014 FOR TEST.LDS.S01.DATA
ADR209I (001)-DFRAG(01), MOVED EXTENT 006 FROM 0013:0000-0013:0014 TO 0007:0000-0007:0014 FOR TEST.LDS.S01.DATA
ADR209I (001)-DFRAG(01), MOVED EXTENT 007 FROM 0015:0000-0015:0014 TO 0008:0000-0008:0014 FOR TEST.LDS.S01.DATA
ADR213I (001)-DANAL(01), ENDING STATISTICS ON SHARE1:
DATA SET EXTENTS RELOCATED 0007
TRACKS RELOCATED 0105
FREE CYLINDERS 0031
FREE TRACKS 0009
FREE EXTENTS 0002
LARGEST FREE EXTENT (CYL,TRK) 0031,0000
FRAGMENTATION INDEX .018
PAGE 0002 DFDSS 5665-327 DATA FACILITY DATA SET SERVICES V2R3.0 86.324 14:28
PAGE 0003 DFDSS 5665-327 DATA FACILITY DATA SET SERVICES V2R3.0 86.324 14:28
ADR212I (001)-DANAL(01), EXTENT DISTRIBUTION MAP FOR SHARE1:
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 2 0.015
2 0.000 0.000 1 0.031
3 0.000 0.000 1 0.055
9 0.000 1 0.018 0.055
14 0.000 0.018 1 0.166
15 6 0.189 0.018 7 1.000
24 1 0.240 0.018 1.000
360 1 1.000 0.018 1.000
465 1.000 1 1.000 1.000
ADR013I (001)-CLTSK(01), 86324 14:28:34 TASK COMPLETED WITH RETURN CODE 0000
ADR012I DSSU (01), 86324 14:28:35 DFDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000

```

Figure 32. Printed Output Resulting from a Successful DEFRAG Run

If RACF DASDVOL class is active and a profile exists for the volume, the DEFRAG task will abend with a system code 913 should DASDVOL authorization fail. This happens regardless of RACF data set access authority.

To provide data set integrity in case of job cancellation by the user or abnormal termination while DEFRAG is running, DFDSS uses a format-1 DSCB with the unique name "SYS1.DFDSS.DEFRAG.xxxxxxxx.volser.DUMMY", where xxxxxxxx is 8 bytes of X'FF', and volser is the serial number of the volume being defragmented. It is used by DEFRAG to keep track of an extent when it is being relocated. This DSCB is not cataloged, and is automatically deleted by DFDSS after a successful run. In the unlikely event that a job cancellation or abnormal termination occurs while DEFRAG is relocating an extent, this DSCB will remain on the volume. After the restart, DADSM functions may fail with message IEC602. To correct this problem or to delete the "SYS1.DFDSS.DEFRAG.xxxxxxxx.volser.DUMMY" entry, rerun DEFRAG on the volume.

DEFRAG issues a RESERVE on the VTOC to serialize access to the VTOC. DEFRAG also issues exclusive enqueues on data sets before relocating an extent. This ensures data integrity on a single processor, but does not ensure integrity of

## DEFRAG Command

	Password- Protected	Defined ERASE <sup>1</sup>	RACF- Protected	Erased on Scratch
User	-	-	-	No
Install	Yes	-	-	Yes
Exit	-	Yes	-	Yes
= ERASE	-	-	Yes (=ERASE)	Yes
(default)	Yes	-	Yes (=NOERASE)	No
	Yes	-	Yes (=ERASE)	Yes
	-	Yes	Yes (=NOERASE)	No
	-	Yes	Yes (=ERASE)	Yes
	Yes	Yes	Yes (=NOERASE)	Yes
	Yes	Yes	Yes (=ERASE)	Yes
	Yes	Yes	Yes (=NOERASE)	Yes
User Install Exit = NOERASE	Does not matter what type of protection the data set has.			NO
<sup>1</sup> The catalog entry contains the ERASE attribute specified when the data set was defined (VSAM only).				

**Figure 33. Data Set Erase Table for DEFRAG (with RACF Version 1 Release 7)**

data sets on DASD shared between multiple processors. To ensure integrity of data sets on shared DASD, you must vary the volume offline from all processors except the one on which DEFRAG runs. After the DEFRAG run terminates, you can vary the volume online for the other processors.

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.

During a DEFRAG operation, the VTOC is serialized to prevent any DADSM activity via a RESERVE macro on the volume. This serialization has to be ended via the DEQ macro prior to generating the ending statistics provided by message ADR213I. Therefore, it is possible for another job to allocate or delete data sets on the processed volume between the time the DEQ is issued and the ending statistics are obtained.

Because DEFRAG does not relocate VSAM data sets not cataloged in the integrated catalog facility catalog, ISAM, unmovable, ABSTR data sets, VTOC index data set, catalogs, or those in the EXCLUDE list, the effectiveness of a DEFRAG run will be affected by the presence of those data sets.

When DEFRAG relocates data set extents, the data set is enqueued (via the ENQ macro). You will notice that LINKLIST data sets and other system data sets may remain open but not enqueued. If defragmenting a volume with these data sets, they should be put in the EXCLUDE list of DEFRAG to prevent their 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 could cause excessive wait time for other jobs to update the VTOC. Therefore, times of low system activity are best for DEFrag runs.

## 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.

Data sets can be selected by searching the catalog or the VTOC. If you specify DATASET and do not specify input volume(s), DFDSS performs a *logical* data set DUMP using information in the catalog(s) to select data sets. If you specify DATASET and either LOGINDDNAME or LOGINDYNAM, DFDSS performs a logical data set DUMP using information in the VTOC(s) to select data sets. If you specify INDDNAME or INDYNAM (specifying either FULL, TRACKS, or DATASET), DFDSS performs a *physical* dump. See “Logical and Physical Processing” on page 3 for more on the subject.

You can use the DUMP command to:

- Dump data sets or a volume(s) in device-independent format, which can then be restored to a like or unlike device.
- Back up data sets selected on the basis of their fully or partially qualified names and numerous data attributes. For example, you can back up only those data sets changed since the last backup (incremental backup).
- Back up multivolume data sets without having to specify any or all of the input volumes, and consolidate the extents in one dump data set.
- Dump only used space for SAM and PAM data sets and employ run-length data compression to save space on the dump tape.
- Dump user catalogs, including the catalogs’ aliases.
- Create multiple (up to 255) dump copies with a single DUMP command.
- Delete (scratch and uncatalog) data sets if DELETE is specified.
- Automatically uncatalog data sets.

DFDSS facilitates backup and recovery procedures for physical data set dumps by printing the names of data sets dumped and the serial numbers and data set sequence numbers of the backup tape volumes on which the dump of a DASD volume begins and ends.

By using “DD DUMMY” for the output data set, you can scratch and uncatalog data sets without dumping them.

### Logical Volumes of a Physical Dump

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. On a *physical* data set RESTORE, data sets from one or more of these 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 parameter to specify the volume serial numbers of the source DASD volumes you want to restore. Data from a logical volume can only be restored to a DASD volume of a like device type.



# DUMP Command

## Error Termination

Unless overridden via the user installation options exit routine, DFDSS continues dumping as long as there is at least one output copy that does not have an output error.

## Authorization Checking

DFDSS performs authorization checking to ensure that you are allowed to access a source volume and/or a data set. See “How DFDSS Checks Authorization for You” on page 167 for authorization checking details.

## Input Serialization

DFDSS uses a number of serialization schemes to ensure data integrity. See “Read/Write Access Serialization Scheme” on page 183 for an overview.

Figure 34 and Figure 35 show the syntax of the DUMP command. Before using the command, you should review “DUMP Command Usage” on page 94.

Command	Parameters
DUMp	<pre> <b>{{FULL   FUL }  </b> <b>{TRACKS   TRKS}{(c1,h1,c2,h2) [...] } }</b>  <b>{{INDDname   IDD}{(ddn)  </b> <b>{INDYnam   IDY} (volser[,unit]) }</b>  <b>OUTDDname   ODD (ddn [...])</b>  Optional parameters that can be used with FULL:  <b>ALLData { (dsn [...])   (*) }</b>  <b>ALLExp   ALLX</b>  <b>CANcelerror</b>  <b>COMpress</b>  <b>OPTimize (n)</b>  <b>PASsword   PSWD ( [ddn]</b> <b>                    [dsn/pswd [...]]</b> <b>                    [catname/pswd [...]])</b>  <b>RESEt</b>  <b>TOLerate(IOERror)</b>  <b>TRACKOverflow   TRKO (dsn [...])</b>  <b>WAIT (numsecs,numretrys)</b>  Optional parameters that can be used with TRACKS:  <b>CANcelerror</b>  <b>COMpress</b>  <b>OPTimize (n)</b>  <b>PASsword   PSWD ( [ddn [...]]</b> <b>                    [dsn/pswd [...]]</b> <b>                    [catname/pswd [...]])</b>  <b>TOLerate(IOERror)</b>  <b>TRACKOverflow   TRKO (dsn [...])</b>  <b>WAIT (numsecs,numretrys)</b> </pre>

Figure 34. DUMP Command Syntax—FULL or TRACKS

# DUMP Command

Command	Parameters
<b>DUMp</b>	<p> <b>{DATASet   DS} { {FILterdd   FDD} (ddn)</b>  <b>or</b>  <b>{DATASet   DS} ( [ INClude(dsn[,...]) ]</b>  <b>                  [ EXClude(dsn[,...]) ]</b>  <b>                  [ BY((schar,op,arg) [...]) ] )</b> </p> <p> <b>OUTDDname   ODD (ddn [...])</b> </p> <p> <b>Optional parameters that can be used with DATASET:</b> </p> <p> <b>ALLData { (dsn [...])   (*) }</b> </p> <p> <b>ALLExcp   ALLX</b> </p> <p> <b>ALLMulti</b> </p> <p> <b>CANcelerror</b> </p> <p> <b>COMpress</b> </p> <p> <b>DElete</b> </p> <p> <b>DYNALloc</b> </p> <p> <b>{{INDDname   IDD} (ddn [...])</b>    <b>{INDYnam   IDY} ((volser [,unit]) [...])</b>    <b>{LOGINDDname   LIDD} (ddn [...])</b>    <b>{LOGINDYnam   LIDY} ((volser [,unit]) [...]) }</b> </p> <p> <b>OPTimize (n)</b> </p> <p> <b>PASsword   PSWD ( [ddn</b>  <b>                          [dsn/pswd [...]]</b>  <b>                          [catname/pswd [...]] )</b> </p> <p> <b>PURge   PRG</b> </p> <p> <b>RESEt</b> </p> <p> <b>SHAre   SHR</b> </p> <p> <b>TOLerate({IOERror</b>  <b>                  [ENQFailure])</b> </p> <p> <b>TRACKOverflow   TRKO (dsn [...])</b> </p> <p> <b>UNCATalog</b> </p> <p> <b>WAIT (numsecs,numretrys)</b> </p>

Figure 35. DUMP Command Syntax—DATASET

## Explanation of DUMP Command Parameters

### **FULL | FUL**

specifies that an entire DASD volume is to be dumped. This is the default. Unallocated tracks are not dumped. Unless specified via 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).

### **TRACKS | TRKS ((c1,h1,c2,h2) [...])**

specifies ranges of tracks to be dumped. To restore the data, this entire range or its subset must be specified during a subsequent RESTORE.

*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 may be omitted. For example:

Specified	Results
None	Syntax error
c1	c1,0,c1,maximum head number
c1,h1	c1,h1,c1,maximum head number
c1,h1,c2	c1,h1,c2,maximum head number
c1,,c2	Syntax error
,h1	Syntax error

### **DATASet | DS ({FILterdd | FDD} (ddn))**

or

### **DATASet | DS ( [ INClude(dsn,[...]) ] [ EXClude(dsn,[...]) ] [ BY((schar,op,arg) [...]) ] )**

specifies a data set DUMP operation, using filtering. See "Choosing the Data Sets You Want Processed—Filtering" on page 21 for an explanation of the filtering process used. Unless specified via 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.

where:

### **FILterdd | FDD (ddn)**

*ddn*

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

# DUMP Command

command syntax, that contain the INCLUDE, EXCLUDE, and BY parameters described below.

## **INClude**(*dsn*[,...])

*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 “Using Qualified Data Set Names” on page 21. 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.

## **EXClude**(*dsn*[,...])

*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.

## **BY**((*schar*,*op*,*arg*) [...])

specifies that the data sets selected up to this point, by the processing of the INCLUDE and EXCLUDE parameters, are to be further filtered. To select the data set, **all** BY criteria must be met.

*schar*

specifies the applicable selection characteristic, consisting of one of the following:

- ALLOC (allocation type)
- CATLG (data set cataloged)
- CREDIT (creation date)
- DSCHA (data-set-changed flag)
- DSORG (data set organization)
- EXPDT (expiration date)
- 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)
- MULTI (multivolume data set)
- REFDT (last-referenced date)

*op*

specifies the operator, consisting of one of the following:

- 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*

specifies an argument meaningful for the selection characteristic used.

Figure 36 on page 84 shows the permissible combinations of *schar*, *op*, and *arg*.

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)	If MOV is picked, the data sets to be dumped cannot be allocated as any of the following: <ul style="list-style-type: none"> <li>• PSU (physical sequential unmovable)</li> <li>• POU (partitioned organization unmovable)</li> <li>• DAU (BDAM unmovable)</li> <li>• ABSTR (absolute tracks)</li> <li>• ISAM (indexed sequential access method)</li> </ul>
CATLG	EQ	YES (or 1) NO (or 0)	If YES is specified, only cataloged data sets are processed. If NO is specified, only uncataloged data sets are processed. (Only valid for a physical DUMP.)
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.	If you specify a yy of 00 through 49, DFDSS assumes you want years 2000 through 2049; if you specify a yy of 50 through 99, DFDSS assumes 1950 through 1999. CREDIT and EXPDT: For a multivolume data set, the date from its first volume's VTOC is used for checking. REFDT: For a multivolume data set, the latest date from all its VTOCs is used for checking.
DSCHA	EQ NE	YES (or 1) NO (or 0)	For a multivolume data set, the value used for checking is 1 if any part of the data set has been changed. Otherwise, the value used is 0. VSAM does not set the DSCHA bit in the VTOC for the index component of a KSDS cluster; if the data and index components reside on different volumes, the index component will not be dumped during a physical data set dump.
DSORG	EQ NE	SAM PAM BDAM ISAM VSAM EXCP	EXCP applies to data sets not allocated as any of the listed organizations and not accessed using any of the listed access methods.
EXTNT FSIZE	LT GT EQ NE GE LE	A 1- to 5-digit number (0 to 99999)	This is the number of used extents (EXTNT) or used tracks (FSIZE) for an entire data set on all the volume(s) on which it resides. 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 only processes multivolume data sets. If NO is specified, it only processes single-volume data sets.

Figure 36. BY Parameters—DUMP Command

## DUMP Command

Here are some examples of the BY parameters:

```
BY((ALLOC EQ CYL) (CATLG EQ YES))  
or  
BY(FSIZE GE 100)
```

### INDDname | IDD (*ddn* [...])

*ddn*

specifies the name of your DD statement that identifies the input volume to be dumped. Only one *ddn* is allowed for a full or tracks DUMP; one or more are allowed for a physical data set DUMP. To assure correct processing, each of the DD statements corresponding to a *ddname* (*ddn*) must identify only one volser.

### INDYnam | IDY ((*volser* [,*unit*] ) [...])

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

*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 parameter is optional.

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

### OUTDDname | ODD (*ddn* [...])

*ddn*

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

### ALLData { (*dsn* [...]) | (\*) }

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. See footnote.<sup>6</sup>

\*

specifies that all allocated space is to be dumped for *all* SAM and PAM data sets, and data sets with a null DSORG field, selected in the

---

<sup>6</sup> 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*) or ALLEXCP to dump all its data. Examples of data sets that may have these characteristics are JES2/JES3 data sets, and CICS journal and MSVCJRNL data sets.

## DUMP Command

DUMP operation, 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 | ALLX

Applies to full and data set DUMP operations, and specifies that, for data sets whose DSCB DS1LSTAR field is zero (SAM or PAM data sets, or data sets with a null DSORG field), all allocated space is to be dumped. See footnote.<sup>6</sup>

### ALLMulti

specifies that cataloged, multivolume data sets that have volumes missing from the LOGINDDNAME or LOGINDYNAM volume list will be dumped 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 dumped for applicable data sets. The ALLMULTI parameter is valid only when LOGINDDNAME or LOGINDYNAM is specified.

### CANceIerror

specifies that the DUMP is to be terminated if a track-related permanent read error occurs. If this parameter is *not* specified and a permanent read error occurs, the track image record is flagged on output 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. 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.

The stand-alone restore program cannot be used to restore from tapes produced using the COMPRESS parameter. The system residence volume should not be dumped using the COMPRESS parameter.

### DELeTe

For a *physical* data set DUMP, specifies that expired single-volume non-VSAM data sets that are successfully serialized and dumped are to be scratched. In addition, DFDSS is to uncatalog successfully scratched data sets. DELETE is ignored for VSAM data sets. A data set with a high-level qualifier of "SYS1.", a page data set, or a swap data set cannot be 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 will be deleted only if you also specify PURGE. Do not specify SHARE if you specify DELETE.

### DYNALloc

specifies that DFDSS, in a data set DUMP operation, is to use dynamic allocation, instead of the ENQ macro, to serialize the use of data sets. This



## DUMP Command

allows cross-system serialization in a JES3/MVS environment. This is only of value if the dynamic allocation/JES3 interface is not disabled. For more information, see "DYNALLOC" on page 167.

### LOGINDDname | LIDD (*ddn* [...])

*ddn*

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

### LOGINDYnam | LIDY ((*volser* [,*unit*] ) [...])

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

*unit*

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

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. A multivolume data set that has extents on volumes not specified with LOGINDDNAME or LOGINDYNAM will not be dumped if you omit ALLMULTI.

### OPTimize (*n*)

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 this parameter 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 parameter uses more real and virtual storage (see "Storage Requirements—DFDSS" on page 181). It also keeps the channel busy for longer blocks of time. There is no performance gain on mass storage virtual volumes by specifying OPTIMIZE.

Do not specify COMPRESS if you specify OPTIMIZE. You must choose between performance (specifying OPTIMIZE) or saving space (specifying COMPRESS).

### PASsword | PSWD ( [*ddn*]

[*dsn/pswd* [...]]

[*catname/pswd* [...]] )

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 user authorization exit does not bypass the checks.
- You do not want to be prompted for the password.

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

*ddn*

specifies the name of your DD statement that identifies the sequential data set, or member of a partitioned data set, containing 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 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 your input command stream is suppressed in the SYSPRINT output.

### **PURge | PRG**

for a data set DUMP, specifies deletion of unexpired data sets that are dumped successfully. This parameter is only valid 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. Do not specify SHARE if you specify RESET. You may not want to specify RESET if you use a storage management program, such as Data Facility Hierarchical Storage Manager.

### **SHAre | SHR**

specifies that DFDSS is to share for read access with other programs, the data sets to be dumped. See “Read/Write Access Serialization Scheme” on page 183 for information about data set enqueues. 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.

### **TOLerate([ENQFailure] [IOERror])**

#### **ENQFailure**

specifies that data sets are to be processed even though shared or exclusive access fails. For more information on TOL(ENQF), see “ENQ” on page 165.

#### **IOERror**

specifies that DFDSS is to continue processing even though input errors occur, but to terminate after 100 errors. This applies to all input errors, including equipment check and busout parity. TOL(IOERror) is ignored if CANCELerror is specified. If a permanent read error occurs, the track image record is flagged on output as having an I/O error and the DUMP continues. This track is cleared in a RESTORE operation.

# DUMP Command

## TRACKOverflow | TRKO (*dsn*[,...])

*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 parameter is not needed for data sets indicated as having overflow records.

When performing 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.<sup>7</sup> 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. See "Efficiency Considerations" on page 94 for details.

## UNCATalog

specifies, for a *physical* DUMP, that DFDSS is to uncatlog successfully dumped single-volume non-VSAM data sets that are currently cataloged for the volume being dumped.

specifies, for a *logical* DUMP, that DFDSS is to uncatlog successfully dumped single- or multivolume non-VSAM data sets that are currently cataloged. (For VSAM data sets, see the DELETE parameter.)

## WAIT(*numsecs,numretrys*)

For a *volume, tracks, or physical data set* DUMP, 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.

*numretrys*

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.

For a *logical data set* DUMP, specifies to DFDSS the length of 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 time interval, in seconds, to wait before attempting another pass through the list of selected data sets.

*numretrys*

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.

---

<sup>7</sup> Track packing is a technique used by DFDSS that builds target tracks for any DASD device 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.

## DUMP Command

The default 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 *numretrys*. For more information, see "WAIT(numsecs,numretrys) Option for Serialization" on page 164.

### Examples of Full and Tracks DUMP Operations

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

Examples 1A and 1B below are complemented by Examples 1A and 1B, respectively, in the RESTORE section.

```
//JOB1    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADDRSSU
//SYSPRINT DD   SYSOUT=A
//DASD    DD     UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//TAPE    DD     UNIT=3480,VOL=SER=(TAPE01,TAPE02),
// LABEL=(1,SL),DISP=(NEW,KEEP),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(1,0,1,15) INDDNAME(DASD) -
OUTDDNAME(TAPE)
```

### Examples of Data Set DUMP Operations

Specified data sets on DASD volumes 111111 and 222222 are to be dumped to the first data set of standard label tape volume TAPE02.

Examples 2A through 2H below are complemented by Examples 2A through 2D in the RESTORE section, in any combination.

```
//JOB2    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADDRSSU
//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,KEEP),DSNAME=USER2.BACKUP
//SYSIN   DD     *
command input (see Examples 2A, 2B, ... below)
/*
```

#### *Example 2A: Using the INCLUDE Subparameter*

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

# DUMP Command

## **Example 2B: Using the INCLUDE and EXCLUDE Subparameters**

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

## **Example 2C: Using the INCLUDE, EXCLUDE, and BY Subparameters**

```
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)
```

*Note:* The following DD 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.

**Example 2H: Dumping data sets that are constantly in use**

```
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)).

**Example 2J: 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.

```
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=ADRSSU
//TAPE      DD   UNIT=3480,VOL=SER=TAPE04,
//          LABEL=(1,SL),DISP=(NEW,KEEP),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 them are multivolume, source DASD volumes are not specified, resulting in data set selection by catalog.

The above example 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=ADRSSU
//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,KEEP),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.

## DUMP Command

The above example 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=ADRDSU
//SYSPRINT  DD  SYSOUT=A
//TAPE      DD  UNIT=3480,VOL=SER=TAPE04,
//          LABEL=(1,SL),DISP=(NEW,KEEP),DSN=USER3.BACKUP
//SYSIN     DD  *
            DUMP DATASET(INCLUDE(**)) -
            ALLMULTI -
            LOGINDYNAM(338001) -
            OUTDDNAME(TAPE) -
            DELETE PURGE
/*
```

### DUMP Command Usage

For other notes on the DUMP operation, see: “Space Management DUMP Techniques” on page 163, “Multivolume Data Sets—Processing” on page 181, and from “DUMP Operation—Dumping to a DASD Sequential Data Set” on page 175 to “DUMP Operation—OPTIMIZE Parameter” on page 176.

#### General Notes

- The default block size for output records written to tape is 32760 bytes (32760 is also a maximum). 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. To have the block size specification included in the tape label, you must specify the BLKSIZE parameter in the DD statement for the tape. See also “Block Size” on page 161.

#### Efficiency Considerations

The amount of backup space used during a DUMP operation can be minimized if you do any or all of the following:

- Perform a data set DUMP, not a full volume DUMP.
- Use the COMPRESS parameter to save dump space.
- Dump only data sets changed since the last backup.
- Dump only the used space (the default), instead of all allocated space, in SAM and PAM data sets, or data sets with a null DSORG field.
- Dump to tape, where larger block sizes can be used.

The number of I/O operations executed during a DUMP operation can be minimized if you do any or all of the following:

## DUMP Command

- Dump to tape, where a block size of 32760 bytes is used.
- 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.

In a DUMP operation, 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, that are not flagged as such in the VTOC's format-1 DSCB control block, are specified in the TRACKOVERFLOW parameter, 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.

### *Physical Data Set DUMP Notes*

1. You should take special care while dumping multivolume data sets to make sure that the segments from all the volumes are dumped at the same time.
2. Data sets that are defined on the source volume with null or zero extents and not selected by the filtering criteria are not copied to the dump tape. Hence, they cannot be restored.
3. Data sets allocated with the MXIG JCL parameter are dumped only up to the space used at dump time. At restore time, if they are not preallocated, they are allocated without the MXIG parameter.
4. If you specify input volume(s) (INDDNAME or INDYNAM), you can uncatalog single-volume non-VSAM data sets as follows:
  - If you specify DELETE, successfully dumped single-volume non-VSAM data sets are scratched. Such successfully scratched data sets are uncataloged. Unmovable data sets are not deleted.
  - The UNCATALOG parameter applies only to single-volume non-VSAM data sets. If specified, successfully dumped data sets that are currently cataloged for the volume being dumped are uncataloged but not scratched. This could be useful if you want a copy of the data set on the dumped volume to remain while the data set is restored to another volume and cataloged there.
  - If DELETE is specified, UNCATALOG is superfluous. If both are specified, the data sets are uncataloged only if they are successfully scratched.
  - The catalog that DFDSS uses to uncatalog a data set is determined by the standard search order (see "Standard Catalog Search Order" on page 24). Hence, you are required to supply the catalogs in a STEPCAT or a JOBCAT DD statement for those data sets which are not cataloged in the master catalog and for those data set names whose first qualifiers are not aliases for user catalog(s) or CVOL(s).
5. You should use the IDCAMS utility to delete (scratch and uncatalog) VSAM data sets.
6. DFDSS does not support key range data sets.



# DUMP Command

## *Logical Data Set DUMP Notes*

1. DFDSS selects data sets via the catalog if input volume(s) are not specified. DFDSS selects data sets via the VTOC if input volume(s) are specified using LOGINDDNAME or LOGINDYNAM.
2. When you dump a multivolume VSAM data set, the size of each data set component must be less than or equal to the size of one source volume less one cylinder.
3. A standard user label for a multivolume data set is not supported.
4. DFDSS cannot be used to dump data sets with a volume serial of MIGRAT.
5. A logical data set dump cannot be used to dump the master catalog.
6. When you dump an ICF user catalog, its aliases are automatically dumped. The user catalog name must be fully qualified. In an MVS/XA DFP Version 2 Release 3 environment, the LOCK attribute of an ICF 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 and the proper access authority).
7. When you dump data sets cataloged in a CVOL, ensure that the number of data sets to be dumped is fewer than 1455.

## PRINT Command

With the PRINT command, you can print DASD data to SYSPRINT or to a sequential data set in print format. You can print:

- A non-VSAM data set, as specified by a fully qualified name. The data set location need not be known.
- VSAM data set components (not clusters). 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.

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 while 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 “How DFDSS Checks Authorization for You” on page 167.

Figure 37 shows the syntax of the PRINT command.

# PRINT Command

Command	Parameters
PRInt or PRT	<pre> {{DATASet   DS} (dsn)   {TRACKS   TRKS} ((c1,h1,c2,h2) [...])     VTOC(a,b) }  {{INDDname   IDD} (ddn)   {INDYnam   IDY} (volser [,unit] ) }           </pre> <p>Optional parameters that can be used with DATASET:</p> <p>ALLData</p> <p>DATALength(<i>n</i>)</p> <p>DYNALloc</p> <p>ERRortracks</p> <p>KEYlength(<i>n</i>)</p> <p>OUTDDname   ODD (<i>ddn</i>)</p> <p>PASsword   PSWD ( [<i>ddn</i>]                   [<i>dsn/pswd</i> [...]] )</p> <p>SHAre   SHR</p> <p>TOLerate([ENQFailure]           [IOERror])</p> <p>WAIT (<i>numsecs,numretrys</i>)</p> <p>Optional parameters that can be used with TRACKS or VTOC:</p> <p>DATALength(<i>n</i>)</p> <p>ERRortracks</p> <p>KEYlength(<i>n</i>)</p> <p>OUTDDname   ODD (<i>ddn</i>)</p> <p>PASsword   PSWD ( [<i>ddn</i> [...]]                   [<i>dsn/pswd</i> [...]] )</p> <p>TOLerate(IOERror)</p> <p>WAIT (<i>numsecs,numretrys</i>)</p>

Figure 37. PRINT Command Syntax

## Explanation of PRINT Command Parameters

### DATASet | DS (*dsn*)

*dsn*

specifies the fully qualified name of the data set to be printed. (See "Using Qualified Data Set Names" on page 21.) The data set is printed in logical sequence.

### TRACKS | TRKS ((*c1,h1,c2,h2*),...)

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 may be omitted. For example:

Specified	Results
None	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(*a,b*)

specifies that all or part of the VTOC is to be printed. Omitting the VTOC, DATASET, and TRACKS parameters 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 may 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')

# PRINT Command

## **INDDname | IDD (*ddn*)**

*ddn*

specifies the name of your DD statement that identifies a volume to be printed. If you want to print a multivolume data set, you must print one volume at a time.

## **INDYnam | IDY (*volser* [, *unit*])**

specifies that the volume 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 parameter is optional.

## **ALLData**

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

## **DATALength(*n*)**

*n*

specifies the logical length 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.

## **DYNALloc**

specifies that DFDSS use dynamic allocation, instead of enqueue, to serialize the use of the data set. This allows cross-system serialization in a JES3/MVS environment. This is only of value if the dynamic allocation/JES3 interface is not disabled. For more information, see "DYNALLOC" on page 167.

## **ERRortracks**

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

## **KEYLength(*n*)**

*n*

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

## **OUTDDname | ODD (*ddn*)**

*ddn*

specifies the name of your DD statement that identifies the (output) print data set. To assure correct processing, each of the DD statements corresponding to a ddname (*ddn*) must identify only one volser. If this parameter is not specified, the default is SYSPRINT. If the DCB parameters LRECL and BLKSIZE are specified on the DD statement, these must be in the ranges of 84 to 137 and 88 to 141 respectively; otherwise, these values are overridden by the standard values 137 and 141, respectively.

## **PASsword | PSWD ( [*ddn*] [*dsn/pswd* [...]] )**

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:

## PRINT Command

- You do not have the required volume-level RACF DASDVOL or RACF DATASET access.
- The user 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.

*ddn*

specifies the name of your DD statement that identifies the sequential data set, or member of a partitioned data set, containing 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 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 via the input stream. They are required on tracks PRINT operations only.

### SHAre | SHR

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

### TOLerate([ENQFailure] [IOERror])

#### ENQFailure

specifies that data sets are to be processed even though shared or exclusive access fails. For more information on TOL(ENQF), see “ENQ” on page 165.

#### IOERror

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

### WAIT(*numsecs*,*numretrys*)

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 time interval, in seconds, between retries.

*numretrys*

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 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 *numretrys*.

# PRINT Command

## 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 15 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=A
//SYSUDUMP DD    SYSOUT=A
//DD1     DD     UNIT=3380,VOL=SER=DSST80,DISP=OLD
//SYSIN    DD     *
PRINT INDD(DD1) DS(PRINT.SAMPLE)
/*
```

The output shown in Figure 38 on page 103 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 DFDSS 5665-327 DATA FACILITY DATA SET SERVICES V2R3.0 87.007 11:07
PRINT DATASET(TEST.SAM.S01) -
ADR101I RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'PRINT '
INDYNAM ( -
D8C020 -
)
ADR109I RI01 (01), 87007 11:07:57 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR006I (001)-SETUP(01), 87007 11:07:57 EXECUTION BEGINS
ADR317I (001)-OPNCL(01), THE DEVICE SIZE FIELD IN THE VTOC IS INCORRECT IF VOLUME D8C020 IS NOT A MINIVOLUME. DEVICE SIZE OF 0020
CYLINDERS WILL BE USED
*** TRACK(CCHH) 00000006 RO DATA 0000000600000000
COUNT 0000000601000320
0000 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9C1C2 C3C4C5C6 *ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEF*
0020 C7C8C9D1 D2D3D4D5 D6D7D8D9 E2E3E4E5 E6E7E8E9 C1C2C3C4 C5C6C7C8 C9D1D2D3 *GHIJKLMNOPQRSTUVWXYZABCDEFHIJKL*
0040 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9C1C2 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 *MNOPQRSTUVWXYZABCDEFGHIJKLMN*
0060 D8D9E2E3 E4E5E6E7 E8E9C1C2 C3C4C5C6 C7C8C9D1 D2D3D4D5 D6D7D8D9 E2E3E4E5 *QRSTUVWXYZABCDEFGHIJKLMN*
0080 E6E7E8E9 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9C1C2 *WXYZABCDEFGHIJKLMN*
00A0 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9C1C2 C3C4C5C6 *ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEF*
00C0 C7C8C9D1 D2D3D4D5 D6D7D8D9 E2E3E4E5 E6E7E8E9 C1C2C3C4 C5C6C7C8 C9D1D2D3 *GHIJKLMNOPQRSTUVWXYZABCDEFHIJKL*
00E0 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9C1C2 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 *MNOPQRSTUVWXYZABCDEFGHIJKLMN*
0100 D8D9E2E3 E4E5E6E7 E8E9C1C2 C3C4C5C6 C7C8C9D1 D2D3D4D5 D6D7D8D9 E2E3E4E5 *QRSTUVWXYZABCDEFGHIJKLMN*
0120 E6E7E8E9 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9C1C2 *WXYZABCDEFGHIJKLMN*
0140 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9C1C2 C3C4C5C6 *ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEF*
0160 C7C8C9D1 D2D3D4D5 D6D7D8D9 E2E3E4E5 E6E7E8E9 C1C2C3C4 C5C6C7C8 C9D1D2D3 *GHIJKLMNOPQRSTUVWXYZABCDEFHIJKL*
0180 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9C1C2 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 *MNOPQRSTUVWXYZABCDEFGHIJKLMN*
01A0 D8D9E2E3 E4E5E6E7 E8E9C1C2 C3C4C5C6 C7C8C9D1 D2D3D4D5 D6D7D8D9 E2E3E4E5 *QRSTUVWXYZABCDEFGHIJKLMN*
01C0 E6E7E8E9 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9C1C2 *WXYZABCDEFGHIJKLMN*
01E0 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9C1C2 C3C4C5C6 *ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEF*
0200 C7C8C9D1 D2D3D4D5 D6D7D8D9 E2E3E4E5 E6E7E8E9 C1C2C3C4 C5C6C7C8 C9D1D2D3 *GHIJKLMNOPQRSTUVWXYZABCDEFHIJKL*
0220 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9C1C2 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 *MNOPQRSTUVWXYZABCDEFGHIJKLMN*
0240 D8D9E2E3 E4E5E6E7 E8E9C1C2 C3C4C5C6 C7C8C9D1 D2D3D4D5 D6D7D8D9 E2E3E4E5 *QRSTUVWXYZABCDEFGHIJKLMN*
0260 E6E7E8E9 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9C1C2 *WXYZABCDEFGHIJKLMN*
0280 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9C1C2 C3C4C5C6 *ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEF*
02A0 C7C8C9D1 D2D3D4D5 D6D7D8D9 E2E3E4E5 E6E7E8E9 C1C2C3C4 C5C6C7C8 C9D1D2D3 *GHIJKLMNOPQRSTUVWXYZABCDEFHIJKL*
02C0 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9C1C2 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 *MNOPQRSTUVWXYZABCDEFGHIJKLMN*
02E0 D8D9E2E3 E4E5E6E7 E8E9C1C2 C3C4C5C6 C7C8C9D1 D2D3D4D5 D6D7D8D9 E2E3E4E5 *QRSTUVWXYZABCDEFGHIJKLMN*
0300 E6E7E8E9 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E9C1C2 *WXYZABCDEFGHIJKLMN*
ADRO13I (001)-CLTSK(01), 87007 11:07:59 TASK COMPLETED WITH RETURN CODE 0000
ADRO12I DSSU (01), 87007 11:08:00 DFDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000

```

Figure 38. Output Resulting from a PRINT Command



# RELEASE Command

The RELEASE command releases, on specified volume(s), 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 the Data Sets You Want Processed—Filtering” on page 21. DFDSS only selects data sets that have releasable space. The RELEASE function requires MVS/XA DFP Version 2 Release 1.0 or later release.

You should exclude, via the EXCLUDE parameter, 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.

Figure 39 shows the RELEASE command syntax.

Command	Parameters
RELease	<pre> {{{FILterdd   FDD} (ddn) }   { [ INClude (dsn [...]) ]     [ EXClude (dsn [...]) ]     [ BY((schar,op,arg) [...]) ] }  { DDName (ddn [...])               DYNAM ((volser [,unit] ) [...]) }  Optional parameters that can be used with RELEASE:  DYNALloc  MINSECqty(n)  MINTRAcksunused   MINTRKS (n)  PASsword   PSWD ( [ddn]                   [dsn/pswd [...]] )  WAIT(numsecs,numretrys) </pre>

Figure 39. RELEASE Command Syntax

## Explanation of RELEASE Command Parameters

### FILterdd | FDD (ddn)

*ddn*

specifies the name of your DD statement that identifies the sequential data set, or member of a partitioned data set, containing 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 parameters described below. If you have more than 255 entries in the INCLUDE, EXCLUDE, or BY sublist, you can use this method.

## INClude(*dsn*[...])

*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 "Using Qualified Data Set Names" on page 21. 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.

## EXClude(*dsn*[...])

*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.

## BY(*schar,op,arg*) [...]

specifies that the data sets selected up to this point, by the processing of the INCLUDE and EXCLUDE parameters, are to be further filtered. To select the data set, *all* BY criteria must be met.

*schar*

specifies the applicable selection characteristic, consisting of one of the following:

- ALLOC (allocation type)
- CATLG (data set cataloged)
- CREDIT (creation date)
- DSCHA (data-set-changed flag)
- DSORG (data set organization)
- EXPDT (expiration date)
- EXTNT (number of allocated extents)
- FSIZE (number of allocated tracks)
- MULTI (multivolume data set)
- REFDT (last-referenced date)

*op*

specifies the operator, consisting of one of the following:

- 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*

specifies an argument meaningful for the selection characteristic used.

Figure 40 shows the permissible combinations of *schar*, *op*, and *arg*.

Here are some examples of the BY parameters:

```
BY((ALLOC EQ CYL) (CATLG EQ YES))  
or  
BY(FSIZE GE 100)
```

## RELEASE Command

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)	If MOV is picked, the data sets to be released cannot be allocated as any of the following: <ul style="list-style-type: none"> <li>• PSU (physical sequential unmovable)</li> <li>• POU (partitioned organization unmovable)</li> <li>• DAU (BDAM unmovable)</li> <li>• ABSTR (absolute tracks)</li> <li>• ISAM (indexed sequential access method)</li> </ul>
CATLG	EQ	YES (or 1) NO (or 0)	If YES is specified, only cataloged data sets are processed. If NO is specified, only uncataloged data sets are processed.
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 one-hundredth 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.	If you specify a yy of 00 through 49, DFDSS assumes you want years 2000 through 2049; if you specify a yy of 50 through 99, DFDSS assumes 1950 through 1999.  For both single- and multivolume data sets, the date in the VTOC of the volume being processed is used for filtering.
DSCHA	EQ NE	YES (or 1) NO (or 0)	For a multivolume data set, the value obtained from the VTOC of the volume being processed is used for checking.
DSORG	EQ NE	SAM PAM	
EXTNT FSIZE	LT GT EQ NE GE LE	A 1- to 5-digit number (0 to 99999)	This is the number of allocated extents (EXTNT) or allocated tracks (FSIZE) for a data set on the volume being processed.
MULTI	EQ	YES (or 1) NO (or 0)	If YES is specified, DFDSS only processes multivolume data sets. If NO is specified, it only processes single volume data sets.

Figure 40. BY Parameters—RELEASE Command

### DDName (*ddn* [...])

specifies the name of your 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 volser.

### DYNAM((*volser*[,*unit*]) [...])

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 parameter is optional.

## **DYNALloc**

specifies that DFDSS use dynamic allocation, instead of the ENQ macro, to serialize the use of data sets. This allows cross-system serialization in a JES3/MVS environment. This is only of value if the dynamic allocation/JES3 interface is not disabled. For more information, see "DYNALLOC" on page 167.

## **MINSECqty(*n*)**

Because all the allocated but unused space would be released, if you do not specify a secondary allocation quantity you would not be able to add records to the data set after the RELEASE operation. MINSECQTY (*n*) 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 | MINTRKS (*n*)**

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.

## **PASsword | PSWD ( [*ddn*] [*dsn/pswd* [,...]] )**

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 user authorization exit does not bypass the checks.
- You do not want to be prompted for the password.

*ddn*

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

## **WAIT(*numsecs,numretrys*)**

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.

## RELEASE Command

### *numsecs*

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

### *numretrys*

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 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 *numretrys*.

### Example of a RELEASE Operation

#### *A RELEASE Operation on Selected Sequential and Partitioned Data Sets*

```
//JOB1      JOB  accounting information,REGION=nnnk
//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:

- The number of unused tracks in the data set is greater than or equal to 10, and
- 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 */
/*
```

### RELEASE Command Usage

The following applies to the RELEASE command:

- A data set with 16 allocated extents does not have any space released.
- Free tracks in cylinder-allocated extents are not released. Only free cylinders are released.
- DFDSS excludes data sets beginning with "SYS1."

## RESTORE Command

With the RESTORE command, you can restore data to DASD volume(s) from DFDSS-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.

If the data being restored has been compressed, it is automatically expanded to its original form during the RESTORE operation.

You can restore the data on dump volume(s) as follows:

- If the dump volume(s) resulted from a *full* DUMP operation, you can do a full or a tracks RESTORE (that is, range(s) of tracks) or a data set RESTORE.
- If the dump volume(s) resulted from a *tracks* DUMP operation (that is, range(s) of tracks), you must do a tracks RESTORE, which can consist of a subset of the dump data.
- If the dump volume(s) resulted from a *physical data set* DUMP operation, you must do a *physical* data set RESTORE, or a tracks RESTORE, which can consist of a subset of the dump data. You must specify the output volume (OUTDDNAME or OUTDYNAM parameter). A physical data set RESTORE restores only the first volume in a list passed in the OUTDDNAME or OUTDYNAM parameter.
- If the dump volume(s) resulted from a *logical data set* DUMP operation, a logical data set RESTORE occurs automatically. Specifying the output volume(s) is optional; if you do not specify output volume(s), the selected data sets are restored to their original volume(s).

With the RESTORE operation, you can, optionally, copy the volume serial number (VOLID) to the output DASD if track 0 is restored.

Passwords can be specified via the input or via a DD card for a data set. See “How DFDSS Checks Authorization for You” on page 167 for more information about authorization checking.

User catalog aliases are restored when present on the dump tape, if the catalog is restored successfully. See “Logical Data Set RESTORE Notes” on page 130 for a full explanation.

Figure 41 and Figure 42 show the syntax of the RESTORE command. Before using the command, you should review “RESTORE Command Usage” on page 126.

# RESTORE Command

Command	Parameters
RESTore	<p> <b>{{FULL   FUL }              {TRACKS   TRKS }((c1,h1,c2,h2) [...]) }</b> </p> <p> <b>INDDname   IDD (ddn)</b> </p> <p> <b>{{OUTDDname   ODD}(ddn)              {OUTDYnam   ODY} (volser [,unit]) }</b> </p> <p>Optional parameters that can be used with FULL:</p> <p> <b>CANcelerror</b> </p> <p> <b>COPYVolid   CPYV</b> </p> <p> <b>PASsword   PSWD ( [ddn]            [dsn/pswd [...]]            [catname/pswd [...]] )</b> </p> <p> <b>PURge   PRG</b> </p> <p> <b>WAIT(numsecs,numretrys)</b> </p> <p> <b>WRIttecheck   WRC</b> </p> <p>Optional parameters that can be used with TRACKS:</p> <p> <b>CANcelerror</b> </p> <p> <b>COPYVolid   CPYV</b> </p> <p> <b>OUTTRacks   OUTTRKS ((c1,h1) [...])</b> </p> <p> <b>PASsword   PSWD ( [ddn [...]]            [dsn/pswd [...]]            [catname/pswd [...]] )</b> </p> <p> <b>PURge   PRG</b> </p> <p> <b>WAIT(numsecs,numretrys)</b> </p> <p> <b>WRIttecheck   WRC</b> </p>

Figure 41. RESTORE Command Syntax—FULL or TRACKS

Command	Parameters
RESTore	<p> <b>{DATASet   DS} ({FILTerdd   FDD} (ddn))</b>  <b>or</b>  <b>{DATASet   DS} ( [ INClude(dsn[,...]) ]</b>  <b>                  [ EXClude(dsn[,...]) ]</b>  <b>                  [ BY((schar,op,arg) [...]) ] )</b> </p> <p> <b>INDDname   IDD (ddn)</b> </p> <p>           Optional parameters that can be used with DATASET:         </p> <p> <b>CANceLerror</b> </p> <p> <b>CATalog   RECATalog(newcatname)   RECATalog(*)</b> </p> <p> <b>DYNALloc</b> </p> <p> <b>FORce</b> </p> <p> <b>LOGicalvolume   LVOL (volser [...])</b> </p> <p> <b>MENtity(modeldsn) [MVOlser(volser)]</b> </p> <p> <b>{{OUTDDname   ODD}(ddn [...])  </b>  <b>{OUTDYnam   ODY}((volser [,unit] ) [...]) }</b> </p> <p> <b>PASsword   PSWD ( [ddn</b>  <b>                          [dsn/pswd [...]]</b>  <b>                          [catname/pswd [...]] )</b> </p> <p> <b>PERCENTUtilized   PCTU ( n [...])</b> </p> <p> <b>REName { (pfx)  </b>  <b>          ((on,nn) [...])  </b>  <b>          ((pfx),(on,nn) [...]) }</b> </p> <p> <b>RENAMEUnconditional   RENUNC { (pfx)  </b>  <b>                                  ((on,nn) [...])  </b>  <b>                                  ((pfx),(on,nn) [...]) }</b> </p> <p> <b>REPlace</b> </p> <p> <b>SHAre   SHR</b> </p> <p> <b>TGTAlloc(BLK   CYL   TRK   SOURCE   SRC)</b> </p> <p> <b>TOLerate(ENQFailure)</b> </p> <p> <b>WAIT(numsecs,numretrys)</b> </p> <p> <b>WRItcheck   WRC</b> </p> <p> <b>Note:</b> If "PURGE" is specified, a message will indicate that "REPLACE" was used instead.         </p>

Figure 42. RESTORE Command Syntax—DATASET



# RESTORE Command

## Explanation of RESTORE Command Parameters

### **FULL | FUL**

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

### **TRACKS | TRKS ((c1,h1,c2,h2) [...])**

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 terminated.

#### *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 may be omitted. For example:

Specified	Results
None	Syntax error
c1	c1,0,c1,maximum head number
c1,h1	c1,h1,c1,maximum head number
c1,h1,c2	c1,h1,c2,maximum head number
c1,,c2	Syntax error
,h1	Syntax error

### **DATASet | DS ({FILterdd | FDD} (ddn))**

or

### **DATASet | DS ( [ INClude(dsn[...]) ] [ EXClude(dsn[...]) ] [ BY((schar,op,arg) [...]) ] )**

specifies a data set RESTORE operation, using filtering. See "Choosing the Data Sets You Want Processed—Filtering" on page 21 for an explanation of the filtering process used.

### **FILterdd | FDD (ddn)**

#### *ddn*

specifies the name of your DD statement that identifies the sequential data set, or member of a partitioned data set, containing 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 subparameters described below. If you have more than 255 entries in the INCLUDE, EXCLUDE, or BY sublist, you can use this method.

### **INClude(dsn[...])**

#### *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 "Using Qualified Data Set Names" on page 21. If INCLUDE is omitted (but EXCLUDE or BY is specified) or if

“INCLUDE(\*\*)” is specified, *all* data sets are eligible to be selected for restoring.

## **EXCLUDE**(*dsn*[...])

*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.

## **BY**((*schar,op,arg*) [...])

specifies that the data sets selected up to this point, by the processing of the INCLUDE and EXCLUDE parameters, are to be further filtered. To select the data set, **all** BY criteria must be met.

*schar*

specifies the applicable selection characteristic, consisting of one of the following:

- ALLOC (allocation type)
- CATLG (data set cataloged)
- CREDIT (creation date)
- DSCHA (data-set-changed flag)
- DSORG (data set organization)
- EXPDT (expiration date)
- 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)
- MULTI (multivolume data set)
- REFDT (last-referenced date)

*op*

specifies the operator, consisting of one of the following:

- 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*

specifies an argument meaningful for the selection characteristic used.

Figure 43 shows the permissible combinations of *schar*, *op*, and *arg*.

Here are some examples of the BY parameters:

```
BY((ALLOC EQ CYL) (CATLG EQ YES))  
or  
BY(FSIZE GE 100)
```

## RESTORE Command

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)	If MOV is picked, the data sets to be restored cannot be allocated as any of the following: <ul style="list-style-type: none"> <li>• PSU (physical sequential unmovable)</li> <li>• POU (partitioned organization unmovable)</li> <li>• DAU (BDAM unmovable)</li> <li>• ABSTR (absolute tracks)</li> <li>• ISAM (indexed sequential access method)</li> </ul>
CATLG	EQ	YES (or 1) NO (or 0)	If YES is specified, only cataloged data sets are processed. If NO is specified, only uncataloged data sets are processed.
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 one-hundredth 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.	If you specify a yy of 00 through 49, DFDSS assumes you want years 2000 through 2049; if you specify a yy of 50 through 99, DFDSS assumes 1950 through 1999. CREDIT and EXPDT: For a multivolume data set, the date from its first volume's VTOC is used for checking. REFDT: For a multivolume data set the latest date from all its VTOCs is used for checking.
DSCHA	EQ NE	YES (or 1) NO (or 0)	For a multivolume data set, the value used for checking is 1 if any part of the data set has been changed. Otherwise, the value used is 0.
DSORG	EQ NE	SAM PAM BDAM ISAM VSAM EXCP	EXCP applies to data sets not allocated as any of the listed organizations and not accessed using any of the listed access methods.
EXTNT FSIZE	LT GT EQ NE GE LE	A 1- to 5-digit number (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. The dump data set determines the number of used or allocated extents or tracks.
MULTI	EQ	YES (or 1) NO (or 0)	If YES is specified, DFDSS only processes multivolume data sets. If NO is specified, it only processes single volume data sets.

Figure 43. BY Parameters—RESTORE Command

INDDname | IDD (*ddn*)

*ddn*

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

## OUTDDname | ODD (*ddn* [...])

*ddn*

specifies the name of your 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 data set RESTORE or spill files, you can specify multiple names, separated by commas.

## OUTDynam | ODY ((*volser* [,*unit*] ) [...])

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 data set RESTORE or 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 parameter is optional.

OUTDDNAME or OUTDYNAM is required when restoring from a physical dump; they are optional for a logical RESTORE.

## CANceIerror

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

## CATalog | RECATalog(*newcatname*) | RECATalog(\*)

specifies that on a *logical* RESTORE, DFDSS is 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 and multivolume 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.

CATALOG or RECATALOG fails if the target data set is already cataloged in the same catalog, and either RENAME is not specified or DELETE or UNCATALOG was not specified for the DUMP. CATALOG and RECATALOG are ignored when the target data set is preallocated.

## COPYVolid | CPYV

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. If you are doing a full RESTORE, and the input has VSAM data sets, the VOLID should be copied.

## RESTORE Command

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

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

For more information on COPYVOLID, see “RESTORE Operation—Full Volume Restore and RACF-Protected Data Sets” on page 183.

### **DYNALloc**

specifies that DFDSS, in a data set RESTORE operation, use dynamic allocation, instead of the ENQ macro, to serialize the use of data sets. This allows cross-system serialization in a JES3/MVS environment. This is only of value if the dynamic allocation/JES3 interface is not disabled. For more information, see “DYNALLOC” on page 167.

### **FORce**

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

### **LOGicalvolume | LVOL (volser[...])**

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 only want to restore some data sets from source volume 222222, specify LOGICALVOLUME (222222). This is useful for restoring multivolume data sets. See “Multivolume Data Sets—Processing” on page 181, for details.

### **MENTity(modeldsn) [MVOLser(volser)]**

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 parameters are used to define the data sets to RACF. Specification of MVOLSER is optional when the model entity (MENTITY) is cataloged in an integrated catalog facility catalog, or it is a non-VSAM data set and that MENTITY is cataloged in the standard catalog search order. When MVOLSER is specified for a VSAM model entity, the *volser* specified should be the volser of the model entities cataloged. If these parameters 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 it was RACF-protected (with a discrete profile) during DUMP and either the data set is not preallocated or it is preallocated but is not RACF protected with a discrete profile. For a description of RACF data sets, refer to *Resource Access Control Facility (RACF) General Information Manual*.

### **OUTTRacks | OUTTRKS ((c1,h1) [...])**

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

**PASsword | PSWD ( [ddn]  
[dsn/pswd [...]]  
[catname/pswd [...]] )**

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 user authorization exit does not bypass the checks.
- You do not want to be prompted for the password.

*ddn*

specifies the name of your DD statement that identifies the sequential data set, or member of a partitioned data set, containing 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 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 your input command stream is suppressed in the SYSPRINT output.

**PERCENTUtilized | PCTU (n [...])**

specifies that DFDSS should 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). This parameter is valid only for logical data set restore operations. PERCENTUTILIZED is ignored when the target data set is preallocated.

**PURge | PRG**

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

**REName {(pfx) |  
(on,n) [...]} |  
(pfx),(on,n) [...]} }**

specifies that, if a data set exists on the output DASD volume with the old name, DFDSS is to allocate a new data set with the new name and restore the data set. If the data set does not exist on the volume with the old name, 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 parameter applies to movable data sets only. See “Data Set RESTORE—RENAME, RENAMEUNCONDITIONAL, and REPLACE Parameters” on page 126, for details. RENAME and RENAMEUNCONDITIONAL are mutually exclusive.

## RESTORE Command

*px*

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 sublist. 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*

The old name used as filtering criteria to check if it matches the data set name.

*nn*

The new name used to derive the new data set name if the data set name matches the corresponding old name filtering criteria.

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 alphameric

The syntax for the **old name** filter is exactly like that of the **INCLUDE** filter, and their rules match.

The syntax for the **new name** filter is as follows:

- **\*\***—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.
- **\*\* .DATA .\*\***—Invalid (level to be replaced is ambiguous).
- **\***—If dsname has one level, then restore with old name.
- **\*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).
- **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".

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.

## RESTORE Command

**RENAMEUnconditional | RENUNC** {(*pfx*) |  
  (*on,nn*) [...] |  
  (*pfx*),(*on,nn*) [...] } }

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 parameter applies to movable data sets only. If the old name filter matches and there is an error in the new name filter, the data set is not restored. See “Data Set RESTORE—RENAME, RENAMEUNCONDITIONAL, and REPLACE Parameters” on page 126 for details.

RENAME and RENAMEUNCONDITIONAL are mutually exclusive. RENAMEUNCONDITIONAL is not supported for physical restore of VSAM data sets.

*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 sublist. 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*

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

*nn*

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

The syntax rules for prefix, old name, and new name are the same as for the RENAME parameter and are explained above.

### REPlace

specifies, for a data set RESTORE, that, if the data set exists on DASD, it can be overlaid by that from the source volume. DFDSS will search the output volumes for usable preallocated data sets. If no output volumes are specified, DFDSS will search 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 as equivalent to REPLACE.

For non-VSAM data sets, if the target data set size is smaller than the source, DFDSS will scratch the target data set and reallocate it with the size of the source data set. REPLACE must be specified to restore preallocated unmovable data sets. This parameter is also required for movable data sets if a preallocated data set is to be overlaid.

### SHAre | SHR

specifies that DFDSS is to share, for read access with other programs, the data sets that are to be restored. See “Read/Write Access Serialization Scheme” on page 183 for information about data set enqueues. The resetting function is bypassed if SHARE is specified on a data set RESTORE.

### TGTAlloc(BLK | CYL | TRK | SOURCE | SRC)

specifies, during a logical data set RESTORE, how DFDSS will allocate the target data set.



## RESTORE Command

**BLK**  
specifies allocate by blocks.

**CYL**  
specifies allocate by cylinders.

**TRK**  
specifies allocate by tracks.

**SOURCE | SRC**  
specifies allocate with the same space allocation type as the source data set.

If the TGTALLOC parameter is omitted, the target allocation defaults to tracks. If BLK is specified for VSAM data sets, TRK is used instead. The final VSAM allocation may be different than the requested one because of VSAM allocation rules.

**TOLerate(ENQFailure)**  
specifies that data sets are to be processed even though shared or exclusive access fails. For more information on TOL(ENQF), see "ENQ" on page 165.

**WAIT(numsecs,numretrys)**  
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.

*numretrys*  
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 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 *numretrys*.

**WRitecheck | WRC**  
Specifies, during a RESTORE, that the data written is to be verified for successful completion. This parameter increases the overall elapsed time. The default is NO WRITECHECK.

### Examples of Full and Tracks RESTORE Operations

DASD volume 111111 is to be restored from the first data set of standard label tape volumes TAPE01 and TAPE02.

## RESTORE Command

Examples 1A and 1B below complement Examples 1A and 1B, respectively, in the DUMP section.

```
//JOB1 JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADDRSSU
//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(1,0,1,15) INDDNAME(TAPE) -
OUTDDNAME(DASD) PURGE
```

### **Example 1C: A Tracks RESTORE—Restore to Different Tracks**

```
//JOBTRKS JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADDRSSU
//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 may 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 Data Set RESTORE Operations

Specified 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 2H in “DUMP Command,” in any combination; for example, the dump tape produced in Example 2C in “DUMP Command” could be used as the input tape for Example 2A below.

```
//JOB2 JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADDRSSU
//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)
/*
```

## RESTORE Command

### **Example 2A: Using the INCLUDE Subparameter to Restore All Data Sets on a Dump Tape**

```
RESTORE INDDNAME(TAPE) OUTDDNAME(DASD1) -  
        DATASET(INCLUDE(**))
```

### **Example 2B: Using the INCLUDE and EXCLUDE Subparameters**

```
RESTORE INDDNAME(TAPE) OUTDDNAME(DASD1) -  
        DATASET(INCLUDE(**) -  
                EXCLUDE(**.LIST))
```

All data sets, except those in the EXCLUDE subparameter, are restored.

### **Example 2C: Using the INCLUDE, EXCLUDE, and BY Subparameters**

```
RESTORE INDDNAME(TAPE) OUTDDNAME(DASD1) -  
        DATASET(INCLUDE(**) -  
                EXCLUDE(**.LIST) -  
                BY((EXPDT,LE,80045)))
```

All data sets that satisfy the BY subparameter, except those specified in the EXCLUDE subparameter, are restored.

### **Example 2D: With Filtering Data in a Data Set**

```
RESTORE INDDNAME(TAPE) OUTDDNAME(DASD1) -  
        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(**) -  
EXCLUDE(**.LIST) -  
BY((DSCHA,EQ,1`))
```

### **Example 2E: Using the LOGICALVOLUME and REPLACE Parameters**

```
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 data sets, movable or unmovable, exist on the volume, replace the data sets. If no movable data sets exist on the volume, allocate the data set and restore.

### **Example 2F: Using the REPLACE and RENAME Parameters**

```
RESTORE INDDNAME(TAPE) OUTDDNAME(DASD1) -  
        DATASET(INCLUDE(**)) LOGICALVOLUME(111111) -  
        REPLACE -  
        RENAME((USER2),(USER4.**,USER3.**))
```

## RESTORE Command

In the example above, data sets with a first level qualifier of USER4 are renamed using 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. Renaming only takes place for those data sets that exist on DASD with the old name.

### *Example 2G: Using the REPLACE and RENAMEUNCONDITIONAL Parameters*

```
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, and REPLACE is used for unmovable data sets. With the RENUNC parameter, 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 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(**))
```

The above example 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 the example above, 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: Restore a VSAM Data Set, Using the RENAME Parameter*

```
//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. 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.

## RESTORE Command

### Example 4: Using the RECATALOG Parameter

```
//JOB4      JOB  accounting information,REGION=nnnnK
//STEP1     EXEC PGM=ADDRSSU
//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)
/*
```

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.

### Example 5: A Logical Data Set RESTORE—Output Volumes not Specified

```
//JOB5      JOB  accounting information,REGION=nnnnK
//STEP1     EXEC PGM=ADDRSSU
//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
/*
```

Data set USER1.MULTVOL is restored. The location to which it is to be restored is not given. The above RESTORE statement also applies when the data set has been scratched inadvertently after the DUMP. A multivolume data set is restored to the volumes it was dumped from, provided the data set is preallocated on the output volumes. If it is not preallocated, then the data set is restored to the first volume it was dumped from that has adequate space to restore it 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 6: Restore an Unmovable Data Set

```
//JOB6      JOB  accounting information,REGION=nnnnK
//STEP1     EXEC PGM=ADRDSSU
//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. You marked the data set as unmovable on volume 338000 either because:

- 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, or
- It was allocated as an ABSTR data set for performance reasons.

## Example 7: 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,KEEP),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 example above, 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 parameter is used to change the name of the data set to USER2.NEWDS.

# RESTORE Command

## Data Set RESTORE—RENAME, RENAMEUNCONDITIONAL, and REPLACE Parameters

Figure 44 describes, in decision table format, the RESTORE actions taken on movable data sets. The restore parameters specified and the existence of the data set are shown in the upper half. The actions taken are shown in the lower half.

	Logical Restore or Non-VSAM Physical Restore												VSAM Physical Restore			
RENAME specified?	N	N	Y	Y	Y	Y	-	-	-	-	N	N	-	-	-	-
RENUNC specified?	N	N	-	-	-	-	Y	Y	Y	Y	N	N	-	-	-	-
REPLACE specified?	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N
Old data set exists on DASD?	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N
New name on volume already?	-	-	N	Y/N	N	Y/N	N	N	N	N	-	-	-	-	-	-
Overlay old data set on DASD	X	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-
ALLOC with old name and restore	-	X	-	X	-	X	-	-	-	-	-	X	-	X	-	X
ALLOC with new name and restore	-	-	X	-	X	-	X	X	X	X	-	-	-	-	-	-
Do not restore	-	-	-	-	-	-	-	-	-	-	X	-	-	-	X	-
Legend:																
Y Yes																
N No																
X Action taken																
- Not applicable																

Figure 44. RESTORE Actions on Movable Data Sets

Figure 45 on page 127 is not a program flowchart. It is used to clarify the actions taken by the RESTORE function on non-VSAM data sets under varying conditions, as described in Figure 44.

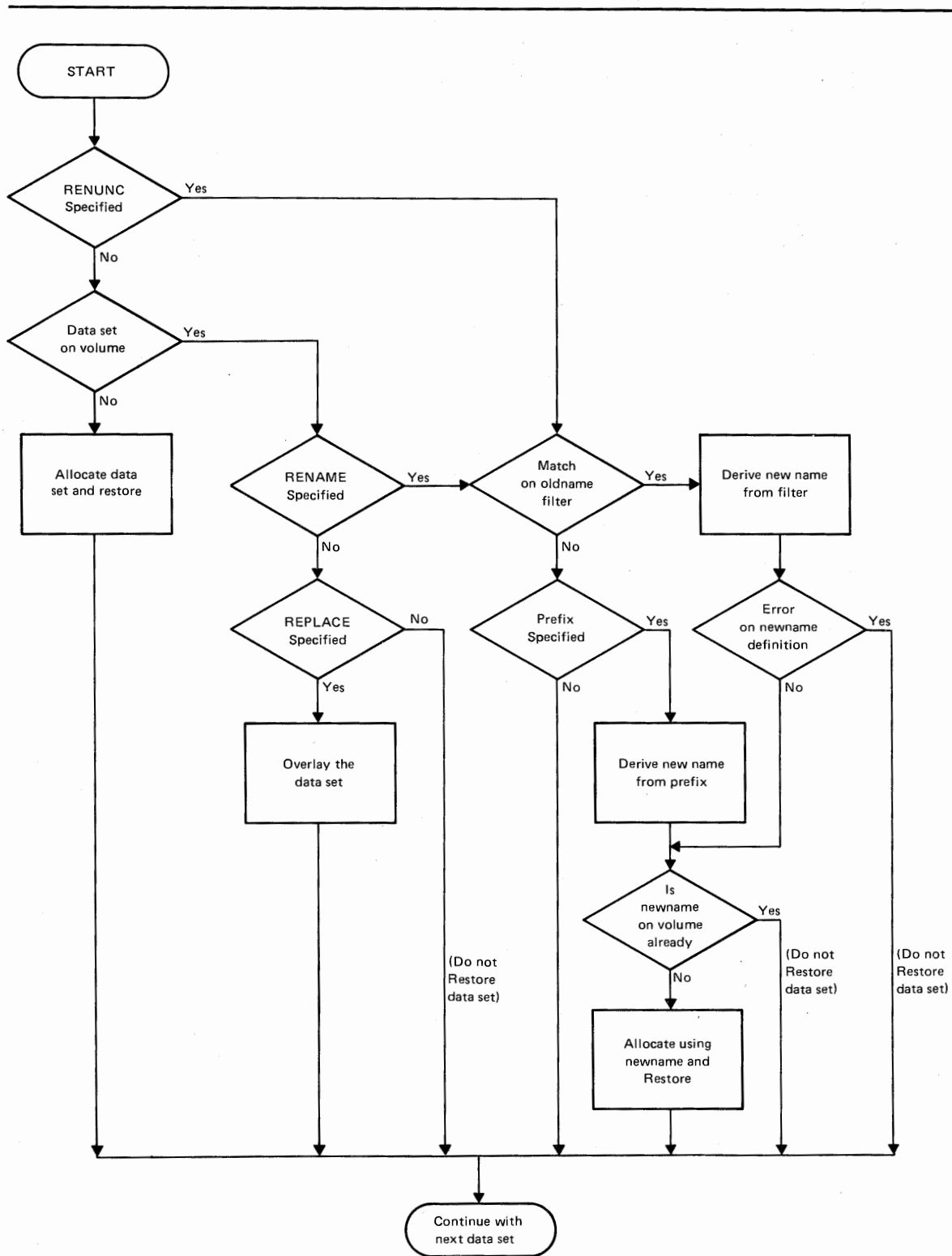
### RESTORE Command Usage

For other notes on the RESTORE operation, see from “RESTORE Operation—Full Volume Restore and RACF-Protected Data Sets” on page 183 to “RESTORE Operation—Tracks RESTORE” on page 186.

#### Full or Tracks RESTORE Notes

For a full or tracks RESTORE operation, there must be a match between the device type of the source volume used in the DUMP and the device type of the target volume used in the RESTORE. However, you can do the following:

- Data from an IBM 3330 Model 1 can be restored to an IBM 3330 Model 11.



**Figure 45. Restore Actions on Non-VSAM Movable Data Sets.** These actions apply to a data set RESTORE with RENAME, RENAMEUNCONDITIONAL, and REPLACE parameters.



## RESTORE Command

- 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.
- Dump 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.

In prior releases, 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 had to allocate a temporary data set and then rebuild the indexed VTOC. Starting with DFDSS Version 2 Release 3, 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.

### *General Data Set RESTORE Notes*

- Fully qualified names are required to restore the following:

- VVDS
- VTOCIX
- SYS1.STGINDEX
- OS catalog
- Integrated catalog facility catalog
- VSAM read-only (temporarily exported with the INHIBITSOURCE parameter) data sets

- Neither the VVDS nor the VTOCIX data set can 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.

- You can restore an ISAM data set, even if it is not preallocated, if the tracks on which it is to be restored are part of the free space and your system is MVS/XA DFP Version 2 Release 1 (or any later release). If the ISAM data set is preallocated, the location and size of the extents on the source and target volumes must match.

## RESTORE Command

### *Physical Data Set RESTORE Notes*

1. For a data set RESTORE operation, there must be a match between the device type of the source volume used in the DUMP operation and the device type of the target volume used in the RESTORE operation. However, the following can also be done:
  - Dump data from an IBM 3330 Model 1 can be restored to an IBM 3330 Model 11, and vice versa.
  - Dump 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, and vice versa.
  - Dump data from a smaller capacity IBM 3380 model can be restored to a larger capacity IBM 3380 model, and vice versa.
  - Dump data from a minivolume or a virtual volume can be restored to a real volume of like device type, and vice versa.
2. 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.
3. The protection and authorization of data sets that are restored from a dump tape created on one system and restored on another are unpredictable. They depend on the profiles and the levels of RACF on the two systems.
4. RACF profiles will not be defined for VSAM clusters.
5. Restored data sets are not automatically cataloged. However, for a data set RESTORE, you can catalog DFDSS-allocated and -restored single volume non-VSAM data sets by specifying the CATALOG parameter.
6. Be sure the segments from all volumes are restored for multivolume data sets via successive RESTORE commands.
7. Restore of a portion of a multivolume non-VSAM data set is allowed only if the volume sequence numbers of the source and destination volumes are the same.
8. DFDSS neither creates entries in the PASSWORD data set for non-VSAM data sets nor does it create entries in the catalog for VSAM password-protected data sets.
9. On a data set RESTORE from a data set dump tape, data sets with null extents are restored if they are selected using the INCLUDE, EXCLUDE, and BY criteria, and if they were selected during the DUMP operation. If you want to prevent this, you should use appropriate INCLUDE, EXCLUDE, or BY (for example, DSORG NE EXCP) criteria in the RESTORE command.
10. When performing a physical restore of many data sets, there is an initial delay while DFDSS allocates the target data sets.
11. DFDSS does not allow the restore of an open integrated catalog facility catalog. Caution should be observed when restoring a catalog. If the entries in the catalog during DUMP do not match the entries during RESTORE, some of the data sets may become inaccessible.
12. VSAM key range data sets are not supported on a data set RESTORE.
13. Restoring a VSAM cluster is allowed if the following match on the source and destination volumes:

## RESTORE Command

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.)

14. 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.
15. Cataloging non-VSAM data sets during RESTORE:

If you specify CATALOG on a data set RESTORE, DFDSS creates catalog entries for single volume non-VSAM data sets which 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 has been allocated and cataloged but encounters errors during the RESTORE operation is neither uncataloged nor scratched by DFDSS.

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 that is 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 (not preallocated) by DFDSS and use IEHPROGM to catalog multivolume non-VSAM data sets.

16. The following data sets must be preallocated if they are to be restored: those with the unmovable attribute, those that are allocated by absolute track, and ISAM data sets. In addition, the size and location of the extents on the dump volume and on the restore volume must match.
17. A dumped catalog can only be restored to its original volume.

### *Logical Data Set RESTORE Notes*

1. DFDSS protects non-VSAM password-protected data sets to provide security.
2. Integrated catalog facility user catalogs can only be restored to the same volumes from which they were dumped. The component names of the source and target user catalog must be the same.

In an MVS/XA DFP Version 2 Release 3 environment, 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*.

## RESTORE Command

User catalog aliases, if present in the dump data set, 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 it is not empty, aliases are not restored.
  - If the target catalog was preallocated and empty, aliases are restored.
3. Multiple output DASD volumes can be specified on a data set RESTORE. This is required for spill volumes when all the data sets to be restored cannot fit on a single volume.
  4. The restored data sets can be cataloged. The CATALOG parameter results in an undirected catalog, and the RECATALOG(catname) parameter results in a directed catalog.
  5. Both VSAM and non-VSAM data sets can, optionally, be renamed. The rules for renaming VSAM clusters are the same as for non-VSAM data sets. You can only rename 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 using a similar mechanism. If the component names were derived using the cluster name (such as clustername.data or clustername.index), the new component names are similarly derived.
  6. ISAM data sets and CVOL data sets cannot be restored to unlike devices.
  7. You can also restore unmovable data sets (those allocated as ABSTR, PSU, POU, or DAU) if they are not preallocated. A preallocated data set is restored if the extents match and REPLACE is specified. It is restored, even if the extents do not match, if both REPLACE and FORCE are specified. If it 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 these data sets have truly location-dependent data, you should prevent DFDSS from restoring them by specifying the names of these data sets 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.

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 any of the above mentioned actions.

8. You can specify multiple target volumes to be used as spill files. You can limit the amount of space that DFDSS is to fill up a volume via the PERCENTUTILIZED(x [,...]) parameter, where x is the percentage, from 1 to 100 (default of 100), of the allocated space to total available space on the volume. When this limit is reached, subsequent data sets are allocated to other volumes. Refer to the description under the COPY command for details.
9. When multivolume data sets are not preallocated and the specified output volumes are different from the input volumes, they are restored to a single volume, space permitting. When multiple target volumes are specified, the process of selection of target volumes is as follows:

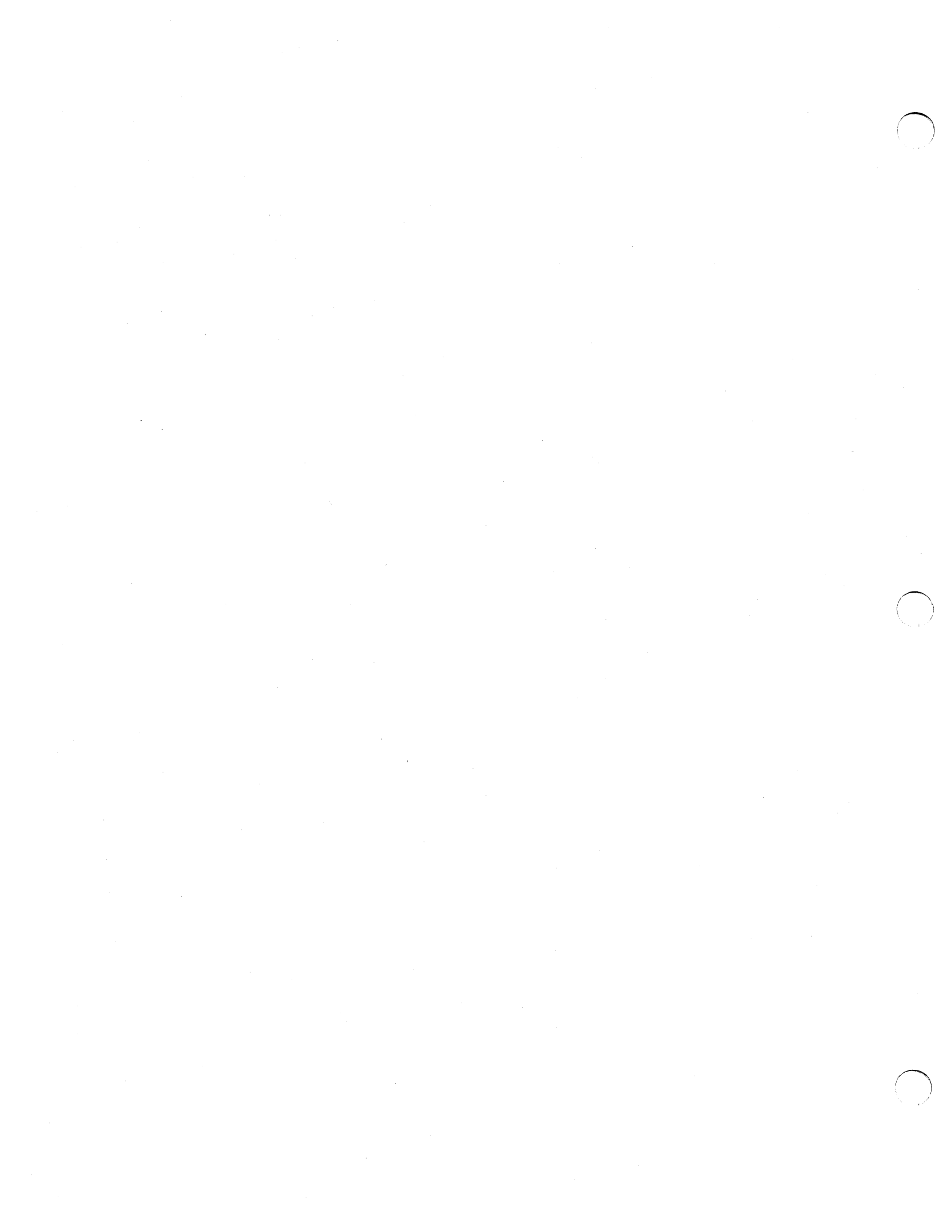
## RESTORE Command

- If a target volume that has the same volume serial as the source volume is available, and if it 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.
10. Target (output) volume specification is optional on data set RESTORE. When not specified, the volume on which the data set currently resides is found from the catalog and is dynamically allocated. This is particularly useful on a data set RESTORE from a data-set-selection-by-catalog dump, because you need not be concerned where the data sets resided at dump time. This is also useful on a data set RESTORE from a full volume dump when the data sets may have been moved to different volumes. The output volume specification is required only if the data set currently:
- 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.
11. VSAM key range data sets can be restored.
12. Single volume data sets are always restored to a single output volume. Multivolume data sets from a *logical* data set dump tape may either be restored to a single volume or to multiple volumes.
13. When restoring a VSAM KSDS or key range data set to an unlike device, the data set must be cataloged in the standard order of search (see “Standard Catalog Search Order” on page 24).
14. If a data set is preallocated, on either a like or unlike device, it will be scratched and reallocated if the target data set is not large enough to contain the dumped data set. VSAM preallocated target data sets will also be 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 RBA of a target VSAM KSDS is not 0.
15. PAM data sets will be restored to both like and unlike devices in a compressed format. If the PAM data set is being restored to an unlike device, the device dependent information (such as TTR pointers and note lists) will be in a usable form after the restore. DFDSS is unable to resolve device-dependent information for all other data set types being restored to unlike devices.
16. When restoring Direct Access Method (DAM) data sets, the following apply:

## RESTORE Command

- If the receiving volume track capacity is equal to or greater than the source volume track capacity, the DAM data set will be allocated on the receiving volume with the same allocation as the source contained. The restore thus yields a usable data set; however, some unused space may remain on each track of the target data set.
  - If the receiving volume track capacity is smaller than the source, then FORCE must be specified. This forces the DAM data set to be restored on any available space by track packing. DFDSS updates the 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 may be unusable.
17. A VSAM KSDS or key range data set that has had CA splits, being restored to an unlike device, will take significantly longer to restore.
  18. DFDSS restores a VSAM alternate index as a normal cluster. You must use IDCAMS to reestablish the connection between the alternate index and its associated cluster. The three steps to do this are:
    - a. Delete the alternate index with NOSCRATCH.
    - b. Define the alternate index with RECATALOG.
    - c. Define the path entry to the alternate index.

See the *Access Method Services Reference* for the syntax of these commands.



## Chapter 4. Controlling Tasks—Auxiliary Commands

The following **auxiliary** commands control tasks.

### Writing to the Operator

With the write-to-operator (WTO) command, you can write a message to the system console. The message is limited to 70 characters. The command syntax is:

<b>WTO</b>	' <i>message</i> '
------------	--------------------

When a WTO command is encountered, DFDSS waits until the last-requested function command completes before issuing the WTO command.

### Operating in Serial or Parallel Mode

With a SERIAL command, you can initiate serial task scheduling (the default); that is, only one task is executed at a time. Tasks are executed in the order in which they appear in the input stream. The command syntax is:

<b>SERial</b>
---------------

With a PARALLEL command, you can initiate parallel task scheduling; that is, two or more tasks are executed concurrently, provided the required system resources (such as virtual storage, DASD, or tape volumes) are available. The commands may not necessarily be executed in the order in which they appear in the input stream. The command syntax is:

<b>PARallel</b>
-----------------

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 execution until the resources are available (that is, when other tasks have terminated). If operating in PARALLEL mode, an IF statement ensures that all prior commands are executed.



## Controlling Task Execution

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. The return code of the last-requested operation (LASTCC), and the maximum return code of all completed operations (MAXCC), can be interrogated with an IF statement, or set with a SET statement. When an "IF MAXCC" or "SET MAXCC" statement is encountered, DFDSS waits for *all previously requested function commands* to complete before the highest return code is interrogated. Also, when an "IF LASTCC" or "SET LASTCC" statement is encountered, DFDSS waits for the *last-requested function command* to complete before the return code is interrogated. Following this, if the condition code is tested and satisfied, DFDSS executes 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.

### Condition Codes

The condition codes that are tested in the IF-THEN-ELSE command sequence are:

- 0 Indicates that the function was executed as directed and expected. Informational messages may have been issued.
- 4 Indicates that some problem was encountered in executing 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 execute, began execution but terminated prematurely, or the job ran without executing 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 execute. No functions were executed.
- 16 Indicates that a function executed which left at least one volume or data set in an unusable condition. For example, a full volume DUMP terminated 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 execution.

### SET Command

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 prior to the first function command.

The syntax of the SET command is:

SET	{MAXCC   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 unexecuted THEN or ELSE clause is not executed.

**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.

**IF-THEN-ELSE Command Sequence**

The IF-THEN-ELSE command sequence is used to control command execution.

The syntax of the IF-THEN-ELSE command sequence is:

<b>IF</b>	<pre>{LASTCC   MAXCC} <i>op number</i> - THEN [<i>command</i>   DO <i>command set</i> END] [ELSE [<i>command</i>   DO <i>command set</i> END]]</pre>
-----------	--

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 executed 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 executed if the tested condition is not satisfied. ELSE can be followed by another IF command.

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. (In other words, each ELSE is matched with the nearest preceding unmatched THEN.) Should there be an IF command that does not require an ELSE clause, follow the THEN clause with a null ELSE clause (see "NULL Commands," below), unless the nesting structure does not require one.

### **DO-END Command Sequence**

#### **DO**

specifies that the group of commands that follow is to be treated as a single unit, that is, to be executed as a result of a single IF command. The set of commands is terminated 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.

### **Null Command**

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  
    ELSE ...  
or
```

```
IF ... THEN;  
    ELSE ...
```

If you want to specify a null ELSE command, specify:

```
IF ... THEN ...  
    ELSE  
or
```

```
IF ... THEN ...  
    ELSE;
```

## Terminating Your DFDSS Job

With the end-of-job (EOJ) command, you can terminate your DFDSS job after the currently executing operation(s) and/or scheduled task(s) is completed. The command syntax is:

```
EOJ
```

## Examples of Controlling Task Execution

The examples that follow show the use of the IF-THEN-ELSE command sequence and the SET command.

### Example 1

In this example, nested IF commands are used to determine whether a COPYDUMP, EOJ, or PRINT command is to be executed.

```
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 executed, but, if the value of MAXCC is 12 or greater, the EOJ command is executed 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 executed.

### Example 2

In this example, nested IF commands are used to determine whether a COPYDUMP or a PRINT command is to be executed.

```
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 executed. 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 executed and a data set is to be printed.

```
IF MAXCC=0 -  
  THEN DO  
    COPYDUMP...  
    PRINT ...  
  END
```

### Example 4

If you want to set the last condition code established to 12, specify:

```
SET LASTCC=12
```

### Example 5

If you want to replace the highest condition code established in processing so far with 8, specify:

```
SET MAXCC=8
```

## Common Continuation Errors

The continuation rules, described in “Specifying DFDSS Commands—Command Syntax” on page 29, must be used carefully when auxiliary commands, comments, or blank records appear in the input stream. You must also be careful when continuing auxiliary commands so that you do not inadvertently specify a null clause.

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 executed.

#### 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 be 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.

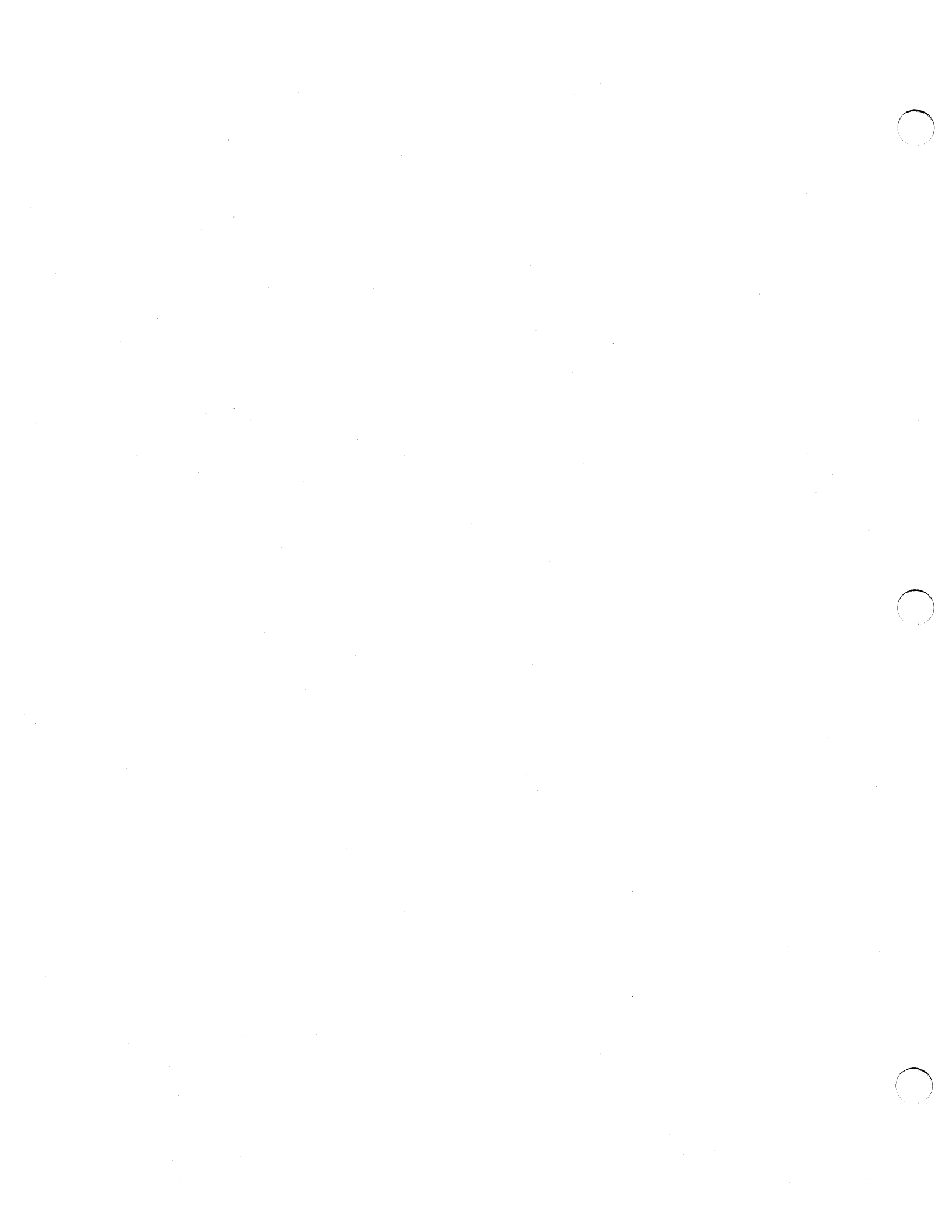
#### Error Example 3

```
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).

## Permissible Truncations of Auxiliary Commands

The keywords PARALLEL and SERIAL can be truncated to PAR and SER, respectively. No other auxiliary command keywords may be truncated. There are no alternative abbreviations for auxiliary commands.



## Chapter 5. Examples of DFDSS Jobs

DFDSS can be invoked by JCL statements or via a LINK or an ATTACH macro from a problem program. Dynamic allocation of real mounted-and-online DASD volumes and virtual DASD volumes is supported.

### Selecting an Example to Use as a Model

You may find it useful to select the example that most closely matches the task(s) you want to accomplish, and use it as a model for your DFDSS job. Possible changes, to accomplish other tasks, are discussed just below the JCL for some of the examples. Figure 46 on page 144 lists the task(s) that are accomplished by each numbered example.

In these examples, a DASD device is presented as "UNIT=3380" and a tape device as "UNIT=3480". This is for purposes of illustration only; you can substitute any supported DASD device in place of "3380" and any supported tape device in place of "3480". Refer to the preface for the publications describing device-type notation to be used in the UNIT parameter of a DD statement. Refer to Chapter 1 for information on supported devices.



Example	Task(s)
1	To copy all the data sets on a volume to another volume of unlike device type (that is, perform <b>device conversion</b> ).
2	To do a full DUMP to tape, followed, if successful, by a DEFrag. A variation of this example shows how to make two copies of the dump.
3	To dump specified data sets to tape (or to a DASD volume), using the INCLUDE subparameter.
4	To dump specified data sets from a single DASD volume to tape, using the INCLUDE, EXCLUDE, and BY subparameters, and the RESET, COMPRESS, and CANCELERROR parameters. A variation of this example is included to show the dumping of a multivolume data set to tape.
5	To restore specified data sets to a DASD volume.
6	To do a full DUMP, followed, if successful, by a COPYDUMP. The VTOC is to be printed if the DUMP fails.
7	To dump specified data sets to tape, using the INCLUDE subparameter. Upon successful dump, each data set is deleted (scratched and uncataloged). To cause these same data sets to be <i>scratched</i> instead of being dumped, a variation of this example is also included.
8	To copy a full DASD volume to a second DASD volume. If this fails, restore the second DASD volume from tape.
9	To illustrate SERIAL and PARALLEL modes.
10A	To back up all VSAM data sets that belong to a particular application, including their catalog.
10B	To recover the integrated catalog facility catalog from the dump taken in example 10A.
10C	To recover all VSAM data sets except the catalog, from the dump taken in example 10A.
10D	To recover one of the data sets from the dump taken in example 10A.
11	To restore and rename specified data sets to a DASD volume.
12	To back up a volume, using a full DUMP, compress the partitioned data sets on a volume, then release unused space in SAM and PAM data sets on the same volume.

Figure 46. Example Table of Contents

### Example 1: Device Conversion—A Data Set COPY to an Unlike Device

This example shows a data set COPY (device conversion) in which all data sets that can be processed by DFDSS on the 3330 source volumes (DASD volumes 111111 and 222222) are copied to the 3380 target volumes (DASD volumes 333333 and 444444), using up to 50% of the space on each target volume. Multivolume data sets that reside in part on volume 111111 or 222222 are copied in their entirety (ALLMULTI). The space allocation type of the source data sets is preserved (TGTTALLOC(SOURCE)). Source data sets such as VVDS and VTOCIX are not processed. If data sets with the same name are on the target volumes, they are not copied. The FORCE parameter ensures that unmovable data

sets are copied. The copied data sets are deleted (DELETE) from the source volumes and are cataloged in catalog USERCAT.

```
//JOB1    JOB    accounting information,REGION=nnnnK8
//STEP1   EXEC   PGM=ADRDSSU
//STEPCAT DD    DSN=USERCAT,DISP=SHR
//SYSPRINT DD   SYSOUT=A
//DASD1   DD    UNIT=3330-1,VOL=(PRIVATE,SER=111111),DISP=OLD
//DASD2   DD    UNIT=3330-1,VOL=(PRIVATE,SER=222222),DISP=OLD
//DASD3   DD    UNIT=3380,VOL=(PRIVATE,SER=333333),DISP=OLD
//DASD4   DD    UNIT=3380,VOL=(PRIVATE,SER=444444),DISP=OLD
//SYSIN   DD    *
COPY DATASET( -
  INCLUDE(**) )          /* INCLUDE ALL DATA SETS */ -
  LOGINDDNAME(DASD1,DASD2) ALLMULTI -
  OUTDDNAME(DASD3,DASD4) -
  PERCENTUTILIZED(50) -
  TGTALLOC(SOURCE) -
  FORCE -
  DELETE -
  CATALOG
/*
```

## Example 2: A Full DUMP, Followed by a DEFRAG

A full DASD volume is to be dumped to a tape volume. If the DUMP return code is zero, the volume is to be defragmented. During the dump, the output records are compressed to reduce the amount of tape used (COMPRESS).

```
//JOB2    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADRDSSU
//SYSPRINT DD   SYSOUT=A
//DASD    DD    UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//TAPEA   DD    UNIT=3480,VOL=SER=(TAPE01,TAPE02),
// LABEL=(1,NL),DISP=(NEW,KEEP)
//SYSIN   DD    *
DUMP INDD(DASD) OUTDD(TAPEA) COMPRESS
IF LASTCC = 0 -
  THEN DEFRAG DDN(DASD)
/*
```

Keep in mind that two tape volumes are specified in the "TAPEA" DD statement in case one volume is not sufficient.

The above example can be modified as follows if you want to make two copies of the dump. Add another DD statement:

```
//TAPEB   DD    UNIT=3480,VOL=SER=(TAPE03,TAPE04),
// LABEL=(1,NL),DISP=(NEW,KEEP)
```

Add this new ddname to the DUMP command as follows:

```
DUMP INDD(DASD) OUTDD(TAPEA,TAPEB) COMPRESS
```

---

<sup>8</sup> See "Storage Requirements—DFDSS" on page 181 for information on how to estimate *nnnn*.

### Example 3: A Physical Data Set DUMP

DASD volumes 111111 and 222222 are to be dynamically allocated (INDYN). All data sets, from the two specified DASD volumes, whose first and last qualifiers are "USER2" and "LIST", respectively, are to be dumped (INCL). The output is to be written to the first data set of an IBM standard label tape (OUTDD).

```
//JOB3      JOB    accounting information,REGION=nnnnK
//STEP1     EXEC   PGM=ADRSSU
//SYSPRINT  DD     SYSOUT=A
//TAPE      DD     UNIT=3480,VOL=SER=TAPE02,
// LABEL=(1,SL),DISP=(NEW,KEEP),DSNAME=USER2.BACKUP
//SYSIN     DD     *
DUMP  INDYN((111111,3350),(222222,3380)) OUTDD(TAPE) -
      DATASET(INCL(USER2.**.LIST))
/*
```

The above example can be modified as follows:

- If you want to run this example without using dynamic allocation, you must add a DD statement for each DASD volume from which data sets are to be dumped. Allocation of these volume(s) occurs at the beginning of the job step. For example, you could add:

```
//DASD1  DD  UNIT=3350,VOL=(PRIVATE,SER=111111),DISP=OLD
//DASD2  DD  UNIT=3380,VOL=(PRIVATE,SER=222222),DISP=OLD
```

In addition, you must replace "INDYN(...)" with an INDDNAME parameter whose subparameters point to these DD statements. For example:

```
INDD(DASD1,DASD2)
```

- If you want to specify a DASD volume as your output device, you can, for example, replace "OUTDD(TAPE)" with "OUTDD(DASD)" and replace the "TAPE" DD statement with the following:

```
//DASD DD UNIT=3380,VOL=SER=333333,DSNAME=USER2.BACKUP,
// DISP=(NEW,CATLG),SPACE=(CYL,(2,1),RLSE)
```

### Example 4: A Physical Data Set DUMP with Various Parameters

DASD volumes 111111 and 222222 (either virtual or real, mounted, and online) are to be dynamically allocated (INDYN). All data sets whose first qualifier is either "USER2" or "USER3" are to be dumped (INCL); however, those that have "USER2" and "REP" as the first and last qualifiers, respectively, are to be excluded (EXCL). Only those data sets that have been updated since the last time their data-set-changed flag was reset are to be dumped (BY). The data-set-changed flag is to be reset for all data sets successfully dumped (RESET). The dump is to be terminated if a permanent read error occurs (CANCEL). The output records are to be compressed to reduce the amount of tape used (COMPR). The output is to be written to the first data set of an IBM standard label tape (OUTDD).

```

//JOB4 JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRSSU
//SYSPRINT DD SYSOUT=A
//TAPE DD UNIT=3480,VOL=SER=TAPE03,
// LABEL=(1,SL),DISP=(NEW,KEEP),DSNAME=USER2.BACKUP
//SYSIN DD *
DUMP INDYN((DSST80,3380),(DSST81,3380)) OUTDD(TAPE) -
DATASET(INCL(USER2.** ,USER3.**)) -
EXCL(USER2.**.REP) -
BY((DSCHA,EQ,1))) -
RESET CANCEL COMPR
/*

```

JOB4 above can be modified as follows:

- If you want to run this example without using dynamic allocation, you must add a DD statement for each DASD volume from which data sets are to be dumped. Allocation of these volume(s) occurs at the beginning of the job step. For example, you could add:

```

//DASD1 DD UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//DASD2 DD UNIT=3380,VOL=(PRIVATE,SER=222222),DISP=OLD

```

In addition, you must replace "INDYN((...))" with an INDDNAME parameter whose subparameters point to these DD statements. For example:

```
INDD(DASD1,DASD2)
```

- If you want to dump a multivolume (VSAM or non-VSAM) data set that spans volumes 111111 and 222222, substitute the following control statements:

```

DUMP INDYN((111111),(222222)) OUTDD(TAPE) -
DATASET(INCL(USER2.MULTIVOL.DATA))

```

To restore and replace the above multivolume data set to volumes 111111 and 222222, do the following:

- To restore the part of the data set on volume 111111:

```

REST INDD(TAPE) OUTDYN(111111) -
DATASET(INCL(USER2.MULTIVOL.DATA)) -
REPLACE LOGICALVOLUME(111111)

```

- To restore the part of the data set on volume 222222:

```

REST INDD(TAPE) OUTDYN(222222) -
DATASET(INCL(USER2.MULTIVOL.DATA)) -
REPLACE LOGICALVOLUME(222222)

```

- If you want to dump and delete (scratch and uncatalog) all single-volume non-VSAM data sets that have not been modified since 87044 (the 44th day of 1987), substitute the following control statements:

```

DUMP INDYN((111111),(222222)) OUTDD(TAPE) -
DS(BY((REFDT,LE,87044) -
(DSORG,NE,VSAM))) -
DELETE

```

## Example 5: A Data Set RESTORE

All data sets whose first qualifier is "USER2" or "USER3" are to be restored (INCL) from the first data set of an IBM standard label tape (INDD). The RESTORE is to be terminated if a permanent read error occurs (CANCEL). Only data sets that were dumped from source DASD volume 111111 are to be restored (LVOL). Data sets might have been dumped from multiple source volumes (111111 and 222222, and so forth). The DASD volume being restored is an IBM 3380 (OUTDD).

```
//JOB5    JOB    accounting information,REGION=nnnnK
//STEP1   EXEC   PGM=ADRDSSU
//SYSPRINT DD   SYSOUT=A
//TAPE    DD     UNIT=3480,VOL=SER=TAPE04,
// LABEL=(1,SL),DISP=(OLD,KEEP),DSNAME=USER2.BACKUP
//DASD    DD     UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//SYSIN   DD     *
          RESTORE INDD(TAPE) OUTDD(DASD) -
          DATASET(INCL(USER2.** ,USER3.**)) -
          CANCEL LVOL(111111)
/*
```

If you want to do a **full RESTORE** (assuming the input tape was created by a full DUMP), remove the "DATASET(...)" and LVOL parameters.

If you want to catalog the single-volume non-VSAM data sets that were not preallocated, add the following after the LVOL parameter:

```
CATALOG
```

## Example 6: A Full DUMP, Followed by a COPYDUMP

A full DASD volume is to be dumped to the first data set of an IBM standard label tape. The dump is to be terminated if a permanent read error occurs (CANCEL).

Regarding the DUMP operation:

- If it fails for any reason, a WTO is to be performed to indicate this failure, and the VTOC is to be printed. If the PRINT operation fails, a WTO is to be performed indicating this.
- If it is successful, a WTO is to be performed to indicate this success, followed by a COPYDUMP to create two tape copies of the backup tape.

Assuming the COPYDUMP operation is performed:

- If it is successful, a WTO is to be performed to indicate this success.
- If it fails for any reason, a WTO is to be performed to indicate this failure.

```

//JOB6   JOB   accounting information,REGION=nnnnK
//STEP1  EXEC  PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=A
//DASD   DD    UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//BACKUP DD    UNIT=3480,VOL=SER=TAPE05,
// LABEL=(1,SL),DISP=(NEW,PASS),DSNAME=V111111.BACKUP
//COPY1  DD    UNIT=3480,VOL=SER=TAPE06,
// LABEL=(1,SL),DISP=(NEW,KEEP),DSNAME=V111111.BACKUP
//COPY2  DD    UNIT=3480,VOL=SER=TAPE07,
// LABEL=(1,SL),DISP=(NEW,KEEP),DSNAME=V111111.BACKUP
//SYSIN  DD    *
DUMP INDD(DASD) OUTDD(BACKUP) -
      CANCEL
IF LASTCC = 0 -
  THEN DO
      WTO 'VOL 11111 DUMP WAS SUCCESSFUL'
      COPYDUMP INDD(BACKUP) OUTDD(COPY1,COPY2)
      IF LASTCC = 0 -
        THEN WTO 'COPYDUMP WAS SUCCESSFUL'
        ELSE WTO 'COPYDUMP FAILED'
      END
  ELSE DO
      WTO 'VOL 11111 DUMP FAILED'
      PRINT VTOC(1,99) INDD(DASD)
      IF LASTCC NE 0 -
        THEN WTO 'ERROR PRINTING VTOC'
      END
/*

```

### Example 7: A Physical Data Set DUMP with the DELETE Parameter

DASD volumes 111111 and 222222 (either virtual or real, mounted, and online) are to be dynamically allocated (INDYN). All data sets, from the two specified DASD volumes, whose first and last qualifiers are "USER2" and "LIST", respectively, are to be dumped to the first data set of an IBM standard label tape. Upon successful dump, each single-volume non-VSAM data set, expired or unexpired (PURGE), is to be deleted (DELETE). You will notice that this example is the same as Example 3, except for the DELETE parameter and the technique for deleting (scratching and uncataloging) data sets without dumping, which is described following the example.

```

//JOB7   JOB   accounting information,REGION=nnnnK
//STEP1  EXEC  PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=A
//TAPE   DD    UNIT=3480,VOL=SER=TAPE02,
// LABEL=(1,SL),DISP=(NEW,KEEP),DSNAME=USER2.BACKUP
//SYSIN  DD    *
DUMP INDYN((111111,3380),(222222,3350)) OUTDD(TAPE) -
      DATASET(INCL(USER2.**.LIST)) -
      DELETE PURGE
/*

```

The above example can be modified as follows:

If you want to **delete** (scratch and uncatalog) the data set specified in the "DATASET(...)" parameter instead of dumping it, replace the "TAPE" DD statement with the following:

```

//TAPE   DD    DUMMY

```

This technique is efficient because:

- No read or write operations will actually be performed on data tracks.
- Specifying one partially qualified name is sufficient to scratch many data sets.

Remember that this technique should be used with extreme caution, because you could, unintentionally, delete (scratch and uncatlog) data sets that you forgot existed.

If you want to uncatlog only the successfully processed single-volume non-VSAM data sets and retain them on the volume, use UNCATALOG instead of DELETE in the above example.

### Example 8: A Full Volume COPY

The DASD volume whose serial number is 111111 (INDYN) is to be copied in full to the DASD volume whose serial number is 222222 (OUTDYN). If the COPY fails, a RESTORE to DASD volume 222222 (OUTDYN) is to be executed from tape (INDD). If the return code (LASTCC) from this RESTORE is greater than 4, a WTO is to be performed to indicate this failure.

```
//JOB8      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
//SYSIN     DD     *
COPY INDYN(111111) OUTDYN(222222)
IF LASTCC > 12 -
  THEN DO
    RESTORE INDD(TAPE) OUTDYN(222222)
    IF LASTCC > 4 THEN -
      WTO 'VOLUME 222222 IS UNUSABLE'
  END
/*
```

The above example can be modified as follows:

If you want to run this example without using dynamic allocation, you must add a DD statement for each DASD volume. Allocation of these volumes occurs at the beginning of the job step. For example, you could add:

```
//DASD1 DD UNIT=3330,VOL=(PRIVATE,SER=111111),DISP=OLD
//DASD2 DD UNIT=3350,VOL=(PRIVATE,SER=222222),DISP=OLD
```

In addition, you must replace INDYN(...) and OUTDYN(...) with INDD and OUTDD parameters whose subparameters point to these DD statements. For example:

```
INDD(DASD1) OUTDD(DASD2)
```

However, if the data sets being copied reside on more than one volume (for example, on DASD volumes 111111 and 333333) you must specify one DD statement per volume. For example:

```
//DASD1 DD UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//DASD2 DD UNIT=3380,VOL=(PRIVATE,SER=222222),DISP=OLD
//DASD3 DD UNIT=3380,VOL=(PRIVATE,SER=333333),DISP=OLD
```

and:

```
INDD(DASD1,DASD3) OUTDD(DASD2)
```

The following is **incorrect**:

```
//DASD1 DD UNIT=(3380,2),VOL=(PRIVATE,SER=111111,333333),
// DISP=OLD
//DASD2 DD UNIT=3380,VOL=(PRIVATE,SER=222222),DISP=OLD
```

and:

```
INDD(DASD1) OUTDD(DASD2)
```

### Example 9: Full DUMPs, Followed by DEFRAgs

In the following example, DUMP commands are executed serially. When the last DUMP terminates, the DEFRAg commands are executed in PARALLEL mode. When DEFRAgs terminate, the VTOCs of the DASD volumes are printed serially.

```
//JOB9 JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRDSU
//SYSPRINT DD SYSOUT=A
//TAPE1 DD UNIT=3480,VOL=SER=TAPE01,
// LABEL=(1,SL),DISP=(OLD,KEEP),DSNAME=BACKUP.DASD1
//TAPE2 DD UNIT=3480,VOL=SER=TAPE02,
// LABEL=(1,SL),DISP=(OLD,KEEP),DSNAME=BACKUP.DASD2
//SYSIN DD *
DUMP INDYN(111111) OUTDD(TAPE1) OPT(4)
DUMP INDYN(222222) OUTDD(TAPE2) OPT(4)
PARALLEL
DEFRAg DYNAM(111111)
DEFRAg DYNAM(222222)
SERIAL
PRINT INDYN(111111)
PRINT INDYN(222222)
/*
```

The above example can be modified as follows:

If you want to run this example without using dynamic allocation, you must add a DD statement for each DASD volume. Allocation of these volumes occurs at the beginning of the job step. For example, you could add:

```
//DASD1 DD UNIT=3330,VOL=(PRIVATE,SER=111111),
// DISP=OLD
//DASD2 DD UNIT=3350,VOL=(PRIVATE,SER=222222),
// DISP=OLD
```

In addition, you must replace INDYN(111111) with INDD(DASD1), replace INDYN(222222) with INDD(DASD2), replace DYNAM(111111) with DDNAME(DASD1), and replace DYNAM(222222) with DDNAME(DASD2).



## Example 10A: Backing Up an Integrated Catalog Facility User Catalog and the VSAM Data Sets Cataloged in It

This example shows the JCL used and the printed output resulting from a command to dump an integrated catalog facility user catalog and all the VSAM data sets cataloged in it belonging to a particular application. All the data sets had "PARTS" as the first level qualifier of the data set name. The master password is required for the catalog and update passwords are required for other data sets. Passwords are not required if you have RACF DASDVOL update access or if the user authorization exit bypasses authorization checking. All data set passwords will be suppressed and will not appear in the printed output.

```
//JOB10A JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRDSSU,PARM='UTILMSG=YES'
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//TAPE DD UNIT=3480,VOL=SER=TAPE01,
// LABEL=(1,SL),DISP=(,KEEP),DSNAME=USER2.BACKUP
//DASD DD UNIT=3380,VOL=SER=338001,DISP=OLD
//SYSIN DD *
DUMP INDD(DASD) OUTDD(TAPE) OPT(4) -
      DS(INCL(PARTS.**)) -
      PSWD(PARTS.ICFCATLG/USERMPW1,PARTS.VSAM1/MASTUPW1, -
           PARTS.VSAM2/MASTUPW1,PARTS.VSAM3/MASTUPW1, -
           PARTS.VSAM4/MASTUPW1)
/*
```

Figure 47 on page 153 shows the printed output that results.

## Example 10B: Restoring an Integrated Catalog Facility User Catalog

This example shows the JCL used and the printed output resulting from a command to restore the integrated catalog facility user catalog that was dumped in Example 10A. The fully qualified name of the catalog must be passed. The catalog is restored to the volume where it resides. The catalog should not be open at the time of the RESTORE. The master password of the master catalog is required. Passwords are not required if you have DASDVOL update access or if the user authorization exit bypasses authorization checking. All data set passwords will be suppressed and will not appear in the printed output.

```
//JOB10B JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRDSSU,PARM='UTILMSG=YES'
//SYSPRINT DD SYSOUT=A
//TAPE DD UNIT=3480,VOL=SER=TAPE01,
// LABEL=(1,SL),DISP=(OLD,KEEP),DSNAME=USER2.BACKUP
//DASD DD UNIT=3380,VOL=SER=338001,DISP=OLD
//SYSIN DD *
RESTORE OUTDD(DASD) INDD(TAPE) -
      DS(INCL(PARTS.ICFCATLG)) REPLACE -
      PSWD(PARTS.ICFCATLG/USERMPW1)
/*
```

In Figure 48 on page 153, the name of the cluster and the data component is "PARTS.ICFCATLG".

```

PAGE 0001 DFDSS 5665-327 DATA FACILITY DATA SET SERVICES V2R3.0 87.007 13:52
DUMP INDD(DASD) OUTDD(TAPE) OPT(4) -
ADR101I RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP '
DS(INCL(PARTS.**)) -
PSMD(PARTS.ICFCATLG/,PARTS.VSAM1/,
PARTS.VSAM2/,PARTS.VSAM3/,
PARTS.VSAM4/)
ADR109I RI01 (01), 87007 13:52:04 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR006I (001)-SETUP(01), 87007 13:52:04 EXECUTION BEGINS
ADR317I (001)-OPNCL(01), THE DEVICE SIZE FIELD IN THE VTOC IS INCORRECT IF VOLUME 338001 IS NOT A MINIVOLUME. DEVICE SIZE OF 0040
CYLINDERS WILL BE USED
ADR329I (001)-DTDS (01), DATA SET DUMP OF VOLUME 338001 BEGINS ON TAPE TAPE01 SEQUENCE 0001
ADR329I (001)-DTDS (02), DATA SET DUMP OF VOLUME 338001 ENDS ON TAPE TAPE01 SEQUENCE 0001
ADR378I (001)-DTDS (01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED FROM VOLUME 338001
PARTS.ICFCATLG
PARTS.VSAM1
PARTS.VSAM2
PARTS.VSAM3
PARTS.VSAM4
ADR013I (001)-CLTSK(01), 87007 13:52:39 TASK COMPLETED WITH RETURN CODE 0000
ADR012I DSSU (01), 87007 13:52:41 DFDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000

```

Figure 47. Example 10A Printed Output

```

PAGE 0001 DFDSS 5665-327 DATA FACILITY DATA SET SERVICES V2R3.0 87.007 13:52
RESTORE OUTDD(DASD) INDD(TAPE) -
ADR101I RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'RESTORE '
DS(INCL(PARTS.ICFCATLG)) REPLACE -
PSMD(PARTS.ICFCATLG/)
ADR109I RI01 (01), 87007 13:52:51 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR006I (001)-SETUP(01), 87007 13:52:52 EXECUTION BEGINS
ADR317I (001)-OPNCL(01), THE DEVICE SIZE FIELD IN THE VTOC IS INCORRECT IF VOLUME 338001 IS NOT A MINIVOLUME. DEVICE SIZE OF 0040
CYLINDERS WILL BE USED
ADR378I (001)-TTDS (02), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED FROM VOLUME 338001
PARTS.ICFCATLG PARTS.ICFCATLG RESTORED ON 338001
PARTS.VSAM1 CATINDEX.TFD5DD7F.VID87007.T9C18231 RESTORED ON 338001
ADR013I (001)-CLTSK(01), 87007 13:53:03 TASK COMPLETED WITH RETURN CODE 0000
ADR012I DSSU (01), 87007 13:53:04 DFDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000

```

Figure 48. Example 10B Printed Output

### Example 10C: Restoring VSAM Data Sets Cataloged in an Integrated Catalog Facility User Catalog

This example shows the JCL used and the printed output resulting from a command to restore the VSAM data sets that were dumped in Example 10A. To prevent the catalog from being restored, the fully qualified names were not passed. The data sets are restored to the volume where they resided, even though they could have been restored to a different volume. Keep in mind, however, that if data sets are restored to a different volume, a message is issued, warning that they may have to be cataloged in the user catalog.

If the catalog and data sets are inaccessible or contain errors, the catalog should be restored first, as in example 10B, because authorization checks are done for VSAM data sets via their catalog. Passwords are not required if you have DASDVOL update access or if the user authorization exit bypasses authorization checking. All data set passwords will be suppressed and will not appear in the printed output.

```

//JOB10C JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRSSU,PARM='UTILMSG=YES'
//SYSPRINT DD SYSOUT=A
//TAPE DD UNIT=3480,VOL=SER=TAPE01,
// LABEL=(1,SL),DISP=(OLD,KEEP),DSNAME=USER2.BACKUP
//DASD DD UNIT=3380,VOL=SER=338001,DISP=OLD
//SYSIN DD *
RESTORE OUTDD(DASD) INDD(TAPE) -
DS(INCL(PARTS.**)) REPLACE -
PSWD(PARTS.ICFCATLG/USERMPW1,PARTS.VSAM1/MASTUPW1, -
PARTS.VSAM2/MASTUPW1,PARTS.VSAM3/MASTUPW1, -
PARTS.VSAM4/MASTUPW1)
/*

```

Figure 49 shows the printed output that results.

Return code “8” is returned because of message ADR421E. To avoid this message and its return code, you could have specified DS(INCL(PARTS.V\*)).

```

PAGE 0001 DFDSS 5665-327 DATA FACILITY DATA SET SERVICES V2R3.0 87.007 13:53
RESTORE OUTDD(DASD) INDD(TAPE) -
ADR101I RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'RESTORE '
DS(INCL(PARTS.**)) REPLACE -
PSWD(PARTS.ICFCATLG/,PARTS.VSAM1/,
PARTS.VSAM2/,PARTS.VSAM3/,
PARTS.VSAM4/)
ADR109I RI01 (01), 87007 13:53:13 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR006I (001)-SETUP(01), 87007 13:53:13 EXECUTION BEGINS
ADR317I (001)-OPNCL(01), THE DEVICE SIZE FIELD IN THE VTOC IS INCORRECT IF VOLUME 338001 IS NOT A MINIVOLUME. DEVICE SIZE OF 0040
CYLINDERS WILL BE USED
ADR421E (001)-FDSRS(02), DATA SET PARTS.ICFCATLG NOT PROCESSED, FULLY QUALIFIED NAME NOT PASSED, 2
ADR378I (001)-TDDS (02), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED FROM VOLUME 338001
PARTS.VSAM1 PARTS.VSAM1.DATA RESTORED ON 338001
PARTS.VSAM2 PARTS.VSAM2.DATA RESTORED ON 338001
PARTS.VSAM3 PARTS.VSAM3.DATA RESTORED ON 338001
PARTS.VSAM3 PARTS.VSAM3.INDEX RESTORED ON 338001
PARTS.VSAM4 PARTS.T900FE5B.VDD87007.T9C18239 RESTORED ON 338001
PARTS.T90100A6.VID87007.T9C18239 RESTORED ON 338001
ADR013I (001)-CLTSK(01), 87007 13:53:29 TASK COMPLETED WITH RETURN CODE 0008
ADR012I DSSU (01), 87007 13:53:30 DFDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0008 FROM:
TASK 001

```

Figure 49. Example 10C Printed Output

### Example 10D: Restoring a Single VSAM Data Set Cataloged in an Integrated Catalog Facility User Catalog

This example shows the JCL used and the printed output resulting from a command to restore only one of the data sets that was dumped in Example 10A. To restore the data set, its fully qualified name (PARTS.VSAM1) is passed.

The data set is restored to the volume on which it resided, even though it could have been restored to a different volume. However, if a data set is restored to a different volume, a message is issued, warning that it may have to be cataloged in the user catalog. Passwords are not required if you have DASDVOL update access or if the user authorization exit bypasses authorization checking. All data set passwords will be suppressed and will not appear in the printed output.

```

//JOB10D JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRSSU,PARM='UTILMSG=YES'
//SYSPRINT DD SYSOUT=A
//TAPE DD UNIT=3480,VOL=SER=TAPE01,
// LABEL=(1,SL),DISP=(OLD,KEEP),DSNAME=USER2.BACKUP
//DASD DD UNIT=3380,VOL=SER=338001,DISP=OLD
//SYSIN DD *
RESTORE OUTDD(DASD) INDD(TAPE) -
DS(INCL(PARTS.VSAM1)) REPLACE -
PSWD(PARTS.VSAM1/MASTUPW1)
/*

```

Figure 50 show the printed output that results.

```

PAGE 0001 DFDSS 5665-327 DATA FACILITY DATA SET SERVICES V2R3.0 87.007 13:53
RESTORE OUTDD(DASD) INDD(TAPE) -
ADR101I RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'RESTORE '
DS(INCL(PARTS.VSAM1)) REPLACE -
PSWD(PARTS.VSAM1/)
ADR109I RI01 (01), 87007 13:53:40 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR006I (001)-SETUP(01), 87007 13:53:40 EXECUTION BEGINS
ADR317I (001)-OPNCL(01), THE DEVICE SIZE FIELD IN THE VTOC IS INCORRECT IF VOLUME 338001 IS NOT A MINIVOLUME. DEVICE SIZE OF 0040
CYLINDERS WILL BE USED
ADR378I (001)-TDDS (02), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED FROM VOLUME 338001
PARTS.VSAM1 PARTS.VSAM1.DATA RESTORED ON 338001
ADR013I (001)-CLTSK(01), 87007 13:53:49 TASK COMPLETED WITH RETURN CODE 0000
ADR012I DSSU (01), 87007 13:53:50 DFDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000

```

Figure 50. Example 10D Printed Output

### Example 11: A Data Set RESTORE With the RENAMEUNCONDITIONAL Parameter

```

//JOB11 JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRSSU
//SYSPRINT DD SYSOUT=A
//TAPE DD UNIT=3480,VOL=SER=TAPE04,
// LABEL=(1,SL),DISP=(OLD,KEEP),DSNAME=USER2.BACKUP
//DASD1 DD UNIT=3350,VOL=(PRIVATE,SER=111111),DISP=OLD
//DASD2 DD UNIT=3350,VOL=(PRIVATE,SER=222222),DISP=OLD
//DASD3 DD UNIT=3350,VOL=(PRIVATE,SER=333333),DISP=OLD
//SYSIN DD *
RESTORE INDD(TAPE) OUTDD(DASD1) -
DATASET(INCL(USER2.** ,USER3.**)) -
RENUNC (OLD)
RESTORE INDD(TAPE) OUTDD(DASD2) -
DATASET(INCL(USER2.** ,USER3.**)) -
RENUNC ((USER2.** ,OLDUSER2.**), -
(USER3.** ,OLDUSER3.**))
RESTORE INDD(TAPE) OUTDD(DASD3) -
DATASET(INCL(USER2.** ,USER3.**)) -
RENUNC ((OLD),(USER2.* ,OLDUSER2.*), -
(USER3.* ,OLDUSER3.*))
/*

```

All three RESTORE commands in this example will restore the data sets whose first qualifier is "USER2" or "USER3" from the first data set of an IBM standard label tape (INDD).

The data sets restored to the DASD volume whose DD name is DASD1 are to be renamed, replacing the first level qualifier of the name with the prefix "OLD". The second through last qualifiers are unchanged. The data sets restored to the DASD volume whose DD name is DASD2 are to be renamed with the first qualifier

changed from "USER2" or "USER3" to "OLDUSER2" or "OLDUSER3", respectively. The second through last qualifiers are unchanged.

The data sets restored to the DASD volume whose DD name is DASD3 are to be renamed only if they consist of exactly two qualifiers, in which case the first qualifier will be changed from "USER2" or "USER3" to "OLDUSER2" or "OLDUSER3", respectively. The second qualifiers will remain unchanged. All selected data sets that do not have exactly two qualifier names will be restored using the names derived by replacing the first level qualifier of the name with the prefix "OLD".

## Example 12: Executing DUMP, COMPRESS, and RELEASE on a Volume

This example shows a method of backing up a volume, using a full DUMP, then executing COMPRESS, RELEASE, and DEFRAG on the volume. It assumes that the job is set up to run once a week. If not, the REFDT parameter has to be changed accordingly.

```
//JOB12 JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=ADRDSSU
//TAPE DD UNIT=3480,VOL=SER=(TAPE04,TAPE05,TAPE06),
// LABEL=(1,SL),DISP=(NEW,KEEP),DSN=USER2.BACKUP
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
DUMP IDY(338000) OUTDD(TAPE) RESET
IF LASTCC = 0 -
  THEN DO
    SERIAL
    COMPRESS DYNAM(338000) -
    EXCLUDE(SYS1.***) -
    BY((REFDT GE *,-7))
    IF LASTCC = 0 -
      THEN WTO 'DFDSS COMPRESS PDS RAN SUCCESSFULLY'
      ELSE DO
        WTO 'DFDSS COMPRESS PDS FAILED.'
        RELEASE DYNAM(338000)
        DEFRAG DYNAM(338000)
      END
  END
ELSE DO
  WTO 'DUMP FAILED'
  EOJ
END
/*
```

A full DASD volume is to be dumped to the first data set of an IBM standard label tape. The data-set-changed flag for each data set serialized and successfully dumped is to be reset (RESET).

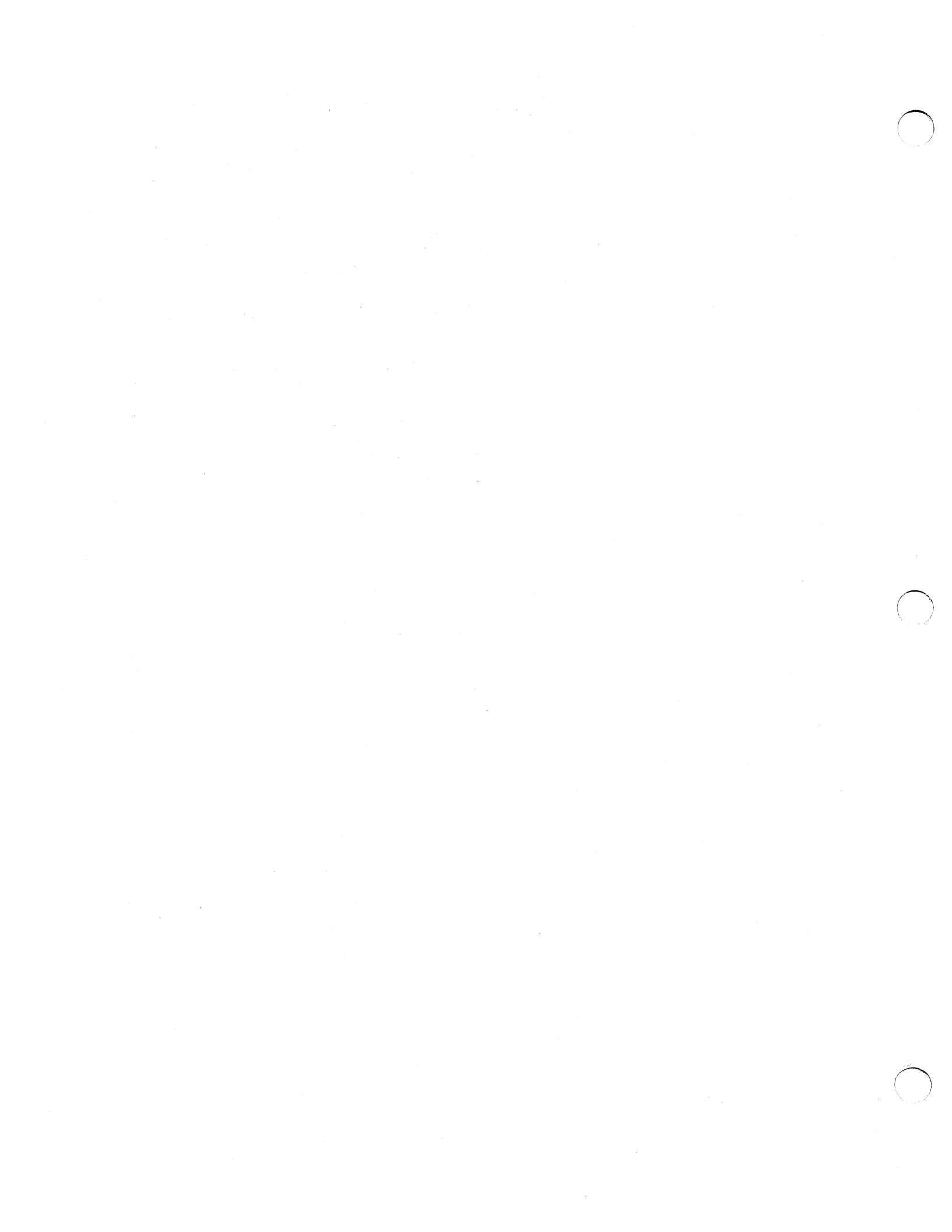
Regarding the DUMP operation:

- If it fails for any reason, a write to operator (WTO) is to be performed to indicate this failure, and the job is to be ended (EOJ).
- If it is successful, the following is to be done:
  - Serial processing (the default)

- Compress the volume, excluding system data sets, and affecting only data sets whose last referenced date is within the last seven days (REFDT GE \*-7).
- If the COMPRESS is successful, perform a WTO to that effect.
- If it fails for any reason:
  - Perform a WTO.
  - Release the allocated but unused space on volume 338000, excluding system data sets.
  - Perform a DEFrag on volume 338000.

If the system abnormally terminates when COMPRESS is executing, the volume should be restored.

If the DEFrag was not performed as a part of the run, a data set DUMP of only the sequential and partitioned data sets could have been taken.



## Chapter 6. Notes on Using DFDSS

Following are notes and hints on using DFDSS.

### Restrictions

DFDSS has the following restrictions (in addition to the restrictions listed with each command):

- DASD volumes used by DFDSS must be initialized.
- The following data sets cannot be processed via a data set COPY, data set DUMP, data set RESTORE, or PRINT operation:
  - VSAM data sets not cataloged in an integrated catalog facility catalog
  - Page, swap, and SYS1.STGINDEX data sets (data set COPY defines, but does not move, these data sets).
- Except for user catalog aliases, which are supported, alias data set names are not supported.
- Split-cylinder data sets are not supported by data set name.
- DFDSS does not support the dump formats produced by any other utilities.
- If a dump tape is created using the COMPRESS option, it cannot be used for restoring by the stand-alone version of DFDSS.
- DFDSS does not support International Organization for Standardization/American National Standards Institute (ISO/ANSI) labels, and, if they are used, there may be unpredictable results.
- In an MVS/Extended Architecture environment, DFDSS load modules must reside below 16 megabytes virtual storage.

### ADRMCLVL Macro—Dynamically Determining the DFDSS Version, Release, and Modification Level

Subsystems that invoke DFDSS dynamically need to determine whether DFDSS is installed on the system, and, if so, need to know its version, release, and modification level. A DFDSS-provided macro attempts to load the DFDSS load module. If 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 executing 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 will contain X'04000000'.
- Otherwise, register 1, byte 0 contains X'00', and the 3 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
  - Byte 3: Modification level (in binary)
    - 0 = Modification level 0
    - 1 = Modification level 1
    - 2 = Modification level 2

The expansion of the macro ADRMCLVL is given in Figure 51.

```

MACRO
ADRMCLVL
  LOAD EP=ADRRELVL,ERRET=ADR1&SYSNDX
  LR 1,0
  L 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 indeterminate)
*   byte 2: release number (0 if errors or indeterminate)
*   byte 3: modification number (0 if errors or indeterminate)

```

Figure 51. Expansion of Macro ADRMCLVL

## Allocating Volumes Dynamically

When allocating volumes dynamically, if no unit type is specified, DFDSS initially attempts to allocate the volume as a real mounted volume. If this allocation fails (with a dynamic allocation return code of X'218'; that is, the volume is not mounted), DFDSS then attempts to allocate the volume as a virtual (unmounted) volume. If the volume is not a virtual volume, message IEF710I will appear in the JOB LOG and the allocation will fail. This message may appear even though there

are no virtual volumes in the installation, and is caused by the way DFDSS allocates volumes. This message should be ignored.

## Block Size

It is recommended that you **do not** code the DCB parameters, specifically the block size in the JCL for the: (1) output DD statement on DUMP, (2) input DD statement on RESTORE, (3) input or output DD statement on COPYDUMP, or (4) LRECL or BLKSIZE for SYSPRINT or for PRINT OUTDD data sets. DFDSS assigns the block sizes, depending on the device type. The minimum allowed block size is 7892 bytes. The maximum block size for tape is 32760 bytes. 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.

For COPYDUMP, the output block size must be at least equal to the input block size. This can be a restriction if the output data set is allocated on a DASD volume.

## DASD Space Utilization

DFDSS provides functions that help you control space utilization on DASD. The most obvious function is DEFrag, but some creative usage of the DUMP command can also help in DASD space utilization.

### DEFrag Function

The DEFrag function has two algorithms, each of which attempts to reduce free space fragmentation by relocating data set extents. DEFrag works on one extent of one data set at a time, and no consolidation of extents for a data set is attempted. There are two passes made in a DEFrag run. The first algorithm is used in the first pass and the second algorithm is used in the second pass.

The first pass attempts to assemble the largest contiguous amount of free space with a minimum amount of data movement. The fragmentation index is associated with the first pass only. This explains why the fragmentation index may be a higher value after a DEFrag run than before the run.

The second pass attempts to assemble multiple groups of large contiguous free space, and will generally move data around more than the first pass. This technique will usually refine the results of the first pass but may result in the largest individual free space actually being smaller than what was on the DASD originally or after the first pass. The fragmentation index is not logically associated with the second pass, but the value of the fragmentation index is set after the second pass and therefore reflects its results.

DFDSS calculates the fragmentation index as follows:

$$\text{FRAGMENTATION INDEX} = \frac{1000 * ((\text{amt} * \ln(\text{amt})) - \text{sigma})}{\text{amt} * (\ln(\text{amt}) - 1)}$$

where:

amt = Sum of free tracks in all extents  
sigma = Sum of products of (free tracks in extent i) \*  
ln(free tracks in extent i), where i = 1 to n,  
and n = number of extents  
ln = Natural logarithm

Although each installation must determine what a good or bad fragmentation index is, a system programmer can build a conditional run based on an evaluation of that index. By use of the FRAGMENTATIONINDEX parameter, DEFRAG can effectively be changed from a full defragmentation run to a NORUN operation, where the output of the run is the same as if the NORUN option had been specified.

The system administrator will require familiarity with the one pass and multiple passes of the DEFRAG operation. There is a trade-off between execution time and completeness of defragmentation resulting from the use of the MAXMOVE parameter. Generally, defragmentation is more complete when DEFRAG makes multiple algorithm passes on a volume. If a volume is badly fragmented with numerous small data sets, the optimized second pass will usually reclaim most of the free space, resulting in very few free space extents. If this higher degree of compaction is not required at a given time or on a given volume, the first algorithm pass will usually give a good defragmentation in much less time.

## 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 execute 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 will terminate at the end of the first pass. For example:

```
DEFRAG DYNAM(V2) MAXMOVE(9999)
```

2. Experimenting with the DEFRAG FRAGI and MAXMOVE parameters will allow you to compare results when operating 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 "V1" the following could be used:

```
DEFRAG DYNAM(V1) FRAGI(3)
```

## Space Management DUMP Techniques

The DUMP command can be utilized in two ways to significantly aid in DASD space management. DUMP can help in deleting unwanted data sets, and, when combined with the RESTORE command, can be used to combine data set extents.

### DASD Space Reclamation

A physical data set DUMP, when used with the DELETE and PURGE parameters and data set filtering, is a useful tool to delete unwanted data sets from DASD. This does not apply to VSAM data sets nor to multivolume non-VSAM data sets.

On a logical data set DUMP, when using the DELETE parameter, VSAM, non-VSAM, and multivolume data sets are deleted.

*Note:* For a *physical* data set DUMP, cataloged VSAM data sets are not uncataloged with the DELETE function.

As an example, delete (scratch and uncatalog) all expired data sets and all data sets that have not been referred to in the last year. Do not actually dump these data sets.

1. JCL requirement:

```
//NOTAPE DD DUMMY
```

The above prevents the dumping of any data sets.

2. Control statements to delete (scratch and uncatalog) all non-VSAM data sets not referenced 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:* The above example can be modified to apply to VSAM and multivolume data sets by omitting the INDD statement. This results in a logical data set DUMP operation.

### Combining Data Set Extents

The DUMP command used with the DELETE and PURGE parameters will scratch and uncatalog 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. Unmovable data sets are not deleted. The volume may 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 ALLDATA and ALLEXCP are not specified for sequential and partitioned data sets, only used spaces will be allocated.

*Note:* Do not use this technique for unmovable data sets such as ABSTR allocated or ISAM.

As an example, 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
```

## Data Integrity

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 via the RESERVE macro. Data set serialization is accomplished via 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; it will not affect the integrity of any data set other than the one currently being processed. Volume serialization, however, cannot be bypassed.

### WAIT(numsecs,numretrys) Option for Serialization

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 (2,2). This means DFDSS is to retry twice at 2-second intervals.

For a data set COPY or logical data set DUMP, the WAIT parameter has a different meaning when serializing data sets and 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 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).

## RESERVE

For volume serialization, DFDSS issues a RESERVE on the SYSVTOC to prevent 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.

A VTOC is reserved when it is accessed during COMPRESS, RELEASE, and COPY (with input volume specified). It is also reserved when updating the RACF-defined flag.

## ENQ

The following is a description of DFDSS enqueue options. Enqueues are done on the data set name for data set COPY, data set DUMP, data set RESTORE, DEFrag, PRINT, COMPRESS, and RELEASE.

A VTOC is reserved when DFDSS accesses it during COPY (with input volumes), logical DUMP with input volumes, COMPRESS, or RELEASE. A VTOC is also reserved when DFDSS updates the RACF protect bit.

The SHARE option applies to non-VSAM data sets only. (VSAM data sets have their own enqueue scheme. See Figure 60 on page 184.)

If SHARE is not specified, DFDSS tries to provide the highest level of data integrity by enqueueing in the exclusive mode. If the enqueue fails, the command is put in a wait state for X seconds, where X is specified in WAIT(X,-), and the enqueue is retried. If the wait-enqueue sequence fails after Y retries, where Y is specified in WAIT(-,Y), processing will terminate for the data set.

If SHARE is specified, an enqueue for share is tried. If it fails, it is retried as in the above paragraph. If the retries all fail, processing will terminate for the data set.

TOL(ENQF) can be specified in addition to the default or the SHARE option. If TOL(ENQF) is specified, DFDSS attempts to get the specified level of enqueue, exclusive or share. If the enqueue fails after the specified or defaulted number of retries, DFDSS processes the data set without an enqueue. You should specify this option 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. 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. TOL(ENQF) cannot apply to those data movements which involve the use of utilities.

Figure 52 shows the data set enqueue options.

TOL(ENQF) is applicable to VSAM and non-VSAM data sets.

	← User Specifications →			
	None	SHARE	TOL(ENQF)	SHARE and TOL(ENQF)
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 52. Data Set Enqueue Options for Non-VSAM Data Sets

## An Example of RESERVE-ENQUEUE Processing

Example: Dump two data sets

```
DUMP INDD(IN1) OUTDD(OUT1) -
DATASET(INCLUDE(MY.DUMP.DATASET1,MY.DUMP.DATASET2)) -
WAIT(5,2) -
SHARE
```

DFDSS processing occurs via the following steps:

1. RESERVE on SYSVTOC
  - If RESERVE fails, retry two times at 5-second intervals.
  - If RESERVE is successful, continue.
2. If either of the above was a VSAM data set, the VVDS would also be serialized. If VVDS serialization fails, none of the data sets are dumped.
3. Issue a shared enqueue on data set name. If the shared enqueue fails, retry two times at 5-second intervals.
  - If the shared enqueue is successful:
    - Add the data set name to the list to be dumped.
    - Loop back to the beginning of this step until an enqueue is tried for all data sets.

- If shared enqueue is unsuccessful:
    - Issue message ADR412E.
    - Set CC = 4.
    - Loop back to the beginning of this step until an enqueue is tried for all data sets.
4. After an enqueue is tried for all specified data sets:
- Dump the VVDS (if an integrated catalog facility data set was selected) and the VTOC and dequeue the VTOC.
  - If VVDS was serialized, it is dequeued.
  - Dump each data set that was successfully enqueued.
  - Dequeue the enqueued data sets.
  - Issue message ADR378I.

Keep in mind that, if TOL(ENQF) was specified (unless overridden by the user enqueue exit routine), the VTOC is not dequeued until all the data tracks for all the data sets are dumped.

## 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, JES3, and the interface between dynamic allocation and JES3 enabled. SVC 99 is used to serialize data sets. Execution time increases because overhead is involved in dynamic allocation and serialization across multiple processors.

Using DYNALLOC to serialize data sets (as opposed to ENQ) will increase job run time. Using INDYNAM instead of DD statements to allocate DASD volumes will not appreciably increase run time and permits easier coding of JCL and command input.

## How DFDSS Checks Authorization for You

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 RACF or password. RACF checking takes precedence over password checking. If System Authorization Facility is installed, it is also supported.

DFDSS checks Resource Access Control Facility (RACF) authorization at the volume and data set levels. If RACF Version 1 Release 5 (RACF 1.5) 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 terminated 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 user authorization exit routine can direct DFDSS to bypass checking for all data sets on a volume, or to bypass it on individual data sets. See “How to Write a User Authorization Exit Routine” on page 232 for details.

## **DASD 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 user authorization exit routine must check your access authority (see “How to Write a User Authorization Exit Routine” on page 232 for details). The user 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.

Figure 53 on page 170 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 via the JSCBPASS bit indicator or your authorization exit routine.

Data sets may 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 under the 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, etc. The access authority required is dependent on the function and type of operation being performed:

### 1. Non-VSAM RACF-protected data sets

If RACF 1.5 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 1.5 and SAF.

If RACF 1.4 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 53 on page 170.

### 2. VSAM data sets.

Authorization for accessing VSAM data sets (except VVDS) is checked via 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 53.

3. 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.
4. Non-VSAM password-protected data sets  
 If the DASD volume is the target volume for the 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 via the PASSWORD parameter of the applicable command.
5. 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 operations attribute:
  - 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.
6. When RACF Version 1 Release 6 or a later release is installed, RACF DASDVOL authorization checking is honored.

Function	Type of Access on Volume or Data Set	Full	Tracks	Dataset
COMPRESS	Input	N/A	N/A	Update
COPY with DELETE	Input Output	N/A N/A	N/A N/A	Update Update
COPY without DELETE	Input Output	Read Alter	Read Alter	Read Read
COPYDUMP	N/A	N/A	N/A	N/A
DEFRAG <sup>1</sup>	Input	Read	Read	Read
DEFRAG <sup>2</sup>	Input	Update	Update	Update
DUMP with DELETE	Input	N/A	N/A	Update
DUMP without DELETE	Input	Update	Update	Update
PRINT	Input	N/A	Read	Read

Figure 53 (Part 1 of 2). Access Authority Requirement—RACF-Protected Data Sets and DAS DVOL

Function	Type of Access on Volume or Data Set	Full	Tracks	Dataset
RELEASE	Input	N/A	N/A	Update
RESTORE	Output	Alter	Alter	Update

Figure 53 (Part 2 of 2). Access Authority Requirement—RACF-Protected Data Sets and DAS DVOL

Notes to Figure 53:

- <sup>1</sup> RACF-protected data set access requirement
- <sup>2</sup> DASDVOL access requirement

### RACF Logging During Authorization Checking

If you specify the RACFLOG=YES parameter in your EXEC statement, or if the user 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 via 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.

## Using RACF for ISMF Authorization

Using the program control feature of RACF Version 1 Release 7, 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 appropriate ISMF data and function are available to users when they need it.

### ISMF Function You May Want to Protect

Because program control allows you to determine **who** can execute **which** ISMF function, you can use it to set up an authorization scheme that applies to both individual users and user groups. The function you can protect falls into four general categories:

1. The entire ISMF component
2. ISMF applications:

- Profile
  - Data Set
  - Volume
3. Line operators
  4. Commands

With program control, you can set up authorization levels for each category, even varying the level within a category to suit the needs of your installation. Before you set up an authorization structure, you may want to consider the following:

- Do you want all users in your installation 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 that you want to limit access to?

## Setting Up the Authorization Structure

RACF program control checks authorization before allowing access to ISMF function. Protection for each function is based on the authorization level of the load module the function is a part of. A user is allowed to execute an ISMF function (the RESTORE list command, for example) when one of the following is true:

- The individual 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, DGTLLIB. They are listed in Figure 54, Figure 55, and Figure 56. Figure 54 lists the module names for DFDSS/ISMF applications. The module names are found in the DGTSMMMD1 member of the panel library. Figure 55 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 56 lists the names for commands. These names are in the DGTTC2D2 member of the load library.

Application	Module Name
Profile	DGTFPF00
Data Set	DGTFDS00
Volume	DGTFVA00

Figure 54. Module Names for DFDSS/ISMF Applications

Line Operator	Data Set Application Module Name	Volume Application Module Name
COMPRESS	DGTFM01	DGTFCS01
COPY	DGTFY01	DGTFV01
DEFRAG	—	DGTFDF01
DUMP	DGTFDP01	DGTFDM01
RELEASE	DGTFRL01	DGTFRV01
RESTORE	DGTFRT01	DGTFRO01

Figure 55. Module Names for DFDSS/ISMF Line Operators

Command	Module Name
COMPRESS	DGTFM01
COPY	DGTFY01
DUMP	DGTFDP01
RELEASE	DGTFRL01
RESTORE	DGTFRT01

Figure 56. Module Names for DFDSS/ISMF Data Set Application Commands

If you want to look at the command table yourself, you need to know the data set names your installation uses for the panel library and the load library. The install process for DFDSS/ISMF puts the panel library in SYS1.DGTPLIB and the load library in SYS1.DGTLLIB. However, your installation's postinstall procedures may involve moving the DFDSS/ISMF libraries. If they have been moved, you can determine the data set name by issuing the TSO LISTALC command and scanning the low-level qualifiers for DGTPLIB and DGTLLIB.

### Protecting DFDSS/ISMF Modules

The two steps to protect DFDSS/ISMF modules are:

1. To define the modules you want to protect, use the RDEFINE command or the ISPF RACF entry panels. When you define the modules to RACF, you 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 will be 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 userids 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 *Resource Access Control Facility (RACF) Security Administrator's Guide*.

## Data Security

DFDSS provides data security by supporting RACF and passwords when these facilities are installed and active on the processor executing DFDSS. The system operator is also queried for authorization in certain instances. For example, 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, accessing the VTOC, VTOC Index, or VVDS). However, installation exits can be used for overriding security checks. DFDSS, including the installation user exit routines, must be stored in an authorized library. The scheduler checks the program properties table, and if password authorization checking is to be bypassed, instructs DFDSS to do so. The intent is to provide automatic use of security features and provide flexibility by giving the installation options to bypass these authorization checks.

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. When RACF Version 1 Release 7 (or later release) and MVS/XA Data Facility Product Version 2 Release 1.0 (or later release) are installed, old extents are also erased for protected data sets during DUMP with DELETE, COPY with DELETE, and RELEASE. This adds time to all the above-mentioned processes.

### Installation Authorization Exit

A user-written routine can allow special privileged users to bypass all or some of the authorization checking. Security processing will increase the time required to perform the 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 greatly increased. Note that, when you invoke this exit, you cannot bypass authorization checking initiated by utilities (such as IEBCOPY) that are invoked by DFDSS.

## Example of Security Processing

Function: Dump data sets by filter argument; assume RACF and password-protected data sets are on the volume.

```
DUMP DATASET(INCL(MY.**))
```

The security processing that occurs is:

- Authorization checking is not done if any of the following are true.
  1. IEFSDPPT program property table indicates DFDSS does not require authorization processing.
  2. Installation exit indicates to bypass authorization checking.
  3. User has RACF DASDVOL update access.
- The following occurs if authorization checking is performed:
  1. Each data set that is RACF-protected will be checked for access authority. If you do not have authority or if the RACF profile is missing for a particular data set, it will not be dumped and message ADR402 will be 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 via both the password and RACF protection mechanisms and RACF is 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. See "Read/Write Access Serialization Scheme" on page 183 for further details.

## Data Set Passwords—Abend Dumps

If you enter data set passwords in the command data set and your job abends, 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 via 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.

## DUMP Operation—Dumping to a DASD Sequential Data Set

If you are dumping to a DASD sequential data set and this data set is on the same volume as the one from which you are dumping, the data set extents of the DASD sequential data set should be large enough to accommodate the data being dumped. If a need arises for a secondary allocation of the data set, the job may result in an S138 abend. This is because the DADSM EXTEND function enqueues on the volume's SYSVTOC while DFDSS has held the ENQ on the SYSVTOC.



For a data set DUMP operation, DFDSS enqueues the SYSVTOC until the VTOC is completely processed, unless TOL(ENQF) is used.

For a full or tracks DUMP operation, DFDSS enqueues the SYSVTOC until the entire volume is processed, unless overridden by an installation-supplied user enqueue exit routine.

## DUMP Operation—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
7. Volume trailer record: identifies the end of the data for the DASD volume

Item 6 is repeated for each data set.

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

1. Tape header
2. List of potential data sets
3. Data set header
4. Volume record (one for each volume or data set)
5. Data track records (one for each track of the data set)

Items 3, 4, and 5 are repeated for each data set.

See "ADRTAPB Data Area" on page 302 for the dump tape header record formats.

## DUMP Operation—OPTIMIZE Parameter

If OPTIMIZE(2), (3), or (4) is specified and the source DASD volume has some paths that do and others that do not support the Read Multiple Count, Key, and Data channel command, it is possible that the operator will receive message IEA000I (command reject on X'5E') from the control unit that does not support the channel command. This will not cause any loss of data. It can cause performance degradation. It also causes unnecessary entries to LOGREC. To avoid this, either use OPTIMIZE(1) or vary the alternate path off the volume that does not support the Read Multiple Count, Key, and Data command. You can also avoid this by upgrading your control units to support X'5E' channel commands.

## Dump/Restore Techniques and Considerations

Dump techniques can be changed from only full volume operations to a combination of both full volume and data set dumps. The filter "BY (DSCHA,EQ,1)" will allow you to dump only the data sets that were changed since the last dump was taken. This data-set-changed (DSCHA) 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 user exit IDATMSTP with one that sets RC = 4.

A possible implementation of a dump strategy could be:

1. Dump a full volume at a given interval—perhaps once a month. This dump should be taken using the RESET option to reset the data-set-changed flag. This results in a full volume dump and all DSCHA bits on the volume are reset.

Full dump, two volumes at once (in parallel; most effective if tapes are on separate channels):

```
PARALLEL
DUMP INDYNAM(111111) OUTDD(TAPE1) RESET OPTIMIZE(3)
DUMP INDYNAM(222222) OUTDD(TAPE2) RESET OPTIMIZE(3)
```

2. Dump only changed data sets at a shorter interval—perhaps every day.

Changed Data Set dump, two volumes

```
DUMP INDYNAM((111111),(222222)) OUTDD(TAPE3) RESET -
OPTIMIZE(3) DATASET(INCLUDE(**) -
BY((DSCHA,EQ,1)))
```

3. By using naming conventions for your data sets, you can set up a dumping scheme more in keeping with the relative importance of the data sets. For example, include the characters "CRIT" anywhere in the DSNAM for your "critical" data sets. Now DSNAM filtering can be used in your dumping scheme. Other naming conventions can be used, such as department number, charge number, user initials, project code, and so forth.

Changed Critical Data Set dump, two volumes

```
DUMP INDYNAM((111111),(222222)) OUTDD(TAPE4) RESET -
OPTIMIZE(3) DATASET(INCLUDE(**.*CRIT*.**) -
BY(DSCHA,EQ,1))
```

The SYSOUT will contain 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 should misplace the previous dump SYSOUT you can discover what data sets are on a particular dump tape by executing a RESTORE operation with the "NORUN" parameter and the command:

```
RESTORE DATASET(INCLUDE(**)) INDD(ddname1) OUTDD(ddname2)
```

This will print 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.

## 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, SVC 99 is used to serialize the data sets. The time taken to serialize, using SVC 99, is much greater than that taken 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 via ENQ is the same as via SVC 99; ENQ takes much less time to serialize data sets.

When you use DYNALLOC, the selected data sets are allocated to DFDSS (similar to a data set being allocated through the JCL for the duration of the function). If some data sets are allocated to the system, they can only be processed by DFDSS with DYNALLOC if SHARE is also specified. Typically, these are data sets in the LINKLIST.

### RESERVE-ENQUE-DYNALLOC Matrix

Function	Data Set Type	Volume Level Serialization		Data Set Level Serialization (ENQ or DYNALLOC)
		VTOC	VVDS	
COMPRESS	Non-VSAM	Yes	N/A	DSName
Data Set COPY	Non-VSAM VSAM	Yes Yes	N/A Yes	DSName Component and Cluster names
DEFRAG	Non-VSAM VSAM	Yes Yes	N/A Yes	DSName Component and Cluster Names
Full DUMP	Non-VSAM VSAM	Yes Yes	N/A Yes	N/A N/A
Data Set DUMP	Non-VSAM VSAM	Yes Yes	N/A Yes	DSName Component and Cluster Names
Data Set PRINT	Non-VSAM VSAM	N/A Yes	N/A Yes	DSName Component name

Figure 57 (Part 1 of 2). Resource Serialization

Function	Data Set Type	Volume Level Serialization VTOC VVDS	Data Set Level Serialization (ENQ or DYNALLOC)
Tracks PRINT		Yes N/A	N/A
RELEASE	Non-VSAM	Yes N/A	DSName
Full RESTORE	Non-VSAM VSAM	Yes N/A Yes Yes	N/A N/A
Data Set RESTORE	Non-VSAM VSAM	N/A N/A N/A Yes	DSName Component and Cluster Names

Figure 57 (Part 2 of 2). Resource Serialization

Notes to Figure 57:

- <sup>1</sup> VSAM data sets must be cataloged in an integrated catalog facility catalog.
- <sup>2</sup> N/A means not applicable.

### Full or Tracks DUMP/RESTORE Integrity Considerations

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

#### DUMP, Full Volume or Tracks

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 parameter 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.

#### Restore, Full Volume or Tracks

1. Specify 'VOL=(PRIVATE,SER=xxxxxx)'

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

## Shared DASD Considerations

Shared DASD presents volume and data set serialization problems not encountered in nonshared DASD environments. Care should be taken when doing data set operations if programs operating in another processor may be accessing the data sets at the same time.

A data set may 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.

DEFRAG in a shared environment presents a more complex problem. The defrag operation moves data set extents while other programs in other processors could be using data in those extents. On a JES3 system, the problem may be avoided by using the DYNALLOC keyword and ensuring that the dynamic allocation function of JES3 is not disabled for any of the data sets that are on the DASD. Results may be unpredictable on JES2 systems. It is highly recommended that the systems be quiesced or the volume be offline to all the processors, except the one executing DEFRAG. The same considerations for shared DASD volumes also apply to the data set COPY operation.

## Filtering Criteria and/or Passwords—Passing via a Data Set

If filtering criteria and/or passwords are passed in a data set, the data set should have the following characteristics: RECFM of F or FB, BLKSIZE of a multiple of 80, and LRECL of 80.

## Lockouts during DFDSS Runs

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 lockout 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.

## Storage Requirements—DFDSS

The following storage estimates can be used as a starting point for determining minimum region sizes in which DFDSS can execute. Figure 58 on page 182 shows the minimum storage requirements, *in bytes*, to execute each DFDSS operation. Figure 59 on page 183 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 may vary depending on your operating system configuration, device characteristics, and data set characteristics. The storage requirement estimates shown for COPY, DUMP, and RESTORE are for a full COPY, DUMP, or RESTORE only; they may vary for a data set operation.

If you use buffers above 16 megabytes virtual storage, the buffer size is allocated independently of the region size.

When specifying a PARALLEL operation, the total storage requirement is the sum of the storage requirements of all functions to be executed in parallel.

## Multivolume Data Sets—Processing

For more information on processing multivolume data sets, see the description of the ALLMULTI parameter of the COPY or DUMP command.

During a *physical* data set DUMP, a data set, if selected, is dumped from all the volumes that are passed. The output dumped data will contain a logical file for each selected volume. During a *physical* data set RESTORE, if the data set was already allocated on the output volume, then the RESTORE will take place from the particular logical volume that contains the data set with the same volume sequence number as that on the output volume. However, if the data set is not preallocated on the output volume, the data set will be restored from the first logical volume that has that data set, provided it matches the selection criteria. In this case, if you wanted to restore from a particular source volume, you would specify the volume serial number in the LOGICALVOLUME parameter.

During a *logical* data set DUMP, if LOGINDDNAME or LOGINDYNAM is not specified, data set extents are dumped in their logical sequence. The catalog(s) 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.

During a multivolume data set COPY or RESTORE:

- For a VSAM key range data set that has as many volumes as there are key ranges, DFDSS preserves the volume spread if enough target volumes are specified.
- For a VSAM key sequence data set whose index and data components are on different source volumes, DFDSS preserves the volume spread if enough target volumes are specified.
- For a non-VSAM or a VSAM data set whose component spans multiple volumes, DFDSS merges the data set to one volume, space permitting.

DFDSS Command	With I/O Buffers Below 16 Mb Virtual	With I/O Buffers Above 16 Mb Virtual	
		Storage Below 16 Mb Virtual	Storage above 16 Mb Virtual
COMPRESS:	965K + (2 * trksize of largest device) + (1152k *number of additional volumes, up to four)	N/A	N/A
COPY (FULL):	820K + (trksize * 5)	820K	(trksize * 5)
COPYDUMP:	810K + (buffersize * 5)	810K	(buffersize * 5)
DEFRAG:	845K + (trksize * 5) + (16K if VSAM present)	845K	(trksize * 5)
DUMP (FULL):			
OPT(1)	820K + (buffersize * 7)	820K	(buffersize * 7)
OPT(2)	820K + (buffersize * 6)	820K	(buffersize * 6)
OPT(3)	820K + (buffersize * 15)	820K	(buffersize * 15)
OPT(4)	820K + (buffersize * (3 * trk/cyl))	820K	(buffersize * (3 * trks/cyl))
PRINT:	810K + (3 * trksize)	810K	(3 * trksize)
RELEASE:	810K + trksize	810K	trksize
RESTORE (FULL):	810K + (buffersize * 6)	810K	(buffer size * 6)
Legend			
K	1024 bytes		
buffersize	The maximum of trksize and default maximum blocksize (32K).		
trksize	The size, in bytes, of a track on your DASD volume.		
trks/cyl	The number of tracks per cylinder on your DASD volume.		

Figure 58. Minimum Storage Requirements for Each DFDSS Operation

- When INDDNAME or LOGINDDNAME is specified, DFDSS requires separate DD statements for each input volume of a multivolume data set.

When DFDSS invokes IEHMOVE to copy multivolume non-VSAM data sets, IEHMOVE requires that the first DD statement in the job stream 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) ...
```

#### Restore of PAM Data Sets to an Unlike Device:

$$800K + (((\text{tracksize} + 64) * 5) + 8K)$$

where: 800K = Load module size  
tracksize = Track size of largest device  
64 = Header record  
5 = Number of buffers used for Restore  
8K = Size of PDS member list + size of PDS TTR translate list

#### Restore of VSAM Data Sets to an Unlike Device:

$$800K + (((\text{tracksize} + 64) * 5) + 8K) + (2 * \text{maximum record size}) + (3 * \text{bufferspace})$$

where: 800K = Load module size  
tracksize = Track size of largest device  
64 = Header record  
5 = Number of buffers used for Restore  
8K = Size of PDS member list + size of PDS TTR translate list  
3 = Minimum number of buffers needed  
bufferspace = Specified in DEFINE command when data set was allocated

Figure 59. Minimum Storage Requirements for a Restore to an Unlike Device

## PARM Statement—“SIZE=”

Specifying “SIZE=” in the PARM statement is redundant if the storage you want to use can as easily be specified in the REGION keyword. The “SIZE=” parameter is intended for use if DFDSS is attached by another problem program and you want to limit the virtual storage acquired by DFDSS, including storage that is above 16 megabytes virtual storage.

## Read/Write Access Serialization Scheme

Figure 60 on page 184 shows the serialization scheme used by all operations, except COPYDUMP, for read/write access.

## RESTORE Operation—Full Volume Restore and 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.



	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		SYSTEM	
Input access			S		
Output access			E		
If DYNALLOC is used:					
For non-VSAM data sets, instead of ENQ, the data set is allocated dynamically 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.					
<b>Note: The minor name consists of:</b>					
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 60. Read/Write Access Serialization Scheme

When the volume serial number is changed via 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 RACF-protected DASD volume.

## RESTORE Operation—Full Volume or Data Set Restore and RACF Generic Profiles

The protection status of data sets that are restored via a full or 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

## RESTORE Operation—Logical Data Set RESTORE and RACF-Protected Data Sets

See Figure 61. See also Figure 27 on page 60.

	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 61. RACF-Protected Data Sets in a Logical Data Set RESTORE

Notes to Figure 61:

1. Data sets protected by both generic and discrete profiles at DUMP time are considered to be discretely protected.
2. When DFDSS defines discrete profiles, if MENTITY is not specified, default discrete profiles are defined, which gives ALTER authority to the userid specified on the JOB statement.

## **RESTORE Operation—Renaming RACF-Protected Data Sets During Data Set Restore**

For logical data set processing, or physical processing of non-VSAM data sets, 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, for the RACF protection of the renamed data set to work, this new group of first level qualifiers of newname must have been defined to RACF. Multivolume, non-VSAM data sets are renamed by DFDSS. 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.

## **RESTORE Operation—Restore Unmovable (ISAM and Absolute Track) Data Sets**

On a physical data set RESTORE, ISAM and absolute track data sets and those that are allocated with the unmovable attribute must be preallocated, and the location and size of each data set extent on the source and destination volumes must match exactly.

If a data set is not preallocated, you can do 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.

## **RESTORE Operation—Tracks RESTORE**

Track 0 and the VTOC are always dumped on a tracks DUMP or data set DUMP even if the requested range falls outside track 0 and the VTOC. For this reason, you are required to specify the range of tracks you need to restore. You can do a tracks RESTORE from a data set DUMP tape from the *first* logical volume only. If you need to restore a track from a volume other than the first logical source volume, first execute a COPYDUMP operation to copy only that logical source volume to a separate data set, then execute the “tracks RESTORE” from that data set. If you restore the VTOC tracks from a data set DUMP, the format-1 DSCB is cleared for the data sets that were not dumped and that had null extents.

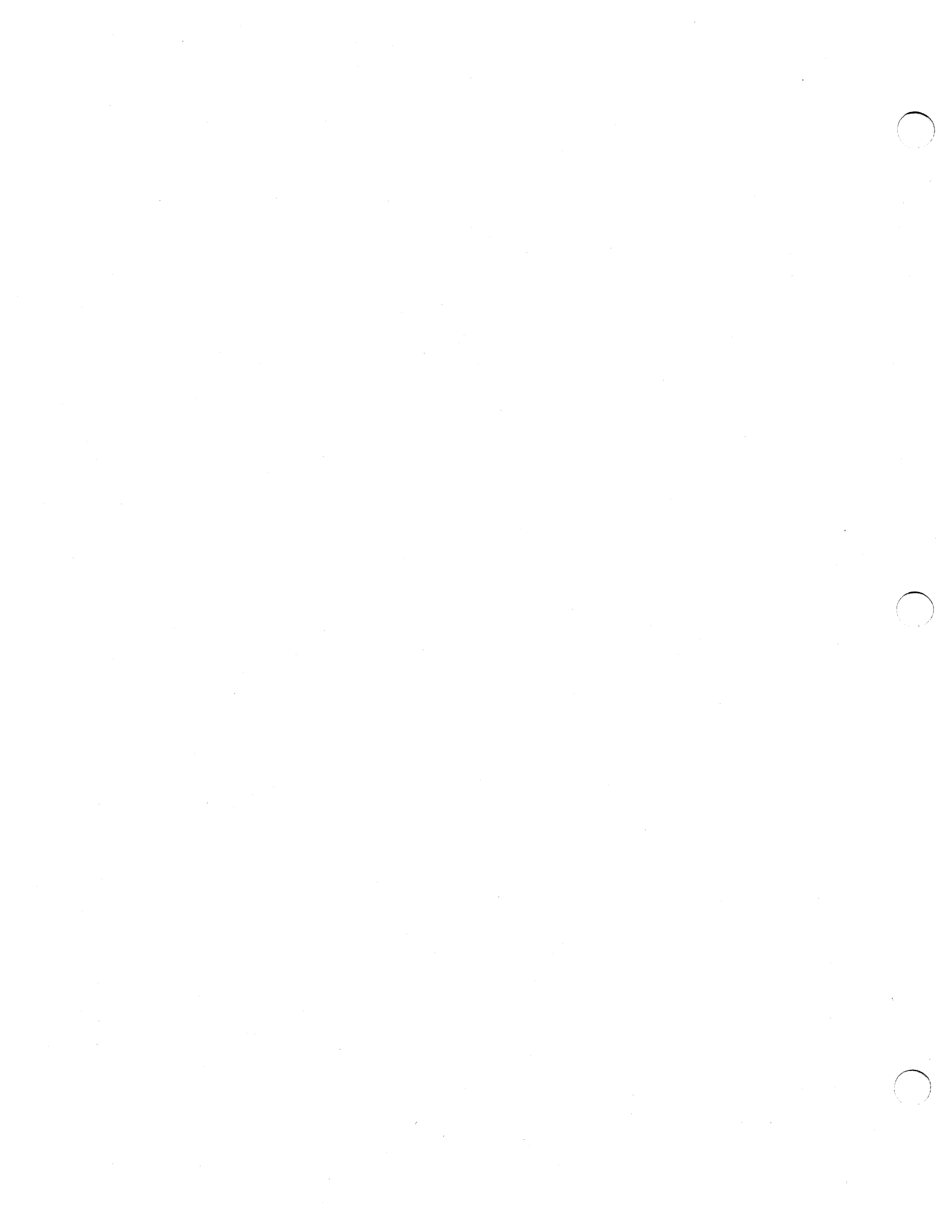
## **SYSPRINT DD Statement**

The following statements are true if you specify a SYSPRINT DD statement (or alternate), a PRINT operation output DD statement (if other than SYSPRINT), or a DCB parameter (not allowed to default):

- If LRECL is specified, it should be equal to or greater than 84 and equal to or less than 137. If LRECL is greater than 137, it is adjusted to 137.
- If BLKSIZE is specified, it must be between 88 and 141, inclusive.
- If RECFM is specified, it must be VA or VBA.

If the LRECL parameter of such a DD statement is less than 84, or BLKSIZE is less than 88, or RECFM is not VA or VBA, the job terminates. For a PRINT operation that uses a print file other than SYSPRINT, the PRINT operation terminates.

If the SYSPRINT data set resides on the volume that is being processed by DFDSS and a secondary allocation is needed for it, the job may result in an S138 abend. This is because the DADSM EXTEND function enqueues on the volume's SYSVTOC while DFDSS holds the enqueue on that SYSVTOC.



## Chapter 7. How You Invoke DFDSS via an Application Program

You can invoke DFDSS from an application program by using the “application interface.” This allows you, for example, to gather statistical or auditing information. This interface allows you to specify control variables.

The application interface allows you to:

- Fully utilize the invocation capabilities of DFDSS if the ATTACH, LINK or CALL system macro is specified
- Optionally, specify a list of parameters to be used by DFDSS during the processing caused by that invocation
- Optionally, interact with DFDSS during user installation options processing after the user 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 (non-supervisor state); any program invoking DFDSS must also be authorized and in non-supervisor state.

### Invocation

DFDSS can be invoked by any of the system macros shown in Figure 62.

```
ATTACH EP=ADDRSSU,PARAM=(OPTPTR,DDPTR,PAGEPTR,UIMPTR,UAPTR),VL=1
```

or

```
LINK EP=ADDRSSU,PARAM=(OPTPTR,DDPTR,PAGEPTR,UIMPTR,UAPTR),VL=1
```

or

```
CALL (15),(OPTPTR,DDPTR,PAGEPTR,UIMPTR,UAPTR),VL
```

Figure 62. Invoking DFDSS via Its Application Interface

A description of the parameters in Figure 62 follows:

## **ADDRSSU**

is the name of the DFDSS module main entry point. Only the lists pointed to by register 1 upon entry to DFDSS are used. This means that if you had JCL EXEC PARM= values specified in the calling program, DFDSS will ignore these parameters.

## **OPTPTR**

is the name of the OPTION list and is similar to the OPTIONADDR description used in *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 (not on a fullword boundary) field 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 may omit them from the list.

## **DDPTR**

is the name of the DDNAME list and is similar to the DDNAMEADDR description used in *Utilities*. This provides a way 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 normally 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 may omit them from the list.

## PAGEPTR

is the name of the Page Number list and is similar to the HDINGADDR description used in *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 will reset 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 will not be changed by DFDSS.

If you do not want to specify subsequent parameters, you may omit them from the list.

## UIMPTR

is the name of the user interaction module (UIM) list. There is no comparable parameter referenced in *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 operations. The UIM list is of variable length, consisting of a halfword field following by a single 4-byte address or 8-character left-justified (right padded 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 the address of the UIM, or the name of the UIM, in the field following the UIM list length field, and fill that length field with the proper binary value of 4 or 8 respectively. See "System Programming Information" on page 193 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 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.

### Calling Block Structure

The parameter structure outlined above can be viewed in block form as shown in Figure 63.

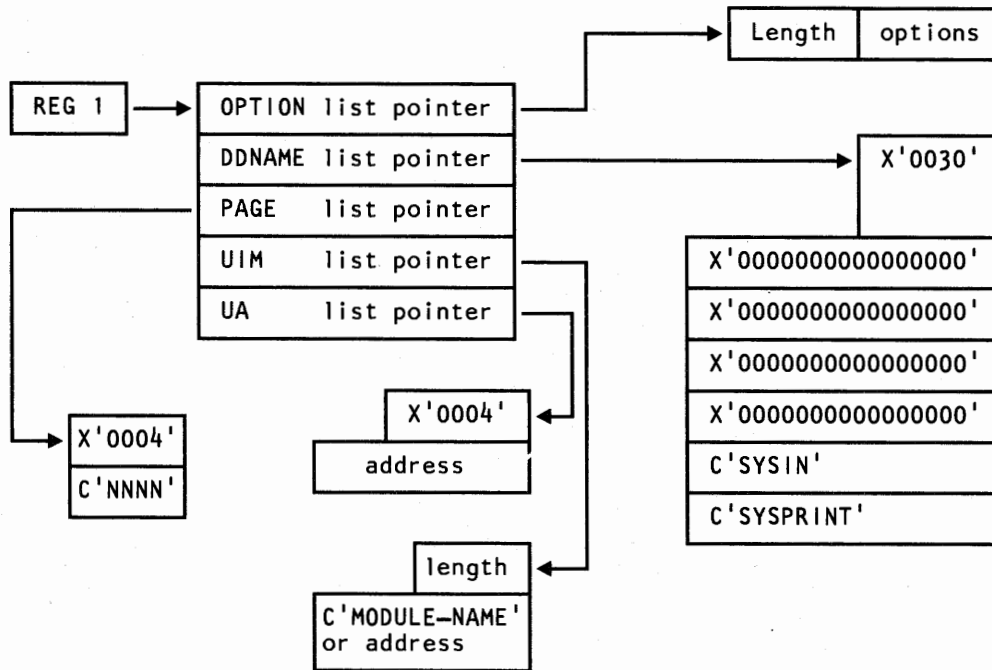


Figure 63. DFDSS Application Interface Structure

## User Interactions

User interactions can only take place when DFDSS has been invoked via the Application Interface and the name of the user interaction module (UIM) exit routine is supplied to DFDSS. As shown in "Invocation" on page 189, a pointer to the UIM exit is passed in the parameter list (UIMPTR). Generally, user interactions may be used for record management or for gathering statistics on DFDSS operations. See Figure 73 on page 203 for a list of the various UIM exits; examples of how to code UIM exits follow the figure.

When a UIM exit routine is specified, DFDSS will process normally, then at each point in the process (hereafter referred to as DFDSS exit points), the UIM exit routine will conditionally be called to allow some types of user operations.

If a DFDSS subtask abends for any reason, it will be unable to make any further calls to the UIM, including the "function termination" call.

### System Programming Information

There are some bit definitions in the user installation options control block (ADRUFO) that permit DFDSS to communicate with the user installation options exit routine. These bits and their meanings can be derived from the formal declaration of the ADRUFO control block in "ADRUFO Data Area" on page 239. However, it is discussed briefly here; examples are given in "Examples of Invoking DFDSS" on page 197. When the user 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 ADRUFO. If there is not to be a SYSIN and/or SYSPRINT data set allocated or all SYSIN/SYSPRINT is to be handled in-storage, the UFSYSIN and UFSYSR specifies this to DFDSS. Three additional bits are referenced—UFAIINV, UFUIMAL and UFUIMCH—which determine allowances within the UIM interaction.

There are two additional bits 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.

When DFDSS has been invoked via 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 will point 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 219 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 is mapped by the macro ADREID0. See Figure 65 on page 194 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 (within the limits allowed to the exit).

**Application Interface Blocks:** The parameter structure described in Figure 63 on page 192 can be viewed in block form as shown in Figure 64.

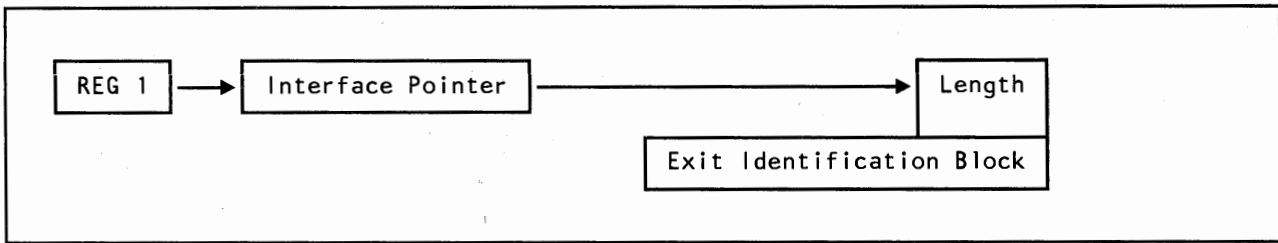


Figure 64. DFDSS Exit Interface Structure

**Exit Identification Block:** The exit identification block is shown in block form in Figure 65.

Control Block Eye-catcher	(supplied by DFDSS)
TASK-ID	(supplied by DFDSS)
User 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 65. 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 219 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 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.

- **User 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 user installation options.
  - Insert data prior to current record.
  - Replace the current data record with an exit supplied one.
  - Delete the current data record.
  - Modify the current data record.
  - Disconnect the exit from further interaction.
  - Recognize when a disallowed option has been attempted.
- **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 type of record being presented to the exit. This number both identifies the type of record being presented and what DFDSS is using this record for.

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 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 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 user installation options record (ADRUFO) 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 when using this code, because multiple record types will 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 issuing the WTOR itself. Note that the UIM must set the original record length to the proper value.
- 32 DFDSS terminates the current functional task and issues message ADR356E. This return code is supported only for full volume DUMP and RESTORE.

- **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 and 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.
- **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.
- **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.

The location of the Original Record, above or below the 16-megabyte (Mb) virtual storage line, is controlled via the User 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 will present 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 will 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 Summary:** If the UIM makes any changes to the presented record (assuming it can validly do so), then return code 16 must be returned or the changes are ignored. If the record is to be replaced in total, then return code 4 must be returned or the original record is used. If a record is to be inserted before the current record, then return code 8 must be returned, or the record is ignored. If the current record is to be deleted, then 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, then 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, then 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 may supply an area and store the address to the record in the original record pointer field or, if allowed, may 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.

## Examples of Invoking DFDSS

In order to demonstrate the different ways DFDSS can be invoked, the following scenarios are presented.

### Invoke DFDSS via Job Control Language

Figure 66 on page 198 shows how you invoke DFDSS via JCL (as described in detail in "How You Invoke DFDSS via Job Control Language" on page 17). It shows that a full DASD volume is to be dumped to tape volume(s). If the dump return code is zero, a defragmentation of the volume is to be done. Two tape volumes are specified in the TAPE DD statement in case one volume is not sufficient.

```

//JOB1      JOB    accounting information,REGION=nnnnK
//STEP1     EXEC   PGM=ADRSSU
//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 DEFRAG DDN(DASD)
/*

```

Figure 66. Invoke DFDSS via JCL

### Invoke DFDSS via an Application Program

The example in Figure 67 is the same as that of Figure 66, except that a user program, MYJOB, is executed. DFDSS is LINKed somewhere in MYJOB to perform the DUMP, and, conditionally, the DEFRAG functions. Notice that the only JCL difference is the name of the program to be executed, MYJOB.

```

//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 DEFR DDN(DASD)
/*

```

Figure 67. Invoke DFDSS from a User Program

Figure 67 does not show how the invocation of DFDSS was brought about, but does show that the user program, MYJOB, was executed. Somewhere within MYJOB it is desired to execute DFDSS to perform the functions specified in the SYSIN data set. At this point, Figure 68 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 LINK parameters.

```

      .
      .
      .
LINK EP=ADRDUSSU,PARAM=(OPTPTR),VL=1
      .
      .
      .
OPTPTR  CNOP 2,4
        DC   H(0)

```

Figure 68. Invoke DFDSS via a LINK

### Invoke DFDSS, Supplying Options

The example in Figure 69 is the same as that of Figure 68, except that EXEC PARM values are to be passed during the invocation.

Figure 67 shows the JCL used to invoke DFDSS. Again, somewhere within MYJOB it is desired to execute DFDSS to perform the functions specified in the SYSIN data set. Figure 69 shows the necessary code used at this point to LINK to DFDSS and specify the OPTION list. The standard SYSIN and SYSPRINT data set names are to be used and are therefore not specified in the LINK parameters. This example uses the option SIZE= to specify that 20480 bytes of storage are to be used by DFDSS and PAGENO= to specify the starting page number 31 to be used for the SYSPRINT data set.

```

      .
      .
      .
LINK EP=ADRDUSSU,PARAM=(OPTPTR),VL=1
      .
      .
      .
OPTPTR  CNOP 2,4
        DC   AL2(OPTLEN)
OPTIONS DC   C'SIZE=20K,PAGENO=31'
OPTLEN  EQU  *-OPTIONS

```

Figure 69. Invoke DFDSS with Options

### Invoke DFDSS, with ddname Overrides

The example in Figure 70 on page 200 is the same as that of Figure 69, except that the MYDATA ddname is to be used as the SYSIN data set during the invocation. Figure 70 shows the JCL used to invoke DFDSS. This JCL is the same as that of Figure 67, except that the SYSIN data set is now named MYDATA. This allows you to use the standard SYSIN DD named data set for your own uses.



```

//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)
//MYDATA   DD     *
          DUMP  INDD(DASD) OUTDD(TAPE)
          IF LASTCC = 0 -
            THEN DEFrag DDN(DASD)
/*

```

Figure 70. Alternate SYSIN for DFDSS

Again, somewhere within MYJOB it is desired to execute DFDSS to perform the functions specified in the MYDATA data set. Figure 71 shows the necessary code used at this point to LINK to DFDSS and specifies the OPTION list and ddname overrides. This example uses the options as in the previous example, along with the input data set named MYDATA and the system output to go to the normal SYSPRINT data set.

```

          .
          .
          .
          LINK EP=ADDRSSU,PARAM=(OPTPTR,DDNPTR),VL=1
          .
          .
          .
          CNOP 2,4
OPTPTR   DC    AL2(OPTLEN)
OPTIONS  DC    C'SIZE=20K,PAGENO=31'
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'SYSPRINT'
DDNLEN   EQU   *-DDNAMES

```

Figure 71. Override SYSIN ddname

## Invoke DFDSS, with Starting Page Number

The example in Figure 72 on page 201 is the same as that of Figure 71, except that the system output page number is to begin with a dynamically determined value.

Figure 70 shows the JCL used to invoke DFDSS. Again, somewhere within MYJOB it is desired to execute DFDSS to perform the functions specified in the MYDATA data set. Figure 72 shows the necessary code used at this point to LINK to DFDSS and specifies the OPTION list, ddname overrides, and a dynamically-defined starting page number for the SYSPRINT data set. This example only uses the "SIZE=" option of the previous example, along with the input data set named MYDATA and the system output to go to the normal SYSPRINT data set. The four instructions prior to the LINK are used to initialize PAGENO with the page number next to be used on the SYSPRINT data set.

```

      .
      .
      L      5,NXTNUM
      CVD    5,WORK
      OI     WORK+7,X'OF'
      UNPK   PAGENO(4),WORK+5(3)
      LINK   EP=ADRDSU,PARAM=(OPTPTR,DDNPTR,PAGEPTR),VL=1
      .
      .
      CNOP   0,8
      WORK   DC      D'0'           UNPACK WORK AREA
      NXTNUM DC      F'0'           NEXT SYSPRINT PAGE NUMBER
      CNOP   2,4
      OPTPTR DC      AL2(OPTLEN)
      OPTIONS DC     C'SIZE=20K'
      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'SYSPRINT'
      DDNLEN EQU     *-DDNAMES
      CNOP   2,4
      PAGEPTR DC     AL2(PAGLEN)
      PAGENO DC     CL4'0000'
      PAGLEN EQU     *-PAGNO
  
```

Figure 72. Dynamic Page Numbering on SYSPRINT

## Sample User Interaction Module (UIM)

Note that the examples shown are written as reentrant code. If you are planning to share a UIM between tasks, you should code the module in reentrant code.

The example in Figure 73 on page 203 demonstrates one possible configuration that you can use when constructing a new UIM. Although the program is written in assembler language here, you are free to use a higher level language as long as the interface is maintained.

Figure 73 shows how the program is basically divided into three parts, the vector intercept processor, the vector table, and the processor routines. This program is

just an example and does not perform any useful function because it only disconnects exits by using return code 20 or allows normal processing of the data records by using a return code zero (0). The following examples expand upon this sample to show how the various interface options can be used.

### **Invoke DFDSS, Supplying a New UIM**

The example in Figure 74 is the same as a previous one, except that a new user interactive module is to be used on this invocation. Figure 70 shows the JCL used to invoke DFDSS. Again, somewhere within MYJOB it is desired to execute DFDSS to perform the functions specified in the MYDATA data set. Figure 74 shows the necessary code used at this point to LINK to DFDSS and specifies the OPTION list, ddname overrides, a dynamically defined starting page number for the SYSPRINT data set, and the name of the new user interaction module, NEWUIM. This module has been defined in Figure 73, and has been assembled and link-edited into the STEPLIB data set MY.LINKLIB prior to the execution of this JCL. This example only uses the "SIZE=" option of the previous example along with the input data set named MYDATA and the system output to go to the normal SYSPRINT data set.

NEWUIM	CSECT		
	USING	*,15	INITIAL ADDRESSABILITY
	STM	14,12,12(13)	SAVE REGISTERS
	BALR	12,0	NEW BASE ADDRESSABILITY
	USING	*,12	
	DROP	15	
	LR	15,13	SAVE CALLER SAVE AREA POINTER
	LA	13,SAVEAREA	SET MY SAVEAREA POINTER
	ST	15,SAVEAREA+4	BACKWARD CHAIN
	ST	13,SAVEAREA+8	FORWARD CHAIN
	USING	ADREIB,2	ADDRESSABILITY TO EIDB
	L	2,0(,1)	ADDRESS OF EIDB
	LH	3,EIOPTION	GET DFDSS PROCESSING OPTION
	SLL	3,2	TIMES FOUR
	LA	15,VECTABLE	POINT TO VECTOR TABLE START
	L	15,0(3,15)	POINT TO PROCESSOR ROUTINE
	BALR	14,15	GO TO PROCESSOR ROUTINE
	L	13,SAVEAREA+4	RESTORE SAVE AREA POINTER
	LM	14,12,12(13)	RESTORE CALLER REGISTERS
	BR	14	RETURN TO DFDSS
STRTUPRC	LH	1,EIRCOO	SET NORMAL PROCESS RETCODE
	STH	1,EIRETCOD	IN EIDB
	BR	14	RETURN TO INTERCEPT PROCESSOR
DISCONRC	LH	1,EIRC20	SET DISCONNECT EXIT RETCODE
	STH	1,EIRETCOD	IN EIDB
	BR	14	RETURN TO INTERCEPT PROCESSOR
TERMRC	LH	1,EIRCOO	SET NORMAL PROCESS RETCODE
	STH	1,EIRETCOD	IN EIDB
	BR	14	RETURN TO INTERCEPT PROCESSOR
SAVEAREA	DC	18F'0'	MY SAVE AREA
VECTABLE	DC	A(STRTUPRC)	FUNCTION STARTUP EXIT
VEC01	DC	A(DISCONRC)	READ SYSIN EXIT
VEC02	DC	A(DISCONRC)	WRITE SYSPRINT EXIT
VEC03	DC	A(DISCONRC)	READ TAPE BLOCK EXIT
VEC04	DC	A(DISCONRC)	READ LOGICAL TAPE EXIT
VEC05	DC	A(DISCONRC)	WRITE LOGICAL TAPE EXIT
VEC06	DC	A(DISCONRC)	WRITE TAPE BLOCK EXIT
VEC07	DC	A(DISCONRC)	READ DASD TRACK EXIT
VEC08	DC	A(DISCONRC)	WRITE DASD TRACK EXIT
VEC09	DC	A(DISCONRC)	READ UTILITY SYSPRINT EXIT
VEC10	DC	A(DISCONRC)	RESERVED
VEC11	DC	A(DISCONRC)	WRITE WTO MESSAGE EXIT
VEC12	DC	A(DISCONRC)	WRITE WTOR MESSAGE EXIT
VEC13	DC	A(DISCONRC)	PRESENT ADRUFO RECORD EXIT
VEC14	DC	A(TERMRC)	FUNCTION TERMINATION EXIT
VEC15	DC	A(DISCONRC)	PRESENT WTOR RESPONSE
VEC16	DC	A(DISCONRC)	OPEN/EOV TAPE VOLUME SECURITY AND VERIFICATION EXIT
VEC17	DC	A(DISCONRC)	OPEN/EOV NON-SPECIFIC TAPE VOLUME MOUNT EXIT
VEC18	DC	A(DISCONRC)	INSERT LOGICAL VSAM RECORD DURING RESTORE EXIT
VEC19	DC	A(DISCONRC)	OUTPUT TAPE I/O ERROR EXIT
	ADREIDO		DEFINE MAPPING FOR EIDB
	END	NEWUIM	

Figure 73. Sample User Interaction Module (UIM)

```

      .
      .
      .
      LINK EP=ADRDSU,PARAM=(OPTPTR,DDNPTR,PAGEPTR,UIMPTR),VL=1
      .
      .
      .
      CNOP 2,4
OPTPTR DC AL2(OPTLEN)
OPTIONS DC C'SIZE=20K'
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'SYSPRINT'
DDNLEN EQU *-DDNAMES
      CNOP 2,4
PAGEPTR DC AL2(PAGLEN)
PAGEPO DC C'31'
PAGELEN EQU *-PAGEPO
      CNOP 2,4
UIMPTR DC AL2(UIMLEN)
UIMNAME DC CL8'NEWUIM'
UIMLEN EQU *-UIMNAME

```

Figure 74. Invoke DFDSS and Use a UIM Module

### Invoke DFDSS Supplying a User Area Pointer

The example in Figure 75 shows the invocation of DFDSS supplying a new user area pointer.

```

      .
      .
      .
      LINK EP=ADRDSU,PARAM=(OPTPTR,DDNPTR,PAGEPTR,UIMPTR,UAPTR),VL=1
      .
      .
      .
OPTPTR ...
DDNPTR ...
PAGEPTR ...
UIMPTR ...
UAPTR DC AL2(UALEN)
UAADDR DC AL4(USERAREA)
UALEN EQU *-UAADDR
USERAREA DC CL80

```

Figure 75. Invoke DFDSS Supplying a User Area Pointer

## Disable UIM Usage

There are four ways to disable usage of the UIM:

1. Execute DFDSS from JCL ( see “Invoke DFDSS via Job Control Language” on page 197).
2. Invoke DFDSS without specifying a UIM (see “Invoke DFDSS via an Application Program” on page 198).
3. Invoke DFDSS with installation exit that disables the UIM (see “Disable the UIM From the User Installation Options Exit”).
4. Invoke DFDSS with the UIM that disables the UIM at the installation exit (see “Disable UIM Usage via the UIM”).

### Disable the UIM From the User Installation Options Exit

The example in Figure 76 is the same as that described in “Invoke DFDSS via an Application Program” on page 198, except that the user installation options exit has been modified to disallow the use of a UIM. Figure 76 shows how the exit has accomplished this, by setting the UFUIMAL bit in the ADRUFO, and passes the return code 4 to indicate the change.

```
ADRUIXIT CSECT
      .
      USING UFO,1
      OI  UFAIFLGS,UFUIMAL  DISALLOW UIM PROCESSING
      .
      LA  15,FOUR           NOTIFY DFDSS OF CHANGES
      RETURN (14,12),RC=(15),T
      .
      FOUR EQU 4           UFO CHANGED - RETURN CODE
```

Figure 76. Disable UIM from ADRUIXIT

### Disable UIM Usage via the UIM

The example in Figure 77 on page 206 is the same as that described in “Invoke DFDSS, Supplying a New UIM” on page 202, except that the UIM is used to disallow its subsequent use. Figure 77 shows how the UIM exit routine has been modified to accomplish this.

NEWUIM	CSECT		
		.	
INSTALRC	L	4,EIRECPTR	POINT TO ADRUFO RECORD
	USING	UFO,4	ADDRESSABILITY TO UFO
	OI	UFAIFLGS,UFUIMAL	DISALLOW UIM PROCESSING
	DROP	4	
	LH	1,EIRC16	SET RECORD MODIFIED RETCODE
	STH	1,EIRETCOD	IN EIDB
	BR	14	RETURN TO INTERCEPT PROCESSOR
		.	
VEC13	DC	A(INSTALRC)	PROCESSOR - PRESENT ADRUFO RECORD EXIT
		.	
		.	
		ADRUFO	
		.	
		.	

Figure 77. Disable UIM from a UIM User Installation Options Exit

## Prevent Command Execution via the UIM

The example in Figure 78 on page 207 is the same as that described in “Invoke DFDSS, Supplying a New UIM” on page 202, except that the UIM is used to disallow a command from being scheduled. It is assumed that the “Write SYSPRINT exit” processor has stored sufficient information to select the correct command to be bypassed (see “Create a Function-Taskid Table via SYSPRINT Data” on page 213). Figure 78 shows how the UIM exit routine has been modified to accomplish this.

NEWUIM	CSECT		
		.	
INSTALRC	L	4,EIRECPTR	POINT TO ADRUFO RECORD
	USING	UFO,4	ADDRESSABILITY TO UFO
	SR	5,5	CLEAR WORK REGISTER
	ICM	5,3,UFFUNCT1	GET FUNCTION BYTES
	LTR	5,5	Q. FUNCTION ENTRY?
	BZ	SETRCOO	A. NO, NORMAL EXIT
	LH	5,TSK2STOP	GET TASKID OF FUNCTION TO STOP
	CL	5,EITSKID	Q. FUNCTION FOUND?
	BNE	SETRCOO	A. NO, NORMAL EXIT
	OI	UFAIFLGS,UFSTOP	INDICATE FUNCTION TO STOP
	DROP	4	
	LH	1,EIRC16	SET RECORD MODIFIED RETCODE
	B	RCEXIT	FINISH EXIT SETUP
SETRCOO	LH	1,EIRCOO	SET NORMAL PROCESS RETCODE
RCEXIT	STH	1,EIRETCOD	IN EIDB
	BR	14	RETURN TO INTERCEPT PROCESSOR
		.	
VEC13	DC	A(INSTALRC)	PROCESSOR - PRESENT ADRUFO RECORD EXIT
		.	
TSK2STOP	DC	H'06'	TASK-ID OF FUNCTION TO STOP
		.	
		.	
		ADRUFO	
		.	
		.	

Figure 78. Disable a Function from a UIM User Installation Options Exit

### Alter an Installation Option via the UIM

The example in Figure 79 on page 208 is the same as that described in "Disable UIM Usage via the UIM" on page 205 except that the UIM is used to modify the default for all functions from "TOL(ENQF)" to "no TOL(ENQF)." Figure 79 shows how the UIM exit routine has been modified to accomplish this.



```

NEWUIM   CSECT
          .
          .
INSTALRC L    4,EIRECPTR      POINT TO ADRUFO RECORD
          USING UFO,4        ADDRESSABILITY TO UFO
          SR    5,5          CLEAR WORK REGISTER
          ICM  5,3,UFFUNCT1  GET FUNCTION BYTES
          LTR  5,5          Q. FUNCTION ENTRY?
          BZ   SETRCOO        A. NO, NORMAL EXIT
          OI   UFO2FLGS,UFO2ENQN INDICATE NO TOLERATE ENQ FAILURE
          DROP 4
          LH   1,EIRC16      SET RECORD MODIFIED RETCODE
          B    RCEXIT        FINISH EXIT SETUP
          SETRCOO LH 1,EIRCOO  SET NORMAL PROCESS RETCODE
          RCEXIT STH 1,EIRETCOD IN EIDB
          BR   14           RETURN TO INTERCEPT PROCESSOR
          .
          .
VEC13    DC   A(INSTALRC)    PROCESSOR - PRESENT ADRUFO RECORD EXIT
          .
          .
          ADRUFO
          .
          .

```

Figure 79. Alter UFO from a UIM Exit

## Supply In-Storage SYSIN Data and Delete SYSPRINT Records

The example in Figure 80 is the same as that described in "Disable UIM Usage via the UIM" on page 205, except that the UIM is used to supply the SYSIN data and delete all SYSPRINT records. Figure 80 shows the JCL used, Figure 74 shows the code used to invoke DFDSS, and Figure 81 shows how the UIM exit routine has been modified to permit the SYSIN and SYSPRINT data to be handled totally in-storage (no data sets specified nor allocated by the calling program). Note that the SYSPRINT exit causes all data to be deleted; this of course need not be the case in any practical application.

```

//JOB1    JOB   accounting information,REGION=nnnnK
//STEP1   EXEC  PGM=MYJOB
//STEPLIB DD   DSN=MY.LINKLIB,DISP=SHR
//DASD    DD   UNIT=3380,VOL=(PRIVATE,SER=111111),DISP=OLD
//TAPE    DD   UNIT=3480,VOL=SER=(TAPE01,TAPE02),
//         LABEL=(1,NL),DISP=(NEW,KEEP)

```

Figure 80. Invoke DFDSS for In-Storage I/O

NEWUIM	CSECT		
INSTALRC	L	4,EIRECPTR	POINT TO ADRUFO RECORD
	USING	UFO,4	ADDRESSABILITY TO UFO
	OI	UFAIFLGS,UFSYSIN+UFSYSPR	FORCE SYSIN/SYSPRINT IN-CORE
	DROP	4	
	LH	1,EIRC16	SET RECORD MODIFIED RETCODE
	STH	1,EIRETCOD	IN EIDB
	BR	14	RETURN TO INTERCEPT PROCESSOR
SYSINRC	TM	SYSINFL,SYSIFRST	Q. FIRST ENTRY TO SYSIN EXIT?
	BO	NXTSYS	A. NO, GET NEXT INPUT.
	OI	SYSINFL,SYSIFRST	INDICATE FIRST TIME TAKEN
	LA	1,FIRSTCOM	POINT TO FIRST DFDSS COMMAND
	ST	1,CURCOMM	SAVE POINTER
NXTSYS	L	1,CURCOMM	GET CURRENT COMMAND POINTER
	ST	1,EIRECPTR	INITIALIZE THE POINTER
	SR	5,5	CLEAR WORK REGISTER
	IC	5,0(,1)	GET COMMAND LENGTH
	LTR	5,5	Q. END OF SYSIN DATA?
	BZ	SYSDEL	A. YES, BYPASS SYSIN DS
	ST	5,EIRECLEN	SET RECORD LENGTH
	LA	1,1(5,1)	POINT TO NEXT RECORD
	ST	1,CURCOMM	SAVE NEW POINTER
	LH	1,EIRCO8	SET RECORD INSERTED RETCODE
	B	SYSEXIT	CONTINUE SETUP
SYSDEL	L	5,EIRECLEN	GET CURRENT RECORD LENGTH
	LTR	5,5	Q. EOF CONDITION?
	BZ	SYSRCOO	A. YES, PROCESS IT.
	LH	1,EIRC12	DELETE CURRENT SYSIN RECORD
	B	SYSEXIT	CONTINUE SETUP
SYSRCOO	LH	1,EIRCOO	USE EOF CONDITION
SYSEXIT	STH	1,EIRETCOD	IN EIDB
	BR	14	RETURN TO INTERCEPT PROCESSOR
CURCOMM	DC	F'0'	ADDRESS OF CURRENT COMMAND
FIRSTCOM	DC	AL1(COM1)	LENGTH OF FIRST COMMAND
	DC	' DUMP INDD(DASD) OUTDD(TAPE)'	
COM1	EQU	*-FIRSTCOM-1	
SECNDCOM	DC	AL1(COM2)	LENGTH OF COMMAND
	DC	' IF LASTCC = 0 -'	
COM2	EQU	*-SECNDCOM-1	
THIRDCOM	DC	AL1(COM3)	LENGTH OF COMMAND
	DC	' THEN DEFRAG DDN(DASD)'	
COM3	EQU	*-THIRDCOM-1	
ENDCOMM	DC	X'00'	END OF COMMAND FLAG
SYSINFL	DC	X'00'	SYSIN FLAG
SYSIFRST	EQU	X'80'	FIRST ENTRY SYSIN PERFORMED
SYSPRRC	LH	1,EIRC12	SET DELETE SYSPRINT RETCODE
	STH	1,EIRETCOD	IN EIDB
	BR	14	RETURN TO INTERCEPT PROCESSOR
VEC01	DC	A(SYSINRC)	PROCESSOR - READ SYSIN EXIT
VEC02	DC	A(SYSPRRC)	PROCESSOR - WRITE SYSPRINT EXIT

Figure 81 (Part 1 of 2). In-Storage SYSIN and SYSPRINT

```

VEC13   DC   A(INSTALRC)      PROCESSOR - PRESENT ADRUFO RECORD EXIT
          .
          .
          ADRUFO
          .
          .

```

Figure 81 (Part 2 of 2). In-Storage SYSIN and SYSPRINT

## Modify SYSIN Data

The example in Figure 82 is the same as that described in “Disable UIM Usage via the UIM” on page 205, except that the UIM is used to modify the SYSIN data. Figure 70 shows the JCL used, Figure 74 shows the code used to invoke DFDSS, and Figure 82 shows how the UIM exit routine has been modified to permit the SYSIN data to be changed from the DUMP and DEFrag used in prior examples, to a RESTORE and DEFrag. The volume being restored is assumed to not contain any of the data sets that are on the input tape data set. If this were not the case, PURGE would have to be specified. It is also assumed that there are no protected data sets on the tape or PASSWORD parameters would have to be supplied. Note that the SYSPRINT exit is not being used in this example and that none of the checked SYSIN commands are continued on succeeding records.

## Recover from an Incorrect Exit Action

The example in Figure 83 on page 212 demonstrates how a particular UIM exit routine recovers from a disallowed option during write logical tape operations. This exit was chosen for the purposes of clarity; any other exit could be handled similarly. Figure 70 shows the JCL used, Figure 74 shows the code used to invoke DFDSS, and Figure 83 shows how the UIM exit routine has been modified to permit the write logical tape processor to incorrectly attempt to perform a DELETE operation and then rectify the error by returning to CONTINUE NORMALLY operation. The purpose of this exit is to output a statistics flag record whenever a change occurs between data tracks and nondata tracks. At the first occurrence of a data track record, the statistics record is to be inserted and then the data tracks; unfortunately, the coder of the UIM used the DELETE return code 12 instead of the INSERT return code 8. Because DELETES are not allowed for write logical tape data records, DFDSS sets the error flag EIXERR in the EIXALOW3 field for a one-time error correction retry. The coder of the UIM did anticipate the possibility of disallowed functions and specified that DFDSS ignore the changes made on the prior call and just use the originally presented (to the exit) record by using the return code zero (0). Although this is a trivial example, it does show how the UIM can have the flexibility to perform a normal set of functions and still recover when the environment is not as expected.

NEWUIM	CSECT		
SYSINRC	TM	SYSINFL,SYSIFRST	Q. FIRST ENTRY TO SYSIN EXIT?
	BO	NXTSYS	A. NO, GET NEXT INPUT.
	OI	SYSINFL,SYSIFRST	INDICATE FIRST TIME TAKEN
	LA	1,FIRSTCOM	POINT TO FIRST DFDSS COMMAND
	ST	1,CURCOMM	SAVE POINTER
NXTSYS	L	1,CURCOMM	GET CURRENT COMMAND POINTER
	L	4,EIRECPTR	POINT TO RECORD AREA
	SR	5,5	CLEAR WORK REGISTER
	IC	5,0(,1)	GET COMMAND LENGTH
	LTR	5,5	Q. END OF SYSIN DATA?
	BZ	SYSDEL	A. YES, BYPASS SYSIN DS
	ST	5,EIRECLEN	SET RECORD LENGTH
	BCTR	5,0	MAKE EXECUTABLE
	EX	5,MOVCOMM	MOVE COMMAND TO RECORD AREA
	LA	1,2(5,1)	POINT TO NEXT RECORD
	ST	1,CURCOMM	SAVE NEW POINTER
	LH	1,EIRC08	SET RECORD INSERTED RETCODE
	B	SYSEXIT	CONTINUE SETUP
SYSDEL	L	5,EIRECLEN	GET CURRENT RECORD LENGTH
	LTR	5,5	Q. EOF CONDITION?
	BZ	SYSRC00	A. YES, PROCESS IT.
	LA	1,30	LENGTH TO LOOK FOR COMMAND TYPE
SYSCHK	CLC	0(6,4),DUMPC	Q. DUMP COMMAND?
	BE	SYSRC12	A. YES, DELETE IT
	LA	4,1(,4)	NEXT STARTING POSITION IN RECORD
	BCT	1,SYSCHK	LOOP THROUGH LIMIT
	B	SYSRC00	USE CURRENT COMMAND
SYSRC12	LH	1,EIRC12	DELETE CURRENT SYSIN RECORD
	B	SYSEXIT	CONTINUE SETUP
SYSRC00	LH	1,EIRC00	USE EOF CONDITION
SYSEXIT	STH	1,EIRETCOD	IN EIDB
	BR	14	RETURN TO INTERCEPT PROCESSOR
MOVCOMM	MVC	0(0,4),1(1)	MOVE COMMAND TO RECORD AREA
DUMPC	DC	C' DUMP '	DUMP COMMAND
CURCOMM	DC	F'0'	ADDRESS OF CURRENT COMMAND
FIRSTCOM	DC	AL1(COM1)	LENGTH OF FIRST COMMAND
	DC	' REST INDD(TAPE) OUTDD(DASD)'	
COM1	EQU	*-FIRSTCOM-1	
ENDCOMM	DC	X'00'	END OF COMMAND FLAG
SYSINFL	DC	X'00'	SYSIN FLAG
SYSIFRST	EQU	X'80'	FIRST ENTRY SYSIN PERFORMED
VEC01	DC	A(SYSINRC)	PROCESSOR - READ SYSIN EXIT

Figure 82. Modify the SYSIN Data Stream

NEWUIM	CSECT		
WRITLTAP	TM	EIXALOW3,EIXERR	Q. DISALLOWED FUNCTION ATTEMPTED?
	BO	TRKRCOO	A. YES, USE CURRENT RECORD
	L	4,EIRECPTR	POINT TO RECORD AREA
	TM	10(4),X'08'	Q. DATA TRACK?
	BZ	NOTDATA	A. NO, TURN OFF FIRST FLAG
	TM	TRKFLAG,TRKIFRST	Q. FIRST DATA TRACK?
	BO	TRKRCOO	A. NO, USE CURRENT RECORD
	OI	TRKFLAG,TRKIFRST	INDICATE FIRST TIME TAKEN
	LA	1,STATSREC	POINTER TO STATISTICS RECORD
	IC	5,0(,1)	GET COMMAND LENGTH
	ST	5,EIRECLEN	SET RECORD LENGTH
	BCTR	5,0	MAKE EXECUTABLE
	EX	5,MOVSTAT	MOVE STATS RECORD TO RECORD AREA
	LH	1,EIRC12	*** WRONG RETCODE - DELETE USED ***
	B	TRKEXIT	CONTINUE SETUP
NOTDATA	NI	TRKFLAG,255-TRKIFRST	RESET FIRST TRACK RECORD
TRKRCOO	LH	1,EIRC00	USE CURRENT RECORD
TRKEXIT	STH	1,EIRETCOD	IN EIDB
	BR	14	RETURN TO INTERCEPT PROCESSOR
MOVSTAT	MVC	0(0,4),1(1)	MOVE STATS RECORD TO RECORD AREA
STATSREC	DC	AL1(STA1)	LENGTH OF FIRST RECORD
	DC	'BEGINNING OF DATA SET TRACKS'	
STA1	EQU	*-STATSREC-1	
TRKFLAG	DC	X'00'	TRACK FLAG
TRKIFRST	EQU	X'80'	FIRST ENTRY STATS PERFORMED
VEC05	DC	A(WRITLTAP)	PROCESSOR - WRITE LOGICAL TAPE

Figure 83. Recognize a Disallowed Option

## Disable Functional Exit via the UIM

The example in Figure 84 demonstrates how a particular UIM exit routine disconnects itself after having processed. The prior example is used again, but it only writes one statistics record and then disconnects itself. Note that any other exits currently active remain active after this exit has returned its code of 20. Figure 70 shows the JCL used, Figure 74 shows the code used to invoke DFDSS, and Figure 84 shows how the UIM exit routine has been modified to permit the write logical tape processor to INSERT the statistics record (corrected from the prior example) and then disconnect itself at the next call.

NEWUIM	CSECT		
WRITLTAP	TM	EIXALOW3,EIXERR	Q. DISALLOWED FUNCTION ATTEMPTED?
	BO	TRKRC00	A. YES, USE CURRENT RECORD
	L	4,EIRECPTR	POINT TO RECORD AREA
	TM	10(4),X'08'	Q. DATA TRACK?
	BZ	TRKRC00	A. NO, USE RECORD AS IS
	TM	TRKFLAG,TRKIFRST	Q. FIRST DATA TRACK?
	BO	TRKRC20	A. NO, DISCONNECT EXIT
	OI	TRKFLAG,TRKIFRST	INDICATE FIRST TIME TAKEN
	LA	1,STATSREC	POINTER TO STATISTICS RECORD
	IC	5,0(,1)	GET COMMAND LENGTH
	ST	5,EIRECLEN	SET RECORD LENGTH
	BCTR	5,0	MAKE EXECUTABLE
	EX	5,MOVSTAT	MOVE STATS RECORD TO RECORD AREA
	LH	1,EIRC08	*** INSERT RECORD RETCODE USED ***
	B	TRKEXIT	CONTINUE SETUP
TRKRC20	LH	1,EIRC20	USE CURRENT RECORD AND DISCONNECT EXIT
	B	TRKEXIT	CONTINUE SETUP
TRKRC00	LH	1,EIRC00	USE CURRENT RECORD
TRKEXIT	STH	1,EIRETCOD	IN EIDB
	BR	14	RETURN TO INTERCEPT PROCESSOR
MOVSTAT	MVC	0(0,4),1(1)	MOVE STATS RECORD TO RECORD AREA
STATSREC	DC	AL1(STA1)	LENGTH OF FIRST RECORD
	DC	'BEGINNING OF DATA SET TRACKS'	
STA1	EQU	*-STATSREC-1	
TRKFLAG	DC	X'00'	TRACK FLAG
TRKIFRST	EQU	X'80'	FIRST ENTRY STATS PERFORMED
VEC05	DC	A(WRITLTAP)	PROCESSOR - WRITE LOGICAL TAPE

Figure 84. Disconnect a Single UIM Exit

### Create a Function-Taskid Table via SYSPRINT Data

The example in Figure 85 on page 214 demonstrates how a UIM stores task-related information for use by other UIM exit routine points. The technique involves saving 'ADR101I' messages in the SYSPRINT data and creating a table in-storage that cross-references the functions with their task-ids. The purpose of doing this here (in the SYSPRINT exit) is to allow the UIM installation exit to disable any function prior to its being scheduled. The table could have been created at the function startup exit, but by then the function would already have been scheduled for execution. Figure 70 shows the JCL used, Figure 74 shows the code used to invoke DFDSS, and Figure 85 shows how the UIM exit routine has been modified to accomplish the above.

NEWUIM	CSECT		
SYSPRRC	L	1,EITSKID	GET CURRENT TASK ID
	LTR	1,1	Q. NON-SYSIN TASK?
	BNZ	SYSPRCOO	A. YES, USE RECORD
	L	4,EIRECPTR	POINT TO SYSPRINT RECORD
	LA	1,25	SEARCH LENGTH
NXTSYSP	CLC	0(8,4),ADR101	Q. TASK ASSIGNED MESSAGE?
	BE	SYSPASGN	A. YES, TABLE ENTRY
	LA	4,1(,4)	SHIFT OFFSET
	BCT	1,NXTSYSP	TEST AGAIN
SYSPRCOO	LH	1,EIRCOO	USE CURRENT RECORD
	B	SYSEXIT	CONTINUE SETUP
SYSPASGN	LA	1,25	SEARCH LENGTH
	LA	4,7(,4)	ADVANCE POINTER
NXTSYSA	CLC	0(8,4),TASKCON	Q. TASKID LOCATED?
	BE	SYSCVB	A. YES, CONVERT IT
	LA	4,1(,4)	SHIFT OFFSET
	BCT	1,NXTSYSA	TEST AGAIN
	B	SYSPRCOO	WRONG RECORD TYPE
SYSCVB	LA	4,8(,4)	POINT TO TASKID NUMBER
	PACK	WORK(8),0(3,4)	PREPARE FOR CONVERSION
	OI	WORK+7,X'OF'	ASSURE NUMERIC
	CVB	1,WORK	CONVERT TO BINARY
	L	5,CUREND	GET CURRENT TABLE END POINTER
	ST	1,0(,5)	STORE TASK ID
	LA	1,30	SET LENGTH
	LA	4,4(,4)	ADVANCE POINTER
SYSPFUN	CLI	0(4),C''''	Q. FUNCTION FLAG FOUND?
	BE	SYSPCNT	A. YES, COUNT LENGTH
	LA	4,1(,4)	SHIFT OFFSET
	BCT	1,SYSPFUN	TEST AGAIN
	B	SYSPRCOO	WRONG RECORD TYPE
SYSPCNT	LA	6,1(,4)	SAVE START POINTER
	LR	4,6	ADJUST PAST QUOTE
	LA	1,1	SET LENGTH
SYSPCTR	CLI	1(4),C''''	Q. END OF FUNCTION NAME?
	BE	SYSPSTO	A. YES, STORE NAME
	LA	1,1(,1)	COUNT CHARACTERS
	LA	4,1(,4)	ADVANCE POINTER
	CH	1,LIMVAL	Q. EXCEEDS TABLE LENGTH?
	BNE	SYSPCTR	A. NO, KEEP LOOKING FOR END
SYSPSTO	MVC	4(8,5),BLANKS	CLEAR TABLE FUNCTION NAME FIELD
	BCTR	1,0	MAKE EXECUTABLE
	EX	1,MOVFUNC	MOVE FUNCTION NAME TO TABLE
	LA	5,12(,5)	POINT TO NEXT TABLE SLOT
	ST	5,CUREND	SAVE POINTER
SYSRCOO	LH	1,EIRCOO	USE EOF CONDITION
SYSEXIT	STH	1,EIRETCOD	IN EIDB
	BR	14	RETURN TO INTERCEPT PROCESSOR
	CNOP	0,8	
WORK	DC	D'0'	WORK AREA
CUREND	DC	A(TASKTAB)	NEXT EMPTY TABLE AREA
LIMVAL	DC	H'8'	MAXIMUM FUNCTION NAME LENGTH
MOVFUNC	MVC	4(0,5),0(6)	MOVE FUNCTION TO TABLE
ADR101	DC	CL8'ADR1011'	MESSAGE IDENTIFIER
TASKCON	DC	C'TASKID'	TASKID IDENTIFIER
	CNOP	0,4	

Figure 85 (Part 1 of 2). Table Function-Task ID Information

TASKTAB	DC	10XL12'00'	TABLE AREA FOR 10 TASKS
VECO2	DC	A(SYSPRRC)	PROCESSOR - WRITE SYSPRINT EXIT

Figure 85 (Part 2 of 2). Table Function-Task ID Information

### Request Only User Statistics Records

The example in Figure 86 demonstrates how to request only user statistics records for the logical tape read exit. Figure 70 shows the JCL used, Figure 74 shows the code used to invoke DFDSS, and Figure 86 shows how the UIM exit routine has been modified to accomplish the above.

NEWUIM	CSECT		
READLTAP	LH	1,EIRCOO	INDICATE RECORD TO BE USED
	STH	1,EIRETCOD	SET NORMAL USE RETURN CODE
	TM	TRKFLAG,TRKSTAT	Q. STATS RECORDS ALREADY SET?
	BO	TRKUSE	A. YES, USE STATS RECORD
	OI	TRKFLAG,TRKSTAT	INDICATE STATS REQUESTED
	LH	1,EIRC24	RESET TO REQUEST STATS
	STH	1,EIRETCOD	SET REQUEST STATS RETURN CODE
	L	4,EIRECPTR	POINT TO DATA RECORD
	TM	18(4),X'04'	Q. USER STATS RECORD?
	BO	TRKUSE	A. YES, USE STATS RECORD
	B	TRKEXIT	IGNORE CURRENT RECORD
TRKUSE	L	4,EIRECPTR	POINT TO STATISTICS RECORD
		* (ANY STATISTICS RECORD PROCESSING ROUTINE)	
TRKEXIT	BR	14	RETURN TO INTERCEPT PROCESSOR
TRKFLAG	DC	X'00'	TRACK READ EXIT FLAGS
TRKSTAT	EQU	X'80'	STATISTICS RECORDS ONLY, REQUESTED
VECO4	DC	A(READLTAP)	PROCESSOR - READ LOGICAL TAPE

Figure 86. Request Statistics Records Only



## Process WTO/WTOR Records via the UIM

The example in Figure 87 demonstrates a method of processing the WTORs issued by DFDSS. Processing WTO RECORDS is just a subset of this method, because no response is needed and the scanning to recognize the message would be the same. Figure 70 shows the JCL used, Figure 74 shows the code used to invoke DFDSS, and Figure 87 shows how the UIM exit routine has been modified to check each message. The UIM exit routine scans the authorization WTOR type. If a WTOR must be issued from ADRAUTH to resolve the authorization question, it returns the "Not authorized" response of "T," indicating that no protected data sets may be processed.

NEWUIM	CSECT		
WTORRC	L	4,EIRECPTR	GET WTO/WTOR RECORD POINTER
	CLC	12(8,4),MSG369	Q. AUTHORIZE REQUEST?
	BNE	WTORCOO	A. NO, LET DFDSS ISSUE WTO/WTOR
	LA	1,1	LENGTH OF RESPONSE
	ST	1,EIRECLEN	STORE IN EIDB
	MVI	0(4),C'T'	INDICATE NOT AUTHORIZED
	LA	1,EIRC28	EXIT HANDLED RESPONSE
	B	WTOEXIT	FINISH SETUP
WTORCOO	LH	1,EIRC00	SET USE MESSAGE RETURN CODE
WTOEXIT	STH	1,EIRETCOD	IN EIDB
	BR	14	RETURN TO INTERCEPT PROCESSOR
MSG369	DC	C'ADR369D '	AUTHORIZATION WTOR MESSAGE
VEC12	DC	A(WTORRC)	PROCESSOR - WRITE WTOR MESSAGE EXIT

Figure 87. Handle WTOR Messages in a UIM

## Termination Exit Processing in the UIM

The example in Figure 88 demonstrates a method of function termination processing. It is assumed that the UIM may have opened a data set in a prior exit and, if so, this routine scans the DCB pointers (maintained by the UIM) to assure that they are all closed prior to the termination of the function being processed. It is further assumed that the UIM is not being processed by multiple functions in a parallel mode. Figure 70 shows the JCL used, Figure 74 shows the code used to invoke DFDSS, and Figure 88 shows how the UIM exit routine has been modified to accomplish this process. For this exit, EIRECPTR points to a fullword field containing the final return code of the function.



## Link-editing Your UIM

If you write your own UIM routine, see Figure 89 for an example of how to link-edit your routine. The example assumes that module USRUIM 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(USRUIM)
                ENTRY        USRUIM
                NAME          USRUIM(R)
/*
```

Figure 89. Link-editing Your UIM

# ADREID0 Data Area

Figure 90 describes the data area corresponding to the DFDSS exit identification block.

Offsets	Type	Length	Name	Description	
0	(0)	STRUCTURE	38	ADREIB	
0	(0)	SIGNED	2	EIDLLEN	LENGTH OF ADREIB 2
2	(2)	CHARACTER	4	EIID	BLOCK IDENTIFIER EBCDIC "EIDB"
6	(6)	SIGNED	4	EITSKID	TASK ID NUMBER
10	(A)	BITSTRING	4	EIXALLOW	USER EXIT ALLOWANCE OPTIONS
10	(A)	BITSTRING	1	EIXALOW0	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 TERMINATION
11	(B)	BITSTRING	1	EIXALOW1	ALLOWANCE OPTION BYTE 2
11	(B)	BITSTRING	1		RESERVED
12	(C)	BITSTRING	1	EIXALOW2	ALLOWANCE OPTION BYTE 3
12	(C)	BITSTRING	1		RESERVED
13	(D)	BITSTRING	1	EIXALOW3	ALLOWANCE OPTION BYTE 4
		1... ..		EIXERR	DISALLOWED OPTION ATTEMPTED
		.111 1111			RESERVED
14	(E)	SIGNED	2	EIOPTION	PROCESSING OPTION (see "Meanings")
16	(10)	SIGNED	2	EIRETCOD	EXIT RETURN CODE (see "Meanings")
18	(12)	SIGNED	4	EIRECALN	RECORD AREA LENGTH
22	(16)	SIGNED	4	EIRECLEN	ORIGINAL RECORD LENGTH
26	(1A)	A-ADDRESS	4	EIRECPTR	ORIGINAL RECORD ADDRESS
30	(1E)	A-ADDRESS	4	EIUSEPTR	USER DATA AREA ADDRESS
34	(22)	A-ADDRESS	4	EIDDID	EIOPO6 DDNAME/VOLID PTR
38	(26)	BITSTRING	4	EIXFLAGS	OTHER FLAGS
38	(26)	BITSTRING	1	EIXFLAG0	FLAG BYTE 0
		1... ..		EIXABEND	FOR EIOPO2 ONLY, 1=MESSAGE IS ADRO13 INDICATING AN ABEND
		.1.. ..		EIXNTERR	FOR EIOPO2 ONLY, 1=MESSAGE IS TYPE 'E' AND IS NOT 324 OR 347
		..11 1111		RESERVED	
39	(27)	BITSTRING	1	EIXFLAG1	FLAG BYTE 1 - RESERVED
40	(28)	BITSTRING	1	EIXFLAG2	FLAG BYTE 2 - RESERVED
41	(29)	BITSTRING	1	EIXFLAG3	FLAG BYTE 3 - RESERVED
44	(2C)	CHARACTER	0		FORCE SIZE OF CONTROL BLOCK TO WORD BOUNDARY
<b>DDINFO</b>					
0	(0)	STRUCTURE	16	EIDDINFO	UIM-06 INFO
0	(0)	CHARACTER	8	EIDDNAME	O/P DEVICE DDNAME
8	(8)	CHARACTER	6	EIVOLID	O/P VOLSER
14	(E)	UNSIGNED	1	EIRETC	RESERVED FOR RETCODE
15	(F)	UNSIGNED	1		RESERVED FOR FLAGS

Figure 90 (Part 1 of 2). ADREID0 Data Area

EIOPTION	Meaning	EIRETCOD	Meaning
0	Function startup entry	0	Continue normal process
1	Reading SYSIN record	4	Record replaced
2	Printing SYSPRINT record	8	Insert record
3	Reading physical tape	12	Delete record
4	Reading logical tape	16	Record modified
5	Writing logical tape	20	Disconnect exit
6	Writing physical tape	24	Select user statistics records
7	Reading disk track	28	WTOR response
8	Writing disk track	32	Terminate functional task
9	Reading utility SYSPRINT		
10	Writing utility SYSPRINT		
11	Writing WTO message		
12	Writing WTOR message		
13	Presenting ADRUFO record		
14	Function termination		
15	Presenting WTOR response		
16	OPEN/EOV tape volume security & verification		
17	OPEN/EOV nonspecific tape volume mount		
18	Insert logical VSAM record during restore		
19	Output tape I/O error		

### CROSS REFERENCE

ADREIB	0 (0)	EIXALOW1	11 (B)
EIDDID	34 (22)	EIXALOW2	12 (C)
EIDDINFO	0 (0)	EIXALOW3	13 (D)
EIDDNAME	0 (0)	EIXDEL	10 X'20'
EIDLLEN	0 (0)	EIXDIS	10 X'08'
EIID	2 (2)	EIXERR	13 X'80'
EIOPTION	14 (E)	EIXFLAGS	38 (26)
EIRECALN	18 (12)	EIXFLAG0	38 (26)
EIRECLEN	22 (16)	EIXFLAG1	39 (27)
EIRECPTR	26 (1A)	EIXFLAG2	40 (28)
EIRETC	14 (E)	EIXFLAG3	41 (29)
EIRETCOD	16 (10)	EIXINS	10 X'40'
EITSKID	6 (6)	EIXMOD	10 X'10'
EIUSEPTR	30 (1E)	IXNTERR	38 X'40'
EIVOLID	8 (8)	EIXREP	10 X'80'
EIXABEND	38 X'80'	EIXSTAT	10 X'02'
EIXALLOW	10 (A)	EIXTERM	10 X'01'
EIXALOWO	10 (A)	EIXWTOR	10 X'04'

Figure 90 (Part 2 of 2). ADREID0 Data Area

## Chapter 8. Using 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 host system environment. Alternatively, you can run ADRDMPRS in a VM/370 virtual machine. Stand-alone DFDSS Restore operates in either System/370 extended-architecture mode or System/370 mode.

### Devices You Can Use

Stand-alone DFDSS Restore can be used with the following IBM **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

Stand-alone DFDSS Restore can be used with the following IBM **system consoles**:

- 1052 Printer-Keyboard Model 7
- 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
- 3036 Console
- 3056 Remote System Console
- 3138 Processing Unit console
- 3148 Processing Unit console
- 3158 Processing Unit display console keyboard

Stand-alone DFDSS Restore can be used with the following IBM **printers**:

- 1403 Printer
- 1443 Printer
- 3203 Printer
- 3211 Printer
- 3262 Line Printer Model 5
- 3800 Printing Subsystem

4245 Printer  
4248 Printer

Stand-alone DFDSS Restore can be used with the following **IBM card readers**:

2501 Card Reader  
2520 Card Read Punch  
2540 Card Read Punch  
3505 Card Reader

Stand-alone DFDSS Restore can be used with the following **IBM tape units**:

2400 series Magnetic Tape Units  
3400 series Magnetic Tape Units  
3430 Magnetic Tape Unit  
3480 Magnetic Tape Subsystem

## Restrictions

The following restrictions apply to stand-alone DFDSS Restore:

- Stand-alone DFDSS Restore can restore from tape volume(s) created by a 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 tape(s) 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 volume(s).
- 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 restored to an IBM 3330 Model 11.
  - Dump 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.
  - Dump data from a smaller capacity IBM 3380 model can be restored to a larger capacity IBM 3380 model.
- 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 cannot restore from a dump tape created with the COMPRESS option.
- Stand-alone DFDSS cannot restore from dump tapes produced by any other utilities.
- Stand-alone DFDSS 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 System DFDSS

The dump format used by both stand-alone and system DFDSS minimizes the time required for DUMP and RESTORE operations, and facilitates the restoring of multiple data sets. Generally, dump tapes produced by system DFDSS can be used in a RESTORE operation by stand-alone DFDSS Restore.

## 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

Stand-alone DFDSS Restore is controlled by control statements. Because it is an independent program, operating-system job control statements are not used.

### Control Statements

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.

## Continuing Control Statements

Control statements are coded on cards, or 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 cards. 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 or card image.

## Generic Device Names

Generic device names may 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 may be used for tape devices, PRNT for printer devices, and CNSL for consoles.

## JOB Statement

The JOB statement indicates the beginning of a job. The syntax of the JOB statement is:

[label]	<b>JOB</b> [user-information]
---------	-------------------------------

**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:

[label]	<b>MSG</b> TODEV= <i>xxxx</i> ,TOADDR= <i>cuu</i>
---------	---

where:

**MSG** must be preceded and followed by at least one blank. If *label* is specified, it must start in column 1.

**TODEV=xxxx**

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 "Executing Stand-Alone DFDSS Restore" on page 227.)
- 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 226.)

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:

Col.		Col.
16		72
<b>[label] RESTORE FROMDEV=xxxx, FROMADDR=cuu, TODEV=xxxx, C TOADDR=cuu, VOLID=serial [ ,MODEL=n]</b>		

where:

**RESTORE** must be preceded and followed by at least one blank. If *label* is specified, it must start in column 1.

**FROMDEV=xxxx**

specifies the type of the source device; for example, 2400. Eligible device names are 2400, 2420, 3400, 3430, 3480, and **TAPE**. (To use a 3430, specify either 3430 or **TAPE**. Likewise, to use a 3480, specify 3480 or **TAPE**.)

**FROMADDR=cuu**

specifies the channel number, *c*, and unit number, *uu*, of the source device.

**TODEV=xxxx**

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. Exceptions are listed under "Restrictions" on page 222. 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=serial**

specifies the volume serial number of the receiving volume. If *serial* does not match the volume serial number found on the receiving volume, the operator is notified and the job is terminated.

**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, TOADDR=190, VOLID=111111	C

**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 one of the following tape(s):

- A full volume dump. If you omit VDRL, you restore the full volume.
- A tracks (one or more track ranges) dump. If you omit VDRL, you restore all the tracks.
- The first logical volume of a physical data set dump. You must specify VDRL in this case.

The syntax of the VDRL statement is:

<i>[label]</i>	<b>VDRL BEGIN=nnnnn[,END=nnnnn]</b>
----------------	-------------------------------------

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 (cylinder 0, track 0 = 0) track address 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:

[label]	<b>END</b> [user-information]
---------	-------------------------------

**END** must be preceded and followed by at least one blank. If *label* is specified, it must start in column 1.

## Copying Stand-Alone DFDSS Restore to Tape

After DFDSS is installed and accepted by System Modification Program or System Modification Program Extended (as described in the Program Directory), the stand-alone Restore program resides in **SYS1.ASAMPLIB** as member **ADRDMPRS**. The following sample job stream will copy the stand-alone 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.ASAMPLIB(ADRDMPRS),DISP=SHR
//SYSUT2 DD DSN=ASAMPLIB,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.

## Executing Stand-Alone DFDSS Restore

Stand-alone DFDSS Restore (**ADRDMPRS**) 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**.

**ADRDMPRS** must operate in **EC** mode. Before executing **ADRDMPRS** under **VM/370**, you should enter:

```
CP SET ECMODE ON
```

ADRDMPRS is loaded as a card deck or as card images on tape. Control statements can follow the last card, or 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 execute ADRDMPRS:

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 execute 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=xxx, cuu" where xxx 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, 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 depressing a keyboard function key on the processor's system console or depressing a button on the processor's operator control panel.
6. The control statements will now be 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, at the end of each reel, the following message is issued:

ADR561A END OF TAPE. MOUNT TAPE ON cuu. DEPRESS INTERRUPT KEY

At this point, to continue the restore, 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 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	Terminating WAIT state when ADRDMPRS has completed the job
FFFF	Terminating WAIT state when ADRDMPRS has completed the job
FFFFFF	Error WAIT in IPL loader due to one of the following bits 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 Jobs

You can use either IBM standard label or nonlabeled tape(s) as the source volume(s) for a stand-alone RESTORE 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 tape(s) at address 280.

Col. 2	Col. 16	Col. 72
JOB	RESTORE 3380 FROM TAPE	
MSG	TODEV=3210,TOADDR=009	
RESTORE	FROMDEV=3400,FROMADDR=280,TODEV=3380, TOADDR=150,VOLID=111111	C
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 tape(s) at address 280.

Col. 2	Col. 16	Col. 72
JOB	TRACKS RESTORE 3380 FROM TAPE	
MSG	TODEV=3210,TOADDR=009	
RESTORE	FROMDEV=3480,FROMADDR=280,TODEV=3380, TOADDR=150,VOLID=111111	C
VDRL	BEGIN=20,END=219	
END		

### Example 3: Same as Example 2, 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, TOADDR=150,VOLID=111111	C
VDRL	BEGIN=20,END=219	
END		

## Storage Requirement

The storage requirement, in bytes, to run stand-alone DFDSS Restore is:

$220K + \text{tracksize}$

where:

K = 1024 bytes.

tracksize = The size, in bytes, of a track on your DASD volume.

## Appendix A. User Exit Routines

User exit routines supplied with DFDSS do the following:

- The user authorization exit routine forces authorization checking of protected data sets.
- The user enqueue exit routine forces enqueueing of the VTOC for the duration of the full or tracks DUMP or COPY, or tracks PRINT, or, if TOL(ENQF) is specified, for the duration of a data-set DUMP.
- The user installation options exit routine does not override any default or user-specified command options in the input stream.
- The user reblock exit routine allows DFDSS, during a data set COPY, to use the block size it selects for the target data set.

To tailor DFDSS processing, your installation can replace the above exit routines with its own routines as follows:

- An installation-supplied authorization exit routine can bypass authorization checking for some or all users by volume or data set. It can allow DEFRAg to relocate protected data sets without checking access authority.
- An installation-supplied enqueue exit routine can force DFDSS to dequeue the VTOC after processing the VTOC or after processing the volume if the volume is on a shared DASD. For example, you may wish to enqueue the VTOC only during DFDSS's access of the VTOC, or you may not wish to check passwords. This is used for volume-oriented commands.
- An installation-supplied installation options exit routine can force execution in SERIAL or PARALLEL mode, and force or prevent COMPRESS, COPYVOLID, PURGE/REPLACE, RESET, WRITECHECK, ALLDATA, ALLEXCP, CATALOG, UNCATALOG, and DYNALLOC parameters from taking effect. It can force SHARE or TOL(ENQF), or it can change the defaulted or specified values. It can also specify, for multiple dump copies, if the dump is to terminate in case of an output error on any copy, or if the dump is to continue and produce as many copies as possible.
- An installation-supplied reblock exit routine can replace the DFDSS-selected block size for a data set COPY.

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.



# How to Write a User Authorization Exit Routine

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 will not be called. Otherwise, the exit routine supplied with DFDSS causes DFDSS to:

- 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.)
- Check authorization at the data set level for all functions other than COPYDUMP.

If a data set is protected via both RACF and 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 via 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.)

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 with 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.

## 1. The 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.

When your user authorization exit routine is called, it can do one of the following:

- Terminate the operation.
- Request processing without authorization checking at the data set level.
- Request that the user authorization exit be entered at the data set level.

## 2. The 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 this level, you can do one of the following:

- a. Terminate processing of the volume.
- b. Terminate processing of the current data set.

## How to Write a User Authorization Exit Routine (continued)

- c. Bypass authorization checking of the current data set.
- d. Have DFDSS check authorization for the current data set.
- e. Also, for b, c, and d above, you can specify if the exit routine is to be entered for subsequent data sets.

ADRDUSSU will load 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 4-word parameter list contains the following. You can use the ADRUPB mapping macro, which is in SYS1.SAMPLIB, to map this parameter list.

First word	Pointer to the format-1 DSCB for the protected non-VSAM data set or the cluster name for a VSAM data set, or 0 for volume level entry.
Second word	Pointer to the catalog name, or 0 for volume level entry.
Third word	Pointer to the volume serial.
Fourth word	Flags, which are set as follows:

### Byte 0

Bit	Meaning
0	End of the parameter list. This is always set to 1.
1	If it is on, read access is required on the volume. If it is off, write access is required.
2	Is set to 1 for a DUMP operation.
3	Is set to 1 for a RESTORE operation.
4	Is set to 1 for a COPY operation.
5	Is set to 1 for a PRINT operation.
6	Is set to 1 for a DEFRAG operation.
7	Is set to 1 for a COMPRESS operation.

### Byte 1

Bit	Meaning
0	Is set to 1 for DEFRAG, PRINT, DUMP, or RESTORE for a data set operation
1	Is set to 1 for a full volume DUMP, RESTORE, or COPY operation
2	Is set to 1 for a tracks DUMP, RESTORE, COPY, or PRINT operation
3	Is set to 1 for a RELEASE operation.
4 to 7	Reserved

### Byte 2

Bit	Meaning
0	If it is on, volume level entry. If it is off, data set level entry.
1	If it is on, it is entered for a VSAM data set cataloged under the integrated catalog facility. Cluster and catalog names are passed. If it is off, and byte 2, bit 0 is also off, it is entered at the data set level for all data sets except VSAM data sets.

Byte 3 is 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 91 on page 234.

## How to Write a User Authorization Exit Routine (continued)

**1. Volume level entry: All functions except COPYDUMP**

Return Code	Meaning		
	Terminate 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

In case of DEFrag, if a nonzero return code less than or equal to 12 is returned, all data sets on the volume are to be checked for authorization; any nonzero return code greater than 12 terminates the function.

For all other functions, the function is terminated 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			
	Terminate processing for volume?	Terminate 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 91. User Authorization Exit Routine Return Codes**

*Note:* The user 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).

## How to Write a User Authorization Exit Routine (continued)

```
//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(ADRSSU)
           ENTRY    ADRSSU
           NAME     ADRSSU(R)
/*
```

## How to Write a User Enqueue Exit Routine

The user 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

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 user enqueue exit routine will have no effect on the DEFRAG, COPYDUMP, RESTORE, or data set PRINT commands.

DFDSS will load 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 2-word parameter list contains the following. You can use the ADRUNQB mapping macro, which is in SYS1.SAMPLIB, to map this parameter list.

## How to Write a User Enqueue Exit Routine (continued)

First word            Pointer to the volume serial field for the volume being dumped.  
 Second word        Contains flags, which are set as follows:

### Byte 0

Bit	Meaning
0	End of the parameter list. This is always set to 1.
1	Reserved.
2	Is set to 1 for a DUMP operation
3	Reserved.
4	Is set to 1 for a COPY operation.
5	Is set to 1 for a PRINT operation.
6	Reserved.
7	Reserved.

### Byte 1

Bit	Meaning
0	Is set to 1 for a data set operation
1	Is set to 1 for a full volume DUMP or COPY operation
2	Is set to 1 for a tracks DUMP, COPY, or PRINT operation
3 to 7	Reserved

### Byte 2

Bit	Meaning
0	Is set to 1 if TOLERATE(ENQFAILURE) is used
1-7	Reserved

Byte 3 is 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)
/*
```

## How to Write a User Installation Options Exit Routine

The user installation options exit routine can control the execution of tasks or change some of the defaults or specified options. It is given control initially when DFDSS is invoked and also before execution 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 mode. When entered for each task, it can change the following options, defaults, or values (as applicable):

ALLDATA(*)	to no ALLDATA, and vice versa
ALLEXCP	to no ALLEXCP, and vice versa
ALLMULTI	to no ALLMULTI, and vice versa
CATALOG	to no CATALOG, and vice versa
COMPRESS	to no COMPRESS, and vice versa
COPYVOLID	to no COPYVOLID, and vice versa
DELETE	to no DELETE, and vice versa
DYNALLOC	to no DYNALLOC, and vice versa
FORCE	to no FORCE, and vice versa
PURGE (or REPLACE)	to no PURGE (or REPLACE), and vice versa
RACFLOG	to no RACFLOG, and vice versa
REBLOCK	to no REBLOCK, and vice versa
RECATALOG	to no RECATALOG, and vice versa
RESET	to no RESET, and vice versa
SERIAL	to PARALLEL, and vice versa
TOL(ENQF)	to no TOL(ENQF), and vice versa
UNCATALOG	to no UNCATALOG, and vice versa
WRITECHECK	to no WRITECHECK, and vice versa

OPTIMIZE, FRAGI, SHARE, and WAIT count values.

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 via the installation options exit.

The exit routine can force the DUMP function to terminate if a write error occurs on any of the output copies. This is done by setting bit "UFOIACPY" on as described in "ADRUF0 Data Area" on page 239. 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.

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 will load 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.

## How to Write a User Installation Options Exit Routine (continued)

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 execution 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 ADRUFO mapping macro, which is in SYS1.SAMPLIB, to map the parameter list shown in Figure 92 on page 239. 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 UFOFUNCT 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, which 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.

## ADRUF0 Data Area

User Installation Options Exit Parameter Block				
Offsets	Type	Length	Name	Description
0	(0) STRUCTURE	20	ADRUF0B	
0	(0) CHARACTER	20	UFOHDR	HEADER OF UFO PARM LIST
0	(0) CHARACTER	4	UFID	IDENTIFIER EBCDIC "UFO "
4	(4) SIGNED	2	UFLEN	LENGTH OF PARM LIST
6	(6) SIGNED	2	UFBDOFF	OFFSET TO UFOFUNCT OR UFOPARM
8	(8) A-ADDRESS	4	UFVOLI@	ADDR OF INPUT VOL LIST   0
12	(C) A-ADDRESS	4	UFVLO@	ADDR OF OUTPUT VOL LIST   0
16	(10) BITSTRING	2	UFFUNCT	FUNCTION BEING PERFORMED '0000'X = PARM CHANGE ENTRY
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
	.111 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 DATA SET DUMP WITH NO INPUT VOLS OR LOGINDD/LOGINDY
	..... ..111			RESERVED
19	(13) CHARACTER	1	UFAIFLGS	APPLICATION INTERFACE FLAGS
	1... ..		UFAIINV	1=INVOKED BY APPL. INTERFACE
	.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

Figure 92 (Part 1 of 4). ADRUF0 Data Area



## ADRUF0 Data Area (continued)

Offsets	Type	Length	Name	Description
20	(14) CHARACTER	0		
0	(0) STRUCTURE	24	UFOFUNCT	LIST OF OPTIONS FOR FUNCTION. POINTED TO BY THE ADDRESS OF UFOHDR + UFBDOFF. 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	(1) BITSTRING	1	UFO1ALLE	1 = ALLEXCP
	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)
	...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
2	(2) UNSIGNED	1	UFDUOPTM	RESERVED
3	(3) BITSTRING	1	UFOINSOP	OPTIMIZE VALUE (DUMP) (1, 2, 3, OR 4)
	1... ..		UFOERASE	INSTALLATION OPTIONS
	.1.. ..		UFOIACPY	1 = ERASE PROTECTED DASD TRACKS
	..1. ....		UFOBLDIX	1 = DUMP MUST PRODUCE ALL OUTPUT COPIES OR NONE AT ALL
	...1 ....		UFORACLG	1 = INVOKE ICKDSF TO REBUILD VTOC INDEX
	.... 1111			1 = RACFLOG=YES OR FORCE RACF LOGGING RESERVED
4	(4) UNSIGNED	4	UFOFRAGI	FRAGMENTATION INDEX(DEFRAG)
8	(8) UNSIGNED	4	UFOMNGAS	9,90,900=900, 09=90, 009=9
12	(C) UNSIGNED	4	UFOMNSQT	MINIMUM WASTED SPACE ON A PDS COMPRESS COMMAND
16	(10) UNSIGNED	4	UFOMNTUS	MINIMUM SECONDARY ALLOCATION QUANTITY
				MINIMUM UNUSED TRACKS
				RELEASEUNUSED , CHANGEALLOC

Figure 92 (Part 2 of 4). ADRUF0 Data Area

ADRUFO Data Area (continued)

Offsets	Type	Length	Name	Description
20	(14) BITSTRING	1	UF03FLGS	3RD OPTION FLAG BYTE
	1... ..		UF03FORC	1= FORCE UNMOVABLES ON COPY, RESTORE
	.1... ..		UF03REPL	1 = REPLACE(DATASET COPY, RESTORE)
	..1... ..		UF0FRBLK	1 = FORCE REBLOCKING OF DATA SETS (COPY)
	...1... ..		UF0DRBLK	1 = DISABLE REBLOCKING OF DATA SETS (COPY)
	.... 1... ..		UF0ALLMU	1 = ALLMULTI PROCESSING FOR COPY, DUMP
	.... .111			RESERVED
21	(15) UNSIGNED	1	UFOWAITS	WAIT TIME IN SECONDS BETWEEN RESERVE & ENQ RETRIES(ALL)
22	(16) UNSIGNED	1	UFOWAITR	NUMBER OF RETRIES ON RESERVE OR ENQ FAILURES(ALL COMMANDS)
24	(18) BITSTRING	1	UFOTGTAL	TGTALLOC FLAGS
	1... ..		UFOTGTCY	1 = CYLINDER
	.1... ..		UFOTGTTR	1 = TRACK
	..1... ..		UFOTGTBL	1 = BLOCK
	...1... ..		UFOTGTSR	1 = SOURCE
	.... 1111			RESERVED
25	(19) CHARACTER	0		
-----				
0	(0) STRUCTURE	4	UFOPARM	EXECUTE CARD PARAMETER OPTION LIST. POINTED TO BY THE ADDRESS OF UFOHDR + UFBDOFF. 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	BUFFER LOCATION CONTROL FLAG
	1... ..		UFXABUFF	1 = I/O BUFFER ABOVE 16MB LINE
	.1... ..		UFBI31B	1 = AI BUFFER ABOVE 16MB LINE
	..11 1111			RESERVED
2	(2) CHARACTER	2		RESERVED
-----				
4	(4) CHARACTER	0		
-----				
0	(0) STRUCTURE	4	UFOVOL	VOLUME LIST HEADER. POINTED TO BY UFVOLI@ AND UFVLO@.
-----				
0	(0) CHARACTER	4	UFOVHDR	HEADER
-----				
0	(0) SIGNED	2	UFOVCNT	COUNT OF VOLUME LIST ENTRIES
2	(2) CHARACTER	2		RESERVED
-----				
4	(4) CHARACTER	0	UFOVENT	VOLUME LIST ENTRY
-----				
4	(4) A-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

**Note:** Fields UFOVENT, UFOVUCB@, UFODDNAM, and UFOVOLID are repeated for each additional volume list entry.

Figure 92 (Part 3 of 4). ADRUFO Data Area

## ADRUFO Data Area (continued)

### CROSS REFERENCE

Name	Offsets/ Equ Value	Name	Offsets/ Equ Value
ADRUF0B	0 (0)	UFOVOLID	16 (10)
UFAIFLGS	19 (13)	UFOVUCB@	4 (4)
UFAIINV	19 X'80'	UFOWAITR	22 (16)
UFAI31B	1 X'40'	UFOWAITS	21 (15)
UFBDOFF	6 (6)	UFO1ALD	0 X'04'
UFDEFPAR	0 X'40'	UFO1ALDL	0 X'02'
UFDUOPTM	2 (2)	UFO1ALLE	0 X'01'
UFFIFILT	18 X'20'	UFO1COMP	0 X'80'
UFFIFULL	18 X'80'	UFO1CVOL	0 X'40'
UFFIND	18 (12)	UFO1FLGS	0 (0)
UFFIPART	18 X'40'	UFO1PURG	0 X'20'
UFFIPRTV	18 X'10'	UFO1RESE	0 X'10'
UFFLOGCL	18 X'08'	UFO1WRCK	0 X'08'
UFFORSER	0 X'80'	UFO2CTLG	1 X'04'
UFFUCOMP	16 X'01'	UFO2DEL	1 X'08'
UFFUCOPY	16 X'40'	UFO2DYNQ	1 X'80'
UFFUCPYD	16 X'04'	UFO2ENQE	1 X'40'
UFFUDEF	16 X'80'	UFO2ENQN	1 X'10'
UFFUDUMP	16 X'10'	UFO2ENQS	1 X'20'
UFFUNCT	16 (10)	UFO2FLGS	1 (1)
UFFUNCT1	16 (10)	UFO2RECT	1 X'04'
UFFUNCT2	17 (11)	UFO2UNC	1 X'02'
UFFUPRT	16 X'08'	UFO3FLGS	20 (14)
UFFUREST	16 X'20'	UFO3FORC	20 X'80'
UFFURLSE	17 X'80'	UFO3REPL	20 X'40'
UFID	0 (0)	UFSERPAR	0 (0)
UFLEN	4 (4)	UFSTOP	19 X'10'
UFNOIN	19 X'02'	UFSYSIN	19 X'08'
UFNOOUT	19 X'01'	UFSYSRPR	19 X'04'
UFOALLMU	20 X'08'	UFUIMAL	19 X'40'
UFOBLDIX	3 X'20'	UFUIMCH	19 X'20'
UFODDNAM	8 (8)	UFVOLI@	8 (8)
UFODRBLK	20 X'10'	UFVLO@	12 (C)
UFOERASE	3 X'80'	UFXABUFF	1 X'80'
UFOFRAGI	4 (4)	UFXAFLAG	1 (1)
UFOFRBLK	20 X'20'		
UFOFUNCT	0 (0)		
UFOHDR	0 (0)		
UFOIACPY	3 X'40'		
UFOINSOP	3 (3)		
UFOMNGAS	8 (8)		
UFOMNSQT	12 (C)		
UFOMNTUS	16 (10)		
UFOPARM	0 (0)		
UFORACLG	3 X'10'		
UFOTGTAL	24 (18)		
UFOTGTBL	24 X'20'		
UFOTGTCTY	24 X'80'		
UFOTGTSTR	24 X'10'		
UFOTGTTR	24 X'40'		
UFOVCNT	0 (0)		
UFOVENT	4 (4)		
UFOVHDR	0 (0)		
UFOVOL	0 (0)		

Figure 92 (Part 4 of 4). ADRUFO Data Area

## How to Write a User Reblock Exit Routine

The user reblock exit routine will be given control when a SAM or PAM data set is being processed during a data set COPY or RESTORE if:

- You specify REBLOCK(dsn [...]) in the COPY statement, and the user 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), or
- The user installation options exit routine forces all data set COPY functions to use the reblock exit routine.

The user 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(\*\*)).

The DFDSS reblock exit routine passes a return code of zero, which allows DFDSS to use an optimal system generated block size.

Your installation can replace the DFDSS exit routine to override the DFDSS-selected block size. If an invalid block size is passed by the exit routine, the data set is not processed and message ADR453E is written to inform you.

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 will load 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 6-word parameter list contains the following information. You can use the ADRRBLKB mapping macro, which is in SYS1.SAMPLIB, 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 the MVS/XA DFP Version 2 DASDCALC function was used to calculate the optimum block size for the target data set.

## How to Write a User Reblock Exit Routine (continued)

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).

Figure 93 is an example of 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     ADRDSSU
  NAME      ADRDSSU(R)
/*
```

Figure 93. Example of Link-Editing a User Reblock Routine

## Appendix B. DFDSS Messages

**Component Name:** ADR

**Program Producing Message:** ADDRSSU

**Audience and Where Produced:**

For programmer: SYSPRINT data set

For operator: Console

**Message Format:**

ADRnnnt (xxx)-mmmm(yy), message text to SYSPRINT

zzADRnnns (xxx)-mmmm(yy), message text to CONSOLE

The following applies to all messages:

zz = Message reply identification (absent if reply not required).

nnn = Message number.

(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 = The identifier (expressed without the ADR prefix) of the module that issued the message.

(yy) = The occurrence indicator of the message. This number allows IBM service personnel to determine where within the module the message was issued.

s = Console Operator type code:

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 has occurred.

t = SYSPRINT message severity code:

I - Informational message to SYSPRINT.

W - Warning message. Task continues, but an error has occurred.

E - Error message. The particular task may end, or may continue without completing all requests.

T - Termination message. DFDSS terminates.

**Comments:**

As a function completes, two values are set: LASTCC and MAXCC. LASTCC is the return code of the completing task and remains in effect until the task following this task completes. Then the LASTCC value is assigned from the subsequent completed task. MAXCC is set whenever a completing task has a return code greater than the previous value of MAXCC.

**General Return Codes:**

For any task, the return code is set equal to the maximum level message issued for that task: error (16 or 8), warning (4), or information (0).

0 = Operation was successful.

4 = Operation completed, but a warning (W) message was issued during processing.

8 = A function did not execute, or began execution but terminated prematurely or an error (E) message was issued during processing.

12 = A terminating (T) message was issued. No functions were executed.

16 = A function executed that left at least one volume or data set in an unusable condition. For example, a DUMP or a full volume RESTORE prematurely terminated.

**Problem Determination:** Refer to the table following the messages for the explanation of the problem determination codes.

**Note to Application Programmers:** Some of the text under "Explanation," "System Action," and "Programmer Response" is oriented toward system programmers, data administrators, or data management personnel. Therefore, contact appropriate personnel in your facility if an explanation is necessary.

**ADR001T**      **mmmmm(yy), OPEN FAILED FOR DDNAME  
ddname**

**Explanation:** The DD statement ddname does not exist or is coded incorrectly. This message is printed if a SYSPRINT DD (or alternative as indicated in the message) was not found or could not be opened.

**System Action:** The job terminates. The return code is 12.

**Programmer Response:** Probable user error. Verify that a SYSPRINT (or acceptable alternative) DD statement is present in the step. Correct the error, and resubmit the job.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 13, and 29.

**ADR002T**      **mmmmm(yy), INSUFFICIENT MAIN STORAGE  
AVAILABLE TO INITIALIZE PROGRAM**

**Explanation:** Not enough main storage was available to perform the initialization functions of DFDSS. Message ADR018I will follow this message to indicate storage needed.

**System Action:** Processing terminates. The return code is 12.

**Programmer Response:** Probable user error. Increase REGION value in the JOB or EXEC statement to at least the value indicated by message ADR018I and retry. Or, if the "SIZE=" parameter was specified, increase the parameter value ensuring it does not exceed the storage available in the REGION value, and is less than or equal to the REGION value.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, and 29.

**ADR004I**      **mmmmm(yy), USER ABEND 0001 WILL BE  
ISSUED ON MESSAGE ADRnnn**

**Explanation:** The ABEND= keyword was encountered in the PARM statement. An abend will be generated at the first occurrence of the indicated message.

**System Action:** Processing continues until the indicated message is encountered. If the message is not encountered, the abend will not occur.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** None.

**ADR005T**      **mmmmm(yy), NO SYSIN FILE**

**Explanation:** A SYSIN DD statement (or acceptable alternative DDNAME) was not present in the job stream.

**System Action:** Processing terminates. The return code is 12.

**Programmer Response:** Probable user error. Make certain a SYSIN DD statement is present, and retry the job.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, and 29.

**ADR006I**      **(xxx)-mmmmm(yy), date time EXECUTION  
{BEGINS | ENDS}**

**Explanation:** This message gives the date and time a particular task begins or ends execution.

**System Action:** None.

**Programmer Response:** None.

**Operator Response:** None.

**ADR007E**      **(xxx)-mmmmm(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 other than one volume.

**System Action:** The task is not executed. 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)-mmmmm(yy), INSUFFICIENT STORAGE  
AVAILABLE TO SCHEDULE TASK**

**Explanation:** Insufficient storage space was available for this function. Message ADR018I will follow this message to indicate the storage for this task.

**System Action:** The task is not executed. Processing continues with the next control statement. The return code is 8.

**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**      **mmmmm(yy), INVALID {LINECNT | SIZE |  
TYPRUN | PAGENO | ABEND | TRACE |  
UTILMSG | XABUFF | RACFLOG} VALUE  
SPECIFIED IN PARM STATEMENT**

**Explanation:** The indicated parameter on the EXEC statement has been 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.
- 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 terminates. The return code is 12.

**Programmer Response:** Make sure that the value indicated 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.

**ADR010I** mmmmm(yy), SIZE VALUE OF size WILL BE USED FOR GETMAIN

**Explanation:** Indicates the value of the SIZE parameter as specified by the requestor. The value does not include the storage acquired by the programs or utilities called by DFDSS.

**System Action:** The storage used by DFDSS will not exceed the indicated value.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** None.

**ADR011I** mmmmm(yy), LINECNT VALUE OF linecount WILL BE USED

**Explanation:** Indicates the value of the LINECNT parameter as specified by the requester.

**System Action:** SYSPRINT output will be printed at the indicated lines per page. If a value of 9999 is specified, no page eject will occur.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** None.

**ADR012I** mmmmm(yy), date time DFDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS rc [FROM:]

**Explanation:** This message is issued upon completion of a DFDSS job step. The highest condition code (MAXCC) set during the job step is printed (see message ADR013I) and returned to the DFDSS invoker in register 15. If the highest condition code is not zero, a list of the tasks that 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.

**ADR013I** (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 execution 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. Else 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*.

**ADR014I** mmmmm(yy), date time ALL PREVIOUSLY SCHEDULED TASKS COMPLETED. {SERIAL | PARALLEL} MODE NOW IN EFFECT

**Explanation:** A switch has been made from serial mode to parallel mode or parallel mode to serial mode (at the request of the user in the control statement or by the installation options exit routine). All previously scheduled tasks completed prior to the mode switch taking place.

**System Action:** All previously scheduled tasks either executing or waiting to execute have completed. Processing continues.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR015I** (xxx)-mmmmm(yy), {MAXIMUM CONDITION CODE | RETURN CODE OF TASK} HAS BEEN RESET FROM rc1 TO rc2

**Explanation:** A SET command has been encountered and has successfully reset a condition code. The text indicates which taskid was affected, what the return code was prior to resetting, and what the return code was reset to.

**System Action:** None.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR016I** (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 user installation options exit has 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

**Explanation:** Because of a previous error, the requested task was not scheduled. The previous message associated with the same taskid indicates the error.

**System Action:** The task is not executed. 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.



**ADR018I** [(xxx)-mmmm(yy), STORAGE REQUIRED  
nK. STORAGE AVAILABLE nK.

**Explanation:** A task was not executed or the job did not start because of lack of storage. This message will follow either message ADR002T or ADR008E and will indicate 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:** The return code and action taken are indicated by message ADR002T or ADR008E.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** None.

**ADR019E** (xxx)-mmmm(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 executed. 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)-mmmm(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 executed. 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.

**ADR022T** 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 terminates. 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)-mmmm(yy), TRACKS/OUTTRACKS  
VALUE IS INVALID FOR DEVICE. VALID  
CYLINDER RANGE IS 0 TO cccc. VALID  
TRACK RANGE IS 0 TO hhhh

**Explanation:** Either the range specified in the TRACKS keyword of a DUMP, COPY, RESTORE, or PRINT, or the range specified for the OUTTRACKS keyword for COPY or RESTORE is invalid for the specified device type. 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.

**System Action:** The task terminates. Processing continues with the next control statement. The return code is 8.

**Programmer Response:** Correct the cause of the error, and rerun the job.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR025E** (xxx)-mmmm(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, input must be DASD. For a RESTORE, output must be DASD. For a COPY, input and output must both be DASD. For a PRINT, input must be DASD. For a DEFRAg, the volume must be DASD.

**System Action:** The task is not executed. Processing continues with the next control statement. The return code is 8.

**Programmer Response:** Correct the cause of the error, and reissue the command.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR026E** mmmmm(yy), {volser | ddname} IS INVALID AS  
OBJECT DDNAME

**Explanation:** A reference was made to a ddname that was incorrect. The following restrictions apply to DD references from the control statements:

- SYSPRINT (or its alternative) can only be referred to from the PRINT command.
- SYSIN (or its alternative) cannot be referred to at all.
- The indicated ddname has been specified more than once for the OUTDDNAME parameter on a DUMP command.
- The indicated volser has been specified more than once for the INDYNAM parameter on a DUMP command.
- The volser associated with the indicated ddname has been specified more than once for the INDDNAME or OUTDDNAME parameter on a COPY command.

**System Action:** The task terminates. 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 mmmmm(yy), DFDSS IS NOT SUPPORTED BY THE CURRENT SCP**

**Explanation:** DFDSS only executes on MVS/370 with DFP and MVS/XA with DFP. The system identification in the nucleus is not one of these.

**System Action:** The job terminates. 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 mmmmm(yy), TYPRUN=SCAN REQUESTED. ONLY CONTROL CARD SYNTAX CHECKING WILL BE DONE**

**Explanation:** No tasks will be executed. 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 mmmmm(yy), KEYWORD PRECEDING POSITION nn IN PARM STATEMENT IS INVALID**

**Explanation:** The keyword preceding the "=" sign in the indicated position of the parameter of the PARM keyword of the EXEC statement is not recognized as a valid keyword option.

**System Action:** The job terminates. The return code is 12.

**Programmer Response:** Correct the error in the PARM statement, and rerun the job. The valid parameter keywords are those shown in "How You Control DFDSS via PARM Information in the EXEC Statement" on page 19.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, and 29.

**ADR030W mmmmm(yy), DCB VALUES HAVE BEEN MODIFIED FOR ddname**

**Explanation:** The DCB information you specified had either the LRECL or BLKSIZE value outside the allowable range. The LRECL and 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 mmmmm(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 DEFRAG and data set DUMP and RESTORE will bypass execution. Data set DUMP and RESTORE will only perform data set filtering and will not perform DUMP or RESTORE functions. DEFRAG will print the fragmentation index but will not

relocate data set extents. This message will be followed by ADR040I message for the tasks that are scheduled.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR033W (xxx)-mmmmm(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 may appear more than once as the OPTIMIZE value is decreased.

**System Action:** The function will execute 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)-mmmmm(yy), INSTALLATION EXIT ALTERED keyword {DEFAULT | REQUEST | OPTION | VALUE} [TO xxxxx]**

**Explanation:** The installation options exit routine has changed the default or requested option or value to that indicated in the message. See "How to Write a User Installation Options Exit Routine" on page 237 for details.

**System Action:** The DFDSS function will run with the altered options.

**Programmer Response:** Check with your system programmer for the user exit routine operation to get the desired result.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR036I (xxx)-mmmmm(yy), TASK NOT SCHEDULED AT REQUEST OF INSTALLATION EXIT**

**Explanation:** The installation exit has requested that this task not be executed.

**System Action:** The task is not executed. Processing continues with the next control statement. The return code is 8.

**Programmer Response:** Check with your system programmer for restrictions placed by the user exit.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR037E mmmmm(yy), FAILURE WHILE ATTACHING TASK**

**Explanation:** The system attach macro returned a nonzero return code while attaching a task. There was probably not enough storage to attach the task.

**System Action:** Execution is attempted on other tasks. Final return code is set to 8.

**Programmer Response:** Increase the region size, specify a smaller size parameter, and 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 has determined from the CVAF table that no CVAF PARTREL macro support is installed on your system.

**System Action:** The task is not executed. Processing continues with the next control statement. The return code is 8.

**Programmer Response:** None.

**Operator Response:** 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 is terminated.
- COPY DATA SET—Filtering is performed and the selected data sets are printed; data sets are not copied.
- COPYDUMP—Task is terminated.
- DEFRAG—Only volume statistics are printed; data set extents are not relocated.
- DUMP FULL/TRACKS—Task is terminated.
- DUMP DATA SET—Filtering is performed and the selected data sets are printed; data sets are not dumped.
- PRINT—Task is terminated.
- RESTORE FULL/TRACKS—Task is terminated.
- RESTORE DATA SET—Filtering is performed and the selected data sets are printed; data sets are not restored.
- COMPRESS and RELEASE—Filtering is performed and the selected data sets are printed; data sets are not restored.

**System Action:** Further processing is bypassed for the volume/task.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR041I mmmmm(yy), I/O BUFFER {ABOVE | BELOW} 16MB VIRTUAL REQUESTED**

**Explanation:** You requested that the I/O buffers used by DFDSS are to reside above or below 16 megabytes virtual address.

**System Action:** An attempt is made to allocate storage for the buffers above 16 megabytes virtual address, if you requested that.

**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 mmmmm(yy), TASKID xxx HAS BEEN ASSIGNED TO COMMAND 'yyyyyyyy'**

**Explanation:** The indicated taskid has been assigned to the command indicated. 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:** The IF-THEN-ELSE command sequence that was specified caused the command to be bypassed. When specifying an IF-THEN-ELSE command sequence, either the THEN or the ELSE clause will be executed, and the one not executed will be bypassed.

**System Action:** The bypassed portion of the command sequence is checked for syntax errors but is not executed. 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 no IF-THEN-ELSE is in effect will cause no special action to be taken by DFDSS. However, it must have a matching END command. DFDSS processing continues. The return code is 4.

**Programmer Response:** Remove the unnecessary command to prevent the message from occurring.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR104E (xxx)-mmmm(yy), AN IMPROPERLY  
PLACED COMMA HAS BEEN  
ENCOUNTERED**

**Explanation:** An improperly placed comma has been found when scanning the command. The omission of positional parameters cannot be indicated by commas. The omission of leading positional parameters is not permitted.

**System Action:** The command is not executed. 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)-mmmm(yy), COMMAND-END  
DELIMITER APPEARS WITHIN  
APOSTROPHES**

**Explanation:** The optional command delimiter (a semicolon) was found within a quoted string. A closing single quotation mark may have been omitted.

**System Action:** The usage is accepted, and the semicolon is treated as a valid character instead of as 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)-mmmm(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.

**ADR107W (xxx)-mmmm(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.

**System Action:** The omission is accepted. DFDSS processing of the command continues. The return code is 4.

**Programmer Response:** Correct the command syntax.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR108I (xxx)-mmmm(yy), date time EOJ  
ENCOUNTERED. ALL TASKS WILL  
QUIESCE**

**Explanation:** An EOJ command has been encountered. A subsequent message ADR013I will indicate when all tasks have completed.

**System Action:** DFDSS terminates after executing tasks before the EOJ command.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR109I mmm(yy), INITIAL SCAN OF USER  
CONTROL STATEMENTS COMPLETED.**

**Explanation:** All control statements have been 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)-mmmm(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)-mmmm(yy), TOO MANY POSITIONAL  
PARAMETERS AFTER 'xxxx'**

**Explanation:** A parameter list has too many positional parameters specified following the characters xxxx.

**System Action:** The task is not executed. 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)-mmmm(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 executed. 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 executed. 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 naming conventions required. For example, a ddname contains invalid characters. Also, check the password data set for syntax errors.

**System Action:** The task is not executed. 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)-mmmmm(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 may be improperly positioned, or a positional parameter may be missing.

**System Action:** The task is not executed. 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)-mmmmm(yy), LEFT PARENTHESIS MISSING FOLLOWING KEYWORD 'xxxx'**

**Explanation:** The keyword xxxx is not followed by an opening parenthesis, which should begin the required subparameter list or value associated with the command keyword.

**System Action:** The task is not executed. 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)-mmmmm(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 executed. 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)-mmmmm(yy), INVALID PARENTHESIS FOR SPECIFYING REPEATED SUBPARAMETER LIST**

**Explanation:** Parentheses that are used for delimiting repeated subparameter lists are missing or not matched.

**System Action:** The task is not executed. 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)-mmmmm(yy), INVALID LEFT PARENTHESIS AFTER 'xxxx'**

**Explanation:** An opening parenthesis was found that appeared to delimit the positional parameter xxxx, but the positional parameter was not a constant or a list of constants.

**System Action:** The task is not executed. 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)-mmmmm(yy), KEYWORD 'xxxx' IS IMPROPER**

**Explanation:** A misspelled, improperly specified, or inapplicable keyword was encountered in the command.

**System Action:** The task is not executed. 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 has appeared too often in the command. A parameter list may be incorrectly specified.

**System Action:** The task is not executed. 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 was terminated. Messages printed just prior to this message indicate the nature of the error.

**System Action:** Processing continues with the next control statement. If this task was to be executed, 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 'xxxxxxxx'**

**Explanation:** A password, denoted by a slash (/), was encountered where a password was not allowed, a password was expected and was missing, or a password was specified that was longer than 8 characters. This is not a verification of the password but an error in the syntax.

**System Action:** The task is not executed. 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 executed. 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)-mmmm(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 executed. 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)-mmmm(yy), IMPROPER NUMERIC DIGIT FOUND IN 'xxxx'**

**Explanation:** An invalid character was encountered in the constant xxxx. A decimal number can only be specified with the symbols 0 through 9; a hexadecimal number can only be specified with the symbols 0 through 9 and A through F; and a binary number can only be specified with the symbols 0 and 1.

**System Action:** The task is not executed. 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)-mmmm(yy), CONSTANT 'xxxxxxxx' 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 executed. 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)-mmmm(yy), TOO MANY CONSTANTS/ENTRIES IN LIST BEGINNING AT 'xxxx'**

**Explanation:** Too many constants or entries were specified in the command beginning at xxxx.

**System Action:** The command is not executed. 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, identified by xxxx, is missing.

**System Action:** The task is not executed. Processing continues with the next control statement. The return code is 8.

**Programmer Response:** Add the missing parameter, and reissue the command.

**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 that are defined as mutually exclusive. If one parameter is specified, the other parameter is not allowed.

**System Action:** The task is not executed. 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 executed. 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 improperly specified. The length may have been greater than 44 characters, or, if the data set name was not in quotation marks, a qualifier may have been longer than 8 characters, or GDG may have been improperly specified.

**System Action:** The task is not executed. 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 has been ignored for one of the following reasons.

**LOGICALVOLUME**—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 ddn**

**Explanation:** Invalid data exists in the data set associated with the DD statement specified by the FILTERDD keyword.

**System Action:** The task terminates. 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 while scanning the character string contained in the ddname specified as a PASSWORD, FILTERDD, or EXCLUDE parameter. This could be caused by incorrect continuation syntax or by missing records in the input stream.

**System Action:** The task is not executed. 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.

**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 executed. Processing continues with the next control statement. The return code is 8.

**Programmer Response:** Increase REGION size and/or change the SIZE parameter, and retry the job.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR169T mmmmm(yy), NO RECORDS IN SYSIN  
STREAM**

**Explanation:** The SYSIN stream was empty.  
**System Action:** The job terminates. The return code is 12.  
**Programmer Response:** Ensure that the SYSIN (or acceptable alternative ddname) DD statement references a nonempty file, and retry the job.  
**Operator Response:** None.  
**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR170T mmmmm(yy), ERROR OPENING DDNAME  
ddname**

**Explanation:** The SYSIN (or acceptable alternative ddname) file could not be opened.  
**System Action:** The job terminates. 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 mmmmm(yy), MAXIMUM NUMBER OF TASKS  
EXCEEDED**

**Explanation:** More than 255 tasks have been encountered in the SYSIN stream.  
**System Action:** The job terminates. The return code is 12.  
**Programmer Response:** Run multiple job steps if necessary to accommodate the number of tasks requested.  
**Operator Response:** None.  
**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR172T mmmmm(yy), ERROR DURING INITIAL SCAN**

**Explanation:** During the initial scan of the control statements, an error was encountered. A previous message will indicate the error.  
**System Action:** The job terminates. 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 mmmmm(yy), I/O ERROR WHILE READING  
ddname**

**Explanation:** An I/O error was encountered reading the SYSIN file (or acceptable alternative ddname).  
**System Action:** The job terminates. The return code is 12.  
**Programmer Response:** 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 mmmmm(yy), REMAINDER OF COMMAND  
INPUT STREAM WAS IGNORED**

**Explanation:** An error has occurred that prohibits further scanning of the command stream. Messages printed just prior to this message indicate the nature of the error.  
**System Action:** The program terminates. 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 mmmmm(yy), COMMAND 'xxxxxxx' IS  
UNKNOWN**

**Explanation:** The command name is not recognized.  
**System Action:** The program terminates. 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 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 terminates. 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 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 terminates. This message will be followed by message ADR172T. 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)-mmmm(yy), AN 'IF' COMMAND IS  
INVALID**

**Explanation:** The IF-THEN-ELSE statement sequence 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 terminates. This message will be followed by message ADR172T. 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)-mmmm(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)-mmmm(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)-mmmm(yy), TOO MANY LEVELS OF 'IF' COMMAND NESTING**

**Explanation:** Either more than 10 IF statements have been 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)-mmmm(yy), INPUT STREAM END-OF-FILE FOUND BEFORE END OF COMMAND**

**Explanation:** An end-of-file condition was encountered while scanning the command. This could be caused by incorrect command continuation syntax or by missing records in the input stream.

**System Action:** The job terminates. 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)-mmmm(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)-mmmm(yy), VTOC IN ERROR PRIOR TO DEFRAG PROCESSING. TASK IS TERMINATED**

**Explanation:** An error occurred while processing the VTOC. The error condition may have been 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 relocating extents on an indexed volume, a recovery environment is set up by DFDSS. If DEFRAG terminates 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 VTOC DSCBs.
- A failure occurred during LSPACE(SVC 78) processing.
- An incompatible VTOC structure; this message is accompanied by message ADR210E. More than one DSCB extent field, either data set or free space, point to the same location on the volume.
- A prior DEFRAG run or a DADSM function terminated before updating the VTOC free space maps.

**System Action:** The DEFRAG function is terminated. The return code is 8.

**Programmer Response:** Rerun DEFRAG function after doing the following:

- If I/O error messages accompany this message, take corrective action recommended by your installation 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 will invoke DADSM modules to rebuild the free space map or 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 format-4 DSCB in the VTOC. 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 while processing an extent of the data set for a DEFrag operation.

**System Action:** That 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 by your installation for the error.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 29.

**Programmer Response:** If the DOS bit is on, allocate a temporary data set on the volume to allow DADSM to rebuild the format-5 chain, 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), BEGINNING STATISTICS ON volser:**

**FREE CYLINDERS** ccc  
**FREE TRACKS** ttt  
**FREE EXTENTS** xxx  
**LARGEST FREE EXTENT (ccc,ttt)**  
**PERCENT FREE** pp  
**FRAGMENTATION INDEX .iii**

**Explanation:** Issued by the DEFrag function at the beginning of the task. 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 ccc and ttt are the cylinder and track values respectively, in decimal. The xxx indicates the number of free extents on the volume, and (ccc,ttt) 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 .iii is the fragmentation index; this index can be a value between 0.000 and 1.000, with a higher value indicating 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.

**ADR205I (xxx)-mmmmm(yy), EXTENT nnn OF DATA SET dsname IS ALLOCATED BY CYLINDER, BUT IT IS NOT ON A CYLINDER BOUNDARY**

**Explanation:** The format-1 DSCB indicates that the data set is either allocated by cylinder (DS1DSPAC=B'11') or has the round bit set to 1 (DS1ROUND=B'1'), but 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 successful, the cylinder allocation indicator in the first byte of the DSCB extent descriptor remains on (for example, if nnn=001, the first byte of DS1EXT1 remains B'1xxxxxx'). If unsuccessful, the extent is relocated as though it had been originally allocated by tracks, and the cylinder allocation indicator in the first byte of the DSCB extent descriptor is turned off. If this message is issued because the round bit is set, the round bit is not turned off. In either case, the message may 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.

**ADR209I mmmmm(yy), MOVED EXTENT nnn FROM cc:hh-cc:hh TO cc:hh-cc:hh FOR dsname**

**Explanation:** The nnth extent of data set dsname has been 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.

**ADR206E (xxx)-mmmmm(yy), I/O ERROR OCCURRED WHILE READING VTOC DURING DEFrag. TASK IS TERMINATED**

**Explanation:** An I/O error occurred while reading the VTOC.

**System Action:** The DEFrag is terminated. 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.

**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 name(s) and/or FREE SPACE(s) overlap on the volume according to the VTOC. The VTOC is in error.

**System Action:** The DEFrag function is terminated. The return code is 8.

**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.

**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 will appear if the setting or resetting could not be done.

**System Action:** DEFrag is terminated. The return code is 8.

**ADR211I** (xxx)-mmmmm(yy), DATA SET dsname IN USE AND WAS NOT MOVED

**Explanation:** The data set was enqueued or allocated by another job 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:

**Explanation:** The distribution map gives an indication of the free-space fragmentation before and after the run in addition to the distribution of data set extents by size. In the map that follows this message, the number of free-space extents of each given size in tracks is listed, as are the numbers of output free-space and data-set extents. A cumulative percentage of occupancy is also provided for each of the three categories.

**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:

EXTENTS RELOCATED ccc  
TRACKS RELOCATED ttt  
FREE CYLINDERS ccc  
FREE TRACKS ttt  
FREE EXTENTS xxx  
LARGEST FREE EXTENT (ccc,ttt)  
FRAGMENTATION INDEX .iii  
\*\*\* = NON-STANDARD O/S VOLUME

**Explanation:** Printed at the end of the DEFRAG run, this message indicates the numbers of data set extents and tracks that were relocated. The description of the other values is given in message ADR208I. 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 volume has the DIRF bit off in the format-4 DSCB.

**Operator Response:** None.

**Problem Determination:** None.

**ADR220I** mmmmm(yy), INTERVAL BEGINS AT CC:HH cc:hh AND ENDS AT CC:HH cc:hh

**Explanation:** An area of DASD has been selected from which a data set extent may be relocated. The interval begins at the first cchh and ends at the second cchh.

**System Action:** None.

**Programmer Response:** None.

**Operator Response:** None.

**ADR226W** (xxx)-mmmmm(yy), CCHH RANGE cchh1 TO cchh2 IS UNACCOUNTED FOR IN VTOC. IT WILL BE IGNORED.

**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 *Common VTOC Access Facility Diagnosis Reference* for an explanation of the error codes.

**System Action:** The function is terminated. The return code is 8.

**Programmer Response:** If the CVAF function-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-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 that either the volume was not sufficiently fragmented to process the function, there was insufficient free space to process the function, or there was a free extent large enough to meet the MAXMOVE specification. If there is fragmented free space on the volume, DEFRAG could not relocate extents with its algorithms to further reduce the fragmentation. This message also appears if all or a significant number of data sets are excluded (or not selected).

**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 space that was not defined either as 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 format-5 chain will be rebuilt, reclaiming the unaccounted-for 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 exceeded the number and was adjusted.

**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, will default 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 or data set RESTORE. The operation codes (x) can be:

- 0 - Insert VVR
- 1 - Get a VVR for update
- 2 - Put a VVR for update
- 3 - Generic read of components of a cluster
- 4 - Read a VVR

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. This message is accompanied by other messages identifying the data set affected. 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 *System Messages*. Table I, items 1, 2, 3, 4, 13, 20, and 29.

**ADR232E (xxx)-mmmmm(yy), VVDS DOES NOT MATCH VTOC FOR COMPONENT componentname, CLUSTER clustername ON VOLUME volser**

**Explanation:** DEFrag detected a mismatch of the extents between the VTOC and the VVDS for the component. A VSAM EXTEND operation on the component might have been interrupted because of a program or system failure.

**System Action:** The component is not relocated. Final return code is set to 8.

**Programmer Response:** Run Access Method Services DIAGNOSE for the component and fix the problem, using the procedure recommended in *Access Method Services Reference*.

**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 as much as was indicated in the user input for the volume.

**System Action:** DEFrag is terminated. The return code is 4.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**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, no protected data set relocation can 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 (freeing up a DSCB), and rerun the DEFrag job to relocate protected data sets. If not, DUMP a data set specifying DELETE, to free a DSCB, rerun DEFrag to process protected data sets, then RESTORE the data set previously dumped.

**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 could not find enough format-5 or free DSCBs to rebuild the format-5 chain 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.

**ADR301E** (xxx)-mmmm(yy), AN ERROR WAS ENCOUNTERED WHILE FILTERING DATA SETS ON VOLUME volser. VOLUME WAS NOT PROCESSED.

**Explanation:** I/O or logical errors were encountered while accessing the VTOC or VVDS on the specified volume during a data set DUMP or COPY 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 installation'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 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.

**ADR302E** (xxx)-mmmm(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 terminates. 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 you get this message when the device types are the same, see Table I, items 29 and 35.

**ADR303E** (xxx)-mmmm(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 terminates. Processing continues with the next control statement. The return code is 8.

**Programmer Response:** Probable user error. Correct the control statement in error, and rerun the job.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, and 13. If you get this message when the system residence volume was not specified as the output volume, see Table I, items 29 and 35.

**ADR304E** (xxx)-mmmm(yy), ERROR ENCOUNTERED WHILE ACCESSING {VTOC | VVDS} IMAGE OF VOLUME volser ON DDNAME ddname, xxxxx

**Explanation:** An error (logical or I/O) was encountered while analyzing the image of the VTOC or VVDS on the dumped tape during data set RESTORE processing. The types of possible logical errors might be: not a VTOC or VVDS record, invalid format of the VVDS, no data on track,

I/O errors were encountered on the track during the DUMP, end of file during the VTOC or VVDS search, etc. If physical I/O errors were encountered, this message will be preceded by other I/O error messages. 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)-mmmm(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/DEFrag operation, or a TRACKS or VTOC PRINT operation even after a specified or defaulted number of retries. If the resource is VTOC, some DADSM functions may have been executing while DFDSS was trying to serialize the use of the DASD volume. If the resource is VVDS, some catalog or VSAM functions might have been accessing the VVDS.

**System Action:** For all operations except DUMP/RESTORE, the function is terminated with a return code of 8. Processing continues with the next control statement. For a data set DUMP/RESTORE operation, that volume is bypassed; a return code of 8 is set, but processing continues on other volumes, if any.

**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 being allocated (or accessed) to the volume. Change the WAIT parameters to increase the WAIT seconds and the retry count, and rerun the job or resubmit the job when there is less system activity on the volume.

**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)-mmmm(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 the volume is in use by another job. Possibly a data set on the volume is being accessed by another job. This message may 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 specified volume should have an operator-initiated MOUNT command issued for the volume with the USE=PRIVATE subparameter before the related job step begins processing. 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 for the volume with the USE=PRIVATE subparameter to reduce the possibility of other jobs being allocated (or accessing) the volume. In the case of data set COPY, remove the INDD keyword and rerun 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.

**ADR307E (xxx)-mmmmm(yy), UNABLE TO OPEN VOLUME volser, xx-yyy**

**Explanation:** DFDSS was 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 format-4 DSCB.
- 6 = Format 4 DSCB not the first record on track.
- 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 terminates. Processing continues with the next control statement. The return code is 8.

**Programmer Response:** If the reason code is 6: DFDSS does not support volumes whose format-4 DSCB is not the first record on the track.

**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:** The user was unable to obtain the necessary RACF or password access to the volume during a data set DUMP/COPY/PRINT/RESTORE operation, during a COMPRESS or RELEASE operation, or to a data set on the volume for a full or tracks DUMP/RESTORE/COPY/PRINT operation; the installation exit routine denied access to the volume; the operator denied access to a checkpoint/restart, RACF-protected or system data set on the volume; or I/O errors were encountered while checking authorization of the volume and the operator denied further processing.

**System Action:** The task terminates 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 authority to access, 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 xxxx**

**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 terminates. 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 VM/370 minidisk or whether the source volume is a 3330 model 11 and the target volume is a 3330 model 1, or whether the source volume is a 70-megabyte 3340/3344 and the target volume is a 35-megabyte 3340. This is not allowed on a full volume operation, or if the range falls outside the capacity of the volume on a tracks operation. Select the proper 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 will be printed for the following reasons:

- A volume without an indexed VTOC indicates no valid format-5 DSCB(s) exists.
- On a volume without an indexed VTOC, a nonzero return code was received from OBTAIN because of an I/O error, 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 DOS bit is ON, on a volume without an indexed VTOC, or the DIRF bit is ON.

**System Action:** The entire volume, including unallocated space, will be dumped or copied. The return code is 4.

**Programmer Response:** For an I/O error, take corrective action recommended by your installation. If the DOS bit is on, for a volume without an indexed VTOC, allocate a temporary data set to let DADSM rebuild the format-5 chain.

**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:** A TRACKS COPY, or RESTORE was attempted on track 0, VTOC, VVDS, or VTOC index data set without specifying the PURGE parameter.

**System Action:** The task terminates. 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)-mmmm(yy), DURING A TRACKS OPERATION, THE VTOC OR TRACK 0 WAS OVERLAID**

**Explanation:** A TRACKS COPY or RESTORE was attempted on the VTOC or track 0, but 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)-mmmm(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 and it was denied.

**System Action:** The task terminates. 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)-mmmm(yy), ERROR ENCOUNTERED WHILE ACCESSING {VTOC | VVDS} ON volser. PROCESSING CONTINUES**

**Explanation:** An I/O error was encountered while 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 will continue on input errors and will terminate on output errors. If CANCELERROR is specified, processing terminates 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 by your installation 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)-mmmm(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 channel command, or I/O errors were encountered while testing whether the Read Multiple Count, Key, and Data channel command is supported on the device. OPTIMIZE(2), OPTIMIZE(3), and OPTIMIZE(4) are applicable only if this channel command is supported. This may also occur if the device can be accessed via multiple control units, some of which do and some of which do not support the channel command.

**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 channel command.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 29.

**ADR316W (xxx)-mmmm(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. 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 or VTOC, do a TRACKS PRINT of the VVDS or VTOC 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 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 DEV TYP 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 may occur if it is a VM/370 minivolume) and the device size field in the VTOC is less than the one passed by DEV TYP, the value in the VTOC is used.

**System Action:** The device size nnnn (in hexadecimal) will be used.

**Programmer Response:** If it is not a minivolume and the nnnn value is incorrect, the DFDSS function has completed



in error. Rerun the job only after running AMASPZAP to correct the device size field in the VTOC to reflect a full volume. If it is a minivolume, action taken 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 nonoptimize channel program chain because track overflow records were encountered while 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 that are not marked as track overflow data sets in the VTOC should 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 DS1LSTAR may 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)-mmmmm(yy), VOLUME SERIAL NOT CHANGED ON VOLUME volser**

**Explanation:** An I/O error was encountered while trying to read track 0 from the target volume at the end of the COPY or RESTORE. This message is preceded by message ADR348E. The rest of the tracks were successfully copied or restored.

**System Action:** Processing terminates. 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 assign an alternate track. Rerun the job.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 30.

**ADR320I mmmmm(yy), VOLUME SERIAL xxxxxx ON UNIT ccu IS CHANGED TO yyyyyy**

**Explanation:** Volume serial was changed on 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)-mmmmm(yy), DATA SET dsname NOT ON VOLUME volser**

**Explanation:** The requested data set is not on the volume.

**System Action:** The task terminates. Processing continues with the next control statement. 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.

**ADR323E (xxx)-mmmmm(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 terminates. 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)-mmmmm(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. This message is preceded by other messages indicating the cause of the error.

**System Action:** For RESTORE and COPY, the current task is terminated 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 user installation options exit does not force the cancelation of all output upon the occurrence of a single error. If the installation options exit forces such a cancelation, the task is terminated 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 by your installation for such errors and rerun the job. If the error was on the output DASD during a COPY or RESTORE, a rerun may 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 terminates. 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.

**ADR326W** (xxx)-mmmm(yy), FAILURE IN MSS ACQUIRE. POST CODE IS zzzz. PROCESSING CONTINUES.

**Explanation:** An ACQUIRE for MSS failed for the reason indicated by zzzz.

**System Action:** Processing continues without the ACQUIRE. Performance during the task will be degraded, but this condition is not associated with the output of the task. The return code is 4.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 5b, 13, 29, and 35.

**ADR327W** (xxx)-mmmm(yy), NO DATA SETS PROCESSED FROM {VOLUME volser | LOGICAL VOLUME volser}

**Explanation:** A data set DUMP, COPY, or RESTORE did not select any data sets on the indicated volume that matched the filtering criteria, or none of the data sets selected could be processed because of failure to ENQ or allocate, or 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)-mmmm(yy), ERROR DURING DATA SET SELECTION. VOLUME volser IS BYPASSED

**Explanation:** A data set DUMP, COPY, 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 message(s) associated with this taskid.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 29.

**ADR329I** (xxx)-mmmm(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)-mmmm(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 under 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.

**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)-mmmm(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.

**System Action:** The task terminates. 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)-mmmm(yy), DATA SET CHANGE INDICATOR NOT RESET FOR {dsname | clustname componentname} ON volser, {I/O ERROR | OBTAIN ERROR, rc}

**Explanation:** The data set change indicator could not be reset in the format-1 DSCB for the data set in the VTOC on the indicated volume. Either DADSM OBTAIN failed or I/O error occurred while writing the DSCB. 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 again, or reset the indicator using AMASPZAP.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 29.

**ADR335W** (xxx)-mmmm(yy), DATA SET CHANGE INDICATOR NOT RESET FOR ANY DATA SET ON A VTOC TRACK ON VOLUME volser

**Explanation:** Input I/O error encountered on a VTOC track when the data set change indicator was being reset in the format-1 DSCBs.

**System Action:** The data set change indicator is not reset in any of the format-1 DSCBs 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 format-1 DSCBs.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, 25b, and 29.

**ADR336I** (xxx)-mmmm(yy), LOGICAL ERROR WHILE READING VOLUME volser, TRACK cchh

**Explanation:** An unexpected error was encountered during a read operation that used the Read Multiple Count, Key, and Data 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 may be in error or DFDSS may 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 format-1 DSCBs in the VTOC.

**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, and 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 in the format-1 DSCB. 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 resetting the data set change indicator during a full DUMP. The output from the DUMP is still usable. This message is preceded by an I/O error message indicating the nature of the error.

**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 ADR333W. Otherwise, restore the volume.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, 25a, 25b, and 30.

**ADR342I** (xxx)-mmmm(yy), DATA SET CHANGE INDICATOR HAS BEEN RESET FOR ALL QUALIFIED DATA SETS ON VOLUME volser

**Explanation:** The data set change indicator was reset during a full volume DUMP for all format-1 DSCBs in the VTOC that had the indicator on.

**System Action:** None.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 29.

**ADR344I** mmmmm(yy), VOLSER ON UCB ccu 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. This message is preceded by ADR320I.

**Programmer Response:** None.

**Operator Response:** To make the new volume available to the system, the old volume that has this volume serial should be unloaded or varied offline before making the new volume available to the system.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 29.

**ADR345D** REQUEST AUTHORIZATION TO OVERLAY {TRACK ZERO | VTOC | VTOC INDEX | VVDS} ON UCB uch, 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.

**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"

terminates the operation, and processing continues with the next function.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 29.

**ADR346I** (xxx)-mmmmm(yy), VTOC INDEX WILL BE DISABLED ON VOLUME volser

**Explanation:** The user installation exit has 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 may be executed separately from DFDSS.

**Operator Response:** None.

**Problem Determination:** None.

**ADR347E** (xxx)-mmmmm(yy), PERMANENT I/O ERROR ON DDNAME ddname,xxxxxxxxxxxx

**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 is terminated 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 is terminated 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:** A permanent I/O error was encountered on the DASD device. The cchh, opcode, csw, and sense are printed in hexadecimal digits.

**System Action:** This message will be followed by another message explaining the termination action, if any. The return code is 8.

**Programmer Response:** Take appropriate action recommended by your installation 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 (DS4DOCVT) in the format-4 DSCB for the indicated volume should be set on. The format-5 DSCB (free space) should be accurate.

**Programmer Response:** Use the IEHLIST utility program to determine the VTOC accuracy. If there were other DFDSS or allocation error messages, you may need to execute 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:** I/O errors were encountered while reading the VVDS tracks or logical errors were encountered while scanning the VVDS for authorization checking on a TRACKS or FULL volume operation or during selection of data sets during a data set DUMP or RESTORE.

**System Action:** If it is during a FULL volume or TRACKS operation during authorization checking, this message is accompanied by other messages. If it is during data set selection, the volume is bypassed. The return code is 8.

**Programmer Response:** If it is a logical error during DUMP, run Access Method Services DIAGNOSE on the volume to determine the problem and correct it using the procedure recommended by *Access Method Services Reference*. If it is a logical or physical error during RESTORE, use a previous backup copy to restore the desired VSAM data sets. If it is a physical I/O error during DUMP, take the appropriate action recommended by your installation 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 may have been created in a DUMP that abnormally terminated.

**System Action:** The task terminates. 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 yyy, THAT HAD VSAM DATA SETS, HAS BEEN CHANGED FROM volser1 TO volser2 DURING A TRACKS OPERATION

**Explanation:** The output volume has VSAM data sets. The VOLID has been changed on the volume. These data sets may not be accessible because of the change in the VOLID if the tracks mapping these data sets have not been copied or restored.

**System Action:** The VOLID has been 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 the list of data set 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 processing the VVDS or VTOC.

**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:** This is accompanied by other error messages. Take the appropriate action and rerun the job if you want to do a data set RESTORE from the DUMP output.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 29.

**ADR355W** (xxx)-mmmmm(yy), CHANGE VOLID ON UNIT xxx 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 was 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, change the volume serial of the receiving volume with Device Support Facilities. 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, nn, has terminated 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.

**ADR358W** (xxx)-mmmmm(yy), PROPER LEVEL levelnum OF productname NOT ON SYSTEM. {COMMAND | VOLUME volser | DATA SET dsname} WILL BE PROCESSED, rc.

**Explanation:** The proper level of the named licensed program is not installed. The reason code, rc, indicates:

- 1 VSAM linear data set support requires MVS/XA Data Facility Product V2.3.0. The target data set is allocated as ESDS.
- 2 Locked user catalog support requires MVS/XA Data Facility Product V2.3.0. The LOCK status of the user catalog is ignored.

**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.

**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.

**ADR362T** (xxx)-mmmmm(yy), NO BUFFER AVAILABLE & NONE IN WRITE

**Explanation:** The DFDSS I/O modules could not find a free buffer and there were outstanding requests for waits or checks.

**System Action:** The job terminates. 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.

**ADR363W** (xxx)-mmmmm(yy), SOURCE DATA SET dsname {IS EMPTY | HAS AN UNSUPPORTED DSORG}. SOURCE NOT COPIED TO PREALLOCATED TARGET DATA SET

**Explanation:** The source data set is either empty or has an unsupported DSORG, and the target data set selected resides on an unlike device. The source data cannot be copied to the target data set.

**System Action:** Processing continues. The return code is 4.

**Programmer Response:** To copy the data set, rerun the job with the changes specified:

- For an empty SAM, PAM, or EXCP data set, use ALLDATA or ALLEXCP to force the copy of all allocated source space.
- For an empty VSAM or non-VSAM data set type not affected by ALLDATA and ALLEXCP, the source allocated space cannot be copied to the target with COPY DATASET.

- For a source data set with unsupported DSORG, define a preallocated target data set on a same or like device.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR364W** (xxx)-mmmm(yy), TRACK CANNOT BE RESTORED DUE TO DATA CHECK DURING DUMP. CCHH ccchhhh, 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)-mmmm(yy), THE CATALOG newcatname SPECIFIED IN RECATALOG IS NOT AVAILABLE. THE {COPY | RESTORE} TASK IS TERMINATED

**Explanation:** The catalog specified in the RECATALOG parameter does not exist or is not available on the system.

**System Action:** Execution of the task is terminated. The return code is 8.

**Programmer Response:** Ensure the catalog name specified in RECATALOG is correct and available, and rerun the job.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR366W** mmmm(yy), TRACK ccchhhh WRITTEN TO VOLUME volser WITH INVALID TRACK FORMAT

**Explanation:** The track identified by the ccchhhh had invalid track format on the source volume during DUMP or COPY and has been restored or copied with the same condition to the target volume indicated. When this track is accessed on the target volume, the invalid track format condition will recur.

**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** mmmm(yy), {R0 COUNT FIELD | INVALID TRACK FORMAT} ERROR ON {INPUT | OUTPUT} VOLUME volser ccchhhh opcode csw sense

**Explanation:** Either the ccchhhh 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 not specified, processing continues and a return code of 8 is set; otherwise, the function is terminated with a return code of 16.

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.

**Programmer Response:** If the function is terminated 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 track in error may be reconstructed after the RESTORE.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 29.

**ADR368E** (xxx)-mmmm(yy), I/O ERROR ON DDNAME ddname WHILE {PROCESSING VTOC | ADVANCING TO NEXT TRAILER RECORD} xxxxxx

**Explanation:** An I/O error was encountered while processing the VTOC or while bypassing the file for a data set RESTORE. The xxxxxx is the SYNAD information passed by the access method routines.

**System Action:** If the VTOC records were being accessed, the function is terminated. If the operation was to bypass the file that is not required for the requested RESTORE, the function is continued. The return code is 8.

**Programmer Response:** If the VTOC is being processed, data sets cannot be restored from this source file. If the file is being advanced to the trailer record (to process the next source file), permanent errors will be encountered if a RESTORE is attempted from this file in the future.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 30.

**ADR369D** AUTHORIZE FOR {READ | WRITE} ACCESS A RACF PROTECTED/SYSTEM DATA SET 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, or
- A VVDS or VTOCIX data set or an integrated catalog facility-cataloged VSAM data set on a non-integrated catalog facility system

**System Action:** Processing continues according to the operation action. If the operator replies "T", a full or tracks operation is terminated and, on a data set operation, the data set is not processed. 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:** A response of "R zz, U" causes DFDSS to continue processing the data set or volume. A response of "R zz, T" terminates the function, and processing continues 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, then a tape may 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** 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 accessing the VTOC for authorization checking.

**System Action:** Processing continues depending on the response from the operator. A reply of "T" terminates the function with a return code of 8.

**Programmer Response:** Instruct the operator to let DFDSS continue processing, and rerun the job.

**Operator Response:** Reply "R zz, U" to continue the function, if the user has authorization to access the volume. Otherwise, reply "R zz, T" to terminate the function.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 29.

**ADR372W** (xxx)-mmmmm(yy), ALIAS dsn 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 non-empty 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 restoring the alias.

**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 has been 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 need be taken. If it is not, correct the problem. If the alias is required, define it as for reason code 00.

**12** Determine the error from message ADR497E which precedes this message. Correct the problem, and then define the alias as for reason code 00 if desired.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, and 29.

**ADR373E** (xxx)-mmmmm(yy), REQUESTED TRACK  
{ccccchhh | (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 terminates. Processing continues with the next control statement. The return code is 8 if it is a data set RESTORE, or 16 if it is a full or tracks RESTORE and the output volume is unusable.

**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 "ADRTAPB Data Area" on page 302 and "ADRBMB Data Area" on page 301 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,
  - RECLN is not 80, or
  - BLOCKSIZE is not a multiple of the record length.

**System Action:** The task terminates. 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 only be copied or restored to a like device, but no like devices were available or usable.

**System Action:** Function fails for that data set.

**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 terminates. Processing continues with the next control statement. The return code is 8.

**Programmer Response:** Provide adequate storage by increasing the REGION size and/or the SIZE parameter and rerun the job. See "Storage Requirements—DFDSS" on page 181 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 nor non-integrated 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), FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED FROM VOLUME volser

**Explanation:** This message is followed by a list of data set names. 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 xxx | NO F2/F3 DSCB | NOT F2/F3 DSCB | ERROR IN F3 DSCB | TOO MANY EXTENTS} ON VOLUME volser FOR {DSNAME | CLUSTERNAME name COMPONENTNAME name}

**Explanation:** Errors encountered while accessing the extents for the data set. The possible errors are:

- OBTAIN failed while retrieving the extents for the data set.
- The DSCB retrieved was not a format-2 or format-3 DSCB.
- The format-1 DSCB had a forward pointer, but the format-2 or format-3 DSCB could not be located.
- The extent fields of the format-3 DSCB 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 volume serial is the logical volume if it is during a RESTORE.

**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 will not be 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)-mmmmm(yy), DATA SET dsname NOT PROCESSED, {rc | rc-xxx}

**Explanation:** The data set was not copied or restored. The reason codes, rc, are:

- 0 Unmovable/ABSTR/ISAM data set not preallocated and 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
- 1 Replace not specified for the unmovable data set
- 2 Unmatched extents on input and output for an unmovable/ ABSTR/ISAM data set
- 3-xxx Unable to allocate data set. The 'xxx' indicates DADSM (SVC 32) return codes. See your *System-Data Administration* book or *MVS/370 DADSM Diagnosis Reference* or *MVS/XA Data Facility Product Version 2: Diagnosis Reference* for more information.
- 3-004 DSNAME of request already exists on this volume; initial allocation not possible under given name
- 3-008 No room available in VTOC
- 3-012 One of the following errors encountered: a) permanent I/O error b) error returned by CVAF
- 3-016 Direct access absolute track not available



- 3-020 Requested quantity not available
- 3-024 Average record length greater than 65535 bytes
- 3-048 Invalid parameter list
- 3-052 Invalid JFCB or partial DSCB pointer
- 3-072 DOS VTOC cannot be converted to an OS VTOC
- 3-076 No space parameter given for a new data set or, zero space requested at absolute track zero
- 3-100 CYL and CONTIG requests conflict
- 3-104 Invalid space subparameter
- 3-116 User labels not supported
- 3-120 DSSIZE=0 and minimum allocation unit is greater than 0
- 3-124 DSSIZE is not a multiple of the minimum allocation unit
- 3-132 Space request must be ABSTR for a DOS volume.
- 3-148 Overlapping extents in the VTOC
- 3-152 Overlapping DOS split cylinder extents in the VTOC
- 3-156 DADSM allocation terminated because of possible VTOC errors
- 3-164 Allocation terminated because of DOS stacked pack format
- 3-168 RACDEF failed; data already defined
- 3-172 User not authorized to define data set
- 3-176 Installation exit rejected the request with a return code of 8
- 3-180 Installation exit rejected this request with a return code of 4
- 3-184 RACF defined with modeling specified; model not found
- 4 Authorization check failed
- 5 The block sizes of the target and source data sets do not match. This may be caused by attempting to move a data set created in an N\*512 environment to a non-N\*512 environment, or vice versa.
- 6 Volume sequence number mismatch
- 7 DSORG did not match
- 8 Error in rename filter
- 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.
- 10 The movable data set on the receiving volume is smaller than the data set on the source volume. Failure in scratch; cannot be scratched and reallocated.
- 11-xx VVDS calls failed. See return code 50 under message IDC3009I in *System Messages* manual for 'xx.'
- 12 The extents as mapped by the VTOC and VVDS on DASD do not match the VVDS on the input logical volume
- 13-xx Failure retrieving the extents from the VTOC. xx = Obtain error code.
- 14 Component names do not match; either the component has been 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.
- 15 Number of components differs on input and output; either the component has been 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.
- 16 Beginning RBA for the component(s) does not match on input and output. The input and output volumes are different volumes of a multivolume data set.
- 17 Primary/secondary indicators do not match on input and output. The input and output volumes are different volumes of a multivolume data set.
- 18 REPLACE or RENAME (conditional or unconditional) not specified for the movable non-VSAM data set, or REPLACE not specified for a VSAM data set.
- 19 The extents mapped by the VTOC do not match the VVDS on the output volume.
- 20-xxxx,yyyy The data set on the output volume is smaller than the input logical volume. xxxx = Data component size in tracks (hexadecimal). yyyy = index component size in tracks (hexadecimal).
- 21 VVDS, BCS, or VTOCIX cannot be restored to a different volume.
- 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.
- 26 CVOL cannot be replaced.
- 27 Input/output block could not be built because one or more of the following conditions exist:
  - Ddname not found
  - Ddname is a SYSIN or SYSPRINT
  - Device not DASD
  - Device not supported
  - Incorrect ranges specified for DASD
- 28 Unexpected catalog return code
- 29 No output volumes available because one of the following conditions exist:
  - There are no volumes in the output volume list that fulfill the requirements of the target volume(s) (that is, exact, same, or like device type).
  - The data set is preallocated and its attributes don't match those of the source data set (that is, the target data set is single volume and the source data set is multivolume, or visa versa).
  - All volumes of a preallocated multivolume data set are not included in the output volume list.
  - Attempting to restore an unmovable data set while specifying output volume(s) without at least as



much space as required by each corresponding source volume.

- 30 The volume serials of the preallocated, cataloged multivolume data set do not match those of the first volume
  - 31 VSAM data set preallocated on source volumes but catalog entry not found
  - 32 Insufficient space to restore the data set on the output volumes
  - 33 Unexpected allocation errors
  - 35 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.
  - 38 The VVR for a component of the data set was found, but there was no corresponding DSCB.
  - 39 RENAME/RENAMEU is not supported for unmovable data sets.
- 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 and rerun.
  - 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 the 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 the format-1 DSCB in the VTOC. 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 it is a RESTORE and 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 Restore to the correct DASD volume or use the LVOL parameter to restore from the correct logical volume.
  - 22 Specify FORCE or free the corresponding location on the target volume and rerun.
  - 23 If it is a multivolume data set, correct 'the lastvolume indicator' and 'the volume sequence number' fields in the VTOC, or catalog the data set in the proper sequence in the appropriate catalog and rerun.
  - 26 Delete the CVOL from target volume. Then rerun the job.
  - 27 Take one or more of the appropriate actions listed below; then rerun the job:
    - Ensure a DDNAME statement is present in job step
    - Ensure ddname is not a SYSIN or SYSPRINT
    - Ensure device specified is a DASD
    - Ensure device supported on system
    - Ensure correct ranges specified for DASD
  - 28 Refer to message ADR497E printed preceding this message, and take appropriate action.
  - 29 Take one of the actions listed below; then rerun the job:
    - Supply a new choice of output volumes.
    - Ensure the attributes of the source and the preallocated target match.
    - Ensure that the data set is cataloged and the catalog volume(s) and output volume(s) on which it is preallocated match.
    - Restore as a movable data set by specifying the FORCE keyword.

- 30 Ensure 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 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 volume. Then rerun the job.
- 35 Choose a different name for 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.
- 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.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 25b. Take actions 29 and 35 on 3-48, 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 which are at the beginning of the file. See "ADRTAPB Data Area" on page 302 to help identify the VTOC tracks.

**ADR381E (xxx)-mmmmm(yy), ALL OUTPUT VOLUMES INACCESSIBLE. DATA SETS NOT PROCESSED**

**Explanation:** The OUTDD subparameter is missing, or the output volume could not be opened for a data set COPY or RESTORE. This message may be preceded by message ADR307E.

**System Action:** The task is terminated. The return code is 8.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, and 29.

**ADR382E (xxx)-mmmmm(yy), THE FOLLOWING DATA SETS WERE NOT PROCESSED FROM VOLUME volser DUE TO ERRORS**

**Explanation:** The data sets in the list were not processed. The error is identified by other messages which precede this message.

**System Action:** The return code is 8.

**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.

**ADR383I mmmmm(yy), DATA SET dsname NOT SELECTED**

**Explanation:** The data set, whose fully qualified name was passed, was not selected. It did not meet the EXCLUDE or BY filtering criteria.

**System Action:** The data set is not processed.

**Programmer Response:** Change the filtering criteria. Rerun the job.

**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. See "ADRTAPB Data Area" on page 302 to help identify the VTOC tracks.

**ADR384E 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.

**System Action:** The return code is 8.

**Programmer Response:** See your *Catalog Administration Guide* under "Using Catalog Management Macro Instructions for OS CVOLs" for an explanation of return and reason codes and take appropriate action.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 29.

**ADR385E 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.

**System Action:** The return code is 8.

**Programmer Response:** See your *Catalog Administration Guide* under "Using Catalog Management Macro Instructions for OS CVOLs" for an explanation of return and reason codes and take 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. The message is followed by a list of volume serials.

**System Action:** The return code is 8.

**Programmer Response:** If no source volume was processed, use the correct RESTORE file or use 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 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 (for example, a dump tape was created using a data set DUMP and you tried to restore it via a full RESTORE), or input was not produced by DFDSS. If unlabeled tapes were used, you may have mounted them in the wrong sequence.

**System Action:** The task terminates. 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), dsname 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. This message will be followed by message ADR396I if allocation is feasible on this volume or by message ADR380E if allocation is not feasible on this volume.

**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 volume(s) must be specified to restore the data set.

**System Action:** Processing terminates 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:** An attempt was made to allocate the data set using the name generated from the RENAME parameter, but a data set exists on the volume with the new name.

**System Action:** This message may be followed by message ADR380E. The return code is 8.

**Programmer Response:** Either scratch the existing data set with the name that was 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)-mmmm(yy), UNABLE TO ALLOCATE  
dsname ON volser WITH NEWNAME dsname

**Explanation:** An attempt was made to allocate the data set using the name generated from the RENAME parameter, but the allocation failed.

**System Action:** This message will be followed by message ADR380E. The return code is 4.

**Programmer Response:** Take the appropriate action listed under message ADR380E. The return code is 4.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 29.

**ADR395I** (xxx)-mmmm(yy), DATA SET dsname  
ALLOCATED 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.

**System Action:** None.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR396I** (xxx)-mmmm(yy), DATA SET dsname  
ALLOCATED [, 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.

**System Action:** None.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR402E** (xxx)-mmmm(yy), AUTHORIZATION CHECK  
FAILED FOR dsname, ON VOLUME volser, rc

**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 non-integrated 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 User authorization exit rejected the volser.

- 6 User authorization exit rejected the data set name.
- 7 Password not supplied for the data set name.
- 8 READ password specified, but WRITE password was required.
- 9 Password incorrect for the data set name.
- 10 Error while accessing PASSWORD data set.
- 11 OPEN failed while verifying the password for the VSAM data set.
- 12 The catalog entries cannot be located for the data set in the VVDS.
- 13 Expiration date has not passed for the data set name.
- 14-xxx-yyy Authorization check failed for the VSAM data set. The xxx and yyy are the return codes and reason codes from SVC 26. See return code/reason code under message IDC3009I in *System Messages* for an explanation.
- 15 Invalid return code from the user authorization exit.

**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 *System Messages*.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 5b, 13, 20, and 29.

**ADR404E** (xxx)-mmmm(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)-mmmm(yy), DYNAMIC ALLOCATION  
OF VOLUME volser FAILED. ERROR CODE  
eeee. INFORMATION CODE iii.

**Explanation:** DFDSS attempted to dynamically allocate the volume indicated and that attempt failed. The error and information codes are those returned by dynamic allocation and are listed in *OS/VS2 MVS System Programming Library: Job Management* or *MVS Extended Architecture System Programming Library: System Macros and Facilities*, Volume 1. Error code X'049C' may result if the volume is not mounted and online and the unit name was not passed because DFDSS tries to allocate the volume as a virtual volume after allocation as a real volume fails. An undocumented error code may be displayed if an installation validation exit is used to deny allocations for SVC 99, but fail to get the reason code returned by SVC 99.

**System Action:** The task or volume is bypassed. The return code is 8.

**Programmer Response:** Correct the problem as indicated by the error and information codes listed in *OS/VS2 MVS System Programming Library: Job Management* or *MVS Extended Architecture System Programming Library: System Macros and Facilities*, Volume 1.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 29.

**ADR406E (xxx)-mmmmm(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 attempting to restore an ISAM data set, or a CVOL to an unlike device, or when attempting to restore 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 then rerun the job.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR407E (xxx)-mmmmm(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 non-integrated catalog facility VSAM catalog.

**System Action:** The COPY or RESTORE is terminated. 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)-mmmmm(yy), LOCATE FAILED FOR CATALOG catname**

**Explanation:** An error occurred while attempting to locate the volsr of the catalog name containing the VSAM model entity (MENTITY) entry.

**System Action:** The COPY or RESTORE is terminated. 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.

**ADR411W (xxx)-mmmmm(yy), DATA SET {dsname | clustername 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, and TOL(ENQF) was specified by the user or user installation exit.

**System Action:** If DFDSS performs the copy, the data set is processed without an enqueue and the return code is 4. If DFDSS invokes a utility to perform the copy, the data set is not processed further and the return code is 8. If RESET was requested on a data set DUMP, the DS1DSCHA bit is not reset in the format-1 DSCB for this data set.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 29.

**ADR412E (xxx)-mmmmm(yy), DATA SET {dsname | clustername 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, and 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, or you can specify TOL(ENQF).

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR413E (xxx)-mmmmm(yy), DATA SET dsname ON VOLUME volsr WAS NOT {DELETED | UNALLOCATED}; {xx | xx-yyyy : zzzz}**

**Explanation:** The data set could not be scratched, uncataloged, or unallocated via SVC 99. The error codes (xx) are:

- 0 SVC 99 installation validation routine denied request.
- 1 SVC 99 information error code is yyyy. Reason code is zzzz. The codes are explained in *OS/VS2 MVS System Programming Library: Job Management* or *MVS/Extended Architecture System Programming Library: System Macros and Facilities*, Volume 1.

**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 mmmmm(yy), DATA SET dsname ON VOLUME volsr NOT SCRATCHED, {xx | xx-yy,zzz}**

**Explanation:** The data set could not be scratched. The error codes (xx) are:

- 0 DEQ on the data set failed.
- 1 The data set was enqueued by the initiator.

- 2 Scratch failed. The yy is the return code and/or status code and the zzz is the reason code from DADSM SCRATCH, as described in *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.

**ADR415E (xxx)-mmmm(yy), NO DATA SETS  
COPIED/DUMPED/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. Either no data sets matched the filtering criteria, or data set(s) matched the filtering criteria but failed serialization or authorization and none were processed, or 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 8. If there were no I/O errors, and 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. See "ADRTAPB Data Area" on page 302 to help identify the VTOC tracks.

**ADR416E 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 VSAM data set (DELETE not supported for VSAM by DFDSS).
- 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 could not 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 Data set was not processed (probably an empty data set).
- 5 Data set expiration date has not expired. 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 should 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)-mmmm(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 format-2 DSCB 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 format-1 DSCB in the VTOC could not be retrieved for updating the fields in the DSCB.
- 2 The data set was copied or restored but errors occurred while rewriting the format-1 DSCB back on the VTOC to update the fields in the DSCB.
- 3 The data set was copied or restored but I/O errors occurred on one or more tracks of the data set during the operation. If it was a RESTORE, the I/O errors occurred at dump time.
- 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 format-1 DSCB in the VTOC could not be retrieved to set the RACF indicator in the DSCB or to update the format-2 DSCB for the data set.
- 6 The data set was not copied or restored because errors occurred while rewriting the DSCB to reset the RACF indicator, or to update the format-2 DSCB for the data set.
- 7 The target VSAM data set could not 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 format-1 DSCB in the VTOC 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 by your installation 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 before you rerun the job.
- 7 The data set does not have the RACF indicator set. However, in an always call (RACF 1.5 or greater) environment, the data set is protected by the existing 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), 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 has been copied or restored to the volume. VTOC and VVDS entries have been 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. It was either 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 this message is accompanied by message ADR411W, 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 VVDS data set is not allowed along with restore of other VSAM data sets from the same logical volume. Restore of VTOCIX data set is not allowed along 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 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. The volume is bypassed. Data set RESTORE continues with other logical volumes.

**System Action:** None.

**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. See "ADRTAPB Data Area" on page 302 for its description. 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:** If it is a multivolume data set, make sure that the entire data set from all of the volumes on which it resides has been 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)-mmmmm(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 terminated 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)-mmmmm(yy), CONFLICTING BY CRITERIA SPECIFIED

**Explanation:** Either VSAM and non-VSAM data sets not wanted or VSAM data set processing not supported on the system.

**System Action:** The command terminates. 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)-mmmmm(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 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 to see that the directory is in the first extent. This can be done 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)-mmmmm(yy), ERROR IN {VVDS | VVDS IMAGE} ON {VOLUME | LOGICAL VOLUME} volser WHILE ACCESSING COMPONENTS OF CLUSTER clustername IN CATALOG catname

**Explanation:** DFDSS found errors while retrieving components for the cluster the input during a COPY, DUMP, or RESTORE. Some components could not be located for the cluster in the VVDS or VVDS image. Or, if it was a RESTORE, VVDS was not dumped successfully during a full volume DUMP. Some 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** mmmmm(yy), VTOC ENTRIES NOT FOUND FOR CLUSTER clustername 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. If it is a logical volume, the error is in the source volume and the error 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 data set RESTORE, use a previous backup copy (full or data set DUMP) 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.

**ADR430E** (xxx)-mmmmm(yy), ERROR IN CLUSTER clustername IN CATALOG catname ON VOLUME volser

**Explanation:** There was a failure during definition of the components of the cluster 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 recommended 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)-mmmmm(yy), DATA SET dsname [IN CATALOG catname] HAS BEEN DELETED**

**Explanation:** The data set was deleted on user request.

**System Action:** The data set has been deleted from the volume.

**Programmer Response:** If the data set was cataloged, it should be uncataloged 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 could not 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. This message is accompanied by other messages indicating the affected cluster that is not processed. 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 appropriate action recommended. If it is the input to the RESTORE (logical volume), use a previous backup copy of the data set 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 was 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 command is also forcibly terminated via a user ABEND0002.

**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 (UNCATALOG is not supported for VSAM by DFDSS)
- 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 could not 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 (probably a null data set).

**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 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 could not 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 was read into the buffer, and before writing it to tape.

**System Action:** The track is not processed. The command is terminated. 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 clustname [IN CATALOG catname] CANNOT BE {OPENED | CLOSED} retcode rsncode**

**Explanation:** The specified cluster cannot be opened or closed for VSAM I/O processing. VSAM has responded with a return code and a reason code after processing has 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 then 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 has provided a return code and a reason code after processing 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 then rerun the job.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR439E** (xxx)-mmmm(yy), PREALLOCATED DATA SET dsname WAS SELECTED BUT IS NOT USABLE, rc

**Explanation:** DFDSS found a preallocated data set but determined it was unusable for the following reason (rc):

- 04 The preallocated cluster type is not equal to the target cluster type.
- 08 The preallocated cluster has an AIX defined over it.
- 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 REUSEABLE.
- 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 Update of the target format-1 DSCB(s) failed. The target format-1 DSCB must be updated to make the target data set look empty if a utility will do the data transfer. This update failed while either reading or rewriting the target DSCB.
- 52 The preallocated, unmovable data set is not on the same or like device as the source data set.
- 56 The preallocated DSORG is not equal to the target DSORG.
- 60 The preallocated data set space is insufficient.
- 68 The preallocated data set is unusable.
- 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 just one extent.
- 88 I/O error encountered on the preallocated partitioned data sets directory.

**System Action:** The specified data set is not copied. 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 has. 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 format-1 DSCB DSILSTAR field to make the data set appear empty.
- 52 Allocate an unmovable target data set on the same or like device, 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.
- 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 the RELBLOCKADDRESS keyword was specified and that the source:
  - Is not a standard user label data set
  - Has a RECFM of fixed or fixed blockOr scratch and reallocate the target on a same or like device as the source.
- 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.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, 25b, 25c, and 29.

**ADR440E** (xxx)-mmmm(yy), UNEXPECTED RETURN CODE FROM externalname: xxx-yyy, WHILE PROCESSING DATA SET dsname

**Explanation:** While processing the specified data set, the external routine named was called, and it returned an unexpected return and/or reason code.

**System Action:** Processing is terminated 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.

**ADR442I** (xxx)-mmmm(yy), DATA SET dsname PREALLOCATED [, CATALOG catname], VOLUME(S): {volser | volser list}

**Explanation:** The data set has been selected for replacement as a preallocated target data set. The target catalog name is printed if known. 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** mmm(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 may 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. This message will be followed by message ADR469E. The return code is 4.

**Programmer Response:** See previous messages for the specific error of the component alter failure. Run Access Method Services LISTCAT against the target catalog to determine the reason for the Access Method Services ALTER failure. The component name can be altered to match the source, using Access Method Services ALTER NEWNAME.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, 26, and 29.

**ADR444W** mmm(yy), DATA SET dsname NOT DELETED

**Explanation:** During a COPY with DELETE, between unlike device types, the target data set was allocated but no data movement occurred. The source data set was not deleted.

**System Action:** The return code is 4. The data set is not deleted.

**Programmer Response:** Delete the data set by other means.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, 25a, and 29.

**ADR445I** (xxx)-mmmm(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)-mmmm(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)-mmmm(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 When restoring to an unlike device, the specified preallocated target KSDS is not reusable.
- 2 The preallocated target data set and the source data set do not have one of the following matching attributes:
  - CI size
  - IMBED
  - KEY length
  - Record length
  - REPLICATE
  - SPANNED

**System Action:** Processing continues. For reason code 2, 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 using nonoptimized channel programs. This results in performance degradation if the data set is not flagged as track overflow in the VTOC format-1 DSCB or is not in the TRACKOVERFLOW keyword list.

**Programmer Response:** No direct programmer action is required, as the data set has been 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 format-1 DSCB should be specified in the TRACKOVERFLOW keyword list. This will prevent DFDSS 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 DS1LSTAR may have residual track overflow records which 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 could cause the messages for this task to be broken up into groups with messages from other tasks in between.

**System Action:** The message data set is reinitialized and messages will 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 terminated. 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} UTILITY FAILED  
WHILE PROCESSING {DATA SET dsname |  
VOLUME volser}, {ABEND | RETURN} CODE  
IS nnn**

**Explanation:** During execution 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 *System Codes* for an explanation of the abend code. If the abend is because of insufficient storage (DASD or virtual), you should run the utility itself (not through DFDSS), as described in your *Utilities* manual. If IEHMOVE abends with code 0C4, it may be because DFDSS called it with a task I/O table (TIOT) referencing 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, specifying the DFDSS "UTILMSG=YES" parameter. See *MVS/XA System Messages* or *MVS/370 Utilities Messages* for an explanation of the utility messages.
- For IDCAMS error return codes, run the job specifying the DFDSS "UTILMSG=YES" parameter. See *System Messages* for an explanation of the messages.
- For ICKDSF error return codes, run the job and specify the DFDSS "UTILMSG=YES" parameter. See *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 that was 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.

**ADR455I (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. See previous message for specific error(s).

**System Action:** Processing continues.

**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 mmmmm(yy), ERROR ENCOUNTERED WHILE FILTERING ON THE CATALOG**

**Explanation:** This message is preceded by message ADR497E. Refer to IDC3009I 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 execution 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

**ADR461I 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.

**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/or the RENAMEUNCONDITIONAL subparameter was specified.

**System Action:** Processing continues. The return code is 8.

**Programmer Response:** Copy the CVOL catalog specifying DELETE and/or 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 volume(s) failed because the volume(s) 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. You probably have an invalid VTOC.

**System Action:** Execution of the command is terminated. The return code is 8.

**Programmer Response:** See *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:** Execution of the command is terminated. 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 then 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, then 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, then: (1) the DELETE subparameter was not specified, or (2) the RENAMEU subparameter was specified.
- If REPLACE was not specified, then one of the following conditions applies:
  - If DELETE is specified and the entry name is a SYS1., page, or swap data set, then 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, then: (1) the DELETE subparameter was not specified, or (2) the RENAMEU subparameter was specified.
  - If the entry name is a user catalog name, then 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 then:
  1. If the data set is eligible for preallocation, ensure that a qualifying target is available on the target volumes.
  2. 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, then: (1) copy the cluster, specifying DELETE, or (2) copy the cluster, specifying RENAMEU(newname), or (3) copy the cluster, specifying RECAT(catname).

- If the entry name is an alternate index or a user catalog name, then copy the alternate index or user catalog, specifying DELETE, and remove the RENAMEU subparameter.
- If the entry name is a user catalog name, then do not specify INDDNAME or INDYNAM.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR469E**      **mmmmm(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 8.

**Programmer Response:** See earlier messages for specific error(s).

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 29.

**ADR470E**      **(xxx)-mmmmm(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 could not be copied.

**System Action:** The task is terminated. Processing continues with the next control statement. The return code is 8.

**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 and rerun the job.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR471E**      **(xxx)-mmmmm(yy), DUE TO ERRORS, SOURCE {CLUSTER | COMPONENT} name IS NOW NAMED name. CLUSTER clustername, CATALOG catname**

**Explanation:** During COPY processing, the source cluster and components may 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 an attempt is made to restore the source cluster to its original state. 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. This message will be followed by message ADR455E. 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)-mmmmm(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** Not enough space available on any target volume to place the target data set.
- 08** At least one duplicate data set was encountered on a target volume. Target volumes which did not contain duplicate data sets did not have enough space for the data set.
- 12** The data set is non-VSAM and either unmovable or absolute track allocation. DFDSS was only able to select unlike targets for target allocation; however, 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 reading the VVR records for the VSAM data set.
- 28** Duplicate format-1 DSCBs were encountered on only/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 building internal control blocks for the data set.
- 44** The data set was BDAM and only target devices of smaller track capacity than the source were available and the data set was not specified in the RELBLOCKADDRESS keyword list, or it was specified in RELBLOCKADDRESS but the data set is TTR-organized. The IEHMOVE utility is used to copy BDAM data sets to unlike devices when RELBLOCKADDRESS is not specified. When the target device has a smaller track capacity, the IEHMOVE copy results in an unloaded format data set. 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 format 1 DSCB.
- 52** An indexed VTOC was not available.
- 56** Not enough space is available in the VTOC on the target volume(s).

**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, or specify REPLACE.
- 12 Either specify FORCE if the unlike target volumes are desired, or specify additional target volumes of the same device type as the one 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 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 This reason code is accompanied by message ADR497E. 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 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.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, 25b, 26, and 29.

**ADR473E**      **mmmmm(yy), FAILURE WHILE ATTACHING UTILITY MODULE, RC=xx**

**Explanation:** DFDSS was unable to attach a utility module. The reason codes, xx, are documented in *Supervisor Services and Macros Instructions*.

**System Action:** Data set not copied.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR474W**      **mmmmm(yy), DATA SET dsname {IS EMPTY | HAS AN UNSUPPORTED DSORG}. IT WILL BE ALLOCATED BUT NOT COPIED**

**Explanation:** Either an empty data set was allocated but not copied because ALLEXCP or ALLDATA was not specified, or a data set with unsupported DSORG or BLKSIZE=0 was allocated but not copied because the target volume selected is unlike the source device.

**System Action:** Processing continues. This message will be followed by either message ADR395I or message ADR396I indicating the target volume serial number. The return code is 4.

**Programmer Response:** To have the data set copied, delete the target data set and rerun the job doing:

- For an empty data set to a like device, use ALLDATA or ALLEXCP to force copy of all allocated space in the data set.
- For a data set with an unsupported DSORG or BLKSIZE=0, specify as target volumes only the same or like devices.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 2, 3, 4, 13, 20, and 25b.

**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**      **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 terminate.

**System Action:** Utility processing is terminated. 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 encountered errors while reading or updating VVDS/VVR/CELL information.

**System Action:** The data set is not copied. 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 has a component that is larger than the entire source volume. The data set is too large to be RESTORED by DFDSS from a logically formatted dump tape.

**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 has been 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.

**ADR480E (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 8.

**Programmer Response:** Correct the problem according to preceding messages.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR481E (xxx)-mmmmm(yy), UNABLE TO ALTER VSAM CLUSTER clustname [IN CATALOG catname]**

**Explanation:** The VSAM ALTER NEWNAME failed while attempting to alter the source cluster name.

**System Action:** Processing continues.

**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 reading the indicated type of dump tape record during logical data set RESTORE processing.

**System Action:** The data set is bypassed.

**Programmer Response:** Use an alternate backup copy to restore the data set.

**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 will not be cataloged. RECATALOG(\*) was specified and 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.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR485E (xxx)-mmmmm(yy), CATALOG catname IS NOT IN STEPCAT/JOB CAT/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 could not 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.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR487W** (xxx)-mmmm(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. It is possible that the data set name is contained in the data set name list at the beginning of the dump tape but 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.  
**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR488E** (xxx)-mmmm(yy), ERROR ENCOUNTERED WHILE ACCESSING {VTOC | VVDS} FOR DATA SET dsname ON VOLUME volser ON DDNAME dname

**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)-mmmm(yy), {dsname | clustername | component name} WAS {SELECTED | RESTORED}

**Explanation:** After a data set was successfully restored (when TYPRUN=RUN) or selected (when TYPRUN=NORUN), the data set name is printed for 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)-mmmm(yy), ERROR {OPENING | READING} UTILITY SYSPRINT MESSAGES. UTILITY MESSAGES WILL BE LOST

**Explanation:** There was an error either opening or reading the utility SYSPRINT file.

**System Action:** DFDSS continues processing for the task but utility SYSPRINT messages are lost.

**Programmer Response:** None

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR491E** (xxx)-mmmm(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 terminated.

**System Action:** None. The return code is 8.

**Programmer Response:** Increase the SIZE parameter and/or region size, and rerun the job.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, 29, and 35.

**ADR492E** (xxx)-mmmm(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 while exporting the catalog to a temporary sequential data set. The COPY is discontinued and the source catalog remains in usable condition. The second portion of the message is not printed for this reason code.
- 02** A failure occurred while IMPORTing the target catalog. The source catalog has been 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 *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 source catalog problem that prevents EXPORT. Rerun the job.
- 02** The source catalog must be restored before attempting to rerun the job. If the catalog was backed up before the COPY was attempted, the backup may be used to restore the catalog. Otherwise, the catalog may be restored by IMPORTing the catalog from the export data set described in the message. See *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)-mmmm(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" data set, a page data set, or a swap data set that cannot be uncataloged or deleted.

**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. For "SYS1.", page, or swap data sets, remove the UNCATALOG/DELETE subparameter.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR494W** (xxx)-mmmm(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

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR495E** (xxx)-mmmm(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 terminated. 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)-mmmm(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 *Supervisor Services and Macro Instruction* for an explanation of DETACH return codes.

**ADR497E** (xxx)-mmmm(yy), CATALOG ERROR OCCURRED. RETURN CODE IS xxx, REASON CODE IS nn-yy

**Explanation:** This return code, xx, and reason code, yy, were returned by Catalog Management module IGG0CLnn as a result of a catalog error or exceptional condition.

**System Action:** Command processing is terminated. The return code is 8.

**Programmer Response:** See message IDC3009I in *System Messages* for specific return code and reason code definitions. Be sure the data sets to be processed occur in the standard catalog search order. For return code 40 or 44, rerun the job after changing the DFDSS filtering to reduce the number of selected CVOL-cataloged data sets to fewer

than 1455 or increasing the region size if you are already selecting fewer than 1455.

**Operator Response:** None.

**Problem Determination:** Table I, items 1, 3, 4, 13, 20, and 29.

**ADR498E** (xxx)-mmmm(yy), DATA SET dsname CAN NOT BE COPIED/RESTORED. PREALLOCATE THE DATA SET WITH trks TRACKS

**Explanation:** DFDSS could not determine the amount of space required to contain the data set prior to 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 volume(s) 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.

# Problem Determination

Problem determination is the activity required to identify a failing hardware unit or program and determine who is responsible for maintenance.

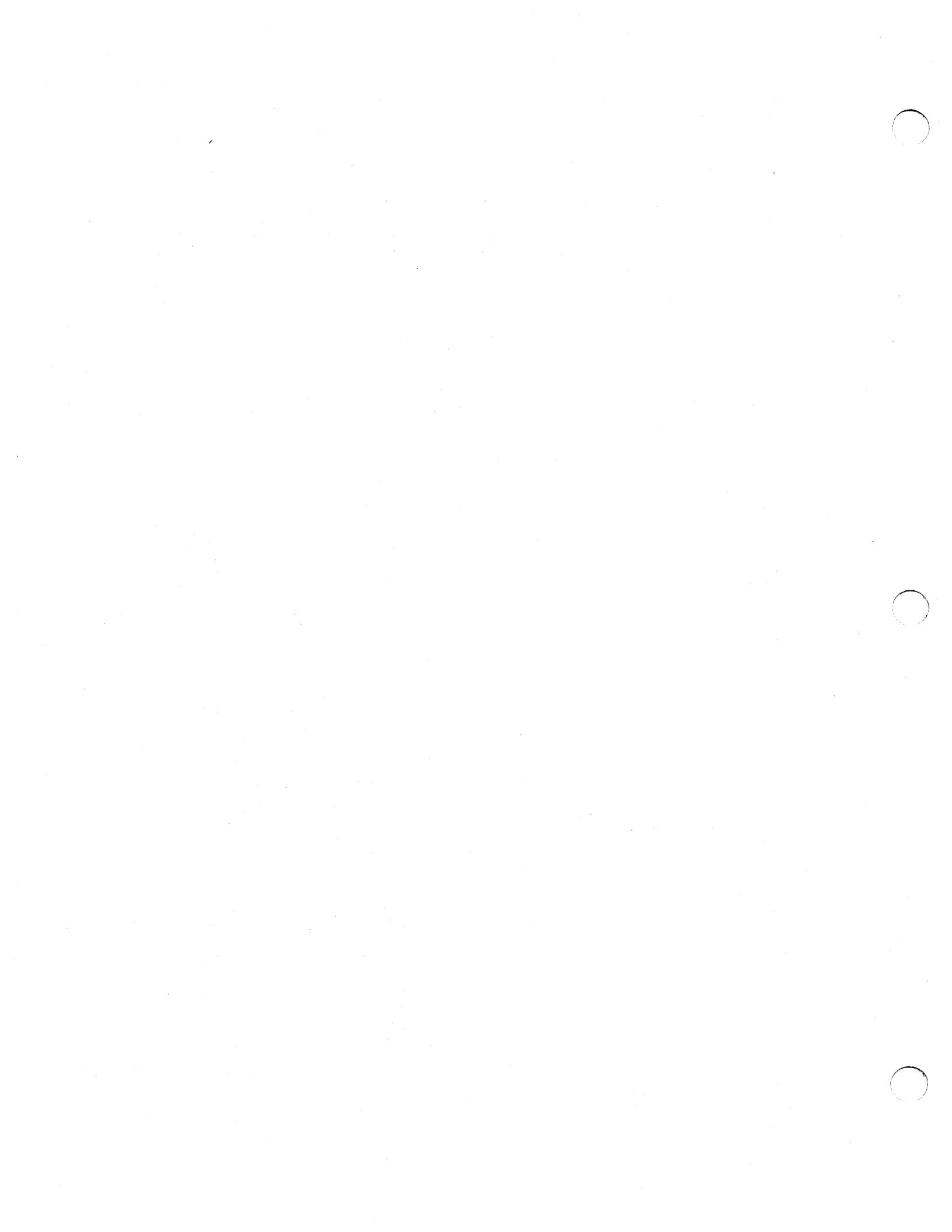
Problem determination is accomplished by using the procedures specified by IBM. In some cases, these procedures may be initiated by a message or code which requires operator or programmer response. The response may include the requirement for additional problem-related data to be collected and attempts, where possible, to indicate "probable" failure responsibility.

Problem determination information is included for applicable messages under the heading "Problem Determination." Standard problem determination actions are identified as items of Table I. Table I provides only the problem determination steps that are referenced in the preceding utilities messages. See "Related Publications" in the Preface for the appropriate manual for a complete description of Table I. Unique actions are identified following the list of standard actions to be taken. In any case, it is intended that the specified actions be taken before calling IBM for support.

## Table I

**Note:** If the problem recurs, follow the problem determination aids specified by the associated message or code before calling IBM for support.

1. Make sure 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. Make sure that the failing job step includes one of the following:
  - a. SYSABEND DD statement
  - b. SYSUDUMP DD statement
6. Not used.
7. Execute 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. Execute 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. Execute the AMDSADMP service aid program to dump the contents of real storage and page data sets on magnetic tape. After restarting the system, execute 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 (should further information from the tape be required) and the listing from AMDPRDMP.
12. Execute 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 execute 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. Execute the IEHLIST system utility program to obtain a list of the:
  - a. Volume table of contents of the associated volume, specifying the FORMAT option
  - b. Volume table of contents of the associated volume, specifying the DUMP option
  - c. Directory of the associated data set
  - d. System catalog
26. Execute 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 3-digit message number on which the dump should be taken, and insert a SYSUDUMP DD statement in the JCL. This will cause a User 0001 abend at the time the message ADRnnnx is issued.
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



## Appendix C. Stand-Alone DFDSS Restore Messages

Component Name: ADR

Program Producing Message: ADRDMPRS

Message Format:

ADRnnns message text

The following applies to all messages:

nnn = Message number

s = Console Operator type code:

A - Action: Operator must perform a specific action.

I - Information: No operator action is required.

W - Warning: No operator action is required, but an error has occurred.

**Problem Determination:** Refer to the table following the messages for the explanation of the problem determination codes.

*Note:* Wherever "interrupt key" is specified, any interrupt mechanism may be used; for example, under VM/370 you would enter "#CP EXTERNAL".

### ADR501W INVALID CARD CODE. CORRECT ERROR. DEPRESS INTERRUPT KEY.

**Explanation:** An invalid card code appeared in the last card processed.

**System Action:** Enters wait state awaiting operator response.

**Programmer Response:** None.

**Operator Response:** Probable user error. Check for and correct keypunch or multipunch errors and reset the card reader. Restart the card reader or rerun the job.

**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 terminated.

**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:** The statements are not in proper sequence or unnecessary statements are present.

**System Action:** The job is terminated.

**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:** If the input tape contains the invalid track, then the RESTORE continues. Otherwise the job is terminated.

**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:** Place control statements in the input device to prepare 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 terminated.

**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 cccchhh**

**Explanation:** A defective track was found at the location specified by cccc, the cylinder number, and hhhh, 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 cccchhhh | NONE**

**Explanation:** An alternate track at the location specified by cccc, the cylinder number, and hhhh, the head number, is the track which replaces the defective track.

**System Action:** None.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** None.

**ADR522W PROGRAM INTERRUPT. JOB TERMINATED.**

**Explanation:** A program interrupt has occurred.

**System Action:** The job is terminated.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** Table I, items 2, 11, 20, and 29.

**ADR533A 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 console interrupt key, and attempt to enter the input message again.

**Problem Determination:** Table I, item 30.

**ADR544A 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 console interrupt key to continue the

program. If the card or transport jam occurs again, reproduce the jammed cards and try again.

**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 may be caused by a forms check, an open interlock, or someone having pressed the STOP key.

**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 console interrupt key.

**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 console interrupt key.

**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 console interrupt key.

**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.
- 4 The COMPRESS option was specified for the DUMP operation.
- 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.

**Problem Determination:** Table I, items 2, 11, 20, 28, and 29.

**ADR559A READER CHECK. CORRECT ERROR.  
DEPRESS INTERRUPT KEY.**

**Explanation:** A reader check has 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. 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 has 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.

**Problem Determination:** Table I, item 30.

**ADR561A END OF TAPE. MOUNT TAPE ON cuu.  
DEPRESS INTERRUPT KEY.**

**Explanation:** End of 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.

**Problem Determination:** None.

**ADR563A END OF JOB.**

**Explanation:** A normal end-of-job condition has occurred.

**System Action:** The job is terminated.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** None.

**ADR565A ATTEMPT TO RESTORE TO WRONG  
DEVICE. JOB TERMINATED.**

**Explanation:** Program attempted to restore data to a device type other than the type from which it was dumped.

**System Action:** The job is terminated.

**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 has been 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 terminated.

**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 has 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 on the console to continue processing.

**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 terminated.

**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 cccchhhh**

**Explanation:** An I/O error was encountered 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 has rejected an incorrect channel command word (CCW) list.

**System Action:** The job is terminated.

**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 make it ready. If the device is not ready without an apparent reason, initiate the following problem determination.

**Problem Determination:** Table I, item 30.

**ADR603W BUS OUT CHECK. c**

**Explanation:** A bus out check has occurred on channel c. Probable hardware error.

**System Action:** The job is terminated.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** Table I, items 12 and 30.

**ADR604W EQUIPMENT CHECK.**

**Explanation:** An equipment failure has occurred. Probable hardware error.

**System Action:** The job is terminated.

**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 has occurred on the specified device. Probable hardware error.

**System Action:** The job is terminated.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** Table I, item 30.

**ADR606W OVERRUN. c**

**Explanation:** An overrun check has occurred on the specified channel.

**System Action:** The job is terminated.

**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 has occurred on the specified device.

**System Action:** The job is terminated.

**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 has occurred on the specified device.

**System Action:** The job is terminated.

**Programmer Response:** None.

**Operator Response:** Rerun the job.

**Problem Determination:** Table I, item 30.

**ADR610W INVALID ADDRESS. device, opcode status, sense bytes, cccchhhh**

**Explanation:** An invalid address has been issued to the specified device.

**System Action:** The job is terminated.

**Programmer Response:** None.

**Operator Response:** Determine if the cccchhhh 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 terminated.

**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 has been detected on the specified tape unit.

**System Action:** The job is terminated.

**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 has occurred.

**System Action:** The job is terminated.

**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 terminated.

**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 has occurred.

**System Action:** The job is terminated.

**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 terminated.

**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 has occurred on the channel specified by cxx.

**System Action:** The job is terminated.

**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 has occurred on the channel specified by cxx.

**System Action:** The job is terminated.

**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 has occurred on the channel specified by cxx.

**System Action:** The job is terminated.

**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 has occurred because of an incorrect CCW.

**System Action:** The job is terminated.

**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 has occurred on the specified device.

**System Action:** The job is terminated.

**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 has occurred on the specified unit.

**System Action:** The job is terminated.

**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 has occurred on the specified unit.

**System Action:** The job is terminated.

**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 has occurred on the channel specified by cxx.

**System Action:** The job is terminated.

**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 terminated.

**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 terminated.

**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 has occurred on the specified tape unit.

**System Action:** The job is terminated.

**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 terminated.

**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 has occurred on the specified device.

**System Action:** The job is terminated.

**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:

- The number of cylinders dumped from the original input volume, or
- The highest cylinder number to be restored, as specified on a VDRL statement.

**System Action:** The RESTORE operation is terminated.

**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 terminated.

**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 execution of write commands.

**System Action:** The job is terminated.

**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 execution of write commands, follow the procedures established by your installation.
- If the WRITE INHIBIT switch is off to allow execution 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 terminated.

**Programmer Response:** None.

**Operator Response:** None.

**Problem Determination:** None.

# Problem Determination

Problem determination is the activity required to identify a failing hardware unit or program and determine who is responsible for maintenance.

Problem determination is accomplished by using the procedures specified by IBM. In some cases, these procedures may be initiated by a message or code which requires operator or programmer response. The response may include the requirement for additional problem-related data to be collected and will attempt, where possible, to indicate "probable" failure responsibility.

Problem determination information is included for applicable messages under the heading "Problem Determination." Standard problem determination actions are identified as items of Table I. Table I provides only the problem determination steps that are referenced in the preceding messages. See "Preface" for the appropriate manual for a complete description of Table I. Unique actions are identified following the list of standard actions to be taken. In any case, it is intended that the specified actions be taken before calling IBM for support.

## Table I

**Note:** If the problem recurs, follow the problem determination aids specified by the associated message or code before calling IBM for support.

1. Make sure 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. Make sure that the failing job step includes a:
  - a. SYSABEND DD statement
  - b. SYSUDUMP DD statement
6. Not used.
7. Execute 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. Execute 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. Execute the AMDSADMP service aid program to dump the contents of real storage and page data sets on magnetic tape. After restarting the system, execute 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 (should further information from the tape be required) and the listing from AMDPRDMP.
12. Execute 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 execute 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. Execute the IEHLIST system utility program to obtain a list of the:
  - a. Volume table of contents of the associated volume, specifying the FORMAT option
  - b. Volume table of contents of the associated volume, specifying the DUMP option
  - c. Directory of the associated data set
  - d. System catalog
26. Not used.
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.



## Appendix D. Data Area Descriptions

This appendix contains two data area descriptions, corresponding to control blocks ADRBMB and ADRTAPB.

For a description of ADRUFO, see "ADRUFO Data Area" on page 239.

For a description of ADREID0, see "ADREID0 Data Area" on page 219.

### ADRBMB Data Area

Offsets	Type	Length	Name	Description
0	(0) STRUCTURE	12	ADRBMB	
0	(0) CHARACTER	12	BMHDR	BITMAP HEADER
0	(0) CHARACTER	4	BMID	BLOCK IDENTIFIER EBCDIC "BMBB"
4	(4) A-ADDRESS	4	BMNPTR	ADDR OF NEXT BITMAP SEGMENT/O
8	(8) SIGNED	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

**CROSS REFERENCE**

ADRBMB	0	(0)
BMBITMAP	12	(C)
BMHDR	0	(0)
BMID	0	(0)
BMNPTR	4	(4)
BMNTRK	8	(8)

Figure 94. ADRBMB Data Area

# ADRTAPB Data Area

Offsets	Type	Length	Name	Description
0	(0) STRUCTURE	16	ADRTAPB	
0	(0) CHARACTER	6	DTPSCHK	
0	(0) SIGNED	4	DTPSEQNO	SEGMENT SEQUENCE NUMBER
4	(4) UNSIGNED	1	DTPNOSEG	NUMBER OF SEGMENTS PER RECORD
5	(5) UNSIGNED	1	DTPSEGNO	SEGMENT NUMBER OF RECORD
6	(6) UNSIGNED	2	DTPSEGLN	SEGMENT LENGTH INCL PREFIX
8	(8) UNSIGNED	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 DATA SET DUMP
	.... 1111			RESERVED
10	(A) BITSTRING	1	DTPRCID1	RECORD IDENTIFIER 1
	1... ..		DTPVHDR	80 VOLUME HEADER
	.1... ..		DTPTHDR	TAPE HEADER
	..1... ..		DTPBTM	40 MAP OF DUMPED TRACKS
	...1... ..		DTPDSNL	DATA SET NAME/CATALOG LIST
	.... 1... ..		DTPTRKO	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	DATA SET TRAILER
	..... ..1... ..		DTPVVDS	02 VVDS TRACK
	..... ..1... ..		DTPFILLR	01
11	(B) BITSTRING	1	DTPRCFL1	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
	.1... ..		DTPDDSP	IF ON, LAST SEGMENT DISPLACED DATA MUST BE REPLACED FIRST (SEE DTPDDISP)
	..11 1111			RESERVED
12	(C) CHARACTER	4		RESERVED
16	(10) CHARACTER	0	DTPBODY	START OF REMAINDER OF RECORD

Figure 95 (Part 1 of 6). ADRTAPB Data Area

## ADRTAPB Data Area (continued)

Offsets	Type	Length	Name	Description
<b>DTVOL</b>				
0	(0) STRUCTURE	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(DS4DEVSZ)
8	(8) UNSIGNED	2	DTVLOGCY	NUMBER OF CYLINDERS IN VOLUME
10	(A) UNSIGNED	2	DTVTRKCY	NUMBER OF TRACKS PER CYLINDER
12	(C) UNSIGNED	2	DTVBLKSZ	MAXIMUM BLOCKSIZE
14	(E) UNSIGNED	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 OOOYDDDC
27	(1B) CHARACTER	4	DTVTIME	TIME OF DAY IN DECIMAL
31	(1F) CHARACTER	4	DTVDEVTY	DEVTYP OF VOLUME(UCBTBYT4)
35	(23) UNSIGNED	1	DTVMODNO	MODEL NUMBER
36	(24) BITSTRING	1	DTVIND1	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) UNSIGNED	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
	...1 ....		DTVGVI	10-VSAM DATA SETS NOT ON ANY VOLUME
	.... 1111			UNUSED
40	(28) UNSIGNED	2	DTVLEN	LEN OF HEADER = SEGMENT LENGTH PREFIX
42	(2A) UNSIGNED	1	DTVVERNO	DFDSS VERSION NUMBER
43	(2B) UNSIGNED	1	DTVLVLNO	DFDSS MODIFIC. NUMBER
44	(2C) UNSIGNED	1	DTVVXTNO	NUMBER OF VVDS EXTENTS
45	(2D) UNSIGNED	1	DTVVXTOF	OFFSET TO VVDS EXTENT
46	(2E) CHARACTER	*	DTVEND	FIELDS FROM DTVOL START
<b>DTVVXTNT</b>				
0	(0) STRUCTURE	0	DTVVXTNT	VVDS EXTENTS, IN DSCB
0	(0) CHARACTER	10	DTVVXLEN	

Figure 95 (Part 2 of 6). ADRTAPB Data Area



# ADRTAPB Data Area (continued)

Offsets	Type	Length	Name	Description
<b>DTHDR</b>				
0	(0) STRUCTURE	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) UNSIGNED	2	DTHLEN	HEADER LENGTH
11	(B) UNSIGNED	1	DTHVERNO	DFDSS VERSION NUMBER
12	(C) UNSIGNED	1	DTHLVLNO	DFDSS MODIFIC. NUMBER
13	(D) UNSIGNED	2	DTHBLKSZ	MAX BLKSIZE
15	(F) UNSIGNED	2	DTHNDS	NO. POTENTIAL DS
17	(11) CHARACTER	1	DTHIND1	INDICATORS
	1... ..		DTHFCMP	FILE COMPRESSED
	.1.. ..		DTHUNLCD	UNALLOCATED SPACE DUMPED
	..11 1111			RESERVED
<b>DTLDSN</b>				
0	(0) STRUCTURE	45	DTLDSN	
0	(0) UNSIGNED	1	DTLLEN	LENGTH OF DATA SET NAME
1	(1) CHARACTER	44	DTLCAT	CATALOG NAME
45	(2D) CHARACTER	*	DTLDCMP	COMPRESSED DATA SET NAME
<b>DTDSHDR</b>				
0	(0) STRUCTURE	59	DTDSHDR	
0	(0) UNSIGNED	1	DTDLEN	LENGTH OF DATA SET NAME
1	(1) UNSIGNED	1	DTDCATLN	LENGTH OF CATALOG NAME
2	(2) CHARACTER	2	DTDDSORG	DATA SET ORGANIZATION (FROM F1)
4	(4) CHARACTER	1	DTDOPTCD	DS OPTION CODE (FROM F1)
5	(5) CHARACTER	1	DTDNVOL	NUMBER VOLS FOR DATA SET
6	(6) CHARACTER	1	DTDIND	DATA SET INDICATOR
	1... ..		DTDIPWD	PASSWORD SUPPLIED 1=YES
	.1.. ..		DTDTPWD	TYPE PASSWORD 0=D.S. 1=CATALOG
	..1. ....		DTDIRACF	RACF PROFILE
	...1 1...		DTDRACFP	RACF PROFILE FLAGS
	...1 ....		DTDRACFD	1 = RACF DISCRETE PROFILE
	.... 1...		DTDRACFG	1 = RACF GENERIC PROFILE
	.... .111			RESERVED
7	(7) CHARACTER	8	DTDPWD	PASSWORD
15	(F) CHARACTER	44	DTDCAT	CATALOG NAME
59	(3B) CHARACTER	*	DTDDSN	COMPRESSED DATA SET NAME

Figure 95 (Part 3 of 6). ADRTAPB Data Area

## ADRTAPB Data Area (continued)

Offsets	Type	Length	Name	Description
<b>DTMVOL</b>				
0	(0) STRUCTURE	28	DTMVOL	
0	(0) CHARACTER	6	DTMVSERL	VOLUME SERIAL ID
6	(6) CHARACTER	4	DTMDEVTY	DEVTYPE (UCBTBYT4)
12	(C) CHARACTER	8	DTMVOLSZ	VOLUME SIZE (DS4DEVSZ)
12	(C) UNSIGNED	4	DTMTRKCP	NUMBER OF BYTES/TRK (TRACK CAPACITY WITH OVERHEAD)
16	(10) UNSIGNED	2	DTMLOGCY	NUMBER OF CYLINDERS PER VOLUME
18	(12) UNSIGNED	2	DTMTRKCY	NUMBER OF TRACKS PER VOLUME
20	(14) UNSIGNED	2	DTMMAXCB	MAX COMPRESS BUF IN WORDS
22	(16) CHARACTER	2	DTMIND	VOLUME INDICATORS
	1... ..		DTMVIRT	VIRTUAL VOLUME
	.1... ..		DTMMINI	MINIVOLUME
	..1. ....		DTMCVAF	VOLUME HAS INDEXED VTOC
	...1 1111			RESERVED
24	(18) UNSIGNED	1	DTM#VVRS	NUMBER OF VVRS DUMPED
25	(19) UNSIGNED	1	DTM#DSCB	NUMBER OF DSCBS DUMPED
26	(1A) UNSIGNED	1	DTM#EXT	NUMBER OF EXTENTS DUMPED
27	(1B) CHARACTER	1	DTMMODNO	MODEL NUMBER
28	(1C) CHARACTER	*	DTMVDATA	DSCBS (1&2) FOR DS, FOLLOWED BY EXTENT LIST, FOLLOWED BY VVRS FOR DATA SET CLUSTER
<b>DTBTMR</b>				
0	(0) STRUCTURE	12	DTBTMR	
0	(0) CHARACTER	12	DTBHDR	RESERVED
0	(0) CHARACTER	4	DTBBMID	ID OF THE BLOCK = BMBB
4	(4) SIGNED	4	DTBADDR	OF NEXT BMBB SEGMENT COPIED FROM ADRBMB BLOCK
8	(8) UNSIGNED	2	DTBTRK#	NUMBER OF TRKS MAPPED IN BMBB SEG
10	(A) CHARACTER	2		RESERVED
12	(C) CHARACTER	*	DTBBITM	BITMAP

Figure 95 (Part 4 of 6). ADRTAPB Data Area

## ADRTAPB Data Area (continued)

Offsets	Type	Length	Name	Description
<b>DTTTRK</b>				
0	(0) STRUCTURE	24	DTTTRK	MAPS THE TRACK
0	(0) CHARACTER	16	DTTHDR	HEADER FOR EACH TRACK SEGMENT
0	(0) UNSIGNED	2	DTTTRKLN	LENGTH OF DATA ON TRK
2	(2) CHARACTER	1	DTTTRKID	TRACK INDICATORS
	1... ..		DTTIOER	I/O ERROR ON TRACK
	.1.. ..		DTTTROVF	LAST REC ON TRK IS OVFLD REC
	..1. ....		DTTTCMP	IF ON, TRACK COMPRESSED
	...1 ....		DTTVFRST	FIRST VVDS RECORD
	.... 1...		DTTINVT	INVALID TRACK FORMAT
	.... .1..		DTTSTAT	USER STATISTICAL RECORD
	.... ..11			RESERVED
3	(3) CHARACTER	4	DTTCCHH	CCHH OF TRACK
3	(3) UNSIGNED	2	DTTCC	CYLINDER NUMBER
5	(5) UNSIGNED	2	DTTHH	TRACK NUMBER
7	(7) CHARACTER	6	DTTVSERL	VOLUME SERIAL
16	(10) CHARACTER	8	DTTRODAT	RCRD 0 DATA, ONLY ON 1ST SEG
24	(18) CHARACTER	0	DTTBODY	R1- RN RECORDS ON TRACK
<b>DTTCKD</b>				
0	(0) STRUCTURE	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) UNSIGNED	2	DTTCCC	CC OF RECORD
2	(2) UNSIGNED	2	DTTCHH	HH OF RECORD
4	(4) UNSIGNED	1	DTTCRCRD	R OF RECORD
5	(5) UNSIGNED	1	DTTKELEN	KEY LENGTH
6	(6) UNSIGNED	2	DTTDATLN	DATA LENGTH
8	(8) CHARACTER	*	DTTKD	KEY AND DATA FIELDS
<b>DTRTLR</b>				
0	(0) STRUCTURE	6	DTRTLR	
0	(0) CHARACTER	6	DTRSERL	VOLUME SERIAL OF FILE

Figure 95 (Part 5 of 6). ADRTAPB Data Area

CROSS REFERENCE

DTBADDR	4 (4)	DTMVOLSZ	12 (C)	DTTTRKLN	0 (0)
DTBBITM	12 (C)	DTMVSERL	0 (0)	DTTTRQVF	2 X'40'
DTBBMID	0 (0)	DTPBODY	16 (10)	DTTVFRST	2 X'10'
DTBHDR	0 (0)	DTPBTM	10 X'40'	DTTVSERL	7 (7)
DTBTMR	0 (0)	DTPDASDD	9 X'20'	DTVBLKSZ	12 (C)
DTBTRK#	8 (8)	DTPDATA	10 X'08'	DTVBMSZ	37 (25)
DTDCAT	15 (F)	DTPDDDSP	11 X'40'	DTVCVAF	36 X'20'
DTDCATLN	1 (1)	DTPDDISP	11 X'80'	DTVDAY	23 (17)
DTDDSN	59 (3B)	DTPDMPID	9 (9)	DTVDEVTY	31 (1f)
DTDDSORG	2 (2)	DTPDSHDR	10 X'20'	DTVEND	46 (2E)
DTDIND	6 (6)	DTPDSNL	10 X'40'	DTVFCMP	36 X'08'
DTDIPWD	6 X'80'	DTPDTRLR	10 X'04'	DTVGNVI	39 X'20'
DTDIRACF	6 X'20'	DTPFILLR	10 X'01'	DTVGI	39 X'10'
DTDLN	0 (0)	DTPFULD	9 X'80'	DTVIND1	36 (24)
DTDNVOL	5 (5)	DTPLQGL	9 X'10'	DTVIND2	39 (27)
DTDOPTCD	4 (4)	DTPNOSEG	4 (4)	DTVLEN	40 (28)
DTDPWD	7 (7)	DTPPARTD	9 X'40'	DTVLNVI	39 X'80'
DTDRACFD	6 X'10'	DTPPFXLN	8 (8)	DTVLOGCY	8 (8)
DTDRACFG	6 X'08'	DTPRCFL1	11 (B)	DTVLVF	36 X'02'
DTDRACFP	6 X'18'	DTPRCID1	10 (A)	DTVLVI	39 X'40'
DTDSHDR	0 (0)	DTPSCHK	0 (0)	DTVLVLNO	43 (2B)
DTDTPWD	6 X'40'	DTPSEGLN	6 (6)	DTVMAXCB	14 (E)
DTHBLKSZ	13 (D)	DTPSEGNO	5 (5)	DTVMINI	36 X'40'
DTHDAY	0 (0)	DTPSEQNO	0 (0)	DTVMODNO	35 (23)
DTHDR	0 (0)	DTPTHDR	10 X'80'	DTVOL	0 (0)
DTHFCMP	17 X'80'	DTPTRKO	10 X'20'	DTVOLSZ	8 (8)
DTHGNVI	8 X'80'	DTPVHDR	10 X'80'	DTVSERL	16 (10)
DTHGVI	8 X'40'	DTPVOLD	10 X'10'	DTVTIMD	23 (17)
DTHIND1	17 (11)	DTPVTOC	10 X'10'	DTVTIME	27 (1B)
DTHIND2	8 (8)	DTPVTRLR	10 X'04'	DTVTOCB	0 (0)
DTHLEN	9 (9)	DTPVVD	10 X'02'	DTVTOCE	4 (4)
DTHLVLNO	12 (C)	DTRSERL	0 (0)	DTVTRKCY	10 (A)
DTHNDS	15 (F)	DTRTLR	0 (0)	DTVUNLCD	36 X'04'
DHTIMD	0 (0)	DTTBDY	24 (18)	DTVVERNO	42 (2A)
DHTIME	4 (4)	DTTCC	3 (3)	DTVVIRT	36 X'80'
DTHUNLCD	17 X'40'	DTTCCC	0 (0)	DTVVTOCI	22 (16)
DTHVERNO	11 (B)	DTTCCCHH	0 (0)	DTVVVDS	36 X'01'
DTLCAT	1 (1)	DTTCCHH	3 (3)	DTVVXLEN	0 (0)
DTLDSCMP	45 (2D)	DTTCCHR	0 (0)	DTVVXTNO	44 (2C)
DTLDSN	0 (0)	DTTCHH	2 (2)	DTVVXTNT	0 (0)
DTLLEN	0 (0)	DTTCKD	0 (0)	DTVVXTOF	45 (2D)
DTM#DSCB	25 (19)	DTTCNT	0 (0)		
DTM#EXT	26 (1A)	DTTCRCRD	4 (4)		
DTM#VVRS	24 (18)	DTTDATLN	6 (6)		
DTMCVAF	22 X'20'	DTTHDR	0 (0)		
DTMDEVTY	6 (6)	DTTHH	5 (5)		
DTMIND	22 (16)	DTTINVT	2 X'08'		
DTMLOGCY	16 (10)	DTTIOER	2 X'80'		
DTMMAXCB	20 (14)	DTTKD	8 (8)		
DTMMINI	22 X'40'	DTTKELN	5 (5)		
DTMMODNO	27 (1B)	DTTRODAT	16 (10)		
DTMTRKCP	12 (C)	DTTSTAT	2 X'04'		
DTMTRKCY	18 (12)	DTTTCMP	2 X'20'		
DTMVDATA	28 (1C)	DTTTRK	0 (0)		
DTMVIRT	22 X'80'	DTTTRKID	2 (2)		
DTMVOL	0 (0)				

Figure 95 (Part 6 of 6). ADRTAPB Data Area



## Appendix E. Quick Reference to DFDSS Function Commands

Figure 96 through Figure 106 serve as a handy reference when you code DFDSS commands.

Command	Parameters
COMpress	<pre> {{FILterdd   FDD} (ddn) }   { [ INClude (dsn [...]) ]   [ EXClude (dsn [...]) ]   [ BY((schar,op,arg) [...]) ] }  { DDName (ddn [...])     DYNAM ((volser [,unit] ) [...]) }  Optional parameters that can be used with COMPRESS:  DYNALloc  PASsword   PSWD ( [ddn]                   [dsn/pswd [...]] )  WAIT(numsecs,numretrys) </pre>

Figure 96. COMPRESS Command Syntax

Command	Parameters
COPY	<pre> <b>{{FULL   FUL}  </b> <b>{TRACKS   TRKS}((c1,h1,c2,h2) [...]) }</b>  <b>{{INDDname   IDD}(ddn)  </b> <b>{INDYnam   IDY}(volser [,unit] ) }</b>  <b>{{OUTDDname   ODD}(ddn)  </b> <b>{OUTDYnam   ODY}(volser [,unit] ) }</b>  Optional parameters that can be used with FULL:  <b>ALLData { (dsn [...])   (*) }</b>  <b>ALLExcp   ALLX</b>  <b>CANcelerror</b>  <b>COPYValid   CPYV</b>  <b>PASsword   PSWD ( [ddn]</b> <b>                    [dsn/pswd [...]]</b> <b>                    [catname/pswd [...]])</b>  <b>PURge   PRG</b>  <b>TOLerate (IOError)</b>  <b>WAIT (numsecs,numretrys)</b>  <b>WRItcheck   WRC</b>  Optional parameters that can be used with TRACKS:  <b>CANcelerror</b>  <b>COPYValid   CPYV</b>  <b>OUTTRacks   OUTTRKS (( c1,h1) [...])</b>  <b>PASsword   PSWD ( [ddn [...]]</b> <b>                    [dsn/pswd [...]]</b> <b>                    [catname/pswd [...]])</b>  <b>PURge   PRG</b>  <b>TOLerate (IOError)</b>  <b>WAIT (numsecs,numretrys)</b>  <b>WRItcheck   WRC</b> </pre>

Figure 97. COPY Command Syntax—FULL or TRACKS

Command	Parameters
COPY	<pre> {{DATASet   DS} ( {FILterdd   FDD} ( ddn) or {DATASet   DS} ( [ INClude(dsn[,...]) ] [ EXClude(dsn[,...]) ] [ BY((schar,op,arg) [,...]) ] ) {{OUTDDname   ODD}( ddn [,...])   {OUTDYnam   ODY}((volser [,unit]) [,...]) }  Optional parameters that can be used with DATASET:  ALLData { (dsn [,...])   (*) }  ALLExp   ALLX  ALLMulti  CANcelerror  CATalog   RECATalog(newcatname)   RECATalog(*)  DELeTe  DYNALloc  FORce  {{INDDname   IDD} ( ddn [,...])   {INDYnam   IDY} ((volser [,unit]) [,...])   {LOGINDDname   LIDD} ( ddn [,...])   {LOGINDYnam   LIDY} ((volser [,unit]) [,...]) }  MENtity (modeldsn) [MVOlser (volser)]  PASsword   PSWD ( [ ddn] [ dsn/pswd [,...]] [ catname/pswd [,...]] )  PERCENTUtilized   PCTU ( n [,...])  PURge   PRG  REBLock ( dsn [,...])  RELBlockaddress   RELBLKA ( dsn [,...])  RENAMEUnconditional   RENUNC { (pfx)   ((on,nn)[,...])   ((pfx),(on,nn) [,...]) }  REPlace </pre>

Figure 98 (Part 1 of 2). COPY Command Syntax—DATASET



Command	Parameters
<b>COPY</b>	<p>Optional parameters that can be used with DATASET (continued):</p> <p><b>SHAre   SHR</b></p> <p><b>TGTAlloc(BLK   CYL   TRK   SOURCE   SRC)</b></p> <p><b>TOLerate([IOERror] [ENQFailure])</b></p> <p><b>TRACKOverflow   TRKO (dsn[,...])</b></p> <p><b>UNCATalog</b></p> <p><b>WAIT (numsecs,numretrys)</b></p> <p><b>WRItcheck   WRC</b></p>

Figure 98 (Part 2 of 2). COPY Command Syntax—DATASET

Command	Parameters
<b>COPYDump or CPYD</b>	<p><b>INDDname   IDD (ddn)</b></p> <p><b>OUTDDname   ODD (ddn[,...])</b></p> <p>The following parameter is optional:</p> <p><b>LOGicalvolume   LVOL (volser[,...])</b></p>

Figure 99. COPYDUMP Command Syntax

Command	Parameters
DEFrag	<p data-bbox="786 239 1097 306"> <b>{ DDName (<i>ddn</i>)    DYNAM (<i>volser</i> [,<i>unit</i>] ) }</b> </p> <p data-bbox="740 338 1357 369">Optional parameters that can be used with DEFrag:</p> <p data-bbox="799 396 1192 459"> <b>BY { (LIST((<i>schar,op,arg</i>) [...]))    (DDName(<i>ddn</i>)) }</b> </p> <p data-bbox="799 485 930 516"><b>DYNALloc</b></p> <p data-bbox="799 543 1136 606"> <b>EXClude { (LIST(<i>dsn</i> [,... ]))    (DDName(<i>ddn</i>)) }</b> </p> <p data-bbox="799 632 1187 663"><b>FRAGmentationindex   FRAGI (<i>n</i>)</b></p> <p data-bbox="799 688 956 720"><b>MAXmove(<i>n</i>)</b></p> <p data-bbox="799 745 1248 808"> <b>PASsword   PSWD ( [<i>ddn</i>]  [<i>dsn/pswd</i> [,...]] )</b> </p> <p data-bbox="799 833 1097 865"><b>WAIT(<i>numsecs,numretrys</i>)</b></p>

Figure 100. DEFrag Command Syntax

Command	Parameters
<b>DUMp</b>	<pre> <b>{{FULL   FUL }  </b> <b>{TRACKS   TRKS}{(c1,h1,c2,h2) [...]} }</b>  <b>{{INDDname   IDD}{(ddn)  </b> <b>{INDYnam   IDY} (volser[,unit]) }</b>  <b>OUTDDname   ODD (ddn [...])</b>  Optional parameters that can be used with FULL:  <b>ALLData { (dsn [...])   (*) }</b>  <b>ALLExcp   ALLX</b>  <b>CANceIerror</b>  <b>COMpress</b>  <b>OPTimize (n)</b>  <b>PASsword   PSWD ( [ddn]</b> <b>                    [dsn/pswd [...]]</b> <b>                    [catname/pswd [...]] )</b>  <b>RESEt</b>  <b>TOLerate(IOERror)</b>  <b>TRACkOverfIow   TRKO (dsn [...])</b>  <b>WAIT (numsecs,numretrys)</b>  Optional parameters that can be used with TRACKS:  <b>CANceIerror</b>  <b>COMpress</b>  <b>OPTimize (n)</b>  <b>PASsword   PSWD ( [ddn [...]]</b> <b>                    [dsn/pswd [...]]</b> <b>                    [catname/pswd [...]] )</b>  <b>TOLerate(IOERror)</b>  <b>TRACkOverfIow   TRKO (dsn [...])</b>  <b>WAIT (numsecs,numretrys)</b> </pre>

Figure 101. DUMP Command Syntax—FULL or TRACKS

Command	Parameters
<b>DUMp</b>	<p> <b>{DATASet   DS} ({FILterdd   FDD} ( <i>ddn</i> ))</b>  <b>or</b>  <b>{DATASet   DS} ( [ INClude(<i>dsn</i>[...]) ]</b>  <b>          [ EXClude(<i>dsn</i>[...]) ]</b>  <b>          [ BY((<i>schar</i>,<i>op</i>,<i>arg</i>) [...]) ] )</b>    <b>OUTDDname   ODD ( <i>ddn</i> [...])</b> </p> <p>Optional parameters that can be used with DATASET:</p> <p> <b>ALLData { (<i>dsn</i> [...])   (*) }</b>    <b>ALLExcp   ALLX</b>    <b>ALLMulti</b>    <b>CANcelerror</b>    <b>COMpress</b>    <b>DELeTe</b>    <b>DYNALloc</b>    <b>{{INDDname   IDD} ( <i>ddn</i> [...])</b>    <b>{INDYnam   IDY} ((<i>volser</i> [,<i>unit</i>]) [...])</b>    <b>{LOGINDDname   LIDD} ( <i>ddn</i> [...])</b>    <b>{LOGINDYnam   LIDY} ((<i>volser</i> [,<i>unit</i>]) [...]) }</b> </p> <p> <b>OPTimize ( <i>n</i> )</b>    <b>PASsword   PSWD ( [ <i>ddn</i> ]</b>  <b>                  [ <i>dsn/pswd</i> [...] ]</b>  <b>                  [ <i>catname/pswd</i> [...] ] )</b> </p> <p> <b>PURge   PRG</b>    <b>RESEt</b>    <b>SHAre   SHR</b>    <b>TOLerate([IOERror]</b>  <b>          [ENQFailure])</b>    <b>TRACkOverFlow   TRKO ( <i>dsn</i> [...])</b>    <b>UNCATalog</b>    <b>WAIT ( <i>numsecs</i>,<i>numretrys</i> )</b> </p>

Figure 102. DUMP Command Syntax—DATASET

Command	Parameters
PRInt or PRT	<pre> {{DATASet   DS} (dsn)   {TRACKS   TRKS} ((c1,h1,c2,h2) [...])   VTOC(a,b) }  {{INDDname   IDD} (ddn)   {INDYnam   IDY} (volser [,unit] ) } </pre> <p>Optional parameters that can be used with DATASET:</p> <p><b>ALLData</b></p> <p><b>DATALength(n)</b></p> <p><b>DYNALloc</b></p> <p><b>ERRortracks</b></p> <p><b>KEYlength(n)</b></p> <p><b>OUTDDname   ODD (ddn)</b></p> <p><b>PASsword   PSWD ( [ddn] [dsn/pswd [...]] )</b></p> <p><b>SHAre   SHR</b></p> <p><b>TOLerate([ENQFailure] [IOERror])</b></p> <p><b>WAIT (numsecs,numretrys)</b></p> <p>Optional parameters that can be used with TRACKS or VTOC:</p> <p><b>DATALength(n)</b></p> <p><b>ERRortracks</b></p> <p><b>KEYlength(n)</b></p> <p><b>OUTDDname   ODD (ddn)</b></p> <p><b>PASsword   PSWD ( [ddn [...]] [dsn/pswd [...]] )</b></p> <p><b>TOLerate(IOERror)</b></p> <p><b>WAIT (numsecs,numretrys)</b></p>

Figure 103. PRINT Command Syntax

Command	Parameters
RELease	<pre> {{FILterdd   FDD} (ddn) }   { [ INClude (dsn [...]) ]   [ EXClude (dsn [...]) ]   [ BY((schar,op,arg) [...]) ] }  { DDName (ddn [...])              DYNAM ((volser [,unit]) [...]) }  Optional parameters that can be used with RELEASE:  DYNALloc  MINSECqty(n)  MINTRAcksunused   MINTRKS (n)  PASsword   PSWD ( [ddn]                   [dsn/pswd [...]] )  WAIT(numsecs,numretrys) </pre>

Figure 104. RELEASE Command Syntax

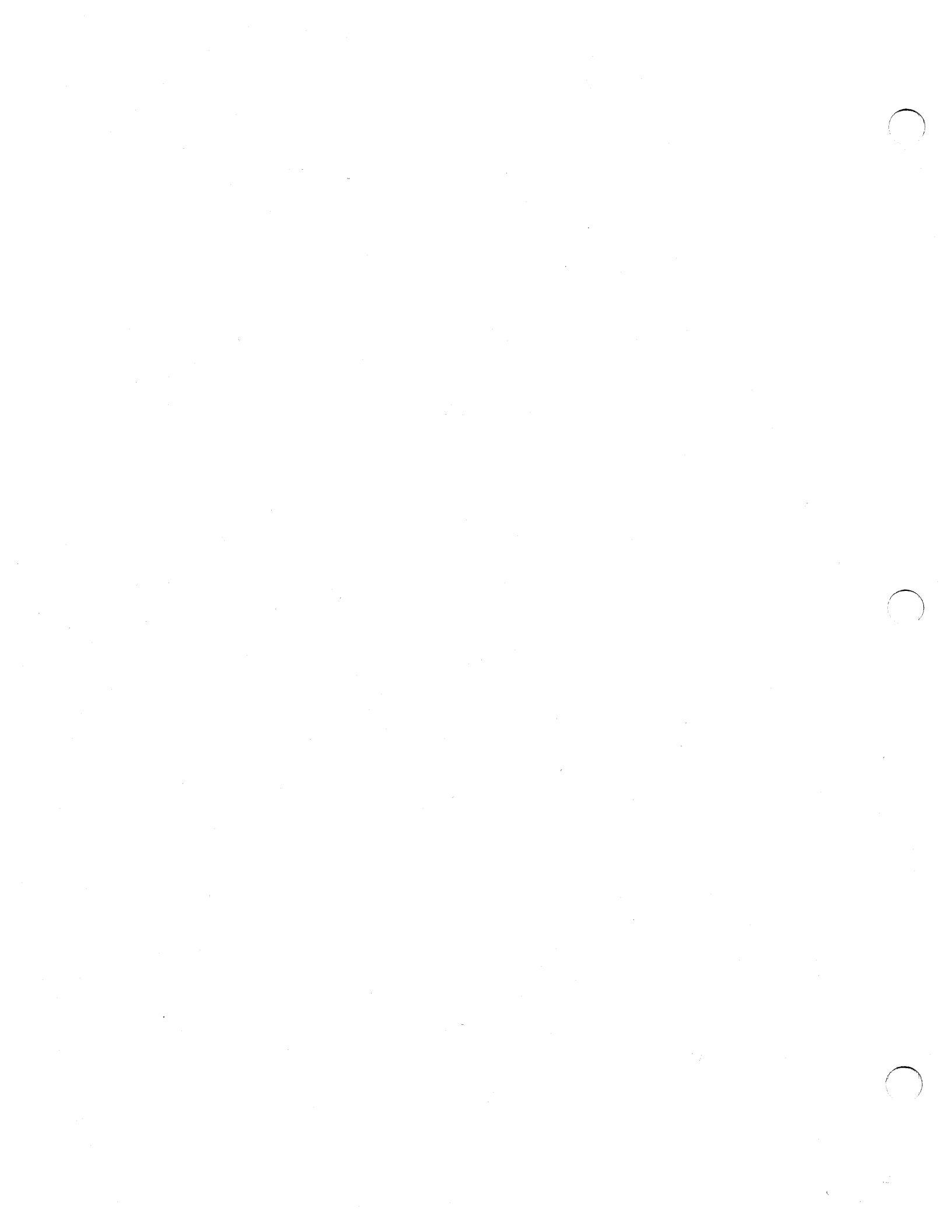
Command	Parameters
RESTore	<pre> <b>{{FULL   FUL }  </b> <b>{TRACKS   TRKS }((c1,h1,c2,h2) [...]) }</b>  <b>INDDname   IDD (ddn)</b>  <b>{{OUTDDname   ODD}(ddn)  </b> <b>{OUTDYnam   ODY} (volser [,unit]) }</b>  Optional parameters that can be used with FULL:  <b>CANcelerror</b>  <b>COPYVolid   CPYV</b>  <b>PASsword   PSWD ( [ddn]</b> <b>                    [dsn/pswd [...]]</b> <b>                    [catname/pswd [...]] )</b>  <b>PURge   PRG</b>  <b>WAIT(numsecs,numretrys)</b>  <b>WRIttecheck   WRC</b>  Optional parameters that can be used with TRACKS:  <b>CANcelerror</b>  <b>COPYVolid   CPYV</b>  <b>OUTTRacks   OUTTRKS ((c1,h1) [...])</b>  <b>PASsword   PSWD ( [ddn [...]]</b> <b>                    [dsn/pswd [...]]</b> <b>                    [catname/pswd [...]] )</b>  <b>PURge   PRG</b>  <b>WAIT(numsecs,numretrys)</b>  <b>WRIttecheck   WRC</b> </pre>

Figure 105. RESTORE Command Syntax—FULL or TRACKS

Command	Parameters
RESTore	<p> <b>{DATASet   DS} ({FILterdd   FDD} (ddn))</b>  <b>or</b>  <b>{DATASet   DS} ( [ INClude(dsn,...) ]</b>  <b>[ EXClude(dsn,...) ]</b>  <b>[ BY((schar,op,arg) [...]) ] )</b> </p> <p> <b>INDDname   IDD (ddn)</b> </p> <p> Optional parameters that can be used with DATASET: </p> <p> <b>CANcelerror</b> </p> <p> <b>CATalog   RECATalog(newcatname)   RECATalog(*)</b> </p> <p> <b>DYNALloc</b> </p> <p> <b>FORce</b> </p> <p> <b>LOGicalvolume   LVOL (volser [...])</b> </p> <p> <b>MENtity(modeldsn) [MVOlser(volser)]</b> </p> <p> <b>{{OUTDDname   ODD}(ddn [...])  </b>  <b>{OUTDYnam   ODY}((volser [,unit] ) [...]) }</b> </p> <p> <b>PASsword   PSWD ( [ddn]</b>  <b>[dsn/pswd [...]]</b>  <b>[catname/pswd [...]] )</b> </p> <p> <b>PERCENTUtilized   PCTU ( n [...])</b> </p> <p> <b>REName { (pfx)  </b>  <b>((on,nn) [...])  </b>  <b>((pfx),(on,nn) [...]) }</b> </p> <p> <b>RENAMEUnconditional   RENUNC { (pfx)  </b>  <b>((on,nn) [...])  </b>  <b>((pfx),(on,nn) [...]) }</b> </p> <p> <b>REPlace</b> </p> <p> <b>SHAre   SHR</b> </p> <p> <b>TGTALloc(BLK   CYL   TRK   SOURCE   SRC)</b> </p> <p> <b>TOLerate(ENQFailure)</b> </p> <p> <b>WAIT(numsecs,numretrys)</b> </p> <p> <b>WRIttecheck   WRC</b> </p> <p> <b>Note:</b> If "PURGE" is specified, a message will indicate that "REPLACE" was used instead. </p>

Figure 106. RESTORE Command Syntax—DATASET





## Glossary

The following terms are defined as they are used in this book. If you do not find the term you are looking for, refer to the index or to the *IBM Dictionary of Computing*, SC20-1699.

**ABSTR.** A subparameter of the SPACE parameter in a DD statement. It indicates that specified tracks be assigned to a data set.

**ALLOC.** A space allocation parameter which indicates type, such as cylinders, tracks, etc.

**APF.** Authorized program facility.

**block length.** A subparameter of the SPACE parameter in a DD statement, specifying the average block length of the data in a data set. The system computes how many tracks to allocate.

**CYL.** A subparameter of the SPACE parameter in a DD statement, specifying that space is to be allocated by cylinder.

**DADSM.** Direct access device space management program that maintains the VTOC, VTOCIX and the space on a volume.

**data compression (run-length).** A method of encoding repetitive series of identical characters so that they occupy less space on a dump tape. Supported by both physical and logical processing.

**data set change indicator.** Indicates whether the data set was opened for output. This indicator is 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.

**DIAGNOSE.** An access method services command used to scan 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 execution 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.

**DOS bit.** On a volume without an indexed VTOC, this bit indicates that the free space map (format-5 DSCB) is invalid.

**DSCB.** Data set control block.

**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).

**EC mode.** Extended control mode.

**ENQ.** A system macro used to serialize the access to a data set.

**EXCP.** Execute channel program. A macro used to access a data set without specifying the organization.

**format-1 DSCB (F1 DSCB).** An identifier DSCB. 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. Describes the indexes of an ISAM data set. See the appropriate system data areas manual.

**format-3 DSCB (F3 DSCB).** An extension DSCB. 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. 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. 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.

**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.

**like device.** A (DASD) device with identical geometry; that is, the same number of bytes per track, number of tracks per cylinder, and number of cylinders per actuator.

**logical (data set) dump.** A dump that results from a logical DUMP operation.

**logical (data set) DUMP operation.** A DUMP operation in which logical processing is performed.

**logical (data set) processing.** Processing in which data sets are treated as logical entities, through the use of catalogs; that is, DFDSS processes an entire data set before it begins processing the next data set.

**logical (data set) RESTORE operation.** A RESTORE operation which uses as input a data set that was produced by a logical DUMP operation.

**logical volume.** The output produced from a DUMP operation, all of whose data was derived from a single DASD volume.

**minivolume.** When an MVS system runs on VM/370, you can have OS/VS-formatted VM/370 minidisks, with their size equal to or less than that of the real volume. These are known as "minivolumes." DFDSS recognizes them as such. It uses the device size specified in the format-4 DSCB in their VTOC. Minivolumes are supported only by the system version of DFDSS.

**MVS/Extended Architecture (MVS/XA).** An operating system that supports a virtual memory address space larger than 16 megabytes.

**PAM.** Partitioned access method.

**physical (data set) dump.** A dump that results from a physical DUMP operation.

**physical (data set) DUMP operation.** A DUMP operation in which physical processing is performed.

**physical (data set) processing.** Processing in which even multivolume data sets are processed as though they resided on a single volume only, through the use of its VTOC; that is, DFDSS processes only that part of a data set residing on the volume being processed.

**physical (data set) RESTORE operation.** A RESTORE operation which uses as input a data set that was produced by a physical DUMP operation.

**PSW.** Program status word.

**RACF.** Resource Access Control Facility

**RBA.** Relative byte address.

**RELCREDT.** Relative creation date.

**RELEXPDT.** Relative expiration date.

**RELREFDT.** Relative last referenced date.

**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.

**run-length data compression.** See data compression (run-length).

**SAM.** Sequential access method.

**similar device.** A (DASD) device with the same number of bytes per track and tracks per cylinder.

**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, specifying that space is to be allocated by tracks.

**UCB.** Unit control block.

**unlike device.** A (DASD) device with a different number of bytes per track, tracks per cylinder, or both.

**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.

**VVDS.** The VSAM volume data set. It describes the VSAM data sets, on a volume, that are cataloged in the integrated catalog facility catalog. The name of the data set is SYS1.VVDS.Vvolser.

# Index

## Special Characters

- \* (single asterisk) used in partially qualified data set names 22
- \*\* (double asterisk) used in partially qualified data set names 22

## A

- abbreviations permitted in commands and keywords 30
- abend dumps, data set passwords 175
- ABEND=nnn in the EXEC statement 19
- ABOVE16 in the EXEC statement 19
- ABSTR, definition 321
- access authority, compatibility with Version 1 24
- Access Method Services 61
- addressing mode 231
- ADRBMB data area 301
- ADRDUSSU 190
- ADREID0 data area 219
- ADRMCLVL macro to dynamically determine the DFDSS version, release, and modification level 159
- ADRBLKB mapping macro 243
- ADRREBLK routine 243
- ADRTAPB data area 302
- ADRUENQ routine 235
- ADRUFO data area 239
- ADRUFO mapping macro 238
- ADRUM module 193
- ADRUIXIT routine 237
- ADRUNQB mapping macro 235
- ADRUPB mapping macro 233
- ADRUPSWD routine 232
- AIX (alternate index), recataloging 48
- ALLDATA parameter
  - COPY command 46
  - DUMP command 86
  - PRINT command 100
- ALLEXCP parameter
  - COPY command 46
  - DUMP command 87
- ALLMULTI parameter
  - COPY command 47
  - DUMP command 87
- ALLOC subparameter
  - COMPRESS command 34
  - COPY command 44
  - DEFRAG command 72
  - DUMP command 84
  - RELEASE command 105
  - RESTORE command 113
- ALLOC, definition 321
- allocating volumes dynamically 160

- allocation of space 52, 119
- ALLX parameter
  - COPY command 46
  - DUMP command 87
- alter an installation option via the UIM 207
- ALTER LOCK, IDCAMS command 130
- alternate index, recataloging 48
- alternate index, restoring 133
- AMODE=31 191, 231
- APF, definition 321
- application interface 189
- application interface blocks 193
- Application Interface summary 197
- asterisks, rules regarding, in a data set qualifier 22
- auditing information 189
- authorization checking 232
  - data set 169
  - general description 167
  - volume 168
- authorization exit routine, user 232
- auxiliary commands 135-141

## B

- backing up (dumping) data 79
- backup copies, multiple 65, 86, 145
- BDAM data sets, copying 52
- BDAM data sets, supported 5
- BEGIN parameter in VDRL statement (stand-alone DFDSS) 226
- BELOW16 in the EXEC statement 19
- blanks before and after parentheses 29
- BLK subparameter of TGTALLOC
  - COPY command 52
  - RESTORE command 119
- BLKSIZE and LRECL parameters, specifying in a PRINT operation 100
- block length, definition 321
- block size
  - default when dumping to tape or DASD 94
  - minimum 94
  - note on 161
- books, related iv-ix
- BY criteria 21
- BY parameter
  - COMPRESS command 34
  - COPY command 43
  - DEFRAG command 69
  - DUMP command 84
  - RELEASE command 105
  - RESTORE command 113

## C

- calling block structure 192
- CANCELERROR parameter
  - COPY command 47
  - DUMP command 87
  - RESTORE command 115
- card readers supported 5, 222
- CATALOG parameter
  - COPY command 48
  - RESTORE command 115
- catalog recovery 130
- catalog search order, standard 24
- catalog, locking a user 130
- cataloging non-VSAM data sets during RESTORE 130
- CATLG subparameter
  - COMPRESS command 34
  - COPY command 44
  - DEFRAG command 72
  - DUMP command 84
  - RELEASE command 105
  - RESTORE command 113
- changes in DFDSS, summary of xi
- characteristics of data sets 21
- codes, condition
  - See condition codes
- combining data set extents 163
- command
  - comments within 29
  - continuation 30
  - null 138
  - separation 29
  - termination 30
- command data set, following SYSIN DD 17
- command syntax
  - for DFDSS 29
  - understanding 30
- command usage
  - COMPRESS command 37
  - COPY command 58
  - COPYDUMP command 67
  - DEFRAG command 75
  - DUMP command 94
  - RELEASE command 108
  - RESTORE command 126-133
- commands, auxiliary 135-141
- commands, function
  - See function commands
- commas used to separate subparameters 30
- comments within command data set 29
- compatibility
  - among releases of DFDSS 26
  - among releases of stand-alone DFDSS 27
  - between Version 1 and Version 2 of DFDSS 24
  - of stand-alone with system DFDSS 223
- COMPRESS command 33-37
  - example of 37, 156
- COMPRESS parameter, DUMP command 87
- compressing data sets 33
  - condition codes
    - See also return codes
    - general description 136
    - LASTCC 137
    - MAXCC 137
  - consoles supported 5, 221
  - consolidating unused space in data sets 33
  - continuation
    - command 30
    - errors in auxiliary commands 140
  - control blocks
    - See data areas
  - control statement examples
    - See examples of DFDSS jobs
  - control statements, stand-alone DFDSS 223
  - controlling DFDSS
    - using ISMF 8
    - using JCL 17
    - using PARM information in the EXEC statement 19
  - controlling task execution
    - description 136
    - examples 139-140
  - converting to new DASD 1
  - COPY command 38, 64
    - data set COPY notes 58
    - example of 20, 150
    - example of data set 55
    - example of to an unlike device 144
    - example of tracks 54
    - examples of full 54
    - ISAM data sets 59
    - notes on 58
    - RENUNC parameter 52
    - syntax 40, 310
    - user catalog 58
    - VSAM data sets 59
  - COPY operation
    - data mover selection matrix 61
  - COPYDUMP command 65-67
    - examples of 66
    - syntax 65, 312
  - copying stand-alone DFDSS Restore to tape 227
  - COPYVOLID parameter
    - COPY command 48
    - RESTORE command 115
  - CPYV parameter
    - COPY command 48
    - RESTORE command 115
  - create a function-taskid table via SYSPRINT data, example 213
  - CREDIT subparameter
    - COMPRESS command 34
    - COPY command 44
    - DEFRAG command 72
    - DUMP command 84
    - RELEASE command 105
    - RESTORE command 113
  - criteria for filtering 21
  - CVOL, moving 58
  - CYL subparameter of TGTALLOC

COPY command 52  
RESTORE command 119  
CYL, definition 321

## D

DADSM, definition 321  
DASD devices supported 5, 221  
DASD initialized 6  
DASD space fragmentation 68  
DASD space reclamation 163  
DASD space utilization 161-164  
data areas  
    ADRBMB 301  
    ADREID0 219  
    ADRTAPB 302  
    ADRUFO 239  
data compression (run-length), definition 321  
Data Facility Data Set Services  
    See DFDSS  
data integrity  
    DYNALLOC 167  
    ENQ 165  
    example of RESERVE-ENQUE processing 166  
    full or tracks DUMP/RESTORE 179  
    general description 164-180  
    RESERVE 165  
    RESERVE-ENQUE-DYNALLOC matrix 178  
    shared DASD considerations 180  
    WAIT(numsecs,numretrys) option for  
        serialization 164  
data mover selection matrix 61  
data security 75, 174  
data set  
    authorization checking 169  
    change indicator, definition 321  
    characteristics (BY criteria) 21  
    COPY to unlike device 144  
    DUMP  
        compatibility with Version 1 25  
        example of physical 146  
        example of physical, with DELETE 149  
        examples of 91  
    extents, combining 163  
    filtering 21  
    names, examples of fully and partially qualified 21  
    passwords, abend dumps 175  
    qualifier, rules regarding asterisks in 22  
    RESTORE  
        compatibility with Version 1 25  
        example of 121, 148  
        with RENAMEU, example of 155  
data-set-changed flag, DUMP command 89  
DATABASE 2 (DB2) data sets, supported 5  
DATALENGTH parameter, PRINT command 100  
DATASET parameter  
    COPY command 43  
    DUMP command 83  
    PRINT command 99

RESTORE command 112  
DCB parameters LRECL and BLKSIZE, specifying in a  
    PRINT operation 100  
DD statements (JCL) 18  
DDNAME list 190  
ddname overrides, example 199  
DDNAME parameter  
    COMPRESS command 35  
    DEFRAG command 69  
    RELEASE command 106  
DDPTR 190  
default block size when dumping to tape or DASD 94  
definitions 1  
DEFRAG command 68-78  
    algorithms used 161  
    ending statistics 77  
    examples of 74  
    fragmentation index 161  
    notes on 75  
    results of a successful run 75  
    syntax 69, 313  
    when to run 78  
degassing data sets 33  
DELETE parameter  
    COPY command 48  
    DUMP command 87  
DELETE, compatibility with Version 1 24  
detailed notes on using DFDSS  
    See also using DFDSS  
    allocating volumes dynamically 160  
    block size 161  
    DASD space utilization 161-164  
    data integrity 164-167  
    data security 174  
    data set passwords, abend dumps 175  
    DEFRAG function 161  
    DUMP operation 175  
    dump/restore techniques and considerations 177  
    dynamically determining DFDSS's version, release,  
        and modification level 159  
    enqueueing versus dynamic allocation 178  
    filtering criteria and/or passwords, passing via a data  
        set 180  
    installation authorization exit 174  
    integrity of dumped and restored data 179  
    lockouts during DFDSS runs 180  
    multivolume data sets, processing 181  
    PARM statement, SIZE 183  
    RESTORE operation 183  
    security processing, example of 175  
    space management DUMP techniques 163  
    storage requirements 181  
    SYSPRINT DD statement 187  
device conversion 1  
device names, generic 224  
Device Support Facilities utility 6  
devices supported  
    DFDSS 5  
    stand-alone DFDSS Restore 221  
DFDSS  
    command separation 29

- compatibility among releases 26
- continuation of a command 30
- control 17
- data set authorization checking 169
- devices supported 5
- efficiency considerations 94
- examples of jobs 143-157
- features 1
- filtering 21-24
- how authorization is checked for you 167
- input 18
- interactive 7
- invoking 7, 143
- JCL requirements 17
- job example, selecting one as a model 143
- job termination 139
- jobs, a note about examples in this book 32
- lockouts during runs 180
- messages 245-291
- notational conventions used 30
- notes on using
  - See detailed notes on using DFDSS
- output 18
- overview 1
- parameter separation 29
- PARM information in the EXEC statement 19
- specifying commands 29
- storage requirements 181
- subparameter separation 29
- user exit routines 231-244
- Version 2, compatibility with Version 1 24
- volume formats supported 5
- DFDSS module, main entry point 190
- DFDSS version, release, and modification level, dynamically determining 159
- DFDSS, stand-alone Restore 221
  - messages 293-299
  - RESTORE compatibility among releases 27
- DIAGNOSE, definition 321
- diagnosis guide, DFDSS iv
- DIRF, definition 321
- disable functional exit via the UIM 212
- disable UIM usage 205
- disk drives
  - See DASD devices supported
- display panels, ISMF 7
- DO-END group of commands 136-138
  - condition for executing 136
  - syntax 137
- DOS bit, definition 321
- DS parameter
  - COPY command 43
  - DUMP command 83
  - PRINT command 99
  - RESTORE command 112
- DSCB, definition 321
- DSCHA subparameter
  - COMPRESS command 34
  - COPY command 44
  - DEFRAG command 72
  - DUMP command 84
  - RELEASE command 105
  - RESTORE command 113
- DSORG subparameter
  - COMPRESS command 34
  - COPY command 44
  - DEFRAG command 72
  - DUMP command 84
  - RELEASE command 105
  - RESTORE command 113
- DSORG, definition 321
- DS1LSTAR, definition 321
- DUMP command 79, 96
  - compatibility with Version 1 25
  - data-set-changed flag 89
  - example of a full 91
  - example of a full followed by a COPYDUMP 148
  - example of a full with a DEFRAG 145, 151
  - example of a physical data set 146
  - example of a physical, with DELETE 149
  - example of a tracks 91
  - example of data set 20, 91
  - multiple copies 86, 145
  - notes on 94-96
  - scratching data sets without dumping them 79
  - syntax 81, 314
  - using DD DUMMY for the output data set 79
- dump data, multiple copies 65, 86, 145
- DUMP operation
  - dumping to a DASD sequential data set 175
  - format of the output data set 176
  - OPTIMIZE parameter 176
- dump tape header record format 302
- dump tapes from DFDSS Version 1 25
- dump/restore techniques and considerations 177
- dumped data, integrity of 179
- DYNALLOC parameter
  - COMPRESS command 36
  - COPY command 48
  - DEFRAG command 70
  - DUMP command 87
  - PRINT command 100
  - RELEASE command 107
  - RESTORE command 116
- DYNAM parameter
  - COMPRESS command 35
  - DEFRAG command 69
  - RELEASE command 106
- dynamic allocation
  - DASD volumes 143, 160
  - non-VSAM data sets 184
- dynamic allocation of data sets versus enqueueing 178

**E**

efficiency considerations in DFDSS 94  
 ELSE command 138  
 END command, in DO-END 138  
 END parameter in VDRL statement (stand-alone DFDSS) 227  
 END statement in stand-alone DFDSS 227  
 ending your DFDSS job 139  
 ENQ 165  
   definition 321  
 ENQ names 183  
 ENQFAILURE subparameter  
   COPY command 53  
   DUMP command 89  
   PRINT command 101  
   RESTORE command 120  
 enqueue exit routine, user 235  
 enqueue scheme 165, 235  
 enqueueing versus dynamic allocation of data sets 178  
 environment, system 6  
 EOJ command 139  
 erase on scratch for DEFrag 75  
 ERRORTRACKS parameter, PRINT command 100  
 ESDS data sets, supported 5  
 examples in this manual, note about 32  
 examples of  
   common continuation errors 140  
   controlling task execution 139-140  
   qualified data set names 21  
 examples of DFDSS jobs 143  
   backing up an integrated catalog facility user catalog 152  
   COMPRESS and RELEASE on a volume 156  
   compressing selected data sets 37  
   COPY 20, 150  
   COPYDUMP 66  
   data set COPY 55  
   data set COPY for device conversion 55, 144  
   data set DUMP 91  
   data set RESTORE 20, 121, 148  
   data set RESTORE with RENAMEU 155  
   DEFrag 74  
   dumping a data set 20  
   full COPY 54  
   full DUMP 91  
   full DUMP followed by a COPYDUMP 148  
   full DUMP followed by a DEFrag 145, 151  
   full RESTORE 120  
   ISMF used 7-17  
   note about those in this book 32  
   physical data set DUMP 146  
     with DELETE 149  
   PRINT 102  
   RELEASE 108, 156  
   restoring an integrated catalog facility user catalog 152  
   restoring VSAM data sets cataloged in an integrated catalog facility user catalog 153, 154

selecting one as a model 143  
 stand-alone DFDSS Restore 229  
 tracks COPY 54  
 tracks DUMP 91  
 tracks RESTORE 120  
 EXCLUDE criteria 21  
 EXCLUDE parameter  
   COMPRESS command 34  
   COPY command 43  
   DEFrag command 71  
   DUMP command 84  
   RELEASE command 105  
   RESTORE command 113  
 EXCP data sets, supported 5  
 EXCP, definition 321  
 EXEC statement (JCL) 18-20  
 executing stand-alone DFDSS Restore 227  
 exit identification block 194  
 exit routines, user  
   See user exit routines  
 EXPDT subparameter  
   COMPRESS command 34  
   COPY command 44  
   DEFrag command 72  
   DUMP command 84  
   RELEASE command 105  
   RESTORE command 113  
 EXPORT/IMPORT (IDCAMS commands) 62  
 extended architecture usage 191  
 EXTNT subparameter  
   COMPRESS command 34  
   COPY command 44  
   DEFrag command 72  
   DUMP command 84  
   RELEASE command 105  
   RESTORE command 113

**F**

FDD parameter  
   COMPRESS command 33  
   COPY command 43  
   DUMP command 83  
   RELEASE command 104  
   RESTORE command 112  
 features of DFDSS 1  
 FILTERDD parameter  
   COMPRESS command 33  
   COPY command 43  
   DUMP command 83  
   RELEASE command 104  
   RESTORE command 112  
 filtering  
   definition 2  
   general description 21-24  
 FORCE parameter  
   COPY command 49  
   RESTORE command 116



- format-1 DSCB (F1 DSCB), definition 321
- format-2 DSCB (F2 DSCB), definition 321
- format-3 DSCB (F3 DSCB), definition 321
- format-4 DSCB (F4 DSCB), definition 321
- format-5 DSCB (F5 DSCB), definition 321
- FRAGI parameter, DEFRAG command 73, 161
- fragmentation index 161
- FRAGMENTATIONINDEX parameter, DEFRAG command 73, 161
- free-space fragmentation 68
- FROMADDR parameter in RESTORE statement (stand-alone DFDSS) 225
- FROMDEV parameter in RESTORE statement (stand-alone DFDSS) 225
- FSIZE subparameter
  - COMPRESS command 34
  - COPY command 44
  - DEFRAG command 72
  - DUMP command 84
  - RELEASE command 105
  - RESTORE command 113
- FULL parameter
  - COPY command 42
  - DUMP command 83
  - RESTORE command 112
- full-track read 176
- fully qualified data set names, examples of 21
- function commands
  - abbreviations permitted 30
  - detailed description of 29-133
  - truncations permitted 30
- function-taskid table via SYSPRINT data, example 213

## G

- GDG filtering 23
- general notes on DFDSS, including restrictions 150
- generation data group filtering 23
- generic device names 224
- glossary 321-322

## H

- HDINGADDR 191
- how authorization is checked for you 167
- how many subparameters are allowed 31
- how to write user exit routines 232-244

## I

- ICF
  - See integrated catalog facility
- ICKDSF, initialize DASD volumes with 6
- IDCAMS utility 61
- IDD parameter
  - COPY command 44
  - COPYDUMP command 65
  - DUMP command 86
  - PRINT command 100
  - RESTORE command 114
- IDY parameter
  - COPY command 44
  - DUMP command 86
  - PRINT command 100
- IEBCOPY utility 61
- IEBISAM utility 61
- IEHMOVE utility 61
- IF-THEN-ELSE group of commands
  - description 136
  - examples 139-140
  - syntax 137
- INCLUDE criteria 21
- INCLUDE parameter
  - COMPRESS command 33
  - COPY command 43
  - DUMP command 84
  - RELEASE command 105
  - RESTORE command 112
- incorrect exit action, recover from (example) 210
- incremental backup 79
- INDDNAME parameter
  - COPY command 44
  - COPYDUMP command 65
  - DUMP command 86
  - PRINT command 100
  - RESTORE command 114
- indexed VTOCs 6
- INDYNAM parameter
  - COPY command 44
  - DUMP command 86
  - PRINT command 100
- initialize all DASD volumes 6
- input error toleration
  - COPY command 53
  - DUMP command 89
  - PRINT command 101
- input to DFDSS (in DD statement) 18
- installation authorization exit 174
- installation options exit
  - parameter block 239
  - user routine 237
- integrated catalog facility 5
  - backing up a user catalog, example of 152
  - definition 322
  - restoring a user catalog, example of 152
  - restoring VSAM data set(s) cataloged in a user catalog 153, 154

- integrity of dumped and restored data 179
- interactive storage management facility 7-17
- invoking DFDSS
  - using ISMF 8
  - using JCL 17
  - via ATTACH, LINK, or CALL macro 143, 189
- invoking DFDSS via an application program 189-220
  - examples of 197-218
- invoking ISMF 8
- IOERROR subparameter
  - COPY command 53
  - DUMP command 89
  - PRINT command 101
- ISAM data sets, copying 59
- ISAM data sets, supported 5
- ISMF use and examples 7-17

## J

- JCL requirements for DFDSS 18
- JCL, using to invoke DFDSS 17
- job examples
  - See examples of DFDSS jobs
- JOB statement
  - DFDSS, JCL 18
  - stand-alone DFDSS Restore 224
- job termination 139

## K

- key range data sets, supported 5
- KEYLENGTH parameter, PRINT command 100
- keyword parameters, how to specify 29-32
- KSDS data sets, supported 5

## L

- LASTCC
  - definition 136
  - in IF-THEN-ELSE command sequence 137
  - in SET command 137
- LDS data sets, supported 5
- LIDD parameter
  - COPY command 49
  - DUMP command 88
- LIDY parameter
  - COPY command 49
  - DUMP command 88
- like device, definition 322
- LINECNT=nnnn in the EXEC statement 19
- LINKLIST data sets in a DEFRAG operation 77
- location-dependent data 61, 131
- locking a user catalog 130

- lockouts during DFDSS runs 180
- logging on to ISMF 8
- logging, RACF 19, 171
- logical data set DUMP
  - definition 322
  - notes 96
- logical data set processing 3
- logical data set RESTORE
  - definition 322
  - notes 130
- logical volume, definition 322
- LOGICALVOLUME parameter
  - COPYDUMP command 65
  - RESTORE command 116
- LOGICALVOLUME, compatibility with Version 1 25
- LOGINDDNAME parameter
  - COPY command 49
  - DUMP command 88
- LOGINDYNAM parameter
  - COPY command 49
  - DUMP command 88
- LRECL and BLKSIZE parameters, specifying in a PRINT operation 100
- LVOL parameter
  - COPYDUMP command 65
  - RESTORE command 116

## M

- magnetic tape devices supported 5, 222
- main storage requirements
  - DFDSS 181
  - stand-alone DFDSS Restore 230
- manuals, related iv-ix
- MAXCC
  - definition 136
  - in IF-THEN-ELSE command sequence 137
  - in SET command 137
- maximum number of subparameters allowed 31
- MAXMOVE parameter
  - DEFRAG command 73
- MENTITY parameter
  - COPY command 50
  - RESTORE command 116
- menu-driven panels, ISMF 7
- messages
  - DFDSS 245-291
  - stand-alone DFDSS Restore 293-299
- migrating to new DASD 1
- minivolume, definition 322
- minivolumes 6
- MINSECQTY parameter
  - RELEASE command 107
- MINTRACKSUNUSED parameter, RELEASE command 107
- MINTRKS parameter, RELEASE command 107
- MODEL parameter in RESTORE statement (stand-alone DFDSS) 226

- modify SYSIN data, example 210
- moving a user catalog 58
- moving data sets (COPY with DELETE) 48
- moving data sets to new DASD 1
- MSG statement in stand-alone DFDSS 224
- MULTI subparameter
  - COMPRESS command 34
  - COPY command 44
  - DEFRAG command 72
  - DUMP command 84
  - RELEASE command 105
  - RESTORE command 113
- multiple backup copies 65, 86, 145
- multiple subparameters, how many are allowed 31
- multiple tracks read, on dump 88
- multivolume data sets, copying 47
- multivolume data sets, processing 181
- MVOLSER parameter
  - COPY command 50
  - RESTORE command 116
- MVS environments supported by DFDSS 6
- MVS/Extended Architecture (MVS/XA)
  - definition 322
  - publications v-viii
- MVS/Extended Architecture interactive storage management facility 7
- MVS/370 publications iv

## N

- names, ENQ 183
- new DASD, moving data to 1
- nonindexed VTOCs 6
- NORUN in the EXEC statement 19
- notational conventions used (command syntax) 30
- notes on commands
  - COMPRESS command 37
  - COPY command 58
  - COPYDUMP command 67
  - DEFRAG command 75
  - DUMP command 94
  - RELEASE command 108
  - RESTORE command 126-133
- notes on using DFDSS
  - See detailed notes on using DFDSS
- null command 138
- number of subparameters allowed 31

## O

- ODD parameter
  - COPY command 46
  - COPYDUMP command 65
  - DUMP command 86
  - PRINT command 100
  - RESTORE command 115

- ODY parameter
  - COPY command 46
  - RESTORE command 115
- old DASD, moving data from 1
- online panels, ISMF 7-17
- operating environment 6
- operating in serial or parallel mode 135
- OPTIMIZE parameter, on DUMP 88, 176
- OPTION list 190
- options, supplying (example) 199
- OPTPTR 190
- OUTDDNAME parameter
  - COPY command 46
  - COPYDUMP command 65
  - DUMP command 86
  - PRINT command 100
  - RESTORE command 115
- OUTDYNAM parameter
  - COPY command 46
  - RESTORE command 115
- output from DFDSS (out DD statement) 18
- output tape error exit 203
- OUTTRACKS parameter
  - COPY command 50
  - RESTORE command 116
- OUTTRKS parameter
  - COPY command 50
  - RESTORE command 116
- overview of DFDSS 1

## P

- page ejection, suppressing 19
- page number list 191
- page numbering, example 200
- PAGENO=nnnn in the EXEC statement 19
- PAGEPTR 191
- PAM data sets, compressing 33
- PAM data sets, supported 5
- PAM, definition 322
- PARALLEL command 135
- parallel mode 135
- parameter separation 29
- parameters, how to specify 29-32
- parenthesis, blanks before and after 29
- PARAM
  - information in the EXEC statement used to control DFDSS 19
    - statement, SIZE= 183
- PARAM values, example 199
- partially qualified data set names, examples of 21
- partitioned data sets, compressing 33
- PASSWORD parameter
  - COMPRESS command 36
  - COPY command 50
  - DEFRAG command 73
  - DUMP command 88
  - PRINT command 100

- RELEASE command 107
- RESTORE command 117
- PCTU parameter
  - COPY command 51
  - RESTORE command 117
- PDS, compressing 33
- PERCENTUTILIZED parameter
  - COPY command 51
  - RESTORE command 117
- physical (data set) processing, definition 322
- physical data set DUMP
  - definition 322
  - notes 95
- physical data set processing 3
- physical data set RESTORE
  - definition 322
  - notes 129
- preallocated data set COPY usage 62
- preallocated target, example 57
- prevent command execution via the UIM 206
- PRG parameter
  - COPY command 51
  - DUMP command 89
  - RESTORE command 117
- PRINT command 97-103
  - examples of 102
  - syntax 98, 316
- PRINT, compatibility with Version 1 25
- printers supported 5, 221
- problem determination
  - stand-alone DFDSS messages 299
  - system DFDSS messages 291
- problem program, invoking DFDSS via
  - See invoking DFDSS via an application program
- process WTO/WTOR records via the UIM,
  - example 216
- processing multivolume data sets, notes on 181
- profiles, defining, with RACF installed 60
- program control
  - See Resource Access Control Facility (RACF)
- PSW, definition 322
- PSWD parameter
  - COMPRESS command 36
  - COPY command 50
  - DEFRAG command 73
  - DUMP command 88
  - PRINT command 100
  - RELEASE command 107
  - RESTORE command 117
- publications, related iv-ix
- PURGE parameter
  - COPY command 51
  - DUMP command 89
  - RESTORE command 117

## Q

- qualified data set names, examples of 21
- quick reference 309-319

## R

- RACF 167, 169
- RACF authorization checking 50, 116, 232
- RACF logging 171
- RACF, definition 322
- RACFLOG=YES in the EXEC statement 19
- RBA, definition 322
- Read Multiple Count, Key, and Data channel
  - command 176
- read/write access serialization scheme 183
- readers, card, supported 5, 222
- reblock exit routine, user 243
- REBLOCK parameter
  - COPY command 51
- RECATALOG parameter
  - COPY command 48
  - RESTORE command 115
- recover from an incorrect exit action, example 210
- recovery of a user catalog 130
- REFDT subparameter
  - COMPRESS command 34
  - COPY command 44
  - DEFRAG command 72
  - DUMP command 84
  - RELEASE command 105
  - RESTORE command 113
- reference, quick 309-319
- related publications iv-ix
- relative generation filtering 23
- RELBLKA parameter
  - COPY command 52
- RELBLOCKADDRESS parameter
  - COPY command 52
- RELCREDIT, definition 322
- RELEASE command 104-108, 156
  - cross-system serialization 107
  - example of 108, 156
  - notes on 108
  - syntax 104, 317
- RELEXPDT, definition 322
- RELREFDT, definition 322
- RENAME parameter, RESTORE command 117
- RENAMEUNCONDITIONAL parameter
  - COPY command 52
  - RESTORE command 119
- renaming RACF-protected data sets during restore 186
- RENUNC parameter
  - COPY command 52
  - RESTORE command 119
- REPLACE parameter
  - COPY command 52

- example 57
- RESTORE command 119
- REPRO (IDCAMS command) 61
- RESERVE 165
  - definition 322
- RESET parameter, DUMP command 89
- Resource Access Control Facility (RACF)
  - protecting ISMF function 171-174
- RESTORE command 109, 133
  - compatibility with Version 1 25
  - cross-system serialization 116
  - data set, with RENAMEU, example 155
  - example of data set 20, 121, 148
  - example of full 120
  - example of tracks 120
  - notes on 126-133, 183, 186
  - RENAME parameter 126
  - RENAMEUNCONDITIONAL parameter 126
  - REPLACE parameter 126
  - syntax 110, 318
- restore limits, stand-alone DFDSS 226
- RESTORE operation
  - data set, RENAME, RENUNC, REPLACE parameters 126
  - data sets with null extents 129
- RESTORE operation, stand-alone 221
- RESTORE statement in stand-alone DFDSS 225-226
- restored data, integrity of 179
- restoring a single VSAM data set cataloged in an integrated catalog facility user catalog, example of 154
- restoring an integrated catalog facility user catalog, example of 152
- restoring VSAM data sets cataloged in an integrated catalog facility user catalog, example of 153
- restrictions
  - in DFDSS 159
  - in stand-alone DFDSS Restore 222
- return codes, DFDSS 245
- RRDS data sets, supported 5
- run-length data compression, definition 321
- running stand-alone DFDSS Restore 227

**S**

- SAF (System Authorization Facility) 167, 169
- SAM data sets, supported 5
- SAM, definition 322
- sample user interaction module (UIM) 201
- SCAN in the EXEC statement 19
- scratching data sets without dumping them 79
- search order, standard catalog 24
- security processing, example of 175
- selecting a JOB example to use as model 143
- SERIAL command 135
- serial mode 135
- serialization scheme, read/write access 183
- SET command 136
- SHARE parameter

- COPY command 52
- DUMP command 89
- PRINT command 101
- RESTORE command 119
- SHR parameter
  - COPY command 52
  - DUMP command 89
  - PRINT command 101
  - RESTORE command 119
- similar device, definition 322
- SIZE=nnnnK in the EXEC statement 19
- SOURCE subparameter of TGTALLOC
  - COPY command 52
  - RESTORE command 119
- source, definition 18
- space allocation 52, 119
- space fragmentation on DASD 68
- space management DUMP techniques 163
- space utilization, percent
  - COPY command 51
  - RESTORE command 117
- specified operating environment 6
- specifying
  - DFDSS commands 29
  - more than 256 subparameters 32
  - tasks (function commands) 29-133
- SRC subparameter of TGTALLOC
  - COPY command 52
  - RESTORE command 119
- stand-alone DFDSS Restore 221-230
  - copying to tape 227
  - devices you can use 221
  - executing 227
  - messages 293-299
  - restrictions 222
  - storage requirements 230
- standard catalog search order 24
- standard label tape recommended 17
- starting page number, example 200
- statements
  - EXEC 18
  - input DD 18
  - JOB 18
  - output DD 18
  - STEPCAT 18
  - SYSIN DD 18
  - SYSPRINT DD 18
- statistical information 189
- STEPCAT DD statement 18
- storage management library vi
- storage requirements
  - DFDSS 181
  - stand-alone DFDSS Restore 230
- subparameter separation
  - commas 30
  - DFDSS 29
- subparameters, how many are allowed 31
- summary of changes xi
- supply in-storage SYSIN data and delete SYSPRINT records 208

- supplying a new UIM, example 202
- supplying a user area pointer, example 204
- supplying PARM values, example 199
- syntax (command) for DFDSS 29
- syntax, understanding 30
- SYSALLDA 58
- SYSDA 58
- SYSIN data, modifying (example) 210
- SYSIN data, supplying in-storage (example) 208
- SYSIN DD statement 18
- SYSPRINT DD statement
  - DFDSS, JCL 18
  - notes on 187
- System Authorization Facility (SAF) 167, 169
- system consoles supported 5, 221
- system DFDSS
  - See DFDSS
- system environment 6
- system generation 2
- system programming information 193

## T

- tape devices supported 5, 222
- tape error exit 203
- tape header record format 302
- tape volume mount exits 203
- tape, copying stand-alone DFDSS Restore to 227
- tape, standard label, recommended 17
- target allocation 52, 119
- target, definition 18
- task execution, controlling 136
- temporary work space. 58
- terminating your DFDSS job 139
- termination exit processing in the UIM 216
- termination of a command 30
- TGTALLOC parameter
  - COPY command 52
  - RESTORE command 119
- THEN command 138
- TOADDR parameter
  - MSG statement in stand-alone DFDSS 225
  - RESTORE statement in stand-alone DFDSS 226
- TODEV parameter
  - MSG statement in stand-alone DFDSS 225
  - RESTORE statement in stand-alone DFDSS 226
- TOLERATE(ENQFAILURE) parameter
  - COPY command 53
  - DUMP command 89
  - PRINT command 101
  - RESTORE command 120
- TOLERATE(IOERROR) parameter
  - COPY command 53
  - DUMP command 89
  - PRINT command 101
- TRACE=YES in the EXEC statement 19
- track packing, definition 90, 322
- TRACKOVERFLOW parameter
  - COPY command 53

- DUMP command 90
- TRACKS parameter
  - COPY command 42
  - DUMP command 83
  - PRINT command 99
  - RESTORE command 112
- tracks read at a time, on dump 88
- TRK subparameter of TGTALLOC
  - COPY command 52
  - RESTORE command 119
- TRK, definition 322
- TRKO parameter
  - COPY command 53
  - DUMP command 90
- TRKS parameter
  - COPY command 42
  - DUMP command 83
  - PRINT command 99
  - RESTORE command 112
- truncations permitted
  - in auxiliary commands 141
  - in function commands 30
- two or more backup copies 65, 86, 145
- TYPRUN=NORUN in the EXEC statement 19
- TYPRUN=SCAN in the EXEC statement 19

## U

- UAPTR 192
- UAPTR, example 204
- UCB, definition 322
- UIMPTR 191
- UNCATALOG parameter
  - COPY command 53
  - DUMP command 90
- unlike device, definition 322
- unmovable data sets, restoring 186
- usage of commands
  - See command usage
- user area list 192
- user area pointer, example 204
- user catalog, moving 58
- user exit routines
  - authorization checking 232
  - enqueue 235
  - installation options 237
  - reblock 243
- user exit routines (overview) 231
- user interaction module (UIM) 193
- user interaction module (UIM), example 201
- user interaction module list 191
- user statistics records, example 215
- using DFDSS 7
  - command syntax 29
  - efficiency considerations in using the DUMP command 94
  - fully and partially qualified data set names 21
  - JCL requirements 17

stand-alone DFDSS Restore 221-230  
user exit routines 231-244  
without an operating system 221-230  
utilities, use of in data set COPY operation 61  
utilization, percent  
COPY command 51  
RESTORE command 117  
UTILMSG=YES in the EXEC statement 19

## V

VDRL statement in stand-alone DFDSS 226  
version, release, and modification level, determining for  
DFDSS dynamically 159  
versions of DFDSS, compatibility 24  
VL 192  
VM/370, running stand-alone DFDSS under 221  
VOLID parameter in RESTORE statement (stand-alone  
DFDSS) 226  
volume authorization checking 168  
volume count subparameter, JCL 18  
volume formats supported by DFDSS 5  
volume restore limits, stand-alone DFDSS 226  
VSAM  
See also alternate index  
data sets, as used in this book 5  
data sets, notes on copying 59  
data sets, notes on dumping 95  
data sets, notes on restoring 129  
key range data sets, compatibility with Version 1 25  
VTOC parameter, PRINT command 99  
VTOC, definition 322  
VTOCIX, definition 322  
VVDS, definition 322

## W

WAIT parameter  
COMPRESS command 36  
COPY command 53  
DEFRAG command 74  
DUMP command 90  
PRINT command 101  
RELEASE command 107  
RESTORE command 120  
when to run DEFRAG 78  
work space. 58  
WORKSIZE, compatibility with Version 1 25  
WRC parameter  
COPY command 54  
RESTORE command 120  
WRITECHECK parameter  
COPY command 54  
RESTORE command 120  
writing a user exit routine 232-244  
writing to the operator 135

WTO command 135  
WTO/WTOR record processing via the UIM,  
example 216

## X

XABUFF=ABOVE16 | BELOW16 in the EXEC  
statement 19

## Numerics

16 megabytes virtual storage, buffers above 19  
2305 DASD device 5  
256 or more subparameters, specifying 32  
31-bit addressing mode 191  
33xx DASD devices supported 5

Data Facility Data Set Services:  
User's Guide and Reference  
SC26-4125-3

This manual is part of a library that serves as a reference source for system 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: Do not use this form to request IBM publications. If you do, your order will be delayed because publications are not stocked at the address printed on the reverse side. Instead, you should 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.**

If you have applied any technical newsletters (TNLs) to this book, please list them here: \_\_\_\_\_

Chapter/Section \_\_\_\_\_

Page No. \_\_\_\_\_

Comments:

Note: Staples can cause problems with automated mail sorting equipment.  
Please use pressure sensitive or other gummed tape to seal this form.

If you want a reply, please complete the following information.

Name \_\_\_\_\_ Phone No. (\_\_\_\_) \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

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 directly to the address in the Edition Notice on the back of the 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, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation  
P.O. Box 50020  
Programming Publishing  
San Jose, California 95150



Fold and tape

Please do not staple

Fold and tape





Data Facility  
Data Set Services:  
User's Guide and Reference

File Number S370-32

SC26-4125-03



Printed in U.S.A.