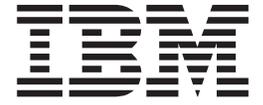


OS/390

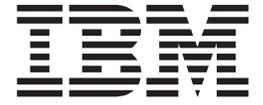


# DFSMSHsm Data Recovery Scenarios

*Release 10*



OS/390



# DFSMSHsm Data Recovery Scenarios

*Release 10*

**Note!**

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 25.

**First Edition (September 2000)**

This edition applies to Version 2 Release 10 of OS/390® (5647-A01) and any subsequent releases until otherwise indicated in new editions.

This edition replaces SC26-4928-03.

Order IBM publications from your IBM representative or the IBM branch office serving your locality.

© **Copyright International Business Machines Corporation 1993, 2000. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Preface</b> . . . . .	v
Related Books . . . . .	v
Accessing OS/390 DFSMS Books on the Internet . . . . .	v
How to send your comments . . . . .	v
<b>Summary of Changes</b> . . . . .	vii
Summary of Changes for GC35-0386-00 OS/390 Version 2 Release 10	
DFSMSHsm . . . . .	vii
New Information . . . . .	vii
Summary of Changes for SC26-4928-02 DFSMS/MVS Version 1 Release 3	vii
New Information . . . . .	vii
<b>Chapter 1. Introduction</b> . . . . .	1
<b>Chapter 2. Data Recovery Scenarios</b> . . . . .	3
Case 1: Damaged CDS, Full Journal . . . . .	3
Prevention . . . . .	6
Case 2: Damaged Journal, Undamaged CDS . . . . .	6
Case 3: Full Journal, Undamaged CDS . . . . .	7
Prevention . . . . .	8
Case 4: Structurally Damaged CDS, Missing Journal Records . . . . .	8
Prevention . . . . .	8
Case 5: Time Sequence Gaps in CDS Records, Journal Records Also Missing	9
Prevention . . . . .	9
Case 6: Overwritten Migration Tape . . . . .	10
Prevention . . . . .	12
Case 7: Overwritten Backup Tape . . . . .	13
Prevention . . . . .	17
Case 8: Head Crash On an ML1 Volume . . . . .	17
Prevention . . . . .	23
Case 9: Reestablish Access to Previously Deleted, Migrated Data Sets (No Backup Exists, ML2 Only) . . . . .	23
Prevention . . . . .	23
Case 10: Restoring Stacked Output Tapes With a Down-Level MVS Image . . . . .	24
Prevention . . . . .	24
<b>Notices</b> . . . . .	25
Programming Interface Information . . . . .	26
Trademarks . . . . .	26
<b>Glossary</b> . . . . .	27
<b>Index</b> . . . . .	41



---

## Preface

This book is intended to assist storage administrators when they need to recover data with DFSMSHsm. It describes a variety of situations where data has been lost, and what steps to take to recover that data. It includes situations where the lost data is a control data set or journal. It also tells storage administrators what they can do to help prevent data loss situations from occurring.

---

## Related Books

You should be familiar with the information presented in the following books:

- *OS/390 DFSMSHsm Implementation and Customization Guide*
- *OS/390 DFSMSHsm Storage Administration Guide*
- *OS/390 DFSMSHsm Storage Administration Reference*
- *OS/390 DFSMS Access Method Services for Catalogs*
- *DFSORT Application Programming Guide R14*

---

## Accessing OS/390 DFSMS Books on the Internet

In addition to making softcopy books available on CD-ROM, IBM provides access to unlicensed OS/390 softcopy books on the Internet. To find OS/390 books on the Internet, first go to the OS/390 home page: <http://www.ibm.com/s390/os390/> From this web site, you can link directly to the OS/390 softcopy books by selecting the Library icon. You can also link to IBM Direct to order hardcopy books.

---

## How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other DFSMSHsm documentation:

- Visit our home page at:  
<http://www.storage.ibm.com/software/sms/hsm/hsmhome.htm> There you will find the feedback page where you can enter and submit your comments.
- Send your comments by e-mail to:
  - IBMLink from US: STARPUBS at SJEVM5
  - IBMLink from Canada: STARPUBS at TORIBM
  - Internet: [starpubs@us.ibm.com](mailto:starpubs@us.ibm.com)

Be sure to include the name of the book, the part number of the book, the version and product name, and if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

- Fill out one of the forms at the back of this book and return it by mail, by fax (520 799-2906), or by giving it to an IBM representative. If the form has been removed, please address your comments to: IBM Corporation, Department 61C, 9000 South Rita Road, Tucson, Arizona 84744-0001, U.S.A.

When you send information to IBM you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.



---

## Summary of Changes

The summary of changes informs you of changes to this book. Revision bars (|) in the left margin of the book indicate changes from the previous edition.

---

### Summary of Changes for GC35-0386-00 OS/390 Version 2 Release 10 DFSMSHsm

This book contains information previously presented in *DFSMS/MVS Version 1 Release 4 Managing Data Availability in a System-Managed Storage Environment*, SC26-4928-03.

The following sections summarize the changes to that information.

#### New Information

This edition includes the following new information:

- Title and form number changes to reflect inclusion in the OS/390 product library.

---

### Summary of Changes for SC26-4928-02 DFSMS/MVS Version 1 Release 3

This book contains information previously presented in *DFSMS/MVS Version 1 Release 3 Managing Data Availability in a System-Managed Storage Environment*, SC26-4928-02.

The following sections summarize the changes to that information.

#### New Information

This edition includes the following new information:

- A new scenario, “Case 10: Restoring Stacked Output Tapes With a Down-Level Processor.”
- Minor editorial changes to several of the procedures.



---

## Chapter 1. Introduction

The data recovery scenarios presented in this book are real-life examples of data loss situations and how to recover from them. Someday, some other computer installations are going to have these same data loss situations. When that happens, the storage administrators will be able to pick up this book and see what was done to restore the data, and apply that information to their own systems.

Not all possible data loss situations are covered in this book. However, if you have had a memorable experience with data loss and recovery, and think that other DFSMSHsm users could benefit from your experience, send us a Reader's Comment form with the details. You may find your story in the next edition of this book. See "How to send your comments" on page v.

Many of the examples in this book use ICETOOL, the multipurpose data processing utility, and reporting utility, provided as part of DFSORT Release 12 and subsequent releases. *DFSORT Application Programming Guide R14* contains complete information on DFSORT and ICETOOL.



---

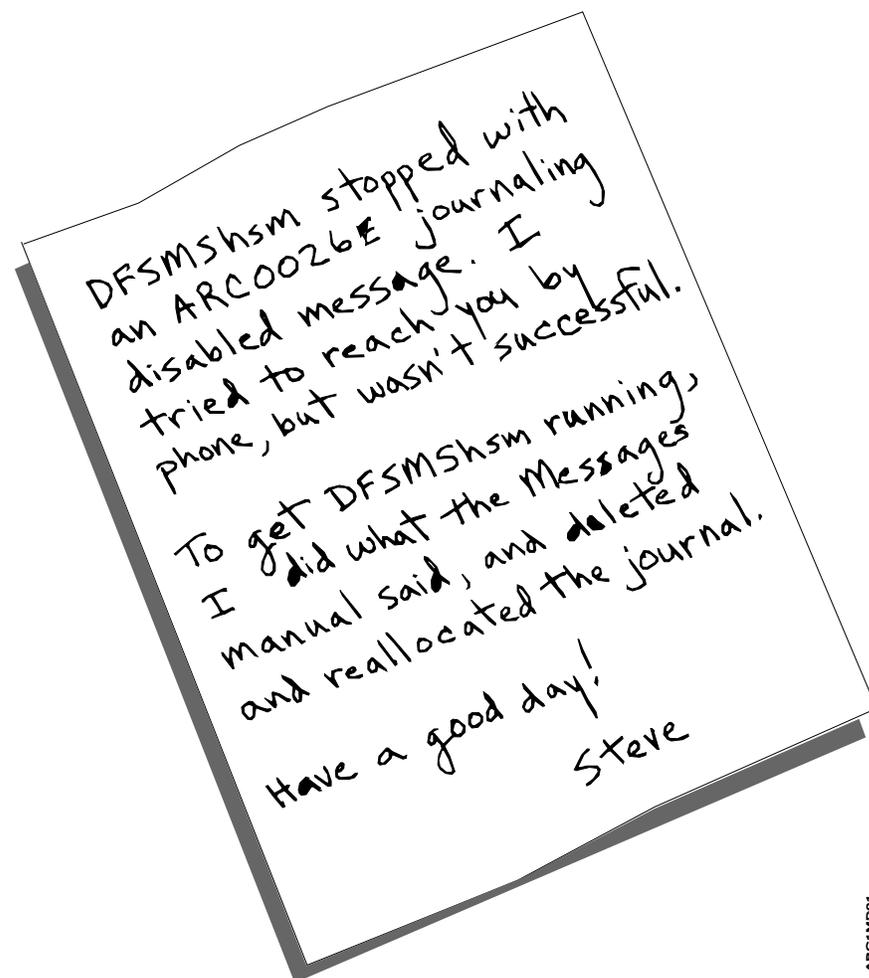
## Chapter 2. Data Recovery Scenarios

In the data loss and recovery scenarios that follow, a description of a problem is presented, followed by the corrective measures you can use for that problem, followed by the preventative measures you can take to prevent that problem from occurring in the future. When you have a data-loss situation, you can quickly scan our examples to see if one of the examples matches your problem. If so, you may be on your way to a speedy recovery.

---

### Case 1: Damaged CDS, Full Journal

It's a cold Monday morning when Rick, a storage administrator, arrives at work to find the following note on his desk:



What does Rick do after he regains consciousness?

The first thing Rick does is review the DFSMSHsm command activity log to see what caused the ARCO026E journaling error. He sees that the ARCO026E error specifies an EOVS problem; the journal has been completely filled up. Rick then examines the backup activity log to find out what caused the journal to completely fill up. He sees that control data set (CDS) backups fail each time the BACKVOL CDS command is issued. Since the BACKVOL CDS command is failing, the journal

isn't getting backed up. Rick also sees that IDCAMS has issued error messages indicating out-of-sequence records in the backup control data set (BCDS). The failures recorded in the backup activity log indicate a "broken" BCDS.

To review the backup activity log, Rick used the DFSMSHsm RELEASE HARDCOPY command. HARDCOPY is an optional parameter of the RELEASE command used to specify that DFSMSHsm close any of the four activity logs if they contain data, and either print the logs or send them to DASD. The DFSMSHsm SETSYS ACTLOGTYPE command determines whether the logs are printed or sent to DASD.

When the activity logs are closed, new copies of the logs are allocated for use.

Once Steve had reset the journal, DFSMSHsm was able to operate because DFSMSHsm addresses the VSAM control intervals directly by key. Backup of the CDS failed because a valid VSAM sequence set is required to read CDSs in key sequence.

What happened to the BCDS? Two MVS images, both having DFSMSHsm installed, updated the shared, BCDS concurrently. When this happened, duplicate records were created in the BCDS, and the index structure of the BCDS was damaged.

After Rick figures out what has happened and what damage has been done, he plans his salvage operation. Although the journal records had been lost when Steve deleted and reallocated the journal, the same information still exists in the BCDS. The BCDS, however, has duplicate records. So the first thing Rick must do is to delete the duplicate records in the BCDS and back up the CDSs. To delete the duplicate records in the BCDS and back up the CDSs, Rick performs the following tasks:

1. Stops DFSMSHsm on all images.
2. Uses the IDCAMS REPRO command to copy the data component of the BCDS to a sequential file. This can be performed by one of the following methods:
  - a. Direct the IDCAMS REPRO command to make a copy of the data component of the BCDS, as in the following example:

```
REPRO IDS(HSM.BCDS.DATA) ODS(RICK.BCDS.DATA)
```
  - b. Direct the IDCAMS REPRO command to make a copy of the BCDS by relative byte address (RBA), as in the following example:

```
REPRO IDS(HSM.BCDS) ODS(RICK.BCDS.DATA) FADDR(0)
```
3. Uses DFSORT's ICETOOL utility to:
  - a. Sort the sequential file by key, eliminating records with duplicate keys. Where there are duplicates, the record most recently updated is the one that is retained.
  - b. Reload the BCDS. The damaged BCDS must be deleted and a new BCDS defined.
4. Starts DFSMSHsm, and issues the DFSMSHsm BACKVOL CDS command to create a good copy of the BCDS and nullify the journal.

Rick used the following JCL to accomplish the preceding steps:

```

//RECOVER JOB
//*****
//* Please change all the following to the appropriate values:
//*      ?USERID - to your user id
//*      ?DFSMSHsm - to the DFSMSHsm prefix for your BCDS
//*****
//* The first step will create a copy of all of the records contained
//* in the data component of the BCDS. These records will not be in
//* order, and may contain multiple records with the same key. In
//* addition, a recovery BCDS will be created in this step. The user
//* must supply the proper space information, volume information, and
//* data set information for the define cluster.
//*****
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//DDIN1 DD DISP=SHR,DSN=?DFSMSHSM.BCDS.DATA
//DDOUT1 DD DISP=(,CATLG),UNIT=SYSDA,
//          DCB=(LRECL=2052,BLKSIZE=20524,RECFM=VB),
//          SPACE=(CYL,(50,10),RLSE),DSN=?USERID.BCDS.RECORDS
//SYSIN DD *
REPRO INFILE(DDIN1) OUTFILE(DDOUT1) FADDR(0)
DEFINE CLUSTER(NAME(?USERID.RECOVER.BCDS) -
  CYL(???) SPEED BUFFERSPACE(530432) -
  VOL(XXXXXX) -
  CISZ(12288) -
  FREESPACE(0 0) -
  KEYS(44 0) -
  RECORDSIZE(2040 2040)) -
  DATA(NAME(?USERID.RECOVER.BCDS.DATA)) -
  INDEX(NAME(?USERID.RECOVER.BCDS.INDEX))
/*
//*****
//* The next step uses DFSORT's ICETOOL utility as follows:
//* - The SORT operator will sort the records extracted from
//* the BCDS data component. They will be sorted by their
//* VSAM keys, and by the time stamp of the last update made
//* to the record. The keys will be sorted in ascending order,
//* and the update time in descending order; the most current
//* updated record being first in the output data set.
//* - The SELECT operator (new in DFSORT R12) will sort the
//* records by their VSAM keys. The order established by the
//* SORT operator, where the most recently updated version of
//* a duplicate record is placed first in the file, is maintained.
//* The duplicate record that is not the most recently updated
//* version is discarded. The output is loaded into the recovery
//* copy of the BCDS that was defined in the first step.
//*****
//STEP2 EXEC PGM=ICETOOL,REGION=4096K,COND=(0,NE)
//TOOLMSG DD SYSOUT=*
//DFSMSG DD SYSOUT=*
//OLDBCDS DD DISP=SHR,DSN=?USERID.BCDS.RECORDS
//TEMP DD DISP=(,PASS),DSN=&&TFILE1,
//        UNIT=SYSDA,SPACE=(CYL,(50,10),RLSE),
//        DCB=(LRECL=2052,BLKSIZE=20524,RECFM=VB)
//NEWBCDS DD DISP=OLD,DSN=?USERID.RECOVER.BCDS
//TOOLIN DD *
* Sort the old BCDS by ascending VSAM key and
* descending time stamp of last update.
  SORT FROM(OLDBCDS) TO(TEMP) USING(SRT1)
* Load only the most recently updated records into the new BCDS.
  SELECT FROM(TEMP) TO(NEWBCDS) ON(5,44,CH) FIRST
/*
//SRT1CNTL DD *
  SORT FIELDS=(5,44,CH,A,53,8,BI,D)
/*

```

## Prevention

Rick's salvage operation is complete. Now, however, Rick wants to prevent the same problem from occurring again, so he:

1. Ensures that no reorganization can occur if DFSMSHsm is running on any image in the complex. To accomplish this Rick does two things:
  - a. He makes sure that SYSDSN enqueues are in the system's inclusion list for global resource serialization (GRS), which allows the SYSDSN enqueues to be passed to all images in the complex that share the CDSs.
  - b. He modifies the JCL used for reorganization of the CDSs by using a disposition of OLD whenever a DD statement refers to a CDS.
2. Defines the DFSMSHsm CDSs with VSAM SHAREOPTIONS(2 3). Information about defining VSAM SHAREOPTIONS for DFSMSHsm can be found in the *OS/390 DFSMSHsm Implementation and Customization Guide* under the headings "VSAM SHAREOPTIONS for Control Data Sets," and "Control Data Set Considerations."

3. Implements a procedure where the backup of the CDSs is closely monitored. One method of monitoring the backups of the CDSs is to review the CDS backup and export messages in the DFSMSHsm backup activity log. Specifically, review the number of records processed by the export function in message IDC0005I, for each CDS. Compare the record count of the current CDS backup with the record count from the preceding days. The record count may not be the same, but it should be close.

Normal update and deletion activity causes some fluctuation in the number of records in the CDSs. However, when a large decrease in records is detected, you should be able to account for the large difference.

Some large fluctuations in the number of CDS records are normal, and are the result of processes that reduce the number of records in a CDS. For example, running the EXPIREBV command deletes records in the BCDS. DFSMSHsm secondary space management deletes records in the MCDS. However, some large fluctuations in the number of records in a CDS can indicate that the CDS has been damaged. For example, in a multiple-MVS-image environment, the reorganization of a CDS occurring on one image while DFSMSHsm is running on another image can damage the structure of the CDS, and cause a large fluctuation in CDS records.

To eliminate any doubt about the soundness of the CDS structure, run the IDCAMS EXAMINE command against the CDS in question. Running the EXAMINE command against your CDSs every day minimizes the chance of CDS damage going undetected.

---

## Case 2: Damaged Journal, Undamaged CDS

A damaged journal is often the result of a hardware-caused error on the journal volume. DFSMSHsm issues error message ARC0026E when an I/O error is encountered in the journal data set. Message ARC0026E states "JOURNALING DISABLED DUE TO I/O ERROR. MIGRATION, BACKUP, DUMP, TAPECOPY, TAPERPL, RECYCLE, AND EXPIREBV HELD."

Once the problem has occurred:

1. Verify the DFSMSHsm CDSs are structurally sound.

Using IDCAMS, run EXAMINE against each of the CDSs. The procedure runs very quickly, and does not require that you bring DFSMSHsm down. The sample

JCL to perform this procedure (which follows) comes from *OS/390 DFSMS Access Method Services for Catalogs*.

```
//EXAMCDS JOB
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
EXAMINE -
  NAME(?hlqua1.MCDS) -
  ERRORLIMIT(0)
//STEP2 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
EXAMINE -
  NAME(?hlqua1.BCDS) -
  ERRORLIMIT(0)
//STEP3 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
EXAMINE -
  NAME(?hlqua1.OCDS) -
  ERRORLIMIT(0)
```

If there are no errors reported by EXAMINE continue to the next step.

If errors **are** reported by EXAMINE, do not proceed to the next step. Go to “Case 1: Damaged CDS, Full Journal” on page 3.

2. Stop DFSMShsm on all images.
3. Do **not** copy the journal.
4. Rename the current journal.
5. Allocate a new journal with the old name. The following JCL allocates a new journal with the old name.

```
//JRNLDEF JOB
//STEP1 EXEC PGM=IEFBR14
//JRNLDD DD DISP=(,CATLG),UNIT=?3390,VOL=SER=?JNLVOL,
// SPACE=(CYL,(100),,CONTIG),DSN=?hlqua1.JRNL
```

6. Start DFSMShsm in EMERG=YES mode, a proclib option.
7. Issue the DFSMShsm BACKVOL CDS command to keep your CDSs in synchronization with the new journal.
8. Delete the old journal.
9. Issue the DFSMShsm SETSYS NOEMERGENCY command and then the DFSMShsm RELEASE command to resume normal processing.

---

## Case 3: Full Journal, Undamaged CDS

The problem of a full journal with an undamaged CDS occurs because either the journal is not large enough, or the CDSs are not being backed up often enough. DFSMShsm issues error message ARC0909E when the DFSMShsm journal threshold is met, and continues to issue the error message each time an additional one percent (or more) of the journal is filled. When the journal is full, ARC0909E states: “JOURNAL DATA SET IS ABOUT 100% FULL.” When the journal is 100% full, message ARC0026E is also issued, which states: “JOURNALING DISABLED DUE TO EOVS ERROR ON JOURNAL. MIGRATION, BACKUP, DUMP, TAPECOPY, TAPERPL, RECYCLE, AND EXPIREBV HELD.”

Once the problem has occurred:

1. Issue the BACKVOL CDS command. The BACKVOL CDS command will back up the journal and the control data sets, and null the journal.

## Prevention

To help prevent the problem of a full journal from occurring, you should ensure that your journal is of adequate size to record all the activity between backups of the CDSs. In conjunction with that, you should schedule CDS backups frequently enough that the journal can handle the activity. Most installations run automatic backup once a day, seven days a week, in which case the journal data set needs to be large enough to contain data for a 24-hour time period.

You should also use the SETSYS MONITOR (JOURNAL) command to send warnings when the journal data set is getting full. When the journal data set is filled to the specified threshold, which you can set to any percentage from 0 to 100 or leave at the DFSMSHsm default value of 80%, a warning message is issued to the system operator. Messages continue to be issued until the journal data set is 100% full. You need to set your threshold at a value that gives you adequate time to back up your CDSs and journal, and allocate a new journal.

Customers that have NetView or another automation product can monitor the ARC0909E and ARC0026E messages, keying in on the 100% and EOVS message variables. When 100% and EOVS are issued, NetView can automatically issue the BACKVOL CDS command.

---

## Case 4: Structurally Damaged CDS, Missing Journal Records

A structurally damaged CDS is usually caused by multiple updates occurring at the same instant in time within the same CDS. Simultaneous updates to the same CDS can occur, in a multiple-MVS-image environment, if a job running on one image reorganizes the CDS while DFSMSHsm is still running on another image. Without proper share options and serialization methods the problem of a structurally damaged CDS along with missing journal records can easily happen.

Problems arise when there are no CDS and journal backup versions prior to when the damage occurred. This most commonly occurs when damage to a CDS is not immediately detected. As new CDS and journal backups are being created, the older versions are being deleted. By the time the damaged CDS is detected, the backup versions that were created prior to the error have been deleted.

When you have a structurally damaged CDS and missing journal records, you must work with the records currently in your damaged CDS. To repair your damaged CDS, follow the same steps that Rick followed in “Case 1: Damaged CDS, Full Journal” on page 3, substituting your damaged CDS for Rick’s BCDS.

## Prevention

To prevent the problem of structural damage to the CDS along with missing journal records from occurring in the future:

- Schedule reorganizations of CDSs to occur only when necessary. If the CDS is of adequate size, you should not have to reorganize often.
- Use the proper procedures to reorganize your CDSs. See *OS/390 DFSMSHsm Implementation and Customization Guide* under the heading “Reorganizing the Control Data Sets.”

- Ensure that no reorganization can occur when DFSMSHsm is running on any MVS image in the complex. To do this:
  - Make sure that SYSDSN enqueues are in the system’s inclusion list for global resource serialization (GRS), which allows the SYSDSN enqueues to be passed to all images in the complex that share the CDSs.
  - Modify the JCL used for reorganization of the CDSs by using a disposition of OLD whenever a DD statement refers to a CDS.
- Define the DFSMSHsm CDSs with VSAM SHAREOPTIONS(2 3). Information about defining VSAM SHAREOPTIONS for DFSMSHsm can be found in the *OS/390 DFSMSHsm Implementation and Customization Guide* under the headings “VSAM SHAREOPTIONS for Control Data Sets,” and “Control Data Set Considerations.”
- Use the IDCAMS EXAMINE command to do preventive examines.
- Reassess the number of backup versions you keep for the CDSs and journal. The more backup versions you have, the greater your chances are of having a good CDS and journal backup version to use should the need arise.

---

## Case 5: Time Sequence Gaps in CDS Records, Journal Records Also Missing

One way the problem of time sequence gaps in the CDS records occurring along with missing journal records is user error during a reorganization.

Take the case where an installation uses generation data sets (GDSs) when reorganizing the CDSs. The user attempts to use IDCAMS REPRO to create an output data set for an MCDS. The MCDS is not successfully copied to the output data set, but the user doesn’t notice. Then the user uses IDCAMS REPRO to create a new, reorganized MCDS, not knowing that the input data set is actually the last version of the MCDS successfully copied, and is not the current MCDS at all.

After the reorganization completes, the administrator thinks that everything went as planned, and continues with everyday processing.

DFSMSHsm runs with an old copy of the MCDS for a week before a high number of data set recall failures signal that there is a problem. The old MCDS that is being used, of course, does not contain the more recently migrated data sets.

When the storage administrator finally discovers what has happened, he looks to see if he can recover from a backup version. He backs up once a day, and keeps three backup versions. The reorganization, however, was performed seven days ago. That means he doesn’t have any backups prior to when the reorganization occurred, for either the journal or the MCDS.

Recovery in this instance is very complicated. The best course of action is to call your IBM Area Storage Specialist for recovery assistance.

## Prevention

To prevent the problem from occurring:

- Write a reorganization job that:
  1. Calls on IDCAMS REPRO to create an output data set
  2. Checks for a successful completion code on the creation of the output data set
  3. Creates an input data set

If a successful completion code is not received for the creation of the output data set, the job ends, and the input data set is not created.

- Make CDS data sets large enough that the need for a reorganization is infrequent.

If your CDS is currently too small and you want to change that, the next time you reorganize the data set, add 10 cylinders. If the data set is still too small, the next time you reorganize add another 10 cylinders. Do this until the data set is large enough that the need for reorganizations is infrequent.

- Before every reorganization do a backup of the CDS, and keep those backup tapes until it's time for the next reorganization.
- Use the proper procedures to reorganize your CDSs. See *OS/390 DFSMSHsm Implementation and Customization Guide* under the heading "Reorganizing the Control Data Sets."
- Protect DFSMSHsm's resources with RACF, a component of the SecureWay Security Server for OS/390, or a comparable product.

---

## Case 6: Overwritten Migration Tape

The main reason for this type of damage is that the wrong tape volume gets mounted to satisfy a mount request for a scratch tape. The problem usually goes undiscovered unless you need the overwritten tape volume to recall a data set.

The use of automated tape libraries eliminates this problem, since the problem is caused by human error. Most tape management systems also prevent this problem.

Use the following recovery procedure to restore access to DFSMSHsm data on tape cartridges. This procedure won't work for reel-type tape because of a difference in file format.

1. Copy the damaged tape volume.

This step is optional. If the user data that was written over the DFSMSHsm data is wanted, you will need to create a copy of that data. At the end of this procedure, the damaged tape volume will be returned to scratch or to DFSMSHsm for its later use, and the user data that was written on this volume will be lost.

2. Reinitialize the damaged tape volume.

Reinitialize the tape volume with the data set name used by DFSMSHsm for all of its tape volumes. Write a dummy data set with the first file on the volume named:

*prefix*.HMIGTAPE.DATASET

The *prefix* can be any valid qualifier. For example you can use the migration prefix as specified on the SETSYS MIGRATEPREFIX command, or the UID. This step will create a good tape label on the damaged tape. Once the tape label is valid, the volume can be processed by RECYCLE in step 5 on page 11 of this procedure.

IEBGENER can be used to perform the reinitialization task.

The sample JCL that follows can be used to reinitialize a damaged migration volume. Modify the JCL to fit your needs by:

- Changing the UNIT parameter on the SYSUT2 DD statement, if needed
- Changing *prefix* to the appropriate prefix

- Changing *ml2volser* to your volume serial number

Included in the example that follows are the various messages generated from this job and the appropriate replies.

```
//GENR      JOB
//STEP1     EXEC PGM=IEBGENER
//SYSUT1    DD DUMMY,LRECL=16384,BLKSIZE=16384,RECFM=F
//SYSUT2    DD DISP=(NEW,KEEP),DCB=*.SYSUT1,UNIT=?3480,
//          DSN=prefix.HMIGTAPE.DATASET,VOL=SER=?ml2volser
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
/*
                                J E S 2  J O B  L O G

JOB09371  $HASP373  HAMMERTA  STARTED - INIT 3 - CLASS A - SYS ..
JOB09371  *IEF233A  M 5A8,HAMMRT,,HAMMERTA,STEP1
JOB09371  IEC512I  LBL ERR 5A8,SMFDMP,SL,HAMMRT,SL,HAMMERTA,STEP1
JOB09371  *IEC507D  E 5A8,SMFDMP,HAMMERTA,STEP1,DMP.T241T240.DATA
JOB09371  IEC507D  REPLY 'U'-USE OR 'M'-UNLOAD
JOB09371  R 78,U
JOB09371  *IEC534D  A 5A8,SMFDMP,SL,HAMMERTA,STEP1
JOB09371  IEC534D  REPLY 'U'-USE OR 'M'-UNLOAD
JOB09371  R 84,U
JOB09371  *IEC704A  L 5A8,HAMMRT,SL,NOCOMP,HAMMERTA,STEP1
JOB09371  IEC704A  REPLY 'VOLSER,OWNER INFORMATION','M'OR'U'
JOB09371  R 87,'HAMMRT,DFSMSHSM'
JOB09371  IEC705I  TAPE ON 5A8,HAMMRT,SL,NOCOMP,HAMMERTA,STEP1
JOB09371  IEF234E  K 5A8,HAMMRT,PVT,HAMMERTA,STEP1
JOB09371  HAMMERTA STEP1 : 00:17:30/ 00:00:00/ 53 - R000
JOB09371  $HASP395  HAMMERTA  ENDED
```

3. Determine if the first data set on the damaged tape volume spanned from another volume.

Issue the LIST TTOC(*volser*) ODS(*dsname*) command to determine if your damaged tape volume is associated with a previous tape volume. If your damaged tape is associated with a previous tape, it will be listed under the heading "PREV VOL." When there is a previous tape, it means the first data set on your damaged tape began on the previous tape.

To verify whether your damaged tape is associated with a previous tape, issue the LIST TTOC(*prev\_volser*) command and check to see that the last data set name on the previous tape is the same as the first data set name on your damaged tape.

4. Delete the first data set on your damaged tape volume if it spanned from a previous tape volume.

To delete the first data set on a damaged migration tape, issue:

```
DELETE dsn
```

5. Issue the following command:

```
RECYCLE VOLUME(volser) EXECUTE
```

Readable, complete, valid data sets are recycled to a new volume.

6. Identify data sets that have been overwritten and build appropriate recovery commands.

Create a list showing which migrated data sets you need to recover (because the RECYCLE command was not able to remove them from the volume), by issuing:

```
LIST TTOC(volser) ODS(dsn)
```

Build RECOVER commands for each data set on the list, but do not process the commands until step 8.

An alternate to using the LIST TTOC command is to use DFSORT. The following is an example of an ICETOOL (DFSORT) job that identifies and builds RECOVER commands for each migrated data set that was overwritten. This job only builds the RECOVER commands, it doesn't process them.

```
//RCVCMDS JOB
//*****
//* Please change all the following to the appropriate values:
//*      ?USERID - to your user id
//*      ?DFSMSHsm - to the DFSMSHsm prefix for your MCDS
//*      ?VLSER - to the volser of your damaged tape volume
//*****
//* This step uses DFSORT's ICETOOL utility to build RECOVER
//* commands.
//*****
//STEP1 EXEC PGM=ICETOOL,REGION=4096K
//TOOLMSG DD SYSOUT=*
//DFSMSG DD SYSOUT=*
//MCDS DD DSN=?DFSMSHSM.MCDS,DISP=SHR
//CMDS DD DSN=?USERID.TEST.SORTMCD,DISP=(NEW,CATLG),
//      DCB=(LRECL=255,BLKSIZE=20520,RECFM=VB),
//      SPACE=(CYL,(5,1),RLSE),UNIT=SYSDA
//TOOLIN DD *
* Build the RECOVER commands.
  COPY FROM(MCDS) TO(CMDS) USING(CPY1)
/*
//*****
//* The INCLUDE statement selects MCD records (X'00'
//* at offset 51), that have the volser of your damaged tape and
//* have the MCDFASN bit on.
//* The OUTREC statement builds a RECOVER command for each
//* migrated data set found.
//*****
//CPY1CNTL DD *
  INCLUDE COND=(51,1,BI,EQ,X'00',AND,69,6,CH,EQ,C'?VLSER',
                AND,75,1,BI,EQ,B'1.....')
  OUTREC FIELDS=(1,4,C' HSEND RECOVER ',5,44)
/*
```

7. Issue the RECYCLE VOLUME(*volser*) FORCE EXECUTE command for the damaged volume.

The FORCE parameter of the RECYCLE command causes all data sets with errors to be deleted from the control data sets and the catalog. The deletion makes the tape empty, and the tape is removed from DFSMSHsm's inventory of tapes that contain data.

If the RECYCLE command fails with return code 16, you must issue a second RECYCLE VOLUME command to recycle the valid data sets while processing in single buffer mode. For an explanation of return code 16, refer to *OS/390 MVS System Messages, Vol 1 (ABA-ASA)*. See the table in the appendix titled "Reason Codes for Message ARC0734I when the Action is RECYCLE and Return Code is 16."

8. Issue the RECOVER commands for the migrated data sets that have been overwritten. The RECOVER commands were generated in step 6 on page 11.

## Prevention

There are two ways to prevent the problem from occurring:

- Protect DFSMSHsm's resources with RACF or a comparable product.

- Protect your alternate volumes if your tape management system doesn't do it for you. DFSMSrmm protects your alternate tape volumes. Some products, however, do not.

---

## Case 7: Overwritten Backup Tape

The main reason for this type of damage is that the wrong tape volume gets mounted to satisfy a mount request for a scratch tape. The problem usually goes undiscovered unless you need the overwritten tape volume to recover a data set.

The use of automated tape libraries eliminates this problem, since the problem is caused by human error. Most tape management systems also prevent this problem.

Use the following recovery procedure to restore access to DFSMSHsm data on single-file-format backup tapes that have been partially overwritten by user data.

1. Copy the user data to a safe place.

This step is optional. If the user data that was written over the DFSMSHsm data is wanted, you will need to create a copy of that data. At the end of this procedure, the damaged tape volume will be returned to scratch or to DFSMSHsm for its later use, and the user data that was written on this volume will be lost.

2. Reinitialize the damaged tape volume.

Reinitialize the tape volume with the data set name used by DFSMSHsm for all of its tape volumes. Write a dummy data set with the first file on the volume named:

*prefix*.**BACKTAPE.DATASET**

The *prefix* is one of the following:

- The backup prefix as specified on the SETSYS BACKUPPREFIX command
- The UID if prefixes are not specified with the SETSYS commands

This step will create a good tape label on the damaged tape. Once the tape label is valid, the volume can be processed by RECYCLE in 5 on page 15 of this procedure.

IEBGENER can be used to perform the reinitialization task.

The sample JCL that follows can be used for a damaged backup volume. Modify the JCL to fit your needs by:

- Changing the VOL=SER parameter, as needed
- Changing the *prefix*, as needed

Included in the example that follows are the various messages generated from this job and the appropriate replies.

```

//GENR      JOB
//STEP1     EXEC PGM=IEBGENER
//SYSUT1    DD DUMMY,LRECL=16384,BLKSIZE=16384,RECFM=F
//SYSUT2    DD DISP=(NEW,KEEP),DCB=*.SYSUT1,UNIT=?3480,
//          DSN=prefix.BACKTAPE.DATASET,VOL=SER=?m12volser
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
/*
                J E S 2  J O B  L O G

JOB09371  $HASP373  HAMMERTA STARTED - INIT 3 - CLASS A - SYS ..
JOB09371  *IEF233A M 5A8,HAMMRT,,HAMMERTA,STEP1
JOB09371  IEC512I LBL ERR 5A8,SMFDMP,SL,HAMMRT,SL,HAMMERTA,STEP1
JOB09371  *IEC507D E 5A8,SMFDMP,HAMMERTA,STEP1,DMP.T241T240.DATA
JOB09371  IEC507D REPLY 'U'-USE OR 'M'-UNLOAD
JOB09371  R 78,U
JOB09371  *IEC534D A 5A8,SMFDMP,SL,HAMMERTA,STEP1
JOB09371  IEC534D REPLY 'U'-USE OR 'M'-UNLOAD
JOB09371  R 84,U
JOB09371  *IEC704A L 5A8,HAMMRT,SL,NOCOMP,HAMMERTA,STEP1
JOB09371  IEC704A REPLY 'VOLSER,OWNER INFORMATION','M'OR'U'
JOB09371  R 87,'HAMMRT,DFSMSHSM'
JOB09371  IEC705I TAPE ON 5A8,HAMMRT,SL,NOCOMP,HAMMERTA,STEP1
JOB09371  IEF234E K 5A8,HAMMRT,PVT,HAMMERTA,STEP1
JOB09371  HAMMERTA STEP1 : 00:17:30/ 00:00:00/ 53 - R000
JOB09371  $HASP395  HAMMERTA ENDED

```

3. Determine if the first data set on the damaged tape volume spanned from another volume.

Issue the LIST TTOC(*volser*) ODS(*dsname*) command to determine if your damaged tape volume is associated with a previous tape volume. If your damaged tape is associated with a previous tape, it will be listed under the heading "PREV VOL." When there is a previous tape, it means the first data set on your damaged tape began on the previous tape.

To verify whether your damaged tape is associated with a previous tape, issue the LIST TTOC(*prev\_volser*) command and check to see that the last data set name on the previous tape is the same as the first data set name on your damaged tape.

4. Delete the first data set on your damaged tape volume if it spanned from a previous volume.

The original data set name and the backup generation number will be needed. To get this information, issue the FIXCDS command to display the MCC record for the backup version. The backup version generation number is a 2-byte field contained in the MCCGEN field at offset X'46' in the FIXCDS display output. For example, issuing:

```
HSEND FIXCDS C DFHSM.BACK.T500718.TONY3.AHRENS.I3006
```

gives you the following results:

```

MCH= 01602400 A6F7B92B F4BBC221 A6DAB593 9C8BBA03
* 7 4 B *
+000 E3D6D5E8 F34BC1C8 D9C5D5E2 4BC2C3C4 E2404040 40404040 40404040 40404040
*TONY3.AHRENS.BCDS *
+020 40404040 40404040 40404040 E7E8E9F9 F0F50003 78048081 18075106 0093006F
* XYZ905 *
+040 00081000 0000001C 000004DD 0121C0E2 0000243C 00003060 0000E3D6
D5E8F0F5
.
.

```

The output gives you the original data set name: TONY3.AHRENS.BCDS and the version number: +040 00081000 000000 **1C** 000004DD

The deletion would incorporate the information as follows:

```
BDELETE DSN(TONY3.AHRENS.BCDS) VERSION(28)
```

**Note:** The version number gets converted from 1C to decimal (28).

To delete the first data set on a damaged backup tape, issue:

```
BDELETE dsn VERSIONS(n)
```

5. Issue the following command:

```
RECYCLE VOLUME(volser) EXECUTE
```

Readable, complete, valid data sets are recycled to a new volume.

6. Identify data sets that have been overwritten and build appropriate recovery commands.

Build BACKDS commands for all the backup data sets that have been overwritten, and that are the most recent backup version. The following ICETOOL (DFSORT) job can identify and build the BACKDS commands for these data sets. This job only builds the BACKDS commands, it doesn't process them.

```
//BDSCMDS JOB
//*****
//* Please change all the following to the appropriate values:
//*      ?USERID - to your user id
//*      ?DFSMShsm - to the DFSMSHsm prefix for your BCDS
//*      ?VLSER - to the volser of your damaged backup volume
//*****
//* This step uses DFSORT's ICETOOL utility to find all
//* associated MCB and MCC records, identifying the most recent
//* backup version that was overwritten on the damaged backup
//* volume. It also generates the BACKDS commands to create a
//* new backup version.
//*****
//STEP1 EXEC PGM=ICETOOL,REGION=4096K
//TOOLMSG DD SYSOUT=*
//DFSMSG DD SYSOUT=*
//BCDS DD DISP=SHR,DSN=?DFSMShsm.BCDS,AMP='BUFND=50'
//BCDSREC1 DD DISP=(NEW,CATLG),DSN=?USERID.BCDSRT1,
// UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE),
// DCB=(LRECL=2048,BLKSIZE=23476,RECFM=VB)
//BCDSRECS DD DISP=(MOD,CATLG),DSN=?USERID.BCDSRT2,
// UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE),
// DCB=(LRECL=255,BLKSIZE=23476,RECFM=VB)
//MERRECS DD DISP=(NEW,CATLG),DSN=?USERID.MERGED.BCDS.RECORDS,
// UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE),
// DCB=(LRECL=255,BLKSIZE=23476,RECFM=VB)
//BKDSREC DD DISP=(NEW,CATLG),DSN=?USERID.BKDS.RECORDS,
// UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE),
// DCB=(LRECL=255,BLKSIZE=23476,RECFM=VB)
//TOOLIN DD *
```

(continued on next page)

```

* Show the minimum, maximum, average and total for the
* record lengths.
  STATS FROM(BCDS) ON(VLEN) VSAMTYPE(V)
* Remove records too short to contain the backup version name.
  COPY FROM(BCDS) TO(BCDSREC1) USING(SBC1)
* Find the MCB records and extract needed information.
  COPY FROM(BCDSREC1) TO(BCDSRECS) USING(SBCB)
* Find the MCC records and extract needed information.
  COPY FROM(BCDSREC1) TO(BCDSRECS) USING(SBCC)
* Flag data set names for which commands are needed.
  SORT FROM(BCDSRECS) TO(MERRECS) USING(MBCD)
* Build BACKDS commands for flagged data set names.
  COPY FROM(MERRECS) TO(BKDSREC) USING(BKDS)
/*
//*****
//* The SBC1 control statement searches the BCDS for all
//* records with a length equal to or greater than X'C1'
//* or decimal 193. This first pass is performed to remove
//* records too short to contain the backup version name.
//* The output is placed in the data set identified by the
//* BCDSREC1 DD statement.
//*****
//SBC1CNTL DD *
  INCLUDE COND=(1,4,BI,GE,X'00C1')
/*
//*****
//* The SBCB control statements search the output from the
//* first COPY operator, looking for all MCB records.
//* For each MCB record found, a record is created as follows:
//* the original data set name is placed in position 5,
//* followed by a blank space, followed by the DFSMSHsm-generated,
//* backup-version name of the most current backup,
//* followed by a counter of '0001'.
//* The output is placed in the data set identified by the
//* BCDSRECS DD statement.
//*****
//SBCBCNTL DD *
  INCLUDE COND=(51,1,BI,EQ,X'20')
  OUTREC FIELDS=(1,4,5,44,X,149,44,X'0001')
/*
//*****
//* The SBCC control statements search the output from the
//* first COPY operator, looking for all MCC records with the
//* volser of the damaged backup volume.
//* For each MCC record found, a record is created as follows:
//* the original data set name is placed in position 5,
//* followed by a blank space, followed by the DFSMSHsm-generated,
//* backup-version name, followed by a counter of '0001'.
//* The output is placed in the data set identified by the
//* BCDSRECS DD statement.
//*****
//SBCCCNTL DD *
  INCLUDE COND=(51,1,BI,EQ,X'24',AND,113,6,CH,EQ,C'?VLSER')
  OUTREC FIELDS=(1,4,69,44,X,5,44,X'0001')
/*
//*****
//* The MBCD control statements sort the data set (identified
//* by the BCDSRECS DD statement) on the DFSMSHsm-generated,
//* backup-version name, and sum the field at offset 94,
//* which is the counter field. They will also delete all
//* duplicate records.
//* The output is placed in the data set identified by the
//* MERRECS DD statement.
//*****

```

(continued on next page)

```

//MBCDCNTL DD *
SORT FIELDS=(50,44,CH,A)
SUM FIELDS=(94,2,BI)
/*
//*****
//*   The BKDS control statements search the data set
//*   (identified by the MERRECS DD statement), looking for
//*   all records with the counter field equal to '0002'.
//*   These records indicate that this backup version on the
//*   damaged backup volume is the most recent backup version,
//*   and needs to have a BACKDS command created. The BACKDS
//*   commands are created by the OUTREC statement and are
//*   placed in the data set identified by the MERRECS DD
//*   statement.
//*****
//BKDSCNTL DD *
INCLUDE COND=(94,2,BI,EQ,X'0002')
OUTREC FIELDS=(1,4,C' HSEND BACKDS ',5,44)
/*

```

7. Issue the DELVOL volser BACKUP(PURGE) command. The DELVOL ... (PURGE) command will perform the same functions as a RECYCLE ... FORCE command against the volume without causing a tape mount. The damaged backup tape volume will be deleted from DFSMSHsm inventory and any other control records describing the contents of the volume. DFSMSHsm deletes any association with a day in the backup cycle or with a set of spill backup volumes.
8. Issue the BACKDS commands that were generated in step 6 on page 15.

**Note:** If you are backing up a data set (with the BACKDS command) that resides on an ML2 tape volume, you need to use the RECALL command to recall the data set prior to issuing the BACKDS command.

## Prevention

There are two ways to prevent the problem from occurring:

- Protect DFSMSHsm's resources with RACF or a comparable product.
- Protect your alternate volumes if your tape management system doesn't do it for you.

---

## Case 8: Head Crash On an ML1 Volume

To handle a head crash on an ML1 volume once it has occurred, perform the following steps:

1. Delete the ML1 volume entry of the damaged volume with the following command:

```

DELVOL volser MIGRATION(PURGE)

```
2. Initialize another DASD volume of the same device type as the damaged volume, using the volume serial number of the damaged volume.
3. Restore the damaged ML1 volume from a dump copy. If DFSMSHsm was used to create the dump copy, you can use the following command to restore the ML1 volume:

```

RECOVER * TOVOLUME(volser) UNIT(unittype) FROMDUMP

```
4. If there was an SDSP data set on the damaged ML1 volume, check to see if there is a backup copy of the SDSP data set that is more recent than the dump copy used to restore the volume. If so, recover the backup copy of the SDSP. If an IDCAMS EXPORT command was used to create the backup copy, use the IDCAMS IMPORT command to recover the data set.

5. Add the restored ML1 volume to the list of volumes that DFSMSHsmowns. Use the ADDVOL command, specifying the DRAIN parameter. The DRAIN parameter prevents the volume from being selected for migration output. For example:

```
ADDVOL volser UNIT(unittype) MIGRATION(ML1 DRAIN)
```

If there was an SDSP data set on your ML1 volume, be sure to specify the SDSP parameter on your ADDVOL command. For example:

```
ADDVOL volser UNIT(unittype) MIGRATION(ML1 DRAIN SDSP)
```

6. Move all valid migration data sets and VTOC copy data sets off the ML1 volume with the following FREEVOL command:

```
FREEVOL MIGRATIONVOLUME(volser) AGE(0) TARGETLEVEL(ML1)
```

It is possible to generate tape mounts for ML2 tape volumes if there are valid, migrated, SMS-managed data sets that have reached the criteria for migrating to ML2.

7. Move all valid backup versions to backup volumes with the following command:

```
FREEVOL ML1BACKUPVERSIONS
```

The FREEVOL command causes all valid backup versions from all ML1 volumes to be moved to backup volumes.

8. Identify all migrated data sets that are lost.

The following is an example of an ICETOOL (DFSORT) job that identifies migrated data sets that are lost, and builds HDELETE and HRECOVER commands for each data set. This job only builds the HDELETE and HRECOVER commands, it doesn't process them.

```

//SMCD    JOB
//*****
//* Please change all the following to the appropriate values:
//*      ?USERID - to your user id
//*      ?DFSMSHsm - to the DFSMSHsm prefix for your MCDS
//*      ?VLSER - to the volser of your damaged MLI volume
//*****
//* This step uses DFSORT's ICETOOL utility to locate all
//* migrated data sets lost from the damaged MLI volume, and
//* build HDELETE and HRECOVER commands for each data set found.
//* The HDELETE commands are placed in the data set named
//* userid.HDELETE.DATA. The HRECOVER commands are placed
//* in the data set named userid.HRECOVER.DATA.
//*****
//STEP1   EXEC PGM=ICETOOL,REGION=4096K
//TOOLMSG DD SYSOUT=*
//DFSMSG  DD SYSOUT=*
//MCDS    DD DSN=?DFSMSHSM.MCDS,DISP=SHR
//HDELCMDS DD DSN=?USERID.HDELETE.DATA,DISP=(NEW,CATLG),
//          DCB=(LRECL=255,BLKSIZE=20520,RECFM=VB),
//          SPACE=(TRK,(5,1),RLSE),UNIT=SYSDA
//HRCVCMDS DD DSN=?USERID.HRECOVER.DATA,DISP=(NEW,CATLG),
//          DCB=(LRECL=255,BLKSIZE=20520,RECFM=VB),
//          SPACE=(TRK,(5,1),RLSE),UNIT=SYSDA
//TOOLIN  DD *
* Build HDELETE commands.
  COPY FROM(MCDS) TO(HDELCMDS) USING(CPY1)
* Build HRECOVER commands.
  COPY FROM(MCDS) TO(HRCVCMDS) USING(CPY2)
/*
//*****
//* The INCLUDE statement selects MCD records ('00' at +51)
//* that contain the volser of your recovered MLI volume
//* (at offset 69), and have the MCDFASN bit on (a value greater
//* than X'80' at offset 75).
//* The OUTREC statement builds an HDELETE command for each MCD
//* record selected. The data set name, which comes from the
//* MCD record at offset 5, is placed following the
//* characters "HDELETE."
//*****
//CPY1CNTL DD *
  INCLUDE COND=(51,1,BI,EQ,X'00',AND,69,6,CH,EQ,C'?VLSER',
                AND,75,1,BI,EQ,B'1.....')
  OUTREC FIELDS=(1,4,C' HDELETE ',5,44)
/*
//*****
//* The INCLUDE statement selects MCD records ('00' at +51)
//* that contain the volser of your recovered MLI volume
//* (at offset 69), and have the MCDFASN bit on (a value greater
//* than X'80' at offset 75).
//* The OUTREC statement builds an HRECOVER command for each MCD
//* record selected. The data set name, which comes from the
//* MCD record at offset 5, is placed following the
//* characters "HRECOVER."
//*****
//CPY2CNTL DD *
  INCLUDE COND=(51,1,BI,EQ,X'00',AND,69,6,CH,EQ,C'?VLSER',
                AND,75,1,BI,EQ,B'1.....')
  OUTREC FIELDS=(1,4,C' HRECOVER ',5,44)
/*

```

## 9. Identify all lost backup versions.

The following is an example of an ICETOOL (DFSORT) job that identifies backup versions of data sets that are lost, and builds HSEND BDELETE commands for those data sets. This job only builds the HSEND BDELETE commands, it doesn't process them.

```

//SMCC      JOB
//*****
//* Please change all the following to the appropriate values:
//*      ?USERID - to your user id
//*      ?DFSMShsm - to the DFSMShsm prefix for your BCDS
//*      ?VLSER - to the volser of your damaged MLI volume
//*****
//* This job extracts data from MCC records to create
//* HSEND BDELETE commands for all backup versions that were lost
//* from the MLI volume.
//* The HSEND BDELETE commands are placed in the data set named
//* userid.BDEL.DATA.
//*****
//* The first step uses DFSORT's ICETOOL utility to extract
//* data from MCC records to produce skeleton HSEND BDELETE
//* commands and backup version values.
//*****
//STEP1     EXEC PGM=ICETOOL,REGION=4096K
//TOOLMSG   DD SYSOUT=*
//DFSMSG    DD SYSOUT=*
//BCDS      DD DISP=SHR,DSN=?DFSMShsm.BCDS,AMP='BUFND=50'
//BCDREC1   DD DISP=(NEW,PASS),DSN=&&SORTOUT1,
//           UNIT=SYSDA,SPACE=(TRK,(15,5),RLSE),
//           DCB=(LRECL=2048,BLKSIZE=23476,RECFM=VB)
//BCDRECS   DD DISP=(NEW,CATLG),DSN=?USERID.MCC.VERS,
//           UNIT=SYSDA,SPACE=(TRK,(10,1),RLSE),
//           DCB=(LRECL=255,BLKSIZE=23476,RECFM=VB)
//VERSION1  DD DISP=(NEW,CATLG),DSN=&&VERS1,UNIT=SYSDA,
//           SPACE=(TRK,(10,1),RLSE)
//VERSION2  DD DISP=(NEW,CATLG),DSN=?USERID.MCC.VERS2,
//           UNIT=SYSDA,SPACE=(TRK,(10,1),RLSE),
//           DCB=(LRECL=251,BLKSIZE=0,RECFM=FA)
//TOOLIN    DD *
* Find MCC records for damaged MLI volume.
  COPY FROM(BCDS) TO(BCDREC1) USING(SMCC)
* Build skeleton HSEND BDELETE command from each record.
  COPY FROM(BCDREC1) TO(BCDRECS) USING(SMCC)
* Extract data set name and backup version number
  from each record.
  DISPLAY FROM(BCDREC1) LIST(VERSION1) -
    ON(5,44,CH) ON(139,2,BI) NOHEADER
* Make information from display look like HSEND BDELETE command.
  COPY FROM(VERSION1) TO(VERSION2) USING(VERS)
/*
//*****
//* Selects all MCC records that point to your damaged MLI volume.
//*****
//SMC1CNTL DD *
  INCLUDE COND=(51,1,BI,EQ,X'24',AND,113,6,CH,EQ,C'?VLSER')
  OPTION VLSHRT
/*
//*****
//* Builds skeleton HSEND BDELETE commands.
//*****
//SMCCCNTL DD *
  OUTREC FIELDS=(1,4,C' HSEND BDELETE ',69,44,C' VERSIONS(0000) ',
    5,44)
/*

```

(continued on next page)

```

//*****
//* Formats output from the DISPLAY operator to resemble the
//* above HSEND BDELETE commands.
//*****
//VERSCNTL DD *
OUTREC FIELDS=(69X,61,4,2X,2,44,251:X)
/*
//*****
//* This step uses IDCAMS to convert the fixed-length records of
//* ?USERID.MCC.VERS2 to variable-length records compatible with
//* ?USERID.MCC.VERS so these two data sets can be further
//* processed together.
//*****
//STEP2 EXEC PGM=IDCAMS,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//IN DD DISP=OLD,DSN=?USERID.MCC.VERS2
//OUT DD DISP=(NEW,CATLG),DSN=?USERID.VB2.VERSION,
// UNIT=SYSDA,SPACE=(TRK,(1,1),RLSE),
// DCB=(LRECL=255,BLKSIZE=23476,RECFM=VB)
//SYSIN DD *
REPRO INFILE(IN) OUTFILE(OUT)
/*
//*****
//* This step uses DFSORT's ICETOOL utility to combine the BDELETE
//* command skeleton in ?USERID.MCC.VERS with the decimal value
//* for the backup version in ?USERID.VB2.VERSION, to produce the
//* final HSEND BDELETE commands in ?USERID.BDEL.DATA.
//*****
//STEP3 EXEC PGM=ICETOOL,REGION=4096K,COND=(0,NE)
//TOOLMSG DD SYSOUT=*
//DFSMSG DD SYSOUT=*
//COMBINE DD DISP=OLD,DSN=?USERID.MCC.VERS
// DD DISP=OLD,DSN=?USERID.VB2.VERSION
//BDELCMDS DD DISP=(NEW,CATLG),DSN=?USERID.BDEL.DATA,
// UNIT=SYSDA,SPACE=(TRK,(5,1),RLSE),
// DCB=*.COMBINE
//TOOLIN DD *
* Build complete HSEND BDELETE commands.
SORT FROM(COMBINE) TO(BDELCMDS) USING(CMBN)
/*
//*****
//* Combines skeleton commands and backup version value to
//* produce the HSEND BDELETE commands.
//*****
//CMBNCNTL DD *
SORT FIELDS=(80,44,CH,A)
SUM FIELDS=(74,4,ZD)
OUTREC FIELDS=(1,4,5,75)
OPTION ZDPRINT
/*

```

10. Identify which of the lost backup-version data sets were the most recent backup versions of the original data sets.

The following is an example of an ICETOOL (DFSORT) job that identifies these data sets, and also builds HSEND BACKDS commands for each of the identified data sets. This job only builds the HSEND BACKDS commands, it doesn't process them.

You will have to assess whether to create new backup versions at this time, and you may want to inform the owners of the data sets that the most recent backup versions of their data sets have been lost.

```

//SMCB      JOB
//*****
//* Please change all the following to the appropriate values:
//*      ?USERID - to your user id
//*      ?DFSMSHsm - to the DFSMSHsm prefix for your BCDS
//*      ?VLSER - to the volser of your damaged ML1 volume
//*****
//* This step uses DFSORT's ICETOOL utility to locate the most
//* recent of any backup versions that were lost from the damaged
//* ML1 volume.
//* It builds HSEND BACKDS commands for each backup version found.
//* The HSEND BACKDS commands are placed in the data set named
//* userid.BACKDS.CMDS.
//*****
//STEP1 EXEC PGM=ICETOOL,REGION=4096K
//TOOLMSG DD SYSOUT=*
//DFSMSG DD SYSOUT=*
//BCDS DD DSN=?DFSMSHSM.BCDS,DISP=SHR
//CMDS DD DSN=?USERID.BACKDS.CMDS,DISP=(NEW,CATLG),
//      DCB=(LRECL=255,BLKSIZE=20520,RECFM=VB),
//      SPACE=(TRK,(5,1),RLSE),UNIT=SYSDA
//TOOLIN DD *
* Build HSEND BACKDS commands.
  COPY FROM(BCDS) TO(CMDS) USING(CPY1)
/*
//*****
//* The INCLUDE statement selects MCB records that contain
//* the volser of your damaged ML1 volume.
//* The OUTREC statement builds an HSEND BACKDS command
//* for each data set found.
//*****
//CPY1CNTL DD *
  INCLUDE COND=(51,1,BI,EQ,X'20',AND,69,6,CH,EQ,C'?VLSER')
  OUTREC FIELDS=(1,4,C' HSEND BACKDS ',5,44)
/*

```

11. Delete all lost migrated data sets by issuing the HDELETE commands generated in step 8 on page 18.  
This step corrects the MCDS so it no longer lists the lost migration data sets as residing on the ML1 volume, and deletes the catalog entries for the data sets.
12. Recover all lost migrated data sets by issuing the HRECOVER commands generated in step 8 on page 18.  
This step recovers the migrated data sets that were lost on the damaged ML1 from backup copies. The data sets are recovered to a level 0 volume.
13. Delete all backup versions that were lost by issuing the HSEND BDELETE commands generated in step 9 on page 19.  
This step corrects the BCDS so it no longer lists the lost backup version data sets as residing on the ML1 volume.
14. Identify all lost VTOC-copy data sets.  
ISMF can be used to easily identify which VTOC-copy data sets are lost. The data sets with the following message are no longer on the ML1 volume:  
DATA SET NOT FOUND ON VOLUME: *volser*  
  
The backup VTOC-copy data set is named:  
  
*bprefix.VTOC.Tssmmhh.Vvolser.Dyyddd*  
  
The dump VTOC-copy data set is named:  
  
*bprefix.DUMPVTOC.Tssmmhh.Vvolser.Dyyddd*

15. Delete the catalog entries for those VTOC-copy data sets that no longer exist on the ML1 volume.
16. Reinitialize the ML1 volume. This will delete all residual data.
17. Define an SDSP data set, if you want one on the ML1 volume.
  - a. If you had an SDSP data set on the damaged volume, you need to delete the catalog entry for that data set by issuing a DELETE NOSCRATCH command.
  - b. Issue the IDCAMS DEFINE command to define a new SDSP data set for the ML1 volume.
18. Change the ML1 volume so it can be selected for migration output by issuing the ADDVOL command with the NODRAIN parameter. For example:
 

```
ADDVOL volser UNIT(unittype) MIGRATION(ML1 NODRAIN)
```

Be sure to specify other parameters as needed, such as SDSP and AUTODUMP.

## Prevention

There are two ways to prevent the problem from occurring:

- Use the IDCAMS EXPORT TEMPORARY command to regularly export SDSPs with high activity.

**Note:** Be sure to use the TEMPORARY parameter with the EXPORT command so the export function does not delete your original data set.

- Dump ML1 volumes regularly.

---

## Case 9: Reestablish Access to Previously Deleted, Migrated Data Sets (No Backup Exists, ML2 Only)

To reestablish access to previously deleted, migrated data sets, do the following:

1. In one of your journal backup versions, locate the journal entry of the MCD record for the migrated data set. Note the migration volume serial number, which is found at offset X'40' in the MCD record.
2. Create a catalog entry for the data set. Use the IDCAMS DEFINE NONVSAM command with VOLSER(MIGRAT) and UNIT(*tapeunit*).

**Note:** For VSAM data sets, defining NONVSAM entries for the base cluster is sufficient for proper recall.

3. Run the DFSMSHsm AUDIT MEDCTL VOLUMES(*ml2volser*) FIX SERIALIZATION(CONTINUOUS) command to build the needed MCD record for recall.

**Note:** Ensure all HIPER APARs for AUDIT MEDCTL have been applied **before** issuing the AUDIT MEDCTL command.

4. Recall the data set.

## Prevention

There are three things you can do to help prevent the problem from occurring:

- Keep multiple CDS backup versions so you can recover from situations such as this.
- Create backup copies of data sets that are important. Without a backup copy, the migration copy of a data set is the only copy you have.

- Use the DCOLLECT function to track what data exists in active, migrated, or backed up format at the time the DCOLLECT function is run. If you run DCOLLECT daily or a couple of times a week, with output going to a generation data group (GDG), the DCOLLECT information could be a valuable asset in data recovery.

---

## Case 10: Restoring Stacked Output Tapes With a Down-Level MVS Image

The DFSMSShsm ABARS file stacking function, introduced with DFSMS/MVS Version 1 Release 4, allows users to place (stack) the ABARS ABACKUP output files from a single aggregate group on a minimum number of tape cartridges. The output tapes are intended to be recovered by ARECOVER with an MVS image running DFSMS/MVS Version 1 Release 4 or a subsequent release.

If, for some reason, you need to ARECOVER the stacked output tapes with an MVS image running DFSMS/MVS Version 1 Release 3, use the following procedure:

1. Ensure that the ABACKUP output files are cataloged at the recovery site.  
Either bring your catalogs from the ABACKUP site or manually catalog the files. This is required for the ARECOVER processing to work. Although this takes some work, it is much quicker than doing a DFSMSdss COPYDUMP, or a REPRO.
2. Issue an ARECOVER command for the stacked output tapes specifying the DATASETNAME and XMIT parameters.  
The XMIT parameter causes ABARS to get the ABACKUP output file information out of the catalog and pick up the appropriate volser and file sequence number from the catalog. This enables DFSMSShsm running at the DFSMS/MVS Version 1 Release 3 level to recover the stacked output.

### Prevention

The DFSMSShsm default is to STACK the ABACKUP output tapes. If you know that you may need to perform ARECOVER processing with an MVS image running at a release prior to DFSMS/MVS Version 1 Release 4, specify the NOSTACK option when you create your ABACKUP output tapes. This can be specified at the system level with a SETSYS ABARSTAPES(NOSTACK) command, or it can be specified with each ABACKUP using the NOSTACK parameter of the ABACKUP command.

---

## Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10594-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Information Enabling Requests  
Department DZWA  
5600 Cottle Road  
San Jose, CA 95193  
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

---

## Programming Interface Information

This book documents information NOT intended to be used as a Programming Interface of DFSMSHsm.

---

## Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX	DFSMSrmm	NetView
DFSMS/MVS	IBM	RACF
DFSMSdfp	MVS/ESA	SMP/E
DFSMSdss	MVS/SP	System/370
DFSMSHsm	MVS/XA	

---

## Glossary

This glossary defines technical terms and abbreviations used in DFSMSHsm documentation. If you do not find the term you are looking for, refer to the index of the appropriate DFSMSHsm manual or view *IBM Dictionary of Computing*, located at

<http://www.ibm.com/networking/nsg/nsgmain.htm>

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

## Numerics

**3480.** An IBM 3480 Magnetic Tape Subsystem device that is capable of recording data only in the 3480 format.

**3480X.** A 3480 XF device or an IBM 3490 Magnetic Tape Subsystem device that is capable of recording data in either the 3480 format or the 3480 XF (IDRC) format. DFSMSHsm can, at the user's option, select either IDRC or non-IDRC recording.

**3490.** An IBM 3490 Magnetic Tape Subsystem device that records data in the 3480-2 XF format. DFSMSHsm always requests IDRC recording.

**3490E.** See *3490*.

**3590.** An IBM Magstar 3590 tape drive that operates as a 3590-B1x. See *3590-B1x* and *3590-E1x*.

**3590B1x.** An IBM Magstar 3590 Model B1x that uses the 3590 High Performance Cartridge, writes in 128 track format, and can emulate the 3490 Magnetic Tape Subsystem.

**3590E1x.** An IBM Magstar 3590 Model E1x that uses the 3590 High Performance Cartridge, can read 128 or 256 track format tapes, and writes in 256 track format. This drive emulates either the IBM 3490 magnetic tape drive or the IBM Magstar 3590 Model B1X tape drive.

## A

**ABARS.** Aggregate backup and recovery support.

**ABR.** Aggregate backup and recovery record.

**accompany data set.** In aggregate backup and recovery processing, a data set that is physically transported from the backup site to the recovery site instead of being copied to the aggregate data tape. It is cataloged during recovery.

**ACCOMPANY keyword.** The keyword used in the selection data set to create an accompany list.

**accompany list.** An optional list in the selection data set that identifies the accompany data sets.

**ACEE.** Access control environment element.

**ACS.** Automatic class selection.

**active data.** Data that is frequently accessed by users and that resides on level 0 volumes.

**activity log.** In DFSMSHsm, a SYSOUT or DASD-type data set used to record activity and errors that occurred during DFSMSHsm processing.

**AG.** Aggregate group.

**aggregate backup.** The process of copying the data sets and control information of a user-defined group of data sets so that they may be recovered later as an entity by an aggregate recovery process.

**aggregate data sets.** In aggregate backup and recovery processing, data sets that have been defined in an aggregate group as being related.

**aggregate group.** A Storage Management Subsystem class that defines control information and identifies the data sets to be backed up by a specific aggregate backup.

**aggregate recovery.** The process of recovering a user-defined group of data sets that were backed up by aggregate backup.

**AIX.** Alternate index.

**allocate data set.** In aggregate backup and recovery processing, a data set name that is listed in the selection data set. The space for this data set is allocated and the data set is cataloged at the recovery location, but the actual data is not restored.

**ALLOCATE keyword.** The keyword used in the selection data set to create an allocate list.

**allocate list.** An optional list in the selection data set that identifies the allocate data sets.

**alternate index.** In systems with VSAM, a collection of index entries related to a given base cluster and organized by an alternate key, that is, a key other than the prime key of the associated base cluster data records. Its function is to provide an alternate directory for locating records in the data component of a base cluster. See also *path*.

**alternate index cluster.** In VSAM, the data and index components of an alternate index.

**alternate tape volume reference.** In DFSMSHsm, additional fields in the TTOC record that record information about the alternate tape volume. These fields provide DFSMSHsm with the necessary information to refer to the alternate tape volume.

**alternate tape volumes.** In DFSMSHsm, copies of original tape volumes created during tape copy processing. The volumes can either be stored on-site or off-site for use later in the event of a disaster. During the tape replace processing, these volumes can replace the original volumes that may be lost.

**ATL.** Automated tape library.

**audit.** A DFSMSHsm process that detects discrepancies between data set information in the VTOCs, the computing system catalog, the MCDS, BCDS, and OCDS.

**AUTH.** The DFSMSHsm command used to identify an authorized user who can issue DFSMSHsm system programmer and storage administrator commands.

**authorized user.** In DFSMSHsm, the person or persons who are authorized through the DFSMSHsm AUTH command to issue DFSMSHsm system programmer, storage administrator, and operator commands.

**automated tape library.** A device consisting of robotic components, cartridge storage frames, tape subsystems, and controlling hardware and software, together with the set of volumes which reside in the library and may be mounted on the library tape drives.

**automatic backup.** In DFSMSHsm, the process of automatically copying eligible data sets from

DFSMSHsm-managed volumes or migration volumes to backup volumes during a specified backup cycle.

**automatic cartridge loader feature.** A feature of the 3480 Magnetic Tape Subsystem providing the operator with the capability of preloading multiple tapes to be used as migration, backup, or dump tapes.

**automatic class selection (ACS).** A mechanism for assigning SMS classes and storage groups.

**automatic dump.** In DFSMSHsm, the process of using DFSMSdss to automatically do a full volume dump of all allocated space on DFSMSHsm-managed volumes to designated tape dump volumes.

**automatic interval migration.** In DFSMSHsm, automatic migration that occurs periodically when a threshold level of occupancy is reached or exceeded on a DFSMSHsm-managed volume during a specified time interval. Data sets are moved from the volume, largest eligible data set first, until the low threshold of occupancy is reached.

**automatic primary space management.** In DFSMSHsm, the process of automatically deleting expired data sets, deleting temporary data sets, releasing unused overallocated space, and migrating data sets from DFSMSHsm-managed volumes.

**automatic secondary space management.** In DFSMSHsm, the process of automatically deleting expired migrated data sets from the migration volumes, deleting expired records from the migration control data set, and migrating eligible data sets from level 1 volumes to level 2 volumes.

**automatic space management.** In DFSMSHsm, includes automatic volume space management, automatic secondary space management, and automatic recall.

**automatic volume space management.** In DFSMSHsm, includes automatic primary space management and automatic interval migration.

**availability management.** In DFSMSHsm, the process of ensuring that a current version (backup copy) of the installation's data sets resides on tape or DASD.

## B

**backup.** In DFSMSHsm, the process of copying a data set residing on a level 0 volume, a level 1 volume, or a volume not managed by DFSMSHsm to a backup volume. See *automatic backup*, *incremental backup*.

**backup control data set (BCDS).** A VSAM, key-sequenced data set that contains information about backup versions of data sets, backup volumes, dump volumes, and volumes under control of the backup and dump functions of DFSMSHsm.

**backup copy.** In DFSMSHsm, a copy of a data set that is kept for reference in case the original data set is destroyed.

**backup cycle.** In DFSMSHsm, a period of days for which a pattern is used to specify the days in the cycle on which automatic backup is scheduled to take place.

**backup frequency.** In DFSMSHsm, the number of days that must elapse since the last backup version of a data set was made until a changed data set is again eligible for backup.

**backup profile.** In DFSMSHsm, a RACF discrete data set profile associated with the backup version of a cataloged data set that is protected by a RACF discrete data set profile.

**backup version.** Synonym for backup copy.

**backup volume.** A volume managed by DFSMSHsm to which backup versions of data sets are written.

**backup volume cleanup process.** A DFSMSHsm process that scratches data set backup versions that are no longer needed on DASD.

**backup VTOC copy data set.** In DFSMSHsm, a copy of the VTOC of a volume that was backed up by DFSMSHsm. This VTOC data set contains only part of the data set VTOC entry for each data set from the original data set. This data set is written on a migration level 1 volume.

**base cluster.** In systems with VSAM, a key-sequenced or entry-sequenced file over which one or more alternate indexes are built. See also *cluster*.

**base data component.** In VSAM, a component of the base cluster containing data of a data set.

**base sysplex.** A base (or basic) sysplex is the set of one or more MVS systems that is given a cross-system coupling facility (XCF) name and in which the authorized programs can then use XCF coupling services. A base system does not include a coupling facility. See also *parallel sysplex* and *sysplex*.

**BCDS.** Backup control data set.

**BDAM.** Basic direct access method.

**BVR.** Backup cycle volume record.

## C

**catalog.** (1) A directory of files and libraries, with reference to their locations. A catalog may contain other information such as the types of devices in which the files are stored, passwords, blocking factors. (A) (ISO) (2) To enter information about a file or a library into a catalog. (A) (ISO) (3) The collection of all data set indexes that are used by the control program to locate a

volume containing a specific data set. (4) To include the volume identification of a data set in the catalog. (5) See *VSAM master catalog*, *VSAM user catalog*.

**CDD.** Common data set descriptor record.

**CDT.** Class descriptor table.

**CF.** Coupling facility.

**changed data set.** In DFSMSHsm, a data set that has been opened for other than read-only access.

**CLIST.** Command list.

**cluster.** In systems with VSAM, a named structure consisting of a group of related components, for example, a data component with its index component. See also *base cluster*.

**command list.** A command procedure containing executable sequences of TSO commands, subcommands, and command procedure statements.

**command procedure.** In TSO, a data set or a member of a partitioned data set containing TSO commands to be performed sequentially by the EXEC command. See also *command list*.

**common data set descriptor record.** The record that precedes a user's data set on a DFSMSHsm-owned volume and that is used to return the data set to the user's format.

**common filter services.** A subcomponent of DFP common services. Common filter services compares data items with filter keys and indicates which data items match the keys and how many matches have been found.

**common service area (CSA).** In OS/VS2, a part of the common area that contains data areas addressable by all address spaces, but protected during its use by the key of the requester.

**compaction.** In DFSMSHsm, a method of compressing and encoding data that is migrated or backed up.

**comprehensive command authority.** In DFSMSHsm, allowing a user to issue the ABACKUP command for all aggregate groups.

**compress.** In DFSMSHsm, to release unused space in a partitioned data set during the migrate/recall and backup/recovery processes.

**computing system catalog.** In DFSMSHsm, the master catalog and any associated user catalogs used as sources during the audit process.

**concurrent copy.** A function to increase the accessibility of data by enabling you to make a consistent backup or copy of data concurrent with the usual application program processing.

**connected set.** A group of tapes volumes that are related to each other because one or more data sets span the physical tape volumes. Or, a single tape volume with a data set or data sets that do not span to any other volumes.

**contiguous space.** An unbroken consecutive series of storage locations.

**control data set.** In DFSMSHsm, one of three data sets (BCDS, MCDS, and OCDS) that contain records used in DFSMSHsm processing.

**control file.** In aggregate backup and recovery processing, one of three aggregate files generated by the aggregate backup process. It contains the catalog, allocation, volume, and related information necessary to perform aggregate recovery.

**converter/interpreter processing.** The job segment that converts and interprets JCL for MVS.

**coupling facility.** A special logical partition that provides high-speed caching, list processing, and locking functions in a sysplex.

**cross-system coupling facility (XCF).** A component of MVS that provides functions to support cooperation between authorized programs running within a sysplex.

**CSA.** Common service area.

**CTC.** Channel-to-channel.

**current backup version.** In DFSMSHsm, a backup copy of the data set that was created on a date after the data set was last updated.

**cycle start date.** In DFSMSHsm, the date a backup cycle, dump cycle, or migration cleanup cycle is started.

## D

**DAE.** Dump analysis elimination.

**daily backup volume.** In DFSMSHsm, a volume associated with a given day in the backup cycle and assigned to contain backup versions created on that cycle day.

**DASD.** Direct access storage device.

**DASD calculation services (DCS).** A subcomponent of DFP common services. DCS retrieves and calculates data set information for both VSAM and non-VSAM data sets based on the user's input request.

**data class.** A list of allocation attributes that the system uses for the creation of data sets.

**data control block (DCB).** A control block used by access method routines in storing and retrieving data.

**data file.** In aggregate backup and recovery processing, one of three aggregate files generated by the aggregate backup process. It contains the backup copies of the data sets to be recovered.

**data migration.** See *migration*.

**data set change indicator.** A bit in the Format 1 data set VTOC entry that indicates whether the data set has been opened for output.

**data set deletion.** In DFSMSHsm, the space management technique of deleting non-SMS-managed data sets that have not been used for a specified number of days and that do not have expiration date protection.

**data set group.** Data sets that have the same set of initial characters in their names.

**data set organization.** The type of arrangement of data in a data set. Examples are sequential organization or partitioned organization.

**data set pool.** One or more volumes managed by DFSMSHsm to which data sets that have migrated can be recalled, depending on the set of initial characters of the data set name.

**data set retirement.** In DFSMSHsm, the space management technique of deleting non-SMS-managed data sets that have not been referred to for a specified number of days, and that have a backup version. See also *delete-if-backed-up*.

**date last referred to.** In DFSMSHsm, the last date when a data set was opened.

**DBA.** Delete-by-age.

**DBU.** Delete-if-backed-up.

**DCB.** Data control block.

**DCS.** DASD calculation services.

**debug mode.** In DFSMSHsm, the method of operation that projects the changes that would occur in normal operation but in which no user data moves.

**decompaction.** In DFSMSHsm, the process of decoding and expanding data that was compacted during daily space management or backup.

**delete-by-age (DBA).** In DFSMSHsm, the space management technique of deleting non-SMS-managed data sets that have not been opened for a specified number of days.

**delete-if-backed-up (DBU).** In DFSMSHsm, the space management technique of deleting non-SMS-managed data sets that have not been opened for a specified number of days, and that have a current backup version. See also *data set retirement*.

**demotion.** The process of one host losing level functions to another. Both original and promoted hosts can go through the demotion process. See also *promotion*.

**DFP common services.** A component of DFP that contains three subcomponents: common filter services (CFS), DASD calculation services (DCS), and device information services (DIS).

**DFSMSdss.** A functional component of DFSMS/MVS used to copy, dump, move, and restore data sets and volumes.

**DFSMSShsm.** A functional component of DFSMS/MVS used to manage volumes and data sets.

**DFSMSShsm-authorized user.** In DFSMSShsm, the person or persons who are authorized through the DFSMSShsm AUTH command to issue system programmer and storage administrator commands.

**DFSMSShsm log.** In DFSMSShsm, a pair of sequential data sets, X and Y, containing a chronological list of transactions and statistics occurring in DFSMSShsm.

**DFSMSdfp.** A functional component of DFSMS/MVS used to manage programs, devices, and data.

**DFSMSShsm-managed volume.** A volume managed by DFSMSShsm containing data sets that are directly accessible to the user.

**DFSMSShsm-owned volumes.** Storage volumes on which DFSMSShsm stores backup versions, dump copies, or migrated data sets.

**DFSMSShsm secondary address space.** A separate address space started and controlled by DFSMSShsm to perform aggregate backup or aggregate recovery processing.

**DFSORT.** An IBM licensed program that serves as a high-speed data processing utility. DFSORT provides an efficient and flexible way to handle sorting, merging, copying and reporting operations as well as versatile data manipulation at the record, field and bit level.

**direct access storage device (DASD).** A device in which the access time is effectively independent of the location of data.

**directed recall.** Moving a migrated data set from a level 1 or a level 2 volume to a level 0 volume and specifying the target volume and unit name where the data set can be allocated.

**disaster.** An unplanned occurrence that keeps a company or organization from conducting its normal business for some time period.

**disaster backup.** A means to protect a computing system complex against data loss in the event of a disaster.

**disaster recovery.** A means to replace lost data at another location with sufficient resources in order to resume operation.

**discrete backup profile.** A RACF profile created when DFSMSShsm backs up a cataloged, RACF-indicated data set.

**discrete profile.** A RACF profile that contains security information about a specific data set on a particular volume.

**disposition processing.** In OS/VS, a function performed by the initiator at the end of a job step to keep, delete, catalog, or uncatalog data sets, or to pass them to a subsequent job step, depending on the data set status of the disposition specified in the DISP parameter of the DD statement.

**Distributed System License Option (DSLO).** A license option available to IBM customers with a basic license that permits them to copy certain IBM-licensed materials for the purpose of installing multiple systems.

**DSLO.** Distributed Systems License Option.

**DSR.** Daily statistics record.

**dump.** See *full volume dump*.

**dump class.** A DFSMSShsm-named set of characteristics that describe how volume dumps are managed.

**dump copy.** In DFSMSShsm, a copy of the volume image produced by the DFSMSdss full volume dump function.

**dump cycle.** In DFSMSShsm, a period of days for which a pattern is used to specify the days in the cycle on which automatic full volume dump is scheduled to take place.

**dump generation.** A successful full volume dump of a volume that may contain one to five identical dump copies.

**dump VTOC copy data set.** In DFSMSShsm, a copy of the VTOC of a volume dumped by DFSMSShsm. This VTOC data set contains only part of the data set VTOC entry for each data set from the original data set. This data set is written on a migration level 1 volume.

**dynamic keyrange CDS.** A multicliques CDS defined without keyranges for which DFSMSShsm dynamically calculates the key boundaries.

## E

**EA.** Extended addressability.

**eligibility age.** The number of days since a data set met its criteria to be migrated.

**emergency mode.** In DFSMSHsm, the method of operation that prevents data set movement and deletion in space management, backup, and recovery processes.

**encode.** (T) To convert data by the use of a code in such a manner that reconversion to the original form is possible.

**erase-on-scratch.** A RACF and DFP/XA function that overwrites the space occupied by a data set when the data set is scratched from a DASD device supported by MVS/XA.

**ESDS.** Entry-sequenced data set.

**esoteric unit names.** The names a user assigns to I/O devices that have the same device type. When the user specifies the assigned unit name to DFSMSHsm, DFSMSHsm associates the unit name to its device type.

**exclude data set.** In aggregate backup and recovery processing, a data set in the selection data set exclude list. This data set is to be excluded from being processed by aggregate backup.

**EXCLUDE keyword.** The keyword used in the selection data set to create an exclude list.

**exclude list.** An optional list in the selection data set that identifies those data sets that are to be excluded from being processed by aggregate backup.

**expiration.** The removal of a user data set from either a user (non-DFSMSHsm-owned) volume, or from a DFSMSHsm-owned volume when the user data set has been migrated. If there is an explicit expiration date, it is found in the data set VTOC entry for a nonmigrated data set, or in the MCD record for a migrated data set. If there is no explicit expiration date, the management class attributes are checked to determine an implicit expiration date.

**extended addressability.** A type of extended format VSAM data set that allows greater than 4GB of data storage.

**extended format compressed data set.** A SAM or VSAM data set whose processing detects errors caused by an incomplete transfer of a physical block of data to a physical record in the data set. Each physical record is ended by a "hidden" suffix. SAM or VSAM processing determines, by examining these suffixes, if a data transfer to a physical record has ended prematurely.

**extended remote copy (XRC).** A hardware- and software-based remote copy service option that provides an asynchronous volume copy across storage subsystems for disaster recovery, device migration, and workload migration.

**extent reduction.** In DFSMSHsm, the releasing of unused space, reducing the number of extents, and compressing partitioned data sets.

**extents.** A continuous space on a direct access storage volume, occupied by or reserved for a particular data set, data space, or file.

## F

**facility class profile.** In DFSMSHsm, the ability to check, through RACF profiles, the console operators' and users' authority to issue ABACKUP or ARECOVER commands.

**fallback.** Pertaining to returning to use of an earlier release of a program after a later release has been installed and used.

**FBA.** Fixed-block architecture.

**fixed-block architecture.** Data stored in blocks of fixed size; these blocks are addressed by block number relative to the beginning of the particular file.

**FMID.** Function modification identifier.

**fragmentation index.** The qualitative measure of the scattered free space on a volume.

**FSR.** Functional statistics record.

**full volume dump.** In DFSMSHsm, the process of using a DFSMSdss function that backs up the entire allocated space on a volume.

**full volume restore.** In DFSMSHsm, the process of using a DFSMSdss function that restores the entire volume image.

**functional statistics record.** A record that is created each time a DFSMSHsm function is processed. It contains a log of system activity and is written to the system management facilities (SMF) data set.

**functional verification procedure.** A procedure distributed with DFSMSHsm that tests to verify that all basic DFSMSHsm functions are working correctly.

## G

**GDG.** Generation data group.

**GDS.** Generation data set.

**general pool.** In a DFSMSHsm environment with JES3, the collection of all DFSMSHsm primary volumes added to that processor that have a mount status of permanently-resident or reserved, that have the automatic recall attribute specified, and that have a mount attribute of storage or private.

**generation data group.** A collection of data sets with the same base name, such as PAYROLL, that are kept in chronological order. Each data set is called a generation data set.

**generic profile.** A RACF profile that contains security information about multiple data sets, users, or resources that may have similar characteristics and require a similar level of protection.

**global scratch pool.** A group of empty tapes that do not have unique serial numbers and are not known individually to DFSMSHsm. The tapes are not associated with a specific device. Contrast with *specific scratch pools*.

**GRSpIex.** One or more MVS systems using global serialization to serialize access to shared resources.

## H

**hardware/software compression facility.** A function for both SAM and VSAM extended format compressed data sets that has the ability to utilize either hardware data compression for best performance, or software data compression for DASD space considerations.

**high threshold of occupancy.** In DFSMSHsm, the upper limit of space to be occupied on a volume managed by DFSMSHsm. Contrast with *low threshold of occupancy*.

**HMT.** HSM Monitor/Tuner.

**HSMplex.** One or more DFSMSHsm hosts that share a set of control data sets. The hosts may be on single or multiple OS/390 images.

## I

**ICETOOL.** DFSORT's multipurpose data processing and reporting utility.

**IDRC.** Improved data recording capability.

**improved data recording capability.** An improved data recording mode that, depending on data set characteristics and machine model, may allow a significant increase in effective cartridge data capacity and performance.

**inactive age.** In DFSMSHsm, the number of days since the data set was last referred to.

**inactive data.** Copies of active or low-activity data that reside on DFSMSHsm-owned dump and incremental backup volumes. See also *low-activity data*.

**include data set.** In aggregate backup and recovery processing, a data set in the selection data set include list. This data set is processed by aggregate backup.

**INCLUDE keyword.** The keyword used in the selection data set to create an include list.

**include list.** A required list in the selection data set that identifies the include data sets that are to be processed by aggregate backup.

**incremental backup.** In DFSMSHsm, the process of copying a data set that has been opened for other than read-only access since the last backup version was created, and that has met the backup frequency criteria.

**incremental recovery.** Recovery of the latest backup copy of a data set or data sets made by incremental backup.

**inline backup.** The process of copying a specific data set to a migration level 1 volume from a batch environment. This process allows you to back up data sets in the middle of a job.

**installation verification procedure (IVP).** A procedure distributed with DFSMSHsm that tests to verify that the basic facilities of DFSMSHsm are functioning correctly.

**installation exit.** A means specified by an IBM software product by which a customer's system programmers may modify or extend the functions of the IBM software product. Such modifications or extensions consist of exit routines written to replace an existing module of an IBM software product, or to add one or more modules or subroutines to an IBM software product for the purpose of modifying the functions of the IBM software product.

**instruction data set.** In aggregate backup and recovery processing, a data set that contains instructions, commands, or any data the aggregate backup site defines as needed for aggregate recovery at the recovery site.

**instruction file.** In aggregate backup and recovery processing, one of three aggregate files generated by the aggregate backup process. It contains the instruction data set.

**Interactive Storage Management Facility (ISMF).** The interactive panels of DFSMSdfp that allow users and storage administrators access to the storage management functions of DFSMSdss and DFSMSHsm.

**Interactive System Productivity Facility (ISPF).** An IBM licensed program used to develop, test, and run application programs interactively. ISPF is the interactive access method for all storage management functions.

**interval migration.** In DFSMSHsm, automatic migration that occurs periodically when a threshold level of occupancy is reached or exceeded on a DFSMSHsm-managed volume during a specified time

interval. Data sets are moved from the volume, largest eligible data set first, until the low threshold of occupancy is reached.

**ISMF.** Interactive Storage Management Facility.

**ISPF.** Interactive System Productivity Facility.

**ISPF/PDF.** Interactive System Productivity Facility/Program Development Facility.

**IVP.** Installation verification procedure.

## J

**JCL.** Job control language.

**JES2.** Job entry subsystem 2.

**JES3.** Job entry subsystem 3.

**JFCB.** Job file control block.

**journal data set.** In DFSMSHsm, a sequential data set used by DFSMSHsm for recovery of the MCDS, BCDS, and OCDS. The journal contains a duplicate of each record in the control data sets that has changed since the MCDS, BCDS, and OCDS were last backed up.

## K

**KB.** Kilobyte; 1024 bytes.

**key-range data set.** A key-sequenced data set that is defined with one or more KEYRANGE groups.

**KSDS.** Key-sequenced data set.

## L

**LE.** Logical end.

**level functions.** (1) Functions that must be performed but need be performed by only one processing unit. Level functions include backing up the control data sets, backing up migrated data sets, deleting expired dump copies, moving backup versions from level 1 volumes to backup volumes, migration cleanup and level-1-to-level-2 migration. (2) See *primary-processing-unit functions*.

**level 0 volume.** A volume that contains data sets directly accessible by the user. The volume may be either DFSMSHsm-managed or non-DFSMSHsm-managed.

**level 1 volume.** A volume owned by DFSMSHsm containing data sets that migrated from a level 0 volume.

**level 2 volume.** A volume under control of DFSMSHsm containing data sets that migrated from a

level 0 volume, from a level 1 volume, or from a volume not managed by DFSMSHsm.

**like device.** Pertaining to (DASD) devices with identical geometry: that is, the same number of bytes per track, the same number of tracks per cylinder, and the same number of cylinders per actuator.

**linear data set.** In VSAM, a named linear string of data, stored in such a way that it can be retrieved or updated in 4096-byte units.

**logical end (LE).** Signifies that a concurrent copy environment has been successfully initialized. After logical end, the data is again available for unrestricted application access. Actual movement of data begins after logical end. Contrast with *physical end*.

**low-activity data.** Data that is infrequently accessed by users and is eligible to be moved or has been moved to DFSMSHsm-owned migration volumes.

**low threshold of occupancy.** The lower limit of space to be occupied on a volume managed by DFSMSHsm. Contrast with *high threshold of occupancy*.

## M

**managed volume.** See *DFSMSHsm-managed volume* and *primary volume*.

**management class.** A list of data set migration, backup, and retention attributes that DFSMSHsm uses to manage storage at the data set level.

**management work element (MWE).** A control block containing the necessary information to direct DFSMSHsm functions.

**manual tape library.** A set of tape drives defined as a logical unit by the installation together with the set of system-managed volumes which may be mounted on those drives.

**master catalog.** A key-sequenced data set or file with an index containing extensive data set and volume information that VSAM requires to locate data sets or files, to allocate and deallocate storage space, to verify the authorization of a program or operator to gain access to a data set or file, and to accumulate usage statistics for data sets or files.

**MB.** Megabyte; 1 048 576 bytes.

**MCB.** BCDS data set record.

**MCC.** Backup version record.

**MCD.** MCDS data set record.

**MCDS.** Migration control data set.

**MCP.** Eligible volume record.

**MCT.** Backup volume record.

**MCV.** Primary and migration volume record.

**MCVT.** Management communication vector table.

**migration.** In DFSMSShsm, the process of moving a cataloged data set from a DFSMSShsm-managed volume to a migration level 1 or migration level 2 volume, from a migration level 1 volume to a migration level 2 volume, or from a volume not managed by DFSMSShsm to a migration level 1 or migration level 2 volume.

**migration cleanup.** In DFSMSShsm, a process of automatic secondary space management that deletes unnecessary records or migration copies.

**migration control data set (MCDS).** A VSAM, key-sequenced data set that contains statistics records, control records, user records, records for data sets that have migrated, and records for volumes under migration control of DFSMSShsm.

**migration level 1 volume.** Level 1 volume.

**migration level 2 volume.** Level 2 volume.

**migration volume.** A volume, under control of DFSMSShsm, that contains migrated data sets.

**minimal discrete profile.** A profile with no access list or model profile. The minimal discrete profile is used when recovering a RACF-indicated data set whose original profile or backup profile no longer exists.

**minimum migration age.** In DFSMSShsm, the number of days a data set must remain unopened before DFSMSShsm can select it to migrate from a volume.

**MIPS.** Million instructions per second.

**ML1.** Migration level 1. Synonym for level 1 volume.

**ML2.** Migration level 2. Synonym for level 2 volume.

**model entity.** A model data set name that defines a discrete data set profile for RACF protection.

**MTL.** Manual tape library.

**multicenter, key-range control data set.** A control data set (MCDS or BCDS) that is represented by two or more key-sequenced data set (KSDS) clusters. Each cluster is a single key range. Contrast with *single-cluster, key range control data set*.

**multiple-file format.** In DFSMSShsm, a 3480 tape format, or the equivalent, that requires a unique standard label data set for each user data set written. When DFSMSShsm writes in multiple-file format, it writes one tape data set for every user data set to all 3480 migration and backup volumes.

**multiple DFSMSShsm-host environment.** Any environment in which two or more DFSMSShsm hosts share a common set of control data sets.

**mutually exclusive parameters.** A set of parameters of which only one can be used. If more than one parameter is specified, only the last parameter specified is used.

**MVS/Enterprise Systems Architecture (MVS/ESA).** An MVS operating system environment that supports accessing of virtual storage in multiple address spaces and data spaces.

**MVS/Extended Architecture (MVS/XA).** An MVS operating system environment that supports 31-bit real and virtual storage addressing, increasing the size of addressable real and virtual storage from 16 megabytes to 2 gigabytes.

**MVS/SP™.** An IBM licensed program used to control the MVS operating system and to establish a base for an MVS/XA or MVS/370 environment.

**MVT.** Mounted volume table.

**MWE.** Management work element.

## N

**non-DFSMSShsm-managed volume.** A volume not defined to DFSMSShsm containing data sets that are directly accessible to users.

## O

**OCDS.** Offline control data set.

**offline control data set (OCDS).** In DFSMSShsm, a VSAM, key-sequenced data set that contains information about tape backup volumes and tape migration level 2 volumes.

**online.** (1) Pertaining to the operation of a functional unit when under the direct control of a computer. (A) (ISO) (2) Pertaining to a user's ability to interact with a computer. (A) (3) Pertaining to a user's access to a computer via a terminal. (A) (4) Controlled by, or communicating with, a computer.

**original tape volume.** In DFSMSShsm, a 3480 or 3490 single-file tape volume, or the equivalent, used to store data during migration or backup processing, and from which a copy (called the alternate volume) is made for disaster recovery.

**OS/VS2.** A virtual storage operating system that is an extension of OS/MVT.

**owned space.** The storage space on a set of volumes to which DFSMSShsm allocates migrated data sets and backup versions, but to which user jobs should not

allocate. Included in this set are migration level 1, migration level 2, and backup volumes.

**owned volume.** A volume on which DFSMSHsm writes dump, migration, or backup data sets.

## P

**parallel sysplex.** A sysplex with one or more coupling facilities. See also *base sysplex* and *sysplex*.

**partitioned data set (PDS).** A data set in DASD that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

**partitioned data set extended (PDSE).** A DFP library structure that is an enhanced replacement for a partitioned data set.

**path.** (1) (T) In a network, any route between any two nodes. (2) In a data base, a sequence of segment occurrences from the root segment to an individual segment. (3) In VSAM, a named, logical entity providing access to the records of a base cluster either directly or through an alternate index. (4) In an online IMS/VS system, the route a message takes from the time it is originated through processing; in a multisystem environment, the route can include more than one IMS/VS system.

**PCDD.** Pseudo common data set descriptor (CDD) record.

**PDA.** Problem Determination Aid.

**PDF.** Program Development Facility.

**PDS.** Partitioned data set.

**PDSE.** Partitioned data set extended.

**PE.** Physical end.

**peer-to-peer remote copy (PPRC).** A hardware-based remote copy option that provides a synchronous volume copy across storage subsystems for disaster recovery, device migration, and workload migration.

**physical data set restore.** In DFSMSHsm, the process of using a DFSMSdss function to restore one data set from a dump copy created by using the DFSMSHsm full volume dump function.

**physical end (PE).** Signifies that the concurrent copy process has finished copying the data to the output device. Contrast with *logical end*.

**physical sequential.** See *sequential data set*.

**pool of volumes.** See *data set pool*, *general pool*, and *volume pool*.

**PPRC.** Peer-to-peer remote copy.

**primary processing unit.** In a multiple processing-unit-environment, the processing unit assigned to do level functions for backup and dump. Level functions for backup are: backup of CDS, move backup versions, and backup of migrated data sets. Level functions for dump are: delete expired dump copies, and delete excess dump VTOC copy data sets.

**primary-processing-unit functions.** (1) The level functions (backing up migrated data sets, deleting expired dump copies, and moving backup versions from level 1 volumes backup volumes) that must be performed by the primary processing unit. (2) See *level functions*.

**primary volume.** A non-SMS volume managed by DFSMSHsm containing data sets that are directly accessible to the user.

**problem determination aid trace.** This data set is used to gather information about DFHSM processing.

**promotion.** The process of one host taking over level functions for another. See also *demotion*.

**PSCB.** Protected step control block.

**PSP.** Preventive service planning.

**PTF.** Program temporary fix.

## Q

**quiesce time.** A time of day after which an automatic function does not start processing any more volumes.

## R

**RACF.** Resource Access Control Facility.

**recall.** The process of moving a migrated data set from a level 1 or level 2 volume to a DFSMSHsm-managed volume or to a volume not managed by DFSMSHsm.

**reconnection.** In DFSMSHsm, the process by which an unchanged data set recalled from a migration Level 2 tape is space managed by reestablishing access to the migration copy from which it was recalled, rather than by moving the data set and creating a new migration copy.

**record-level sharing (RLS).** An extension to VSAM that provides direct shared access to a VSAM data set from multiple systems using cross-system locking.

**recovery.** In DFSMSHsm, the process of copying a backup version of a data set from a backup volume to a specified volume or to the volume from which the backup version was created.

**recycle process.** A DFSMSHsm process that, based on the percentage of valid data on a tape backup or migration level 2 volume, copies all valid data on the tape to a tape spill backup or migration level 2 volume.

**reentrant.** The attribute of a program or routine that allows the same copy of that program or routine to be used concurrently by two or more tasks.

**Resource Access Control Facility (RACF).** An IBM licensed program that provides access control by identifying and by verifying users to the system. RACF authorizes access to protected resources, logs unauthorized access attempts, and logs accesses to protected data sets.

**restart data set.** A data set created by DFSMSHsm if aggregate recovery fails. It contains a list of all the data sets successfully restored during the aggregate recovery and allows the user to restart the aggregate recovery after the cause of the failure has been resolved.

**restore.** In DFSMSHsm, the process of invoking DFSMSdss to perform the program's recover function. In general, it is to return to an original value or image, for example, to restore data in main storage from auxiliary storage.

**restricted command authority.** In DFSMSHsm, allowing a user having at least read authority to each data set within an aggregate group to process ABACKUP or ARECOVER.

**retired version.** In DFSMSHsm, a specially marked backup version that DFSMSHsm created before it deleted the non-SMS-managed original data set during data set retirement.

**retirement.** See *data set retirement*.

**REXX.** Restructured extended executor.

**RLS.** Record level sharing.

**roll off.** When a new backup, dump, or generation data set copy is created, the oldest copy becomes a candidate for deletion, to maintain a customer-specified limit.

**RRDS.** Relative record data set.

## S

**SAF.** System authorization facility.

**SCP.** System control programming.

**SDSP.** Small data set packing.

**secondary address space.** Synonym for DFSMSHsm secondary address space.

**selection data set.** In aggregate backup and recovery processing, a sequential data set or a member of a partitioned data set used to define the data sets that compose the input to the aggregate backup function. It contains any include, exclude, accompany, or allocate lists.

**sequential data set.** A data set whose records are organized on the basis of their successive physical positions, such as on magnetic tape.

**similar device.** A (DASD) device with the same number of bytes per track and tracks per cylinder.

**single-cluster control data set.** A control data set (MCDS or BCDS) that is represented by a single VSAM key-sequenced data set (KSDS) cluster. This can be a key-range or non-key-range cluster.

**single-file format.** In DFSMSHsm, a 3480 or 3490 format, or the equivalent, consisting of one standard-label data set that spans up to 255 tape volumes.

**single DFSMSHsm-host environment.** Any environment in which a single DFSMSHsm host has exclusive use of a set of control data sets.

**small data set packing (SDSP).** In DFSMSHsm, the process used to migrate data sets that contain equal to or less than a specified amount of actual data. The data sets are written as one or more records into a VSAM data set on a migration level 1 volume.

**small-data-set-packing data set.** In DFSMSHsm, a VSAM key-sequenced data set allocated on a migration level 1 volume and containing small data sets that have migrated.

**SMF.** System Management Facilities.

**SMP.** System Modification Program.

**SMP/E.** System Modification Program Extended.

**SMS.** Storage Management Subsystem.

**SMS class.** A list of attributes that SMS applies to data sets having similar allocation (data class), performance (storage class), or availability (management class) needs.

**SMS-managed data set.** A data set that has been assigned a storage class.

**space manager.** Synonym for storage administrator.

**space management.** In DFSMSHsm, the process of managing aged data sets on DFSMSHsm-managed and migration volumes. The three types of space management are: migration, deletion, and retirement.

**spanning data sets.** Data sets that span one or more tape volumes. The tape volumes that are spanned are part of a connected set. See also *connected set*.

**specific scratch pool.** A group of empty tapes with unique serial numbers that are known to DFSMSHsm as a result of being defined to DFSMSHsm with the ADDVOL command.

**spill backup volume.** A volume owned by DFSMSHsm to which all but the latest backup version of a data set are moved when more space is needed on a DASD daily backup volume or all valid versions are moved when a tape backup volume is recycled.

**spill process.** A DFSMSHsm process that moves all but the latest backup version of a data set from a DASD daily backup volume to a spill backup volume.

**SSI.** Subsystem interface.

**SSM.** Secondary space management.

**SSSA.** Subsystem option block extension for SMS.

**storage administrator.** In DFSMSHsm, the person who is authorized through the DFSMSHsm AUTH command to issue DFSMSHsm system programmer and storage administrator commands, who can affect the authority of other DFSMSHsm users, and who controls the ways DFSMSHsm manages DASD space.

**storage class.** A named list of data set storage service attributes that identifies performance and availability requirements. SMS uses these attributes to control data placement.

**storage group.** (1) A named list of DASD volumes used for allocation of new SMS-managed data sets, or a dummy storage group, or a VIO storage group. (2) A list of real DASD volumes, or a list of serial numbers of volumes that no longer reside on a system but that end users continue to refer to in their JCL.

**storage hierarchy.** An arrangement in which data may be stored in several types of storage devices that have different characteristics such as capacity and speed of access.

**Storage Management Subsystem (SMS).** An operating environment that helps automate and centralize the management of storage. To manage storage, SMS provides the storage administrator with control over data class, storage class, management class, storage group, and ACS routine definitions.

**suballocated file.** A VSAM file that occupies a portion of an already defined data space. The data space may contain other files. Contrast with *unique file*.

**subsystem interface (SSI).** The means by which system routines request services of the master

subsystem, a job entry subsystem, or other subsystems defined to the subsystem interface.

**sysplex.** A set of MVS or OS/390 systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads. This term is derived from *system complex*. See also *base sysplex* and *parallel sysplex*.

**sysplex timer.** An IBM unit that synchronizes the time-of-day (TOD) clocks in multiple processors or processor sides.

**system-managed storage.** An approach to storage management in which the system determines data placement and an automatic data manager handles data backup, movement, space, and security.

## T

**take back.** Process performed by a demoted host to take back the level functions that have been taken over by a promoted host.

**threshold of occupancy.** A limit of occupied space on a volume managed by DFSMSHsm.

**time sharing option (TSO).** An option on the operating system for a System/370 that provides interactive time sharing from remote terminals.

**TIOT.** Task input/output table.

**trace.** (1) A record of the execution of a computer program that exhibits the sequence in which the instructions were executed. (2) To record a series of events as they occur.

**TSO.** Time sharing option.

**TSO/E.** Time sharing option/extended.

**TTOC.** Tape table of contents record.

**TVT.** Tape volume table.

## U

**UID.** The DFSMSHsm authorized-user identification in 1 to 7 characters.

**undirected recall.** In DFSMSHsm, moving a migrated data set from a level 1 or level 2 volume to a level 0 volume without specifying the target volume or unit where the volume can be allocated. Undirected recall can be automatic or by command.

**unique file.** A VSAM file that occupies a data space of its own. The data space is defined at the same time as the file and cannot contain any other file. Contrast with *suballocated file*.

**unlike device.** A DASD device with a different number of bytes per track and tracks per cylinder, or both.

**user catalog.** An optional catalog used in the same way as the master catalog and pointed to by the master catalog. Use of user catalogs lessens the contention for the master catalog and facilitates volume portability.

**user exit.** A programming service provided by an IBM software product that may be requested during the execution of an application program that transfers control back to the application program upon the later occurrence of a user-specified event.

## V

**virtual DASD.** In DFSMSHsm, this refers to the 3850 Mass Storage System (MSS).

**virtual storage access method (VSAM).** An access method for indexed or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

**virtual storage constraint relief (VSCR).** A function that increases the amount of storage available for the user's application program.

**volume.** (1) A certain portion of data, together with its data carrier, that can be handled conveniently as a unit. (ISO) (2) A data carrier that is mounted and demounted as a unit, for example, a reel of magnetic tape, a disk pack. (ISO) (3) That portion of a single unit of storage that is accessible to a single read/write mechanism, for example, a drum, a disk pack, or part of a disk storage module. (4) A storage medium that is mounted and demounted as a unit; for example, magnetic tape or diskette.

**volume pool.** In DFSMSHsm, a set of related primary volumes. When a data set is recalled, if the original volume that it was on is in a defined volume pool, the data set can be recalled to one of the volumes in the pool.

**volume serial number.** An identification number in a volume label that is assigned when a volume is prepared for use in the system.

**volume table of contents (VTOC).** (1) A table on a direct access volume that describes each data set on the volume. (2) An area on a disk or diskette that describes the location, size, and other characteristics of each file and library on the disk or diskette.

**VSAM.** Virtual storage access method.

**VSAM sphere.** A sphere containing the following eight components: base cluster, base data object, base index object, base path, alternate index, alternate index data object, alternate index index object, and alternate index path.

**VSCR.** Virtual storage constraint relief.

**VSF.** Volume statistics record.

**VTOC.** Volume table of contents.

**VTOC copy data set.** A data set that contains a copy of the VTOC entry for each data set that DFSMSHsm backs up or dumps.

**VTG.** Virtual tape server.

**VVDS.** VSAM volume data set.

## W

**WPCDD.** ABACKUP/ARECOVER pseudo CDD parameter list.

**WWFSR.** ABACKUP/ARECOVER backup and recovery record.

## X

**XCF.** Cross-system coupling facility.

**XRC.** Extended remote copy.



---

# Index

## B

backup tape, overwritten 13

## C

case 1: damaged CDS, full journal 3  
case 10: restoring stacked output tapes with a down-level MVS image 24  
case 2: damaged journal, undamaged CDS 6  
case 3: full journal, undamaged CDS 7  
case 4: structurally damaged CDS, missing journal records 8  
case 5: time sequence gaps in CDS records, journal records also missing 9  
case 6: overwritten migration tape 10  
case 7: overwritten backup tape 13  
case 8: head crash on an ML1 volume 17  
case 9: reestablish access to previously deleted, migrated data sets (no backup exists, ml2 only) 23  
CDS damaged, full journal 3  
CDS damaged, missing journal records 8  
CDS records with time sequence gaps, journal records missing 9  
CDS undamaged, journal damaged 6  
CDS undamaged, journal full 7

## D

damaged CDS, full journal 3  
damaged CDS, missing journal records 8  
damaged journal, undamaged CDS 6  
data sets, reestablishing access to deleted migrated 23  
deleted migrated data sets 23  
down-level MVS image, restoring stacked output tapes with 24

## F

full journal, undamaged CDS 7

## H

head crash on an ML1 volume 17

## J

journal damaged, CDS undamaged 6  
journal full, damaged CDS 3  
journal full, undamaged CDS 7  
journal records missing, damaged CDS 8  
journal records missing, time sequence gaps in CDS records 9

## M

migrated data sets that got deleted 23  
migration level one volume, head crash on 17

migration tape, overwritten 10  
missing journal records, damaged CDS, 8  
ML1 volume, head crash on 17

## O

overwritten  
backup tape 13  
migration tape 10

## R

restoring stacked output tapes with a down-level MVS image 24

## S

stacked output tapes, restoring with a down-level MVS image 24

## T

tape  
backup overwritten 13  
migration overwritten 10  
time sequence gaps in CDS records, journal records missing 9

## U

undamaged CDS, damaged journal, 6  
undamaged CDS, full journal, 7

## V

volume, head crash on an ML1 17



---

# Readers' Comments — We'd Like to Hear from You

OS/390  
DFSMSHsm Data Recovery Scenarios  
Release 10

Publication No. GC35-0386-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>				

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>				
Complete	<input type="checkbox"/>				
Easy to find	<input type="checkbox"/>				
Easy to understand	<input type="checkbox"/>				
Well organized	<input type="checkbox"/>				
Applicable to your tasks	<input type="checkbox"/>				

Please tell us how we can improve this book:

Thank you for your responses. May we contact you?  Yes  No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

---

Name

---

Address

---

Company or Organization

---

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



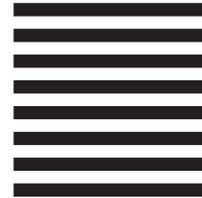
NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation  
Information Development  
Department 61C  
9000 South Rita Road  
TUCSON AZ 85775-4401



Fold and Tape

Please do not staple

Fold and Tape





Program Number: 5647-A01



Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

GC35-0386-00

