

# From Corruption to Restoration: The Power of HSM's Enhanced CDS Recovery

Derek Erdmann  
DFSMSshm Support Team Lead  
Robert Gensler  
DFSMSshm Product Owner  
Feb 25th, 2026

# Agenda

- CDS Overview
- CDS Backup Overview
- Signs of Potential CDS Breakage
- CDS Recovery when a Successful Backup Copy exists
- CDS Recovery when a Successful Backup Copy does not exist



# CDS OVERVIEW

- DFSMSHsm has 3 CDSs (Control Data Sets) and a journal
  - MCDS – Migration Control Data Set (required, up to 4 clusters)
    - 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 DFSMSHsm
  - BCDS – Backup Control Data Set (optional, up to 4 clusters)
    - 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
  - OCDS – Offline Control Data Set (optional, 1 cluster)
    - A VSAM, key-sequenced data set that contains information about tape backup volumes and tape migration level 2 volumes
  - Journal (optional, but highly recommended)
    - A sequential data set used by DFSMSHsm for recovery of the MCDS, BCDS, and OCDS



# CDS BACKUP OVERVIEW

# CDS Backup Overview

- SETSYS CDSVERSIONBACKUP used to configure the CDS Backup environment
  - **DATAMOVER(DSS | HSM)**
    - Whether to use DSS or IDCAMS to backup the CDSs
    - DSS creates **D**xxxxxxx, IDCAMS creates **V**xxxxxxx, **X**xxxxxxx created if backup copy failed
  - **BACKUPDEVICECATEGORY(DASD | TAPE)**
    - Designates what medium backups are stored on
    - When DASD is specified, you must pre-allocate the backup copies to be used (example in ALLOCBK1)
  - **BACKUPDEVICECATEGORY(TAPE(RETENTIONPERIOD(retentionperiod)))**
    - The number of days DFSMSHsm must keep the backup copy
  - **BACKUPCOPIES(x)**
    - Number of backup versions to keep
  - **xxxxBACKUPPSN(dsname)**
    - The initial characters of the data set name when allocating the backup data set for a CDS or journal
  - <https://www.ibm.com/docs/en/zos/3.1.0?topic=opsc-cdsversionbackup-specifying-whether-control-data-sets-should-be-backed-up-multiple-data-sets>
- Recommendations
  - DATAMOVER(DSS) – CDSs will be validated when backed up
  - BACKUPCOPIES(14) – 2 weeks worth of backup copies
  - BACKUPDEVICECATEGORY(TAPE(RETENTIONPERIOD(14))) – prevents overwriting of successful backups with failed backups and retaining backup copies for 2 weeks

# CDS Backup Overview

- Why is BACKUPDEVICECATEGORY(TAPE) recommended?
  - Prevents over-writing of successful backups with failed backups
    - If BACKUPDEVICECATEGORY(DASD) is used:
      - Before CDS Backup
        - DFHSM.MCDS.BACKUP.D0000127
        - DFHSM.MCDS.BACKUP.D0000128
        - DFHSM.MCDS.BACKUP.D0000129
        - DFHSM.MCDS.BACKUP.D0000130
      - After CDS Backup
        - DFHSM.MCDS.BACKUP.X0000131
        - DFHSM.MCDS.BACKUP.D0000128
        - DFHSM.MCDS.BACKUP.D0000129
        - DFHSM.MCDS.BACKUP.D0000130
    - If BACKUPDEVICECATEGORY(TAPE) is used:
      - Before CDS Backup
        - DFHSM.MCDS.BACKUP.D0000127
        - DFHSM.MCDS.BACKUP.D0000128
        - DFHSM.MCDS.BACKUP.D0000129
        - DFHSM.MCDS.BACKUP.D0000130
      - After CDS Backup
        - DFHSM.MCDS.BACKUP.D0000127
        - DFHSM.MCDS.BACKUP.D0000128
        - DFHSM.MCDS.BACKUP.D0000129
        - DFHSM.MCDS.BACKUP.D0000130
        - DFHSM.MCDS.BACKUP.X0000131

Will be uncatalogued



# SIGNS OF POTENTIAL CDS BREAKAGE

# Signs of Potential CDS Breakage

- ARC0744E dsid COULD NOT BE BACKED UP, RC=retcode, REAS=reascode  
MIGRATION, BACKUP, FRBACKUP, DUMP, and RECYCLE HELD
  - Issued during CDS Backup when either an MCDS cluster, BCDS cluster, OCDS cluster or journal was not backed up successfully
  - Research must be done to determine the root cause of the failure
  - 99% (an estimate) of cases requiring CDS Recovery show this error was surfaced
  - Recommend setting up auto-ops to monitor for this message
  - <https://www.ibm.com/docs/en/zos/3.1.0?topic=messages-arc0744e>

# Signs of Potential CDS Breakage

- Other signs
  - ARC0063I DFSMSHsm WILL {HOLD FUNCTIONS SHUTDOWN} DUE TO A VSAM TRAP FINDING A CORRUPT INDEX IN THE {MCDS | BCDS | OCDS}
  - During REORG process, IDCAMS EXAMINE of INDEX/DATA component fails
    - Recommendation – take action based on the messages surfaced
    - If issues surfaced not able to be resolved, use Enhanced CDS Recovery Function




# CDS RECOVERY WHEN A SUCCESSFUL BACKUP COPY EXISTS

# CDS Recovery when a Successful Backup Copy exists

- Utilize the Enhanced CDS Recovery Function
- The example used in the following steps make the following assumptions:
  - Last successful MCDS backup version:  
DFHSM.MCDS.BACKUP.**D**0000130
  - Subsequent MCDS backup failed:  
DFHSM.MCDS.BACKUP.**X**0000131
  - Journal backup version:  
DFHSM.JRNL.BACKUP.**D**0000131
  - Online journal:  
DFHSM.JRNL

# CDS Recovery when a Successful Backup Copy exists

- Step 1) Stop HSM on all hosts utilizing the CDS
- Step 2) Determine the last successful backup copy of the CDS cluster being recovered
  - In this example:
    - The last successful backup is:  
DFHSM.MCDS.BACKUP.**D**0000130
    - The subsequent backup is:  
DFHSM.MCDS.BACKUP.**X**0000131  
  
Failed copy

## CDS Recovery when a Successful Backup Copy exists

- Step 3) Verify the journal backup(s) created since that CDS backup version are also successful (last qualifier begins with 'D' or 'V')
  - In this example:  
DFHSM.JRNL.BACKUP.**D**0000131
  - If a journal backup had failed:  
DFHSM.JRNL.BACKUP.**X**0000131
    - If failed, see section 'CDS Recovery when no Successful Backup Copy exists'

# CDS Recovery when a Successful Backup Copy exists

- Step 4) Run ARCBJRNL to create the merged journal to be used in this recovery process

```
000006 //STEP4      EXEC PGM=ARCBJRNL,PARM=' BOTH'
000007 //STEPLIB    DD  DSNAME=SYS1.LINKLIB,
000008 //              DISP=SHR
000009 //SYSPRINT    DD  SYSOUT=*
000010 //JRNLO       DD  DSNAME=DFHSM.JRNL,
000011 //              DISP=SHR
000012 //BKUPJRNL    DD  DSNAME=DFHSM.JRNL.BACKUP.D0000131,DISP=SHR
000013 //JRNLBK      DD  DSNAME=DFHSM.JOURNAL.BACKUP.AUG7,
000014 //              UNIT=3390,
000015 //              VOL=SER=WRK172,
000016 //              SPACE=(CYL,(10,10)),
000017 //              DISP=(,CATLG,DELETE)
000018 /*
```

Both the active journal and journal backup data sets used as input

The online journal name

The journal backup version(s) in ascending order, oldest to newest

The merged journal to be created as output

# CDS Recovery when a Successful Backup Copy exists

- Step 5) Sort the merged journal created by ARCBJRNL

```
000005 //STEP5      EXEC PGM=SORT
000006 //SYSOUT     DD SYSOUT=*
000007 //SORTIN      DD DSN=DFHSM.JOURNAL.BACKUP.AUG7,
000008 //              DISP=SHR
000009 //SORTOUT      DD DSN=DFHSM.JOURNAL.BACKUP.AUG7.SORTED,
000010 //              UNIT=3390,
000011 //              VOL=SER=WRK172,
000012 //              SPACE=(CYL,(10,10)),
000013 //              DCB=(RECFM=VB,BLKSIZE=0,LRECL=6560),
000014 //              DISP=(,CATLG,DELETE)
000015 //SYSIN         DD *
000016     SORT        FIELDS=(17,44,CH,A)
000017                OPTION EQUALS
000018 /*
```

The merged journal created in Step 4

The sorted journal created as output

Records sorted in ascending order by key, retaining their input sequence

## CDS Recovery when a Successful Backup Copy exists

- Step 6a) Rename the broken CDS cluster
  - This is done to ensure analysis can be done against the broken CDS later

```
000002 //STEP6A EXEC PGM=IDCAMS
000003 //SYSPRINT DD SYSOUT=*
000004 ALTER DFHSM.MCDS -
000005     NEWNAME(DFHSM.MCDS.OLD)
000006 ALTER DFHSM.MCDS.DATA -
000007     NEWNAME(DFHSM.MCDS.OLD.DATA)
000008 ALTER DFHSM.MCDS.INDEX -
000009     NEWNAME(DFHSM.MCDS.OLD.INDEX)
000010 /*
```

# CDS Recovery when a Successful Backup Copy exists

- Step 6b) Create a new CDS cluster

```
000002 //STEP6B EXEC PGM=IDCAMS
000003 //HSMCDS DD UNIT=3390,VOL=SER=WRK172,DISP=SHR
000004 //SYSPRINT DD SYSOUT=*
000005 //SYSIN DD *
000006 DEFINE CLUSTER ( -
000007     NAME(DFHSM.MCDS) -
000008     VOLUMES(WRK172) -
000009     CYLINDERS(5) FILE(HSMCDS) -
000010     RECORDSIZE(435 2040) FREESPACE(0 0) -
000011     INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
000012     SPEED BUFFERSPACE(530432) -
000013     UNIQUE NOWRITECHECK) -
000014 DATA ( -
000015     NAME(DFHSM.MCDS.DATA) -
000016     CONTROLINTERVALSIZE(12288)) -
000017 INDEX ( -
000018     NAME(DFHSM.MCDS.INDEX) -
000019     CONTROLINTERVALSIZE(2048))
000020 /*
```

CDS being created

## CDS Recovery when a Successful Backup Copy exists

- Step 7) Create exported copy of CDS backup version being recovered from
  - **Only needed if DSS was used to create the CDS backup copy**
  - Step 7a – Use DSS to restore the last successful CDS backup version
  - Step 7b – Use IDCAMS to create an exported copy

# CDS Recovery when a Successful Backup Copy exists

- Step 7a) Use DSS to restore the last successful CDS backup version

```
000006 //STEP7A EXEC   PGM=ADRDSSU
000007 //SYSPRINT DD  SYSOUT=*
000008 //INPUT      DD  DSN=DFHSM.MCDS.BACKUP.D0000130,DISP=OLD
000009 //OUTPUT     DD  UNIT=3390,DISP=OLD,VOL=SER=WRK172
000010 //SYSIN       DD  *
000011          RESTORE DATASET(-
000012          INCLUDE(DFHSM.MCDS)) -
000013          INDDNAME(INPUT) -
000014          OUTDDNAME(OUTPUT) -
000015          REPLACE
000016 /*
```

The last successful CDS backup version

The CDS being restored

## CDS Recovery when a Good Backup Copy exists

- Step 7b) Use IDCAMS to create an exported copy

```
000003 //STEP7B EXEC PGM=IDCAMS
000004 //SYSPRINT DD SYSOUT=*
000005 //OUTDD1 DD DISP=(,CATLG,DELETE),BUFNO=26,BLKSIZE=28332,
000006 //          SPACE=(CYL,(10,10),RLSE),
000007 //          UNIT=3390,VOL=SER=WRK172,DSN=DFHSM.MCDS.AUG7.EXPORTED
000008 //SYSIN DD *
000009          EXPORT DFHSM.MCDS -
000010          OUTFILE(OUTDD1) -
000011          TEMPORARY
000012 /*
```

The exported data set created as output

The CDS that was restored in Step 7a

# CDS Recovery when a Successful Backup Copy exists

- Step 8) Use ARCIMPRT to import and merge the exported CDS copy with the sorted journal records

```
000006 //STEP8      EXEC PGM=ARCIMPRT,PARM='MCDS'
000007 //STEPLIB    DD  DSNAME=SYS2.LINKLIB,
000008 //              DISP=SHR
000009 //SYSPRINT     DD  SYSOUT=*
000010 //AMSDUMP      DD  SYSOUT=*
000011 //JOURNAL     DD  DSNAME=DFHSM.JOURNAL.BACKUP.AUG7.SORTED,
000012 //              DISP=SHR
000013 //EXPORT       DD  DSNAME=DFHSM.MCDS.AUG7.EXPORTED,
000014 //              DISP=SHR
000015 //SYSIN        DD  *
000016     DELETE (DFHSM.MCDS) CLUSTER PURGE
000017     IMPORT INFILE(EXPORT) OUTDATASET(DFHSM.MCDS)
000018 /*
```

The CDS being recovered

The sorted journal output from Step 5

The exported version of the CDS (either created as output in Step 7B or during CDS backup)

## CDS Recovery when a Successful Backup Copy exists

- Step 9) Start HSM
  - Verify startup completes successfully
- Step 10) Run `AUDIT DATASETCONTROLS(MCDS|BCDS) ODS(xx)`
- Step 11) If MCDS was recovered, issue `QUERY CDSV` and compare to LISTCAT output to ensure last final qualifier versions match. If they do not, the last final qualifier can be updated using the following command:

```
FIXCDS S MHCR PATCH(X'B1' 0000131) <-would set to V0000131
```

- Step 12) Issue `BACKVOL CDS` and verify CDS Backup completes successfully



# CDS RECOVERY WHEN A SUCCESSFUL BACKUP COPY DOES NOT EXIST

## CDS Recovery when a Successful Backup Copy does not exist

- What does this mean?
  - All backup versions of the “broken” cluster show a last qualifier of **X**nnnnnnnn
- How can this happen?
  - Upon receiving the ARC0744E, an administrator issues BACKVOL CDS repeatedly to see if any attempt creates a successful backup
  - Upon receiving the ARC0744E, an administrator issues RELEASE BACKUP

**DO NOT DO THIS, ESPECIALLY IF CDS BACKUP DIRECTED TO DASD!**

- Instead, investigate the root cause of the ARC0744E
  - Only attempt a BACKVOL CDS if the error surfaced was temporary and the condition that caused it to surface was resolved
  - This is just one example

## CDS Recovery when a Successful Backup Copy does not exist

- What action should be taken?
  - Utilize records in “broken” cluster to build a new “good” cluster
- The example used in the following steps make the following assumptions:
  - The broken cluster is: DFHSM.MCDS

## CDS Recovery when a Successful Backup Copy does not exist

- Step 1) Stop HSM on all hosts utilizing the CDS
- Step 2) Rename the broken CDS cluster
  - This is done to ensure analysis can be done against the broken CDS cluster later

```
000002 //STEP2      EXEC PGM=IDCAMS
000003 //SYSPRINT DD  SYSOUT=*
000004  ALTER DFHSM.MCDS -
000005          NEWNAME(DFHSM.MCDS.OLD)
000006  ALTER DFHSM.MCDS.DATA -
000007          NEWNAME(DFHSM.MCDS.OLD.DATA)
000008  ALTER DFHSM.MCDS.INDEX -
000009          NEWNAME(DFHSM.MCDS.OLD.INDEX)
000010 /*
```

## CDS Recovery when a Successful Backup Copy does not exist

- Step 3) Use IDCAMS to copy the records from the DATA component to a sequential data set

```
000002 //STEP3      EXEC PGM=IDCAMS
000003 //SYSPRINT DD  SYSOUT=*
000004 //DDIN1       DD  DISP=SHR,DSN=DFHSM.MCDS.OLD.DATA
000005 //DDOUT1      DD  DISP=(NEW,CATLG),UNIT=3390,
000006 //              DSN=DFHSM.MCDS.RECORDS.AUG7,
000007 //              SPACE=(CYL,(50,10)),VOL=SER=WRK172,
000008 //              DCB=(LRECL=2044,BLKSIZE=0,RECFM=VB)
000009 //SYSIN        DD  *
000010              REPRO INFILE(DDIN1) OUTFILE(DDOUT1) FADDR(0)
000011 /*
```

Data component of the renamed CDS

Sequential output data set

- Step 3) Use IDCAMS to copy the records from the DATA component to a sequential data set
  - Depending on the breakage of the CDS, the REPRO may fail when encountering a record it is unable to process
    - If this occurs, make note of the key that failed, increment the key and issue another REPRO using this new key in the FROMKEY(key) parameter
    - Repeat as needed until the REPRO completes with an RC0

## CDS Recovery when a Successful Backup Copy does not exist

- Step 3) Use IDCAMS to copy the records from the DATA component to a sequential data set

```
000002 //STEP3      EXEC PGM=IDCAMS
000003 //SYSPRINT DD  SYSOUT=*
000004 //DDIN1       DD  DISP=SHR,DSN=DFHSM.MCDS.OLD.DATA
000005 //DDOUT1      DD  DISP=MOD,UNIT=3390,
000006 //              DSN=DFHSM.MCDS.RECORDS.AUG7,
000007 //              VOL=SER=WRK172
000008 //SYSIN       DD  *
000009             REPRO INFILE(DDIN1) OUTFILE(DDOUT1) FROMKEY(DERDMANN.TESTFILE)
000010 /*
```

Data component of the renamed CDS

Sequential output data set

Key to start copying from

## CDS Recovery when a Successful Backup Copy does not exist

- Step 4) Create a new CDS cluster

```
000002 //STEP4 EXEC PGM=IDCAMS
000003 //HSMCDS DD UNIT=3390,VOL=SER=WRK172,DISP=SHR
000004 //SYSPRINT DD SYSOUT=*
000005 //SYSIN DD *
000006 DEFINE CLUSTER ( -
000007     NAME(DFHSM.MCDS) -
000008     VOLUMES(WRK172) -
000009     CYLINDERS(5) FILE(HSMCDS) -
000010     RECORDSIZE(435 2040) FREESPACE(0 0) -
000011     INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
000012     SPEED BUFFERSPACE(530432) -
000013     UNIQUE NOWRITECHECK) -
000014 DATA ( -
000015     NAME(DFHSM.MCDS.DATA) -
000016     CONTROLINTERVALSIZE(12288)) -
000017 INDEX ( -
000018     NAME(DFHSM.MCDS.INDEX) -
000019     CONTROLINTERVALSIZE(2048))
000020 /*
```

CDS being created

## CDS Recovery when a Successful Backup Copy does not exist

- Step 5) Sort the records in the sequential data set and store them in the new CDS cluster defined in Step 4

```
000005 //STEP5A EXEC PGM=SORT,REGION=4096K
000006 //SYSOUT DD SYSOUT=*
000007 //SORTIN DD DISP=SHR,DSN=DFHSM.MCDS.RECORDS.AUG7
000008 //SORTOUT DD DISP=(,PASS),DSN=&&TFILE1,
000009 //          UNIT=SYSDA,SPACE=(CYL,(50,10),RLSE),
000010 //          DCB=(LRECL=2044,BLKSIZE=0,RECFM=VB,DSORG=PS)
000011 //SYSIN DD *
000012     SORT FIELDS=(5,44,CH,A,53,8,BI,D)
000013 /*
000014 //STEP5B EXEC PGM=SORT,REGION=4096K,COND=(0,NE)
000015 //SYSOUT DD SYSOUT=*
000016 //SORTIN DD DISP=SHR,DSN=&&TFILE1
000017 //SORTOUT DD DISP=OLD,DSN=DFHSM.MCDS
000018 //SYSIN     DD *
000019     SORT FIELDS=(5,44,CH,A),EQUALS
000020     SUM FIELDS=(NONE)
000021 /*
```

Output data set from Step 3

Ascending order by key,  
descending by update time

CDS created in Step 4

Removes duplicates

## CDS Recovery when a Successful Backup Copy does not exist

- Step 6) – Start HSM
  - Verify startup completes successfully
- Step 7) Run `AUDIT DATASETCONTROLS(MCDS|BCDS) ODS(xx)`
- Step 8) If MCDS was recovered, issue `QUERY CDSV` and compare to LISTCAT output to ensure last final qualifier versions match. If they do not, the last final qualifier can be updated using the following command:

```
FIXCDS S MHCR PATCH(X'B1' 0000131) <-would set to V0000131
```

- Step 9) Issue `BACKVOL CDS` and verify CDS Backup completes successfully

# CDS Recovery when a Successful Backup Copy does not exist

- Takeaways
  - **Practice these steps before a CDS breakage occurs**
  - Ideally, have the JCL ready to go with only parameters needing to be modified
  - Keep a copy of the broken CDS for analysis by DFSMSHsm Support
  - <https://www.ibm.com/docs/en/zos/3.1.0?topic=scenarios-case-1-damaged-cds-full-journal>

# Experience more with IBM



## Visit us at the IBM Booth #113

After a full day of technical sessions, take a break with us!

Connect with our experts, snap a photo with the z17 Plexi or the latest Telum II, and get an up-close look at our Spyre Accelerator.

Come back each day for fresh topics and demos at our expert stations.

## Think 2026

Join 5000+ senior business and technology leaders who are seizing the AI revolution to unlock unprecedented growth and productivity at **Think 2026**.

Find out more information using the QR code below.



## IBM Digital Asset Haven

IBM Digital Asset Haven is the operational backbone for financial institutions and regulated enterprises entering the digital asset economy.

Find out more information using the QR code below.





**QUESTIONS?**