DFSMShsm CDS Deep Dive

Glenn Wilcock
IBM

March 3, 2015
Session 17101

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

SHARE in Seattle 2015

Agenda

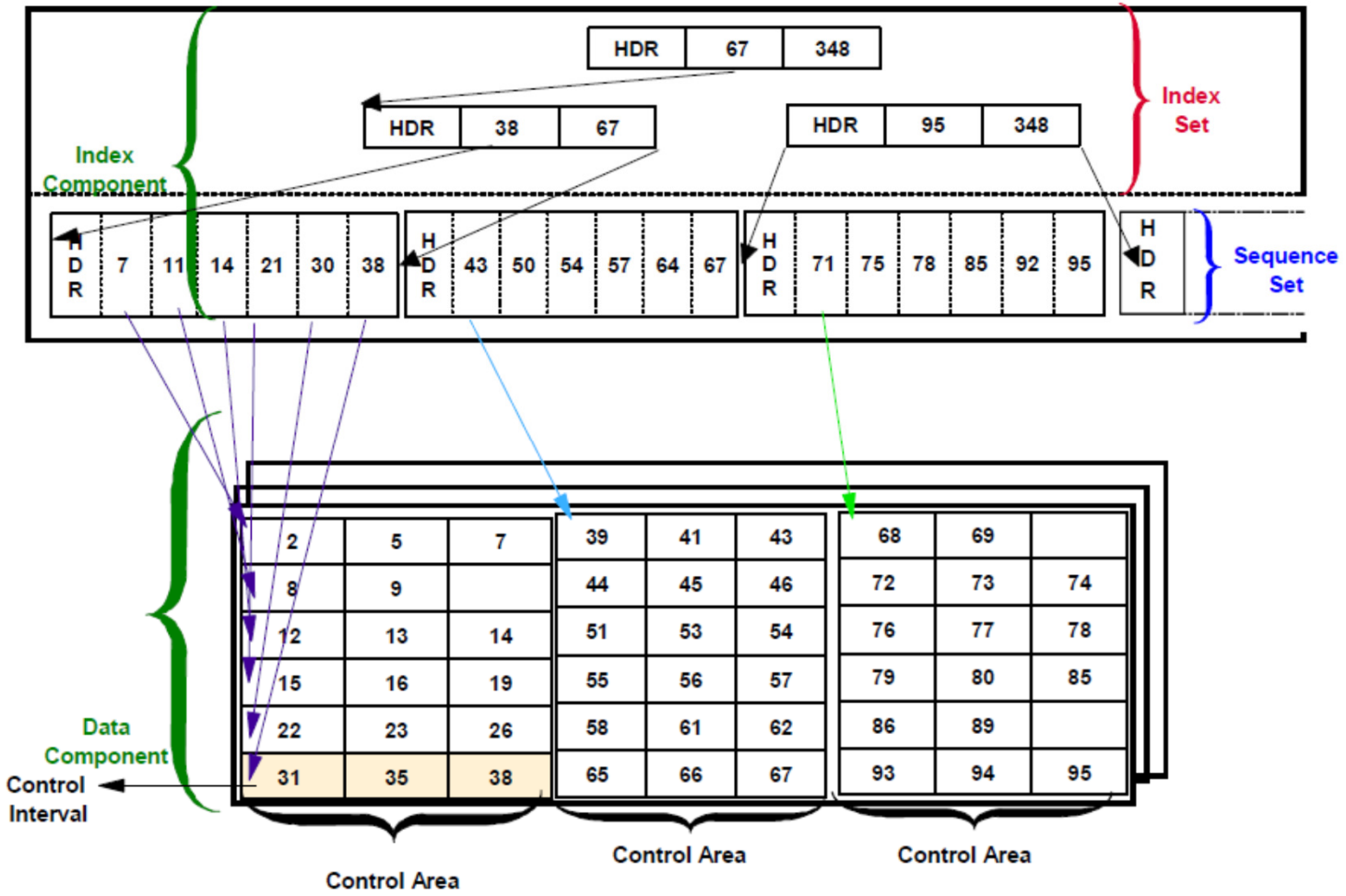
- Overview
- CDS Structure
- Accessing
- Reorganizing
- Backup & Recovery
- Auditing
- Repairing & Updating

Overview

- DFSMShsm maintains an inventory of the data sets and volumes that it manages within three Control Data Sets
  - Migration control data set (MCDS)
    - Migration Environment, Statistics records, Volume records, Function control records
    - *Required* for DFSMShsm to function
  - Backup control data set (BCDS)
    - ABARS, Backup and Dump environment
    - Optional
  - Offline control data set (OCDS)
    - Tape volume information
  - Journal
    - A log of every critical update and delete to each of the CDSes

Overview

- **CDSes are VSAM Key Sequenced**
  - Each record is prefixed with a *unique key*
  - VSAM maintains an index for direct access to records
  - HSM uses both direct access and sequential access, dependent on the function being performed



Overview

- CDSes can be defined as **Extended Addressable**, which enables a data set to be greater than 4GB

- Journal can be defined as **Large Format** sequential, which enables it to be greater than 64K tracks
  - Must be a contiguous, single extent data set

Overview
CDS Record Format

Key 44 Bytes	**Byte 1:** Type of record, except for data set names **Bytes 2 – 44:** Unique identifier
Header 20 Bytes	**Bytes 45 – 46:** Overall Record Length **Byte 47:** Record type, when Key is a data set name, this uniquely identifies the Record type **Byte 48:** Not Used **Bytes 49-56:** Last Updated **Bytes 57 – 64:** Creation Date
Data 6480 Max	Data portion of the record, mapped by the record type

DSR Key

All statistics records are a type x'10'

DSR Key

Bytes 2-4 are 'DSR'

DSR Key

Followed by date, to make it unique

Table 31 (Page 1 of 3). DSR—Daily Statistics Record

Offsets Actual	/ FIXCDS	Type	Length	Name	Description
0(0)		STRUCTURE	1016	DSR	Daily statistics record.
0(0)		CHARACTER	44	DSRKEY	Daily statistics record key, consisting of X'10'||DSR||*yyddds*.
0(0)		CHARACTER	1	*	Reserved.
1(1)		CHARACTER	6	DSRKEY2	The next 6 bytes contain the rest of the key.
1(1)		CHARACTER	3	*	DSR.
4(4)		CHARACTER	3	DSRDATE	Date in format *X'yyddds'*, in packed decimal.
44(2C)		CHARACTER	20	DSRHDR	Migration control data set record header. (See MCK for details.)
64(40)	0(0)	CHARACTER	118	DSRDATA	Daily statistics record information.
64(40)	0(0)	FIXED	4	DSRIPL	Number of DFSMShsm startups that day.

DSR Data

DSR Header

Overview

- **DFSMShsm Data Areas manual**

  http://www-03.ibm.com/systems/z/os/zos/library/bkserv/v2r1pdf/#IDA

  - Previously required a license
  - Documents the structure of all of the DFSMShsm control data sets and also all of the major control blocks

MCB—Backup Control Data Set Data Set Record

Overview

The backup control data set data set record (MCB) describes a data set that has been backed up and that identifies backup versions. This data area record type is a backup control data set data set record. Part of the base portion (offset 64-140) describes the latest version made, such as dates, flags, counts, and size. This can be confusing and meaningless if several data sets have the same name. Backup control data set records are 144 bytes long, plus 64 bytes for each backup version description. The record type is B.

Fundamental structure of the **FIXCDS** command

The key for a type B backup control data set data set record is the original data set name. An example of the key that is used with a B backup control data set data set record is:

```
FIXCDS B USER.DATA.NAME
```

Table 78 (Page 1 of 2). MCB—Backup Control Data Set Data Set Record

Offsets Actual	/ FIXCDS	Type	Length	Name	Description
0(0)			44	MCK	MCDS data set record key, consisting of the name and padded with blanks. (See MCK for details.)
44(2C)			20	MCH	Control data set record header. (See MCK for details.)
64(40)	0(0)	STRUCTURE	80	MCB	Data portion of the BCDS data set record.
64(40)	0(0)	CHARACTER	6	MCBVSN	MCBVSN contains the volume serial number of the volume containing the most recently created backup version at the time the MCB record was created.

Offset

Length
Structures will have length of entire structure

Description

Table 78 (Page 1 of 2). MCB—Backup Control D

Offsets Actual	/ FIXCDS	Type	Length	Name	Description
0(0)			44	MCK	MCDS data set record key, consisting of the name and padded with blanks. (See MCK for details.)
44(2C)			20	MCH	Control data set record header. (See MCK for details.)
64(40)	0(0)	STRUCTURE	80	MCB	Data portion of the BCDS data set record.
64(40)	0(0)	CHARACTER	6	MCBVSN	MCBVSN contains the volume serial number of the volume containing the most recently created backup version at the time the MCB record was created.
70(46)	6(6)	FIXED	2	MCBMBC	Maximum number of backup copies. –1 means to use the system default.
72(48)	8(8)	CHARACTER	8	MCBTSLU	Time stamp when dataset was last updated
72(48)	8(8)	CHARACTER	4	MCBTLU	Time in packed decimal
76(4C)	12(C)	CHARACTER	4	MCBDLU	Date in packed decimal (PRIORONLY in version to 2.4.0). Also used by HSM utility for its scratch date, w/x'FF' in first byte, and entire time stamp used in vandion for both VSAM and non-VSAM
80(50)	16(10)	CHARACTER	8		Time stamp backup copy made.
80(50)	16(10)	CHARACTER	4		Time when the latest backup version was made. The time is obtained from the TIME macro in hundredths of seconds.
84(54)	20(14)	CHARACTER	4		Date when the latest backup version was made. The date is obtained from the TIME DEC macro in format *X'0cyyddds'*.
88(58)	24(18)	BITSTRING	2	MCBDSORG	Data set organization from the data set control block.
90(5A)	26(1A)	FIXED	2	MCBBLKSZ	Maximum block size of the data set.
92(5C)	28(1C)	FIXED	1	MCBKEYLN	Key length of the data set.
93(5D)	29(1D)	BITSTRING	1	MCBRECFM	Data set record format from the VTOC entry:
		11..		MCBRFTYP	These flags indicate a V, B, or F format.
		..1.		MCBRFTO	When set to 1, the track overflow feature is present.
		...x xxxx	1	*	Reserved.

Type: Address, Bitstring, Fixed, Structure,Character, Signed, Unsigned

CDS Structure

- Refer to Michael Friske's SHARE presentation from Atlanta 2012

10915 Taking a look inside the HSM Control Data Sets

BCDS MCC Record – Backup Version Record

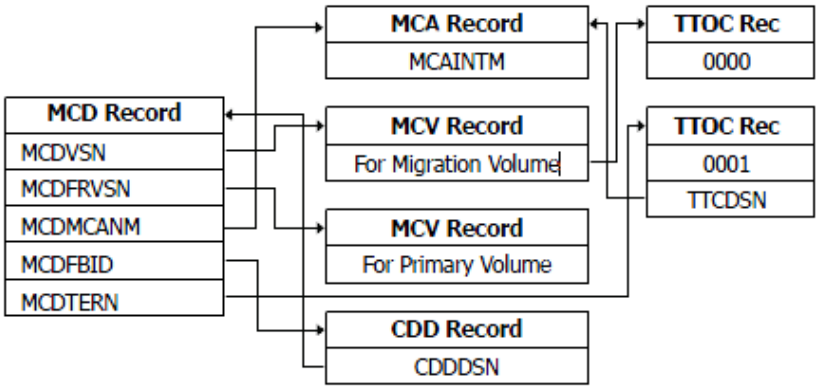
Field Name	Field Length	Field Description	Points to	Usage	Note
MCC	44	Backup copy name		Key of MCC record	1
MCCADSN	44	User data set name	MCB	Key of MCB record	
MCCVSN	6	Volser of backup volume	MCT	Used to build the MCT record	1
MCCGEN	2	Backup version number		Absolute version number	
MCCFLGS	2	Data set type and other info			2
MCCFRVOL	6	User volume	MCV	Used to build the MCV key	1
MCCFRUCB	4	Device type of user volume		Determine the track size	1
MCCTERN	4	TTOC extension record	TTOC	Relative TTOC ext.	3
MCCFBID	4	File Block ID	CDD	Relative block on tape	4

1. This field is required to recover a backup copy.
2. This field is used to recover the correct data set in the event there is an uncataloged data set.
3. This field is required for recovery of a backup copy from tape.
4. This is a performance enhancement.

13

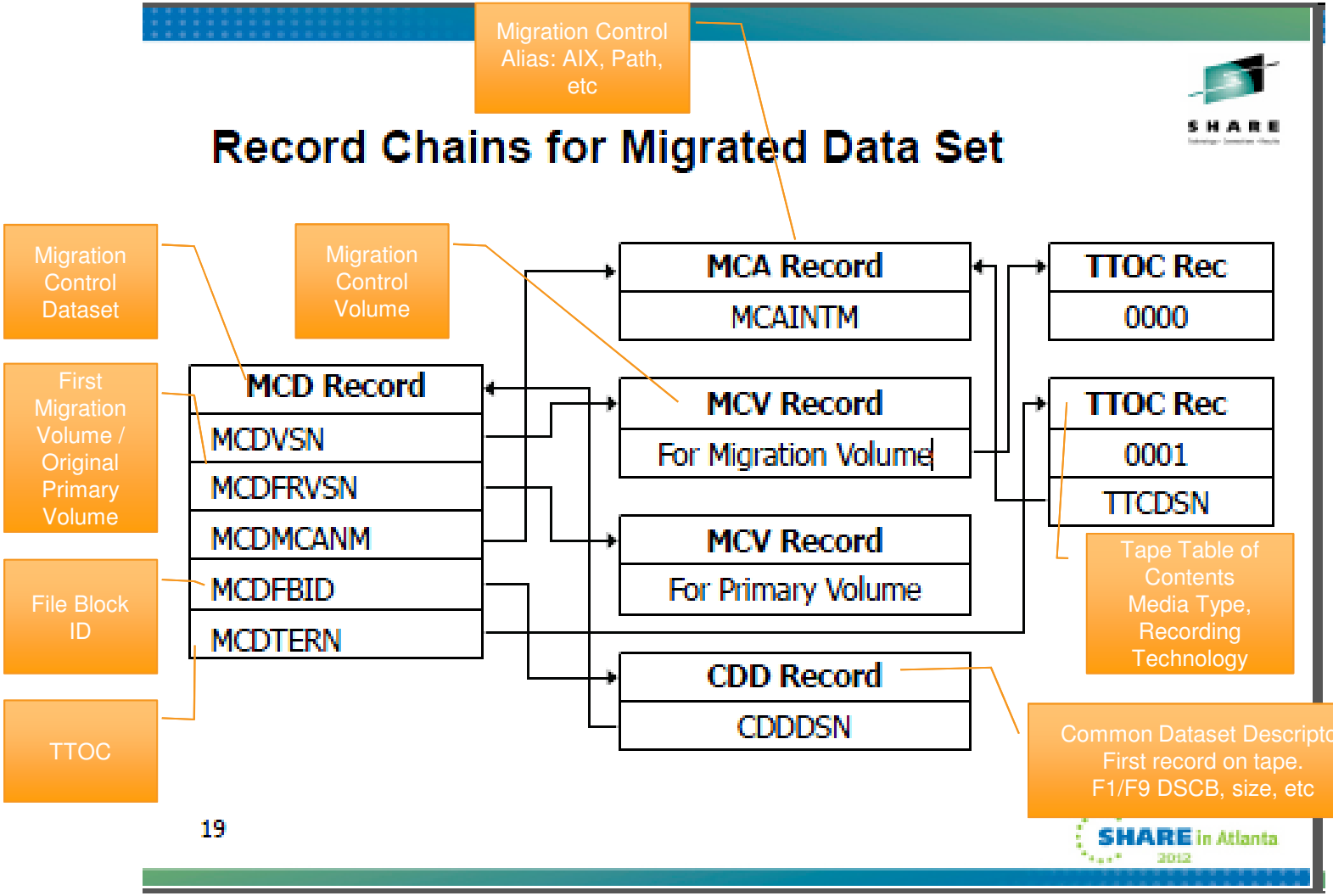
SHARE in Atlanta 2012

Record Chains for Migrated Data Set



19

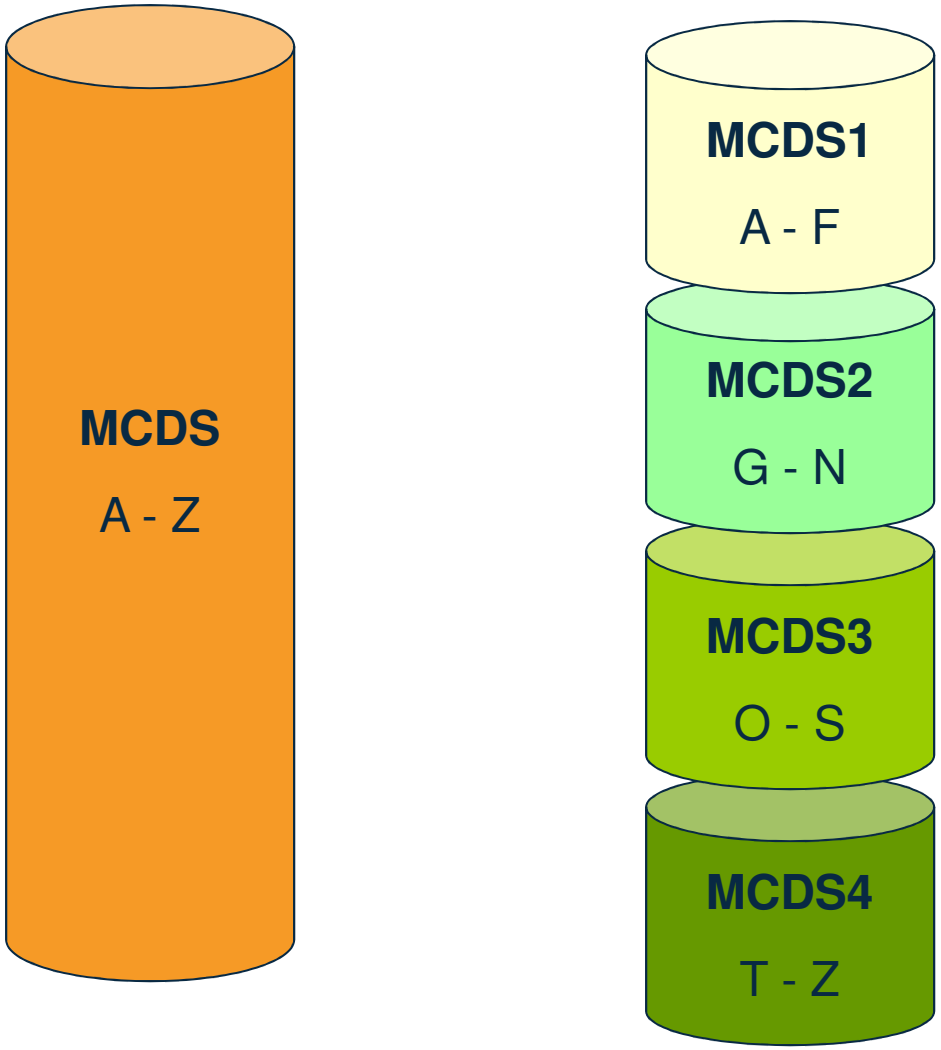
SHARE in Atlanta 2012


Migration Control Alias: AIX, Path, etc
Record Chains for Migrated Data Set
Migration Control Dataset
Migration Control Volume
First Migration Volume / Original Primary Volume
File Block ID
TTOC
MCD Record
MCDVSN
MCDFRVSN
MCDMCANM
MCDFBID
MCDTERN
MCA Record
MCAINTM
MCV Record
For Migration Volume
MCV Record
For Primary Volume
CDD Record
CDDDSN
TTOC Rec
0000
TTOC Rec
0001
TTCDSN
Tape Table of Contents Media Type, Recording Technology
Common Dataset Descriptor First record on tape. F1/F9 DSCB, size, etc
19
SHARE in Atlanta 2012

Multiple Cluster CDSs

- The MCDS and BCDS can be comprised from one to four clusters
- The OCDS can only be one cluster
- Each cluster is a stand-alone VSAM Key Sequence data set
  - Key ranges are no longer supported
- Multiple Clusters enables
  - CDS Backup process to be done at a higher level of parallelism, as opposed to backing up a single larger data set
  - If Recovery is required, only the cluster that is in error needs to be recovered, no all of them

Multiple Cluster CDSs



Multiple Cluster CDSs

- Determining key ranges
  - HSM.SAMPLE.TOOL(SPLITCDS) is a tool that analyzes the current CDS
  - Produces a report for splitting into two, three and four clusters

```
BCDS:
     FROMKEY(X'00')          TOKEY(HSM.BACK.T2)
     FROMKEY(HSM.BACK.T3)    TOKEY(X'FF')

MCDS:
     FROMKEY( X'00')         TOKEY(HSM.HMIG.T4)
     FROMKEY(HSM.HMIG.T5)    TOKEY(X'FF')
```

- Upon starting DFSMShsm determines the key range for each cluster
  - These ranges are stored in the Multiple Host Control Record
  - These ranges are then used to determine which data set to use, based on the key of the record being processed

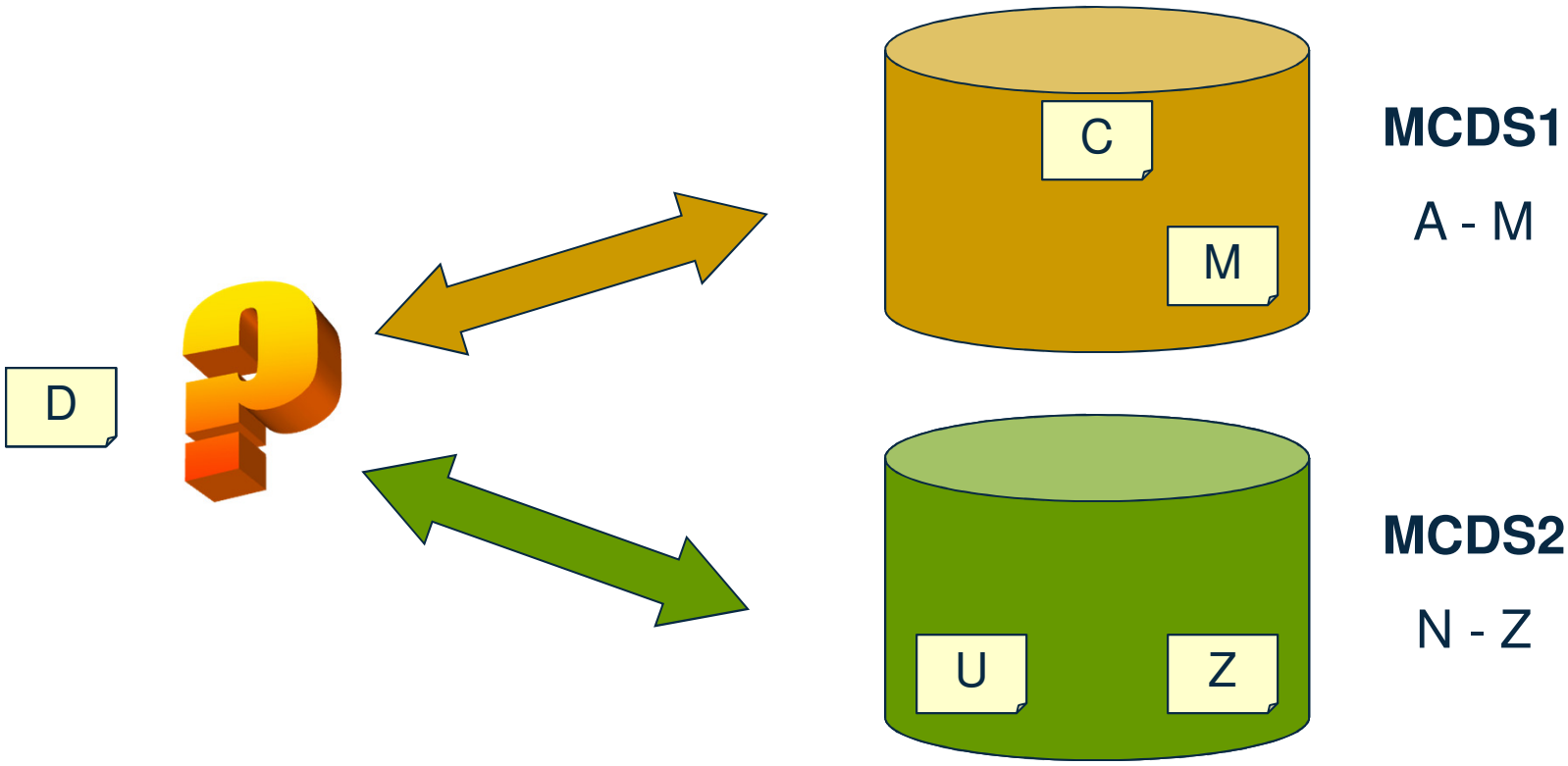
Multiple Cluster CDSs

- When an additional cluster needs to be created, no special processing is needed
  - At startup HSM will automatically determine that there is a new cluster and recalculate the record ranges
- Special care must be taken when records are redistributed within existing records, via a CDS reorg
  - Clusters can grow unevenly based on the record distribution
  - Before shutting down, the MHCR must be patched

```
FIXCDS S MHCR PATCH(X'159' X'FF') /* MCDS */
FIXCDS S MHCR PATCH(X'15A' X'FF') /* BCDS */
```

  - This notifies HSM that the change is intentional
- ✓ ALWAYS immediately backup the CDSs

Multiple Cluster CDSs



Best Practices

- Allocate your CDSs on your fastest disk
  - *Anybody using SSD*?
- Allocate each CDS on a different volume
- Allocate Journal on a different controller
- Don't use Secondary Allocations
- Don't migrate CDSs, CDS backup copies, journal or journal backup copies

Access and Serialization

- **There are three techniques for serialization**
  - **CDSQ**: Enqueues
  - **CDSR**: Volume Reserves
  - **CDSSHR = RLS**: VSAM Record Level Sharing

- Serialization technique is established in the HSM Proclib member and must be consistent

ARC0200I TRAP IN MODULE ARCILOG,
CODE=0099, FATAL ONCE ADDED

Access and Serialization

- **CDSR=YES**
  - Default serialization if none specified
- Local shared Enqueue / Reserve used
  - Major: ARCGPA
  - Minor: ARC*x*CDS (*x*=M, B or O)
- Must ***not*** convert Reserve
- Volume Reserves introduce greater likelihood for lockouts
- A volume shared across multiple sysplexes is an example of when Reserves need to be used

Access and Serialization

- **CDSQ=YES**
  - Preferred over CDSR
- Global enqueues used
  - Major: ARCENQG
  - Minor: ARC*x*CDS (*x*=M, B or O)
- Must propagate to all systems sharing CDSs
- Only as good as your global serialization product
  - *One of the most common causes for HSM CDS corruption*

Access and Serialization

- **CDSSHR=RLS**
  - VSAM RLS, Best Practice
  - CDSQ & CDSR are ignored

- VSAM RLS manages multiple host access in a very efficient manner
  - CDS VERIFY and Buffer Invalidation are not required
  - Serialization performed at the record level

Access and Serialization

- Before and After client data (CDSQ -> RLS) after 1 year

Function	Increase in GB Moved	Elapsed Time
Automatic Backup	33%	-25%
Secondary Space Management	18%	-33%

- Before and After client data for Audit
  - Before: 24+ hours
  - After: 4 hours
- Ensure that your RLS structures are defined large enough

Access and Serialization

- **With CDSQ & CDSR**
  - Only one host can perform CDS I/O at a time
  - While waiting, all CDS I/Os are queued up
  - When it get exclusive access
    - Performs a VERIFY
    - Flushes all of it's existing VSAM buffers because they are no longer valid
    - Performs all of it's outstanding I/Os

- **With RLS**
  - VSAM RLS ensure integrity across all systems
  - All hosts can perform I/O concurrently

Access and Serialization

- **CDSSHR=YES | NO**
  - CDSSHR indicates whether or not HSM is running on multiple systems
  - If CDSSHR is not specified, if the VSAM index of the MCDS is on a Disk genned as shared, then global serialization is used
  - CDSSHR=YES forces global serialization
  - CDSSHR=NO disables global serialization
    - Only use in a single system HSM for which the MCDS resides on a shared disk

Access and Serialization

Redbook: DFSMShsm Primer

Table 8-2 DFSMShsm serialization with startup procedure keywords

CDSQ keyword	CDSR keyword	CDSSHR keyword	Serialization
YES	YES	YES	Both CDSQ and CDSR options are used.
YES	NO or not specified	YES	Only the CDSQ option is used.
With any other combination of specifications		YES	Only the CDSR option is used.
--	--	RLS	Uses VSAM RLS.
With any other combination of specifications		NO	No multiple host serialization.
With any other combination of specifications		Not specified	Performs multi-host serialization of type CDSQ and CDSR (whichever is specified) only if the DASD volume where the index of the MCDS resides has been SYSGENed as SHARED or SHAREDUP.

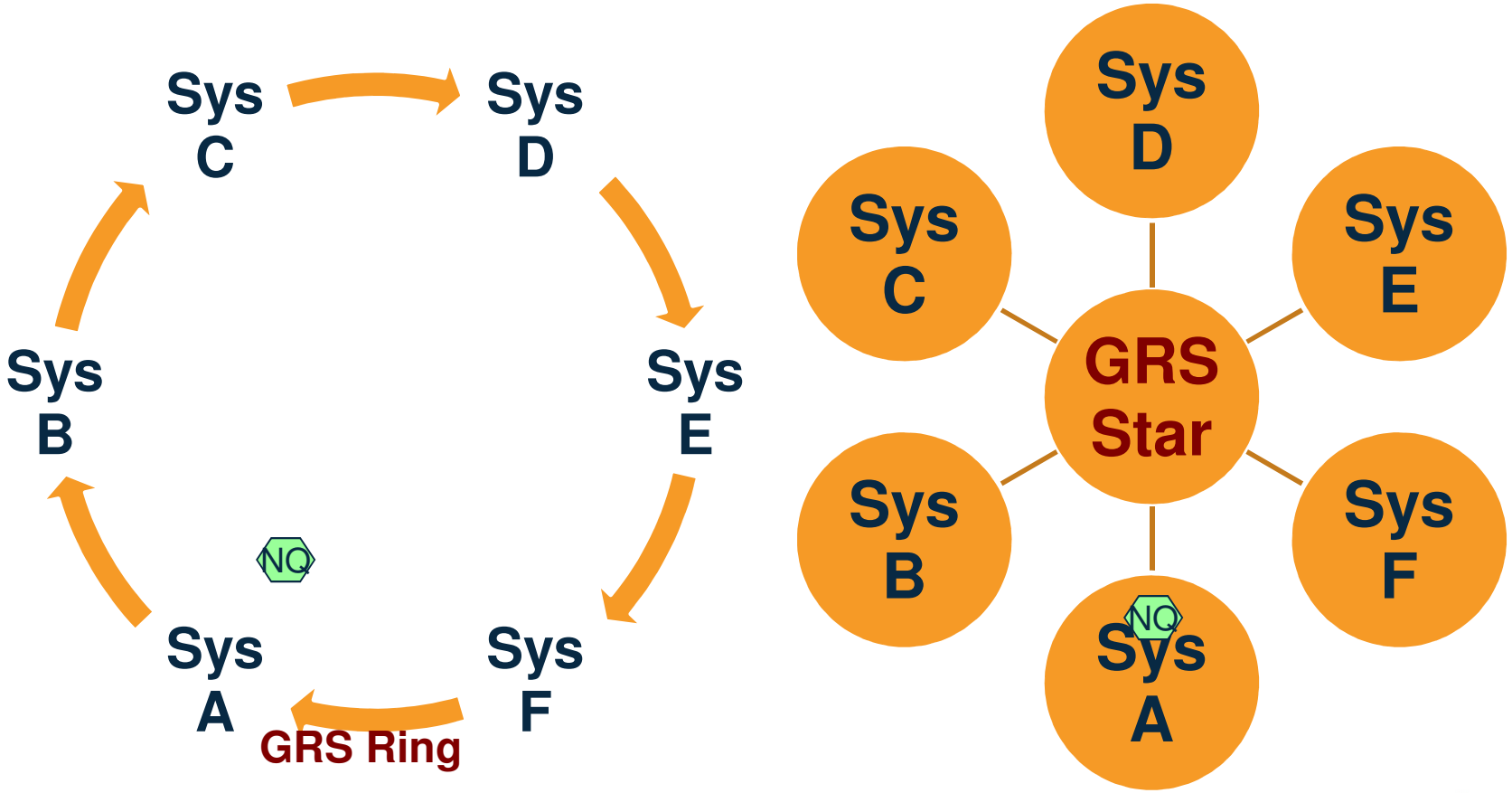
Access and Serialization

- *Internal* performance testing has shown a significant improvement in CDS I/O intensive functions when using ***GRS Star*** as opposed to GRS Ring

  - **GRS Star** – A parallel sysplex implementation of Global Resource Serialization
    - Resource name list is placed in the coupling facility so that any request for a resource can be resolved with a single interaction

  - GRS Ring – A resource request must be passed to every participating member of the sysplex (ring)

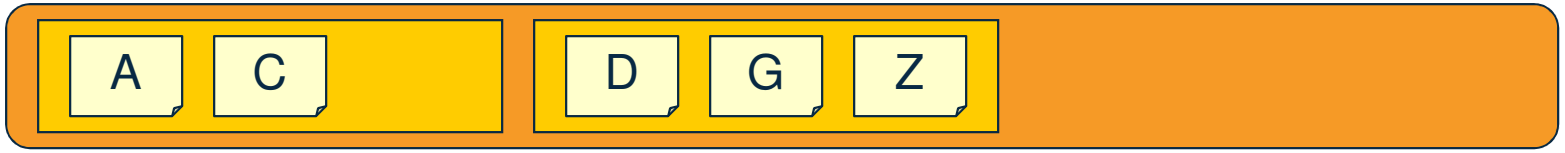
Access and Serialization



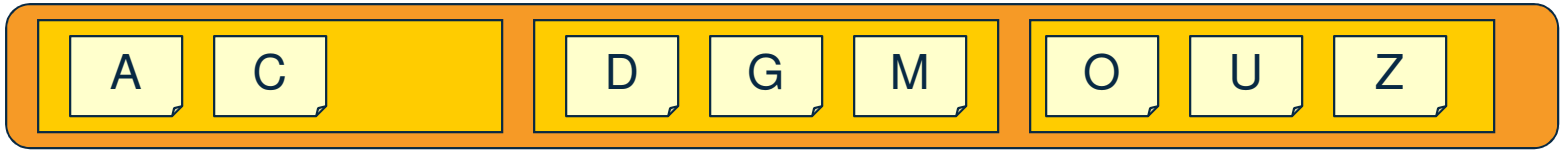
CDS Reorganization, Why??

Time t1: Dataset is loaded with FREESPACE(0,0)

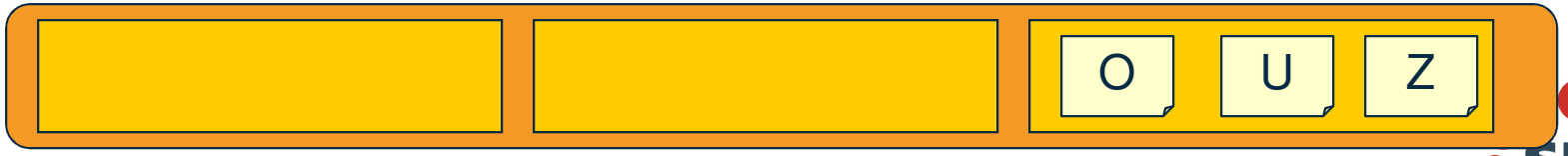
Time t2: Insert Records ‘C’ & ‘Z’. No space, so a split is done.



Time t3: Insert Records ‘M’, ‘O’ & ‘U’. No space, so a split is done.



Time t4: Delete Records ‘A’, ‘C’, ‘D’, ‘G’ & ‘M’

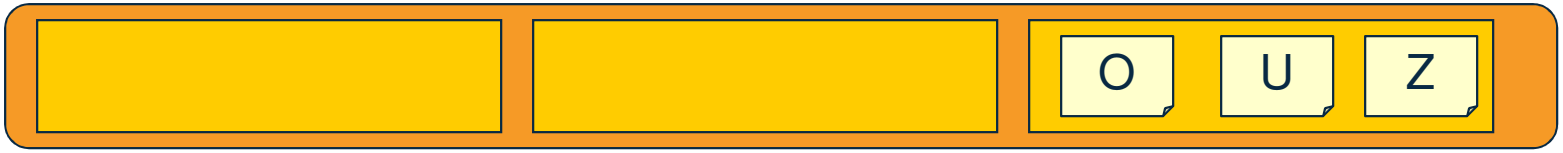


CDS Reorganization

- Necessity to reorganize the DFSMShsm CDSs impacts…
  - Availability – Must stop all DFSMShsm hosts
  - Performance – increased CI/CA splits afterwards
  - Integrity - Doing it incorrectly is a common cause for CDS breakage
  - Best Practice: Run an EXAMINE INDEXTEST after the reorg
- **VSAM CA (Control Area) Reclaim** function
  - Reclaims empty CAs that remain after all records are deleted
  - Common for DFSMShsm
    - Many records written with the time / date in the key
- What this means for DFSMShsm…
  - *Reduces* the need for CDS reorgs
  - *Reduces* the need for SDSP reorgs
  - *Doesn't eliminate* need to reorg

CDS Reorganization, CA Reclaim

After Time t4, with CA Reclaim empty CAs automatically reclaimed



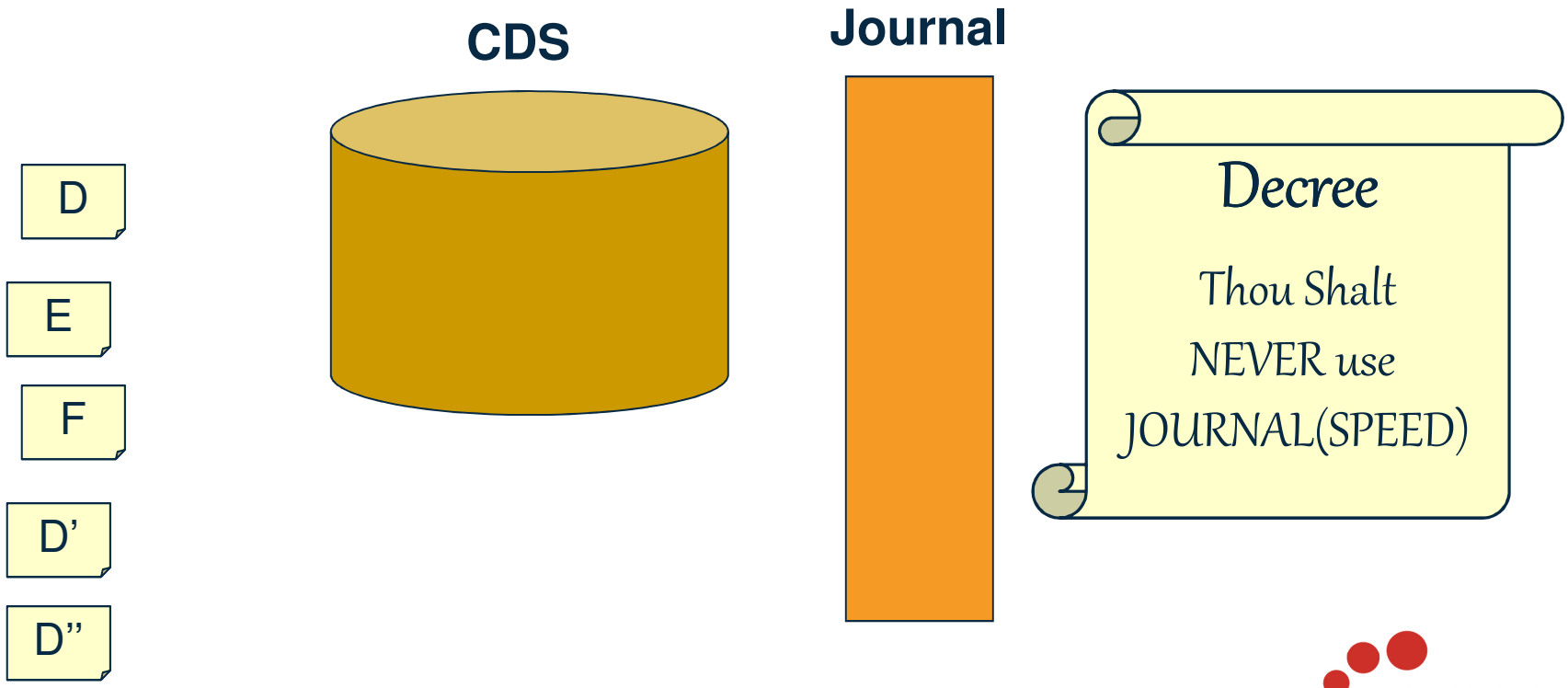
Reorgs are still needed because …

CDS Backup

- CDSs can break …
  - During a reorg, one or more hosts were not stopped
  - During a reorg, a host was started
    - Classic examples
      - HSM is down, so migrated data sets get an 'HSM is not active' message.
      - "Gee, HSM shouldn't be down… let me start it"
    - Use DISP=OLD in reorg jobs
  - After splitting into another cluster, one host's PROCLIB was not updated to show new cluster
    - *This was a doozy to cleanup!!*
  - Physically
    - Serialization product not configured properly
      - Global enqueues were not propagated correctly
    - Serialization product is down

CDS Backup

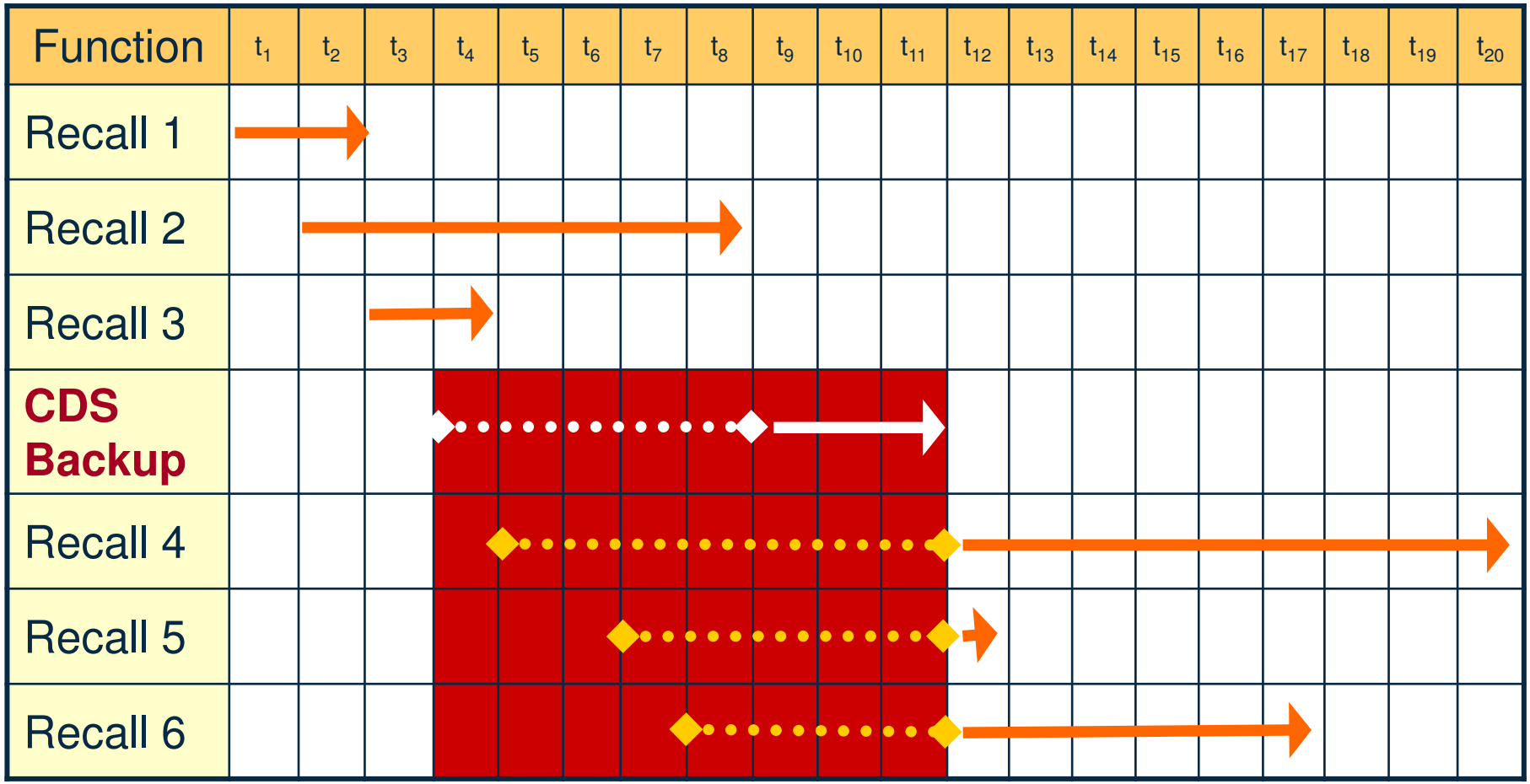
- CDS processing is designed to be able to always recover all of the records through Point in Time backup copies and journaling every critical update



V1R13, Nondisruptive CDS Backup

- CDS Backup can be very disruptive to other HSM activity
  - All other HSM activity must be quiesced before CDS Backup can start
    - Some customers HOLD all HSM activity prior to the start of CDS Backup to ensure that it can begin at its scheduled time
  - Functions that start while CDS Backup is waiting to start have to wait until the completion of CDS Backup
  - Higher impact in an RLS environment than nonRLS
- Journal is backed up using Standard I/O, even when Concurrent Copy is specified
  - Since there is a chance for Concurrent Copy to fail the physical copy after logical completion, Standard I/O is always used for the journal to ensure it is not nulled without being copied in its entirety
  - Outage for CDS backup is at least as long as the time it takes to backup the journal

Nondisruptive CDS Backup



New HSM Activity Quiesced

Pre-V1R13

Nondisruptive CDS Backup

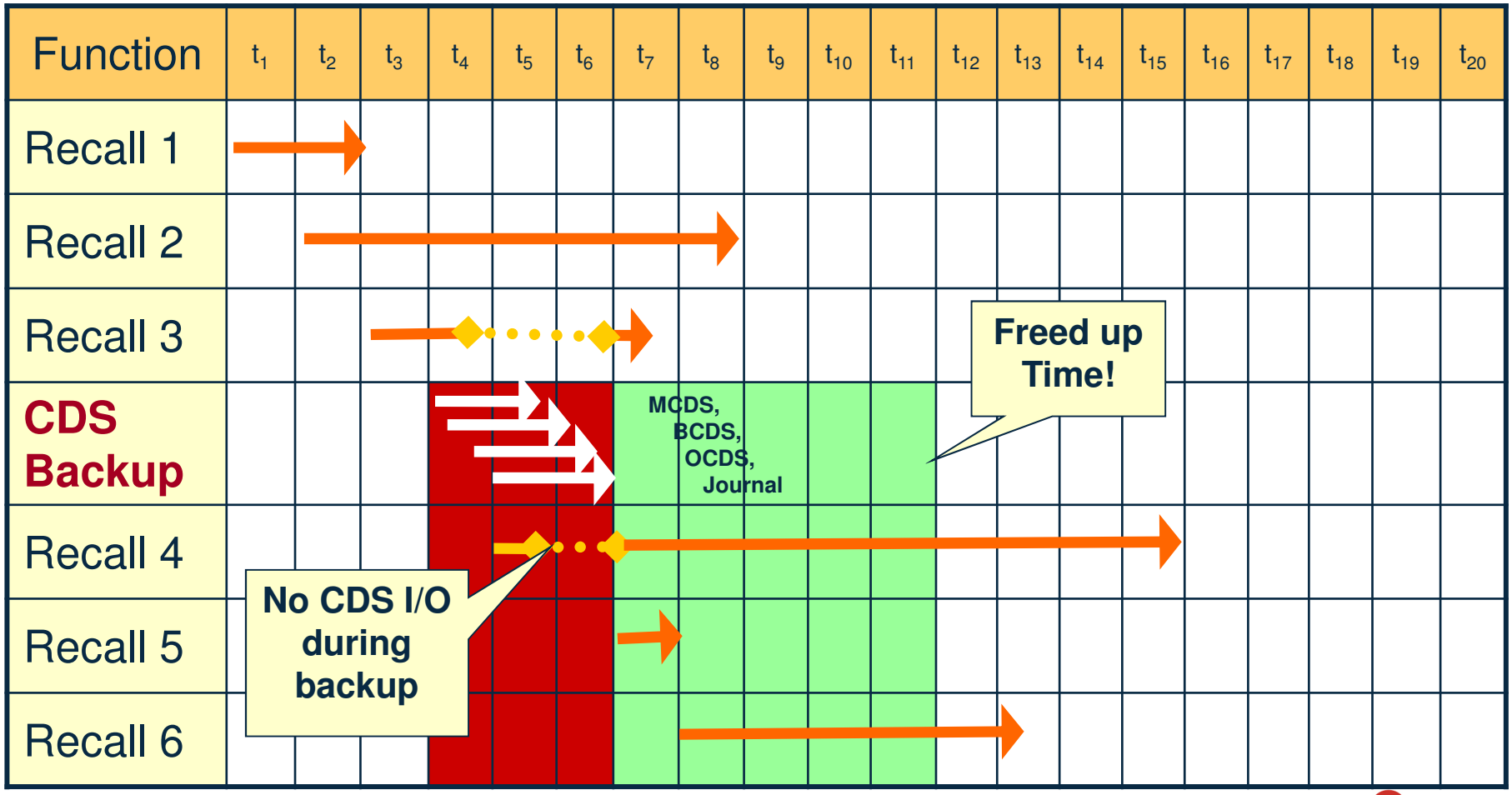
- nonRLS Environment
  - Only HSM activity on the *same LPAR* impacts/is impacted by CDS Backup
    - Serialization scheme uses enqueue scope of SYSTEM and a Reserve to cover other systems

- RLS Environment
  - HSM activity on *any host* in the HSMplex impacts/is impacted by CDS Backup
    - Serialization scheme uses enqueue scope of SYSTEMS

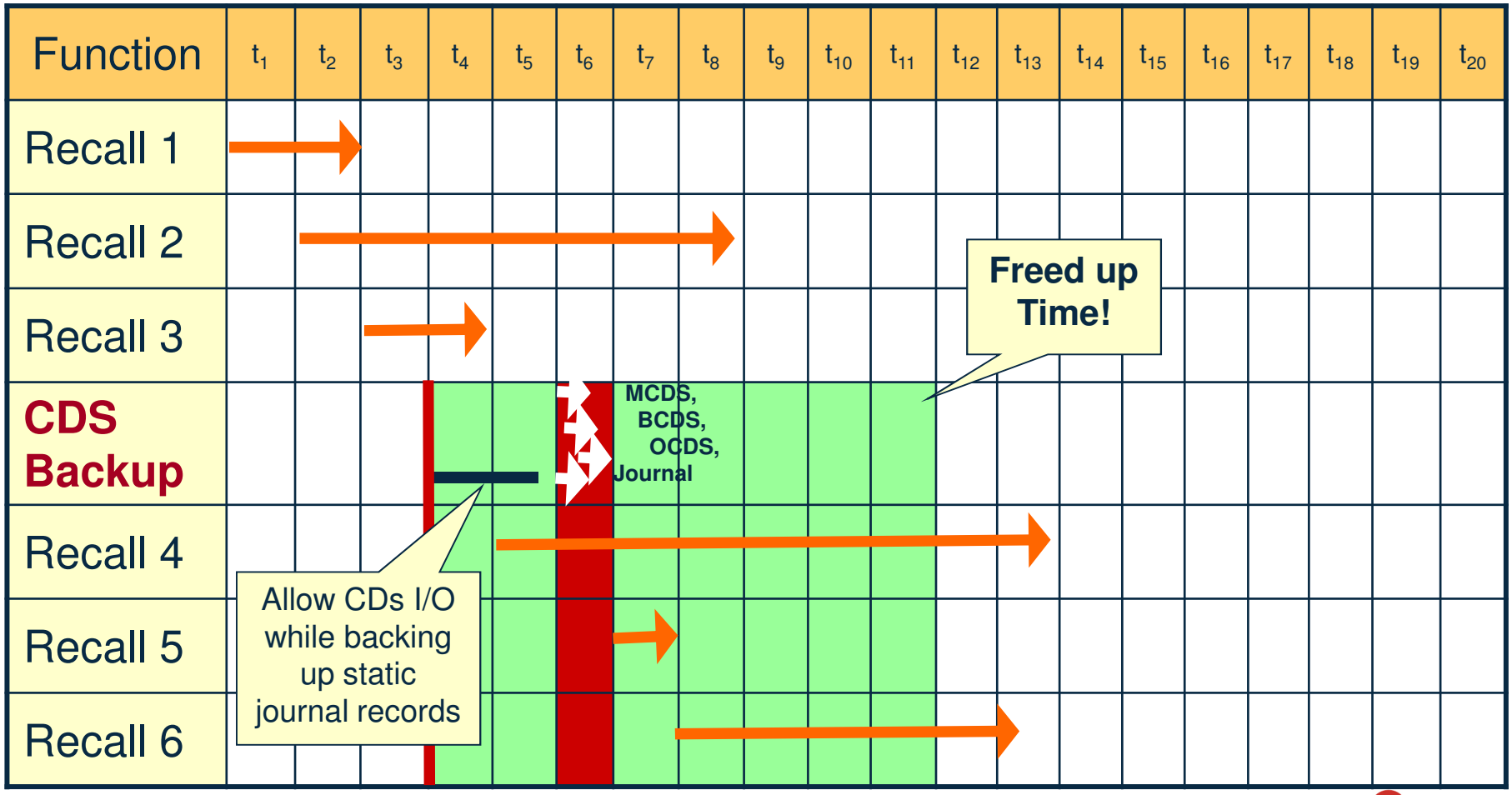
Nondisruptive CDS Backup

- V1R13 Enhancements
  - CDS Backup serialization scheme has been enhanced such that all active HSM activity *does not* have to complete before CDS Backup can begin
    - CDS and Journal I/O is quiesced before and during copy of control data sets and journal to ensure a data consistent backup
    - When concurrent copy is used, this is a brief disruption
  - The backup of the Journal will begin before the CDSes are quiesced
    - HSM activity can continue while the 'static' portion of the journal is backed up
    - Activity is quiesced during brief time required to backup the remainder of the journal
    - Requirements:
      - All CDS clusters are SMS-managed
      - Concurrent Copy specified
      - SETSYS CDSVERSIONBACKUP(DATAMOVER(DSS))
      - SETSYS JOURNAL(RECOVERY)

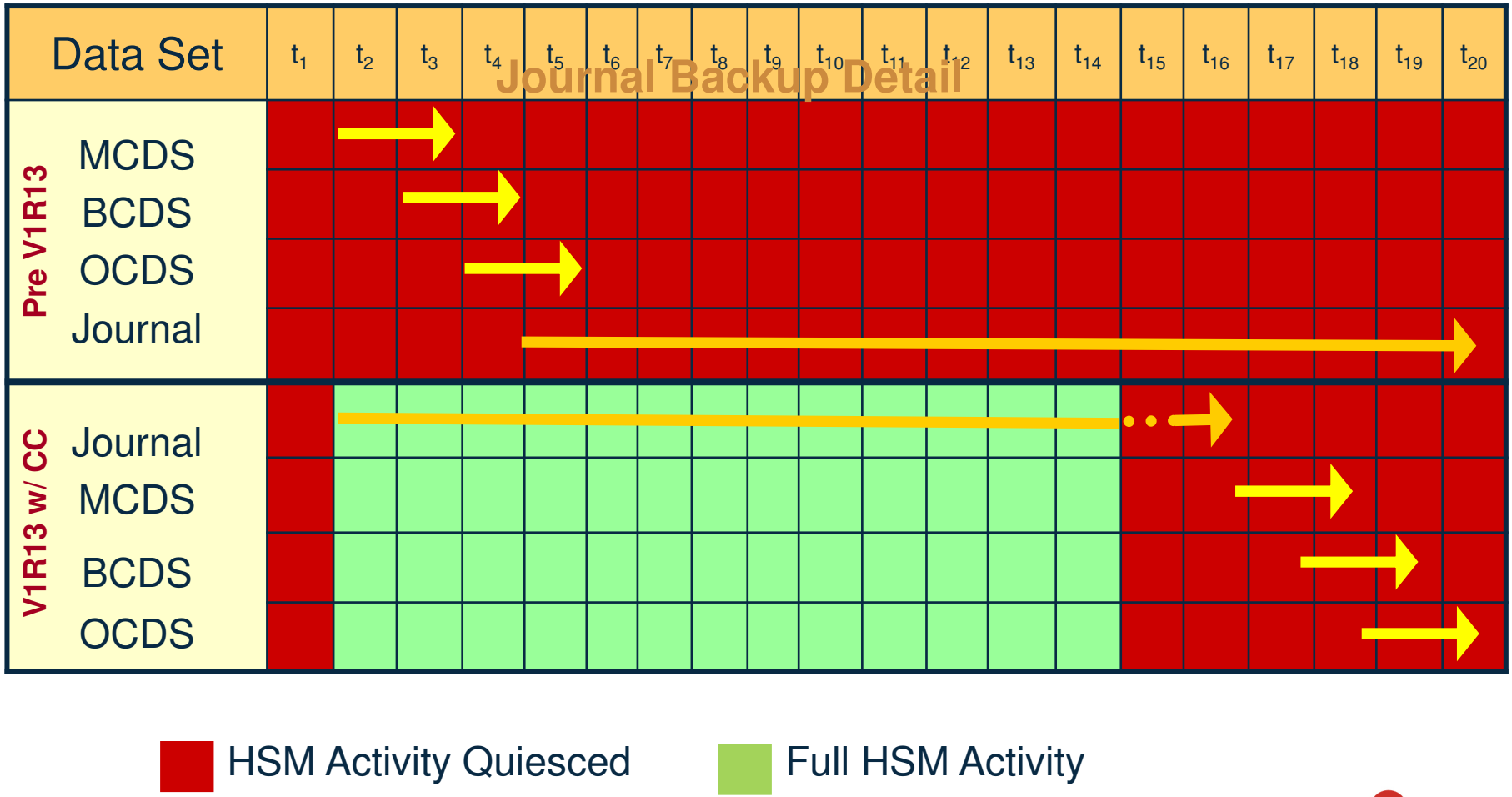
Nondisruptive CDS Backup



Nondisruptive CDS Backup



Nondisruptive CDS Backup



Nondisruptive CDS Backup

- CDS Recovery process is unchanged
- Migrating to new function
  - The journal backup enhancement requires the coexistence APAR to be active for a full backup cycle before becoming fully enabled
  - If you use the function on a V1R13 system that never had the coexistence applied, then not until the *second* and subsequent CDS backups will the function be fully enabled
  - Remove DFSMShsm HOLD commands scheduled before the start time of CDS Backup
- Coexistence
  - When using RLS, the full benefit isn't seen until *all* hosts are at V1R13

CDS Backup Best Practices

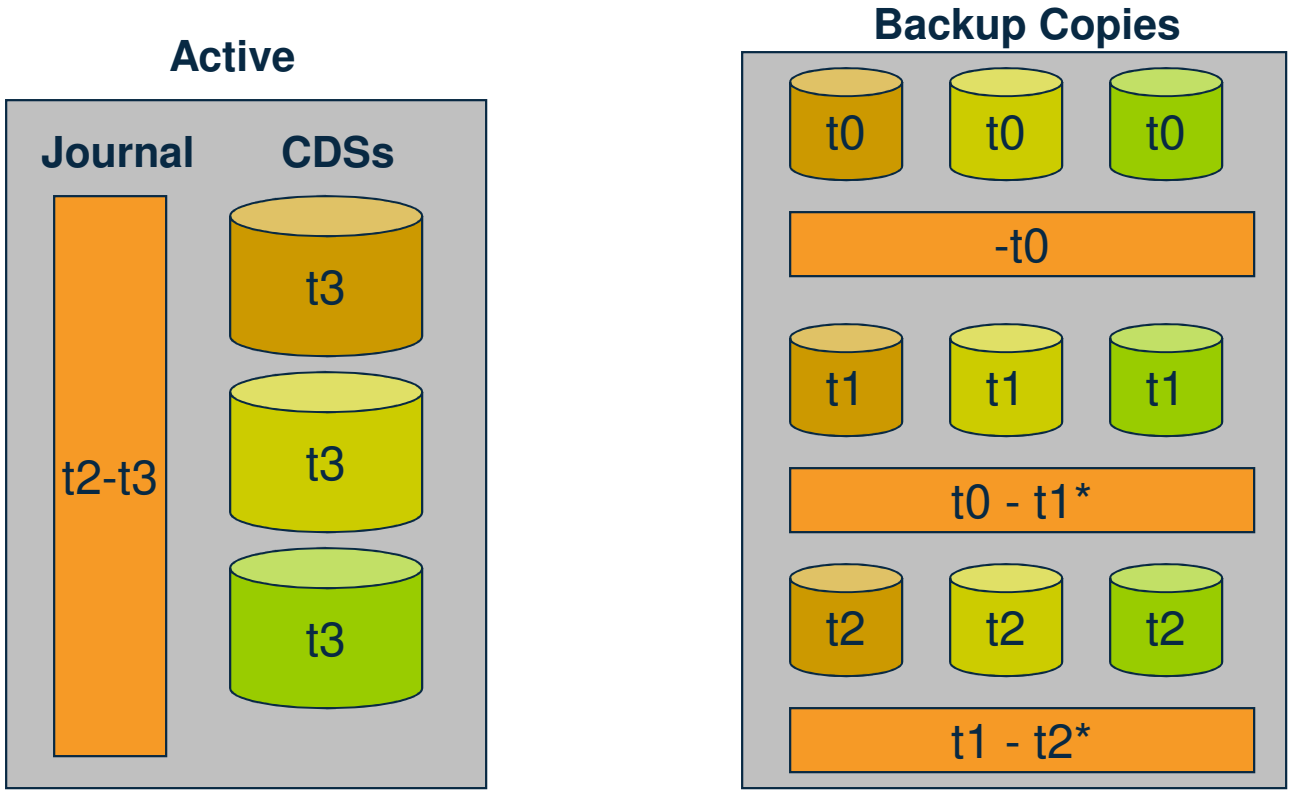
- 14 Backup copies
  - Keep 4 or more on disk and use ARCCBEXT to copy the backup copies to tape
  - Use Tape Management to manage the tape copies
- Use DSS to create the backup copies
  - DSS validates the integrity of the CDSs during backup

CDS Recovery

- Two types
  - Enhanced CDS Recovery
    - Preferred method
    - More manually intensive, but much more efficient
    - Performed while HSM is down
  - UPDATEC
    - Slower
    - Performed while HSM is active

CDS Recovery

- Select the most recent backup copy that was made prior to the introduction of the error (i/o errors start appearing)



*Journal backup contains all changes in-between backups

CDS Recovery

- Step 1a
  - If HSM was used to create the backup copy, no action
  - If DSS was used to create the backup copy, then first you must restore the CDS and then Export the restored data set
    - Enhanced CDS Recovery requires the data to be in Exported format

```
//JOB1 JOB accounting information,REGION=nnnK
//STEP1     EXEC  PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//TAPE      DD    DSN=uid.CDS.BACKUP.D0000106,DISP=(OLD,KEEP)
//DASD      DD    UNIT=3390,VOL=SER=222222,DISP=OLD
//SYSIN     DD    *
            RESTORE DATASET(-
            INCLUDE(uid.dsname))-
            INDDNAME(TAPE)-
            OUTDDNAME(DASD)-
            REPLACE
/*
```

```
//EXPORT JOB accounting information,REGION=nnnnK
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//OUTDD1   DD DISP=(,CATLG,DELETE),BUFNO=26,BLKSIZE=28332,
//         SPACE=(CYL,(200,20),RLSE),
//         UNIT=3390,VOL=SER=222222,
//         DSN=temp_dsname
//SYSIN    DD *
           EXPORT dsname -
           OUTFILE(OUTDD1) -
           TEMPORARY
/*
```

CDS Recovery

- Step 1b
  - Create the ‘backup journal’ using ARCBJRNL
    - All journals with records after the CDS copy being recovered
    - All records will be written to a single data set
    - Journals

If Recovering from Backup t0, need journal backups t0-t1, t1-t2 and the Active

Note – The active journal will always be needed

```
//ARCBJRNL JOB ,MSGLEVEL=(1,1),MSGCLASS=A,REGION=1024K
//*-------------------------------------------------
//* CREATE DATA SET CONTAINING JOURNAL FROM JOURNAL
//* BACKUP COPY DATA SETS AND THE ACTIVE JOURNAL
//*-------------------------------------------------
//STEP01    EXEC PGM=ARCBJRNL,PARM='BOTH'
//STEPLIB   DD DSNAME=location where program ARCBJRNL resides,
//             DISP=SHR
//SYSPRINT  DD SYSOUT=*
//JRNLO     DD DSNAME=online journal data set name,
//             DISP=SHR
//BKUPJRNL  DD DSNAME=backup copy name and version,DISP=SHR
//          DD DSNAME=backup copy name and version,DISP=SHR
//          DD DSNAME=backup copy name and version,DISP=SHR
//JRNLBK    DD DSNAME=backup journal name,
//             UNIT=unit name,
//             VOL=SER=volume where backup journal will reside,
//             SPACE=adequate space to hold journal,
//             DISP=(,CATLG,DELETE)
```

CDS Recovery

- Step 2
  - Sort the ‘backup journal’ so that all operations against the same record are in sequential order (sorted by key order)

```
//SORT JOB
//* ------------------------------------------------
//* SORT THE BACKED UP JOURNAL
//* ------------------------------------------------
//STEP02  EXEC SORTD
//SORTLIB   DD ................
//SYSOUT    DD SYSOUT=*
//SORTWK01  DD ........................
//SORTWK02  DD ........................
//SORTWK03  DD ........................
//SORTWKnn  DD ........................
//SORTIN    DD DSNAME=backup journal name from STEP 1,
//             DISP=SHR
//SORTOUT   DD DSNAME=sorted backup journal name,
//             UNIT=unit name,
//             VOL=SER=volume where sorted backed up journal will reside,
//             SPACE=adequate space to hold sorted backed up journal,
//             DCB=(RECFM=VBS,BLKSIZE=0,LRECL=6560),
//             DISP=(,CATLG,DELETE)
//SYSIN     DD *
  SORT  FIELDS=(17,44,CH,A)
        OPTION EQUALS
/*
```

CDS Recovery

- Step 3
  - Run ARCIMPRT
  - Imports each records from the CDS Backup and applies only the most recent journal action, if any
    - *If a record is updated 100 times, only the latest update is needed, and that is the only one that is applied*
    - FORCE is required is you are recovering from a backup version prior to

```
* * * Top of file * * *
//ARCIMPRT JOB ,MSGLEVEL=(1,1),MSGCLASS=H,REGION=1024K
//*****************************************************************/
//* RECREATE A MULTICLUSTER MCDS                                  */
//*    THE KEY BOUNDARIES FOR THE EXISTING CDS ARE DIFFERENT THAN */
//*    THE KEY BOUNDARIES THAT EXISTED AT THE TIME OF CDS BACKUP. */
//*    FORCE MUST BE SPECIFIED, AND ALL CLUSTERS MUST BE RECOVERED.*/
//*****************************************************************/
//JOBLIB   DD DSNAME=SYS1.LINKLIB,DISP=SHR
//STEP01   EXEC PGM=ARCIMPRT,PARM='BCDS,FORCE'
//SYSPRINT DD SYSOUT=*
//AMSDUMP  DD SYSOUT=*
//JOURNAL  DD DSNAME=DFHSM.JRNL.BACKUP.SORT,DISP=SHR
//EXPORT   DD DSNAME=DFHSM.TEMP.BCDS,DISP=SHR
//OUTCDS   DD DSNAME=DFHSM.RLS.BCDS2,DISP=SHR
//SYSIN    DD *
  DELETE (DFHSM.RLS.BCDS2) CLUSTER PURGE
  IMPORT INFILE(EXPORT) OUTFILE(OUTCDS)
/*
* * * End of file * * *
```

CDS Recovery

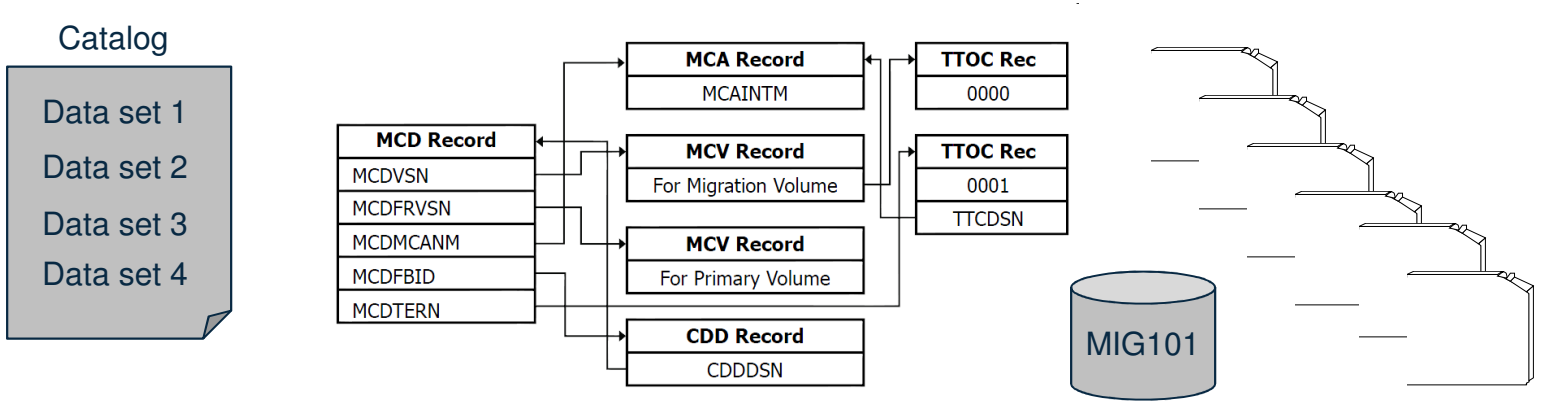
- Step 4
  - Restart HSM
  - Run AUDIT DATASETCONTROLS MIG|BACK NOFIX ODS(…)
- Step 5 (Recovery of MCDS)
  - If needed, Patch MHCR with latest CDS Backup Version
  - QUERY CDSVERSIONBACKUP compared to LISTCAT

FIXCDS S MHCR PATCH(X'B1' 00000150)

- Step *0*
  - Practice, Practice, Practice
  - No one expects to have to perform a CDS Recovery
  - It always happens during the middle of the night
    - Just ask Murphy
  - Have JCL specific to your shop ready to run, so all you have to do is plug in version numbers

Audit

- Validation of CDS records, catalog, actual data, etc



- Audit command should be performed…
  - Periodically
  - After CDS Recovery
  - After Catalog Recovery
  - After an ARC1841I (error reading or writing CDS)
  - ‘Not found’ types of errors

Audit

- AUDIT DATASETCONTROLS(MIGRATION | BACKUP)
  - M|B CDS is read sequentially and verification is done for each Migration | Backup Data Set record and associated records
  - For migration, verifies data set is cataloged as MIGRAT
- AUDIT VOLUMECONTROLS(MIGRATION | BACKUP)
  - HSM volume records are read sequentially and verified against all associated records
- AUDIT VOLUMECONTROLS(RECOVERABLE)
  - Verifies Dump records

Audit

- AUDIT DIRECTORYCONTROLS VOLUMES(tapevol)
  - Verifies each TTOC record and other records associated with the volume
- AUDIT MEDIACONTROLS VOLUMES(tapevol)
  - Tape is read sequentially and all records verified
  - This can run a long time for long tapes
    - Use RESUME option if using FIX and it was previously held
- AUDIT DIRECTORYCONTROLS VOLUMES(ml1)
  - Verifies ML1 VTOC and associated records
- AUDIT MEDIACONTROLS VOLUMES(ml1)
  - Verifies ML1 VTOC and associated records
- AUDIT MEDIACONTROLS(SDSP) VOLUMES(ml1)
  - SDSP records read sequentially

Audit

- AUDIT ABARSCONTROLS(agname)
  - ABARs records
- AUDIT COMMONQUEUE
  - Common Recall Queue structure, not CDSs
- AUDIT COPYPOOLCONTROLS
  - Run with and without copy pool names
- NOFIX | FIX
  - First run with NOFIX and then with FIX
  - One type of Audit may point to other types needing to be run
- *If you haven't run in a while, don't be surprised if it finds alot*

Audit

```
-- DFSMShsm AUDIT -              DFSMShsm CONTROL DATASET -- LISTING- AT 18:48:21 ON 91/01/24

  AUDITING THE MIGRATION CONTROL DATASET

  ERROR                DATA SET NAME                          DATA SET ON    CATALOGED      MIGRATED
  TYPE                                                          VOLUME      TO VOLUME     TO VOLUME

  *ERR 03         D324711.ESDS.R.F40RP123.CLUSTER2             ??????        -NONE-        M2TP02
  *ERR 03         D324711.PSF.N.F40RL016.DSET01                ??????        -NONE-        M2TP02
  *ERR 03         D324711.PSF.N.F40RL055.DSET02                ??????        -NONE-        M2TP02
  *ERR 03         D324711.PSF.R.F40RP123.DSET02                ??????        -NONE-        M2TP02
  **NONE          D324711.PSFB.N.F40EM009.DSET01               ??????        -NONE-      RECALLED
  **NONE          D324711.PSFB.N.F40EM037.DSET01               ??????        -NONE-      RECALLED
  **NONE          D324711.PSFB.TEST.DATASET                    ??????        -NONE-      RECALLED
  *ERR 03         G834921.KSDS.N.F40TM479.CLUSTER1             ??????        -NONE-        M2TP01
  **NONE          G834921.PSF.N.F40TM434.DSET02                ??????        -NONE-      RECALLED
  *ERR 03         G834921.PSF.N.F40TM479.DSET03                ??????        -NONE-        M2TP03
  *ERR 03         H952762.PSFB.F40AU001.DSET03                 ??????        -NONE-        M2TP03
  *ERR 03         H952762.PSFB.F40AU001.DSET04                 ??????        -NONE-        M2TP01
  *ERR 03         M059259.BDAM.N.F40RL025.DSET01               ??????        -NONE-        M2TP02
  **NONE          M059259.KSDS.N.F40RL038.CLUSTER1             ??????        -NONE-      RECALLED
```

Refer to DFSMShsm Storage Administration, Chapter 67

Description	Audit Repair Action	Troubleshooting Hints
The catalog has no entry for the data set that is being audited, but the MCDS record indicates that the data set is on a migration volume.		For AUDIT DSNAMES: Because there is no catalog entry for the data set with the migration copy, that copy is most likely an old one that has not yet been scratched, probably because of a previous error when the data set was being recalled or deleted. There are two methods to correct the

FIXCDS

- Just as the name indicates, the FIXCDS commands operates directly against the control data sets to correct errors
- Audit will produce FIXCDS commands when it is able

```
-DFSMSHSM AUDIT-       ENHANCED AUDIT -- LISTING - AT 11:59:56 ON 12/09/04 FOR
SYSTEM=3090
COMMAND ENTERED:
AUDIT COPYPOOLCONTROLS ODS(DERDMANN.AUDIT.CPC.AFTER2)

/* ERR 180 J RECORD NOT FOUND FOR VOLUME SRC003, IT BELONGS TO COPY POOL CP1 ,
VER=001*/
/* ERR 178 FRTV (I) RECORD FOR TARGET VOLUME TAR001, EXPECTS FRSV (J) RECORD
FOR SOURCE VOLUME SRC003, WHICH WAS NOT FOUND */
/* ERR 202 ORPHANED I (FRTV) RECORD FOUND FOR TARGET VOLUME ABC123, SOURCE
VOLUME ..... */
/* FIXCDS I ABC123 DELETE */
/* ERR 178 FRTV (I) RECORD FOR TARGET VOLUME ABC123, EXPECTS FRSV (J) RECORD
FOR SOURCE VOLUME ....., WHICH WAS NOT FOUND */
/* ERR 178 FRTV (I) RECORD FOR TARGET VOLUME TAR001, EXPECTS FRSV (J) RECORD
FOR SOURCE VOLUME SRC003, WHICH WAS NOT FOUND */
- END OF -     ENHANCED AUDIT - LISTING -
```

- Refer to DFSMShsm Diagnosis

FIXCDS

- FIXCDS *record_type key options*
  - Display, Create, Patch, etc
- Besides Diagnosis Guide, the Data Areas book also provides information and has all of the offsets

The key for a type C backup control data set backup version record is the backup version data set name. An example of the key that is used with a C backup control data set backup version record is:

```
FIXCDS C DFHSM.BACK.T352016.DATA.NAME.H4323
```

Using the DISPLAY GEN(*nn*) parameter or the PATCH GEN(*nn*) parameter with the FIXCDS command allows you to specify the original data set name, along with a generation number, instead of specifying the DFSMShsm-generated name for the backup version data set. An example of displaying a type C backup version record using the original data set name and a generation number is:

```
FIXCDS C JLT7652.REPORT.DATA DISPLAY GEN(0)
```

- Use VERIFY if you are using PATCH!!

FIXCDS

FIXCDS offset is 64 less than the actual because you cannot change the key or header. That's why it's a good idea to use VERIFY, to make sure that you have specified the correct offset.

Table 81 (Page 1 of 4). MCC—BCDS Backup Version Record

Offsets Actual	Offsets FIXCDS	Type	Length	Name	Description
0(0)			44	MCK	Backup control data set backup version record key, consisting of the data set name of a backup version and padded with blanks. (See MCK for details.)
44(2C)			20	MCH	Control data set record header. (See MCK for details.)
64(40)	0(0)	STRUCTURE	316	MCC	Data portion of the BCDS backup version record.
64(40)	0(0)	CHARACTER	44	MCCADSN	Data set name of the original data set.
108(6C)	44(2C)	CHARACTER	6	MCCVSN	Volume serial number of the backup volume containing this version.
114	(72)	FIXED	2	MCCBCN	BACKUP COPY NUMBER NX
114(72)	50(32)		2	*	Reserved.
116(74)	52(34)	CHARACTER	4	MCCUCBTY	The next 4 bytes contain the device type of the backup volume.
116(74)	52(34)	CHARACTER	2	*	Reserved.
118(76)	54(36)	BITSTRING	1	*	This byte contains the following flags:
		1...		MCCTPDEV	When set to 1, this version is on a tape device.
		.xxx xxxx		*	Reserved.
120(78)	56(38)	CHARACTER	8	MCCTSBU	Time stamp when backup copy is created, containing the following:

V2R2 UPDTCDS Command

- New UPDTCDS command for making expected updates to the CDSs
- Today, it can take *hundreds/thousands* of FIXCDS commands to update the expiration date of a single copy pool dump version

UPDTCDS COPYPOOL(*cpname*) VERSION(*ver*)
DUMPEXPIRATION(DCLASS(*class*) NEWDATE(*date*))

- This command updates all of the CDS records that need to be updated to reflect the new expiration date
  - Fast Replication Dump record (FRD)
  - Dump record for each volume dump (DGN)
- This command will be extended to other functions in the future

DFSMShsm CDS Deep Dive

Glenn Wilcock
IBM

March 3, 2015
Session 17101

#SHAREorg

Insert Custom Session QR if Desired.

SHARE is an independent volunteer-run information technology association that provides **education**, professional **networking** and industry **influence**.

Legal Disclaimer

Trademarks

The following are trademarks of the *International Business Machines Corporation:*

IBM, DFSMS/MVS, DFSMShsm, DFSMSrmm, DFSMSdss, DFSMSopt, DFSMS Optimizer, z/OS, eServer, zSeries, MVS, FlashCopy®

The information contained in this presentation is distributed on an 'AS IS' basis without any warranty
either expressed or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. The use of this information is a customer responsibility and depends on the customer's ability to evaluate and integrate it into the customer's operational environment.