
Reclaim Those Empty CAs!

Share Session 10964

Terri Menendez
STSM VSAM/Catalog Development
March, 2012
terriam@us.ibm.com

Agenda

- ❑ VSAM Data set Types.
 - ❑ Example of a Key Sequence Data Set (KSDS).
- ❑ VSAM KSDS Space Management (prior to z/OS 1.12):
 - ❑ Control Interval (CI) Split.
 - ❑ Control Interval (CI) Reclaim.
 - ❑ Control Area (CA split).
- ❑ KSDS Fragmentation
 - ❑ Issues related to KSDS fragmentation.
- ❑ VSAM KSDS Space Management (z/OS 1.12):
 - ❑ Control Interval (CI) Reclaim.
 - ❑ Control Area (CA) Reclaim.
- ❑ Enabling CA Reclaim
- ❑ CA Reclaim Reporting and Recovery
- ❑ Performance Benchmarks
- ❑ “State of the Union”

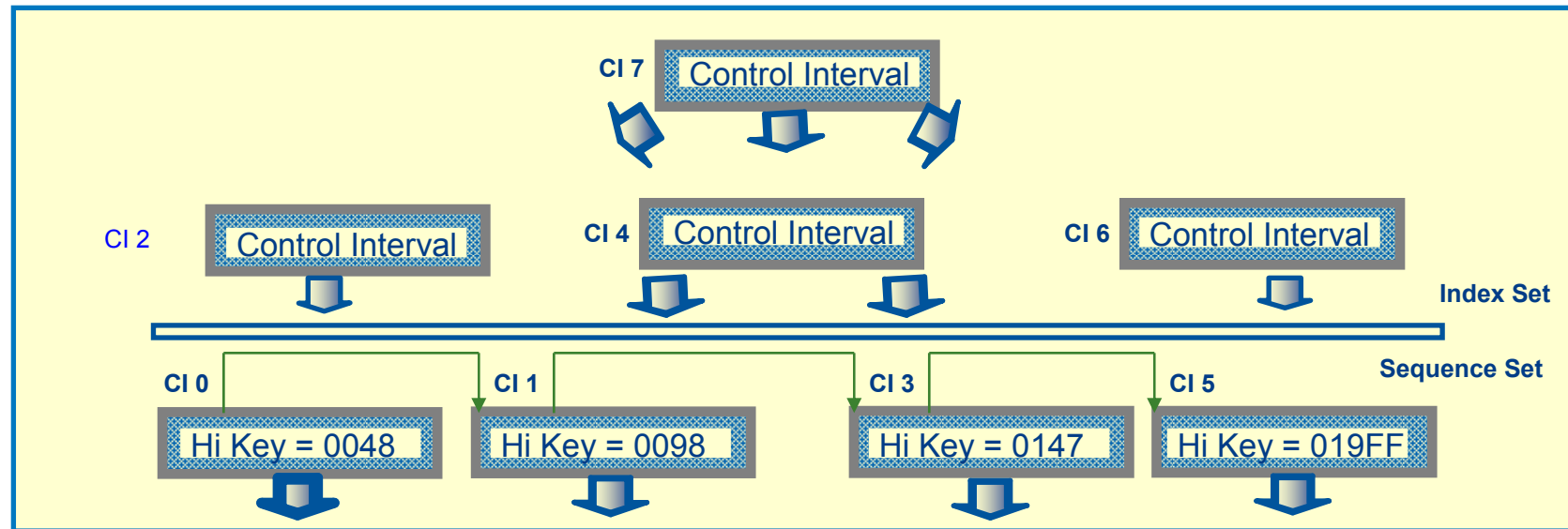
VSAM Data Set Types

VSAM Data Set Types

- VSAM provides 4 possible data set types, ranging from very unstructured to highly structured
- The data set types are:
 - ❑ LDS (Linear Data Set)
 - ❑ Contains data with no record boundaries. User must define the logical format of the space.
 - ❑ ESDS (Entry Sequence Data Set)
 - ❑ Contains records in the order they are entered. Can only add to the end of the data set. Records can be “logically” deleted (user managed).
 - ❑ (V)RRDS (Relative Record Data Set)
 - ❑ Contains records in order by relative record number. Records can only be accessed by its relative record number.
 - ❑ Records can be erased and re-inserted anywhere in the data set.
 - ❑ VRRDS consists of two components: Data and Index.
 - ❑ KSDS (Keyed Sequence Data Set)
 - ❑ Contains records in logically ascending sequence. Can be accessed by key or RBA.
 - ❑ Records can be retrieved, inserted, updated, or erased anywhere in the data set.
 - ❑ Consists of two components: Data and Index Component
 - ❑ Lowest Key Possible = 0... Highest Key = FF...
 - ❑ Keys are stored in the index with both front and rear compression.
 - ❑ Most popular VSAM data set. Used by user applications and system components.

Example of a KSDS

Example Of KSDS – Inserted Keys 0001-0192



CA 0

CA 1

CA 2

CA 3

Index Component

Data Component

Page 6

Note: Data set defined with one record per CI, key length = 4, CISIZE=512, 31 CIs/CA

VSAM KSDS Space Management (Prior to z/OS 1.12)

Control Interval (CI) Split

Control Interval (CI) Split - Prior to z/OS 1.12

- ❑ VSAM allows new (or larger record) inserts anywhere in the data set.
- ❑ Accomplished by inserting space when needed.
- ❑ Space is “added” by obtaining a free CI from a list of free CIs in the index sequence set record. Referred to as a CI split.
- ❑ Current CI split design holds data set level serialization.
- ❑ Two types of CI splits:
 - ❑ “Add to end” occurs when adding records to the end of the data set. New CI obtained for the new record. **Not recorded in the CI split statistics.**
 - ❑ “Not add to end” occurs when adding records anywhere but the end of the data set. Typically existing CI split in half and half the records moved to a new CI.

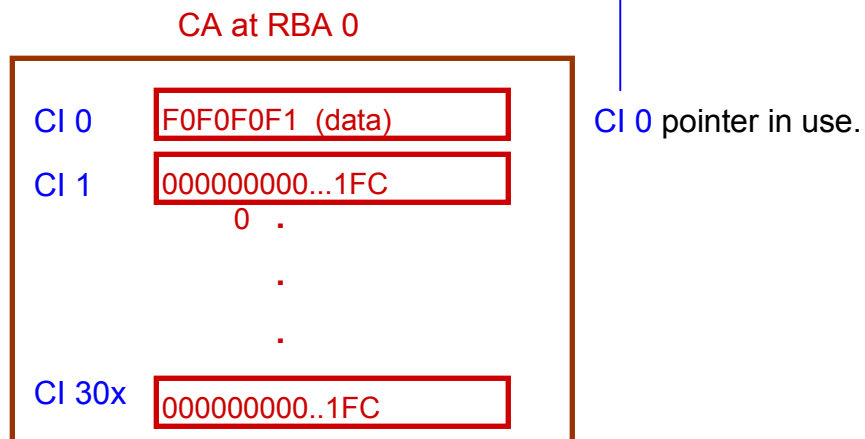
Example of a Sequence Record - Loaded Record 0001

DDS PRINT Output of INDEX Component:

*** TRACK(CCHH) 0001000D R0 DATA 0000000000000000
COUNT 0001000D01000200

0000	01F90301	00000000	00000000	01000044	01E401E4	302F2E2D 2C2B2A29	*.9.....U.U.....*
0020	28272625 24232221	201F1E1D 1C1B1A19	18171615 14131211	100F0E0D 0C0B0A09			*.....*
0040	08070605 04030201	00000000 00000000	00000000 00000000	00000000 00000000	00000000 00000000		*.....*
0060	00000000 00000000	00000000 00000000	00000000 00000000	00000000 00000000	00000000 00000000		*.....*
0080	TO 01DF SAME AS ABOVE						
01E0	00000000 00000000	00000000 00000000	0001F0F0 F0F10004	00001F9 01F90000			*.....0001.....*

01E0 00000000 00000000 00000000 00000000 00000000 00000000 00001F9 01F90000
F L P RDF CIDF



Note: Data set defined with one record per CI, key length = 4, CISIZE=512, 31 CIs/CA

Example of an Add to End CI Split — Insert Record 0002

DDS PRINT Output of INDEX Component:

*** TRACK(CCHH) 0001000D R0 DATA 0000000000000000

COUNT 0001000D01000200

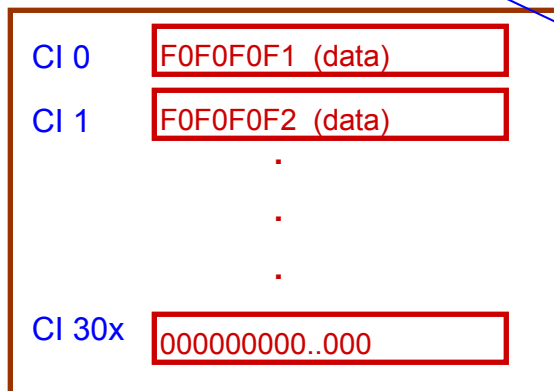
```
0000 01F90301 00000000 00000000 01000044 01E401E4 302F2E2D 2C2B2A29 *9.....U.U.....*
0020 28272625 24232221 201F1E1D 1C1B1A19 18171615 14131211 100F0E0D 0C0B0A09 *.....*
0040 08070605 04030200 00000000 00000000 00000000 00000000 00000000 *.....*
0060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0080 TO 01DF SAME AS ABOVE
01E0 00000000 00000000 00000000 00000000 0001F0F0 F0F10004 00001F9 01F90000 *.....0001.....*
```

01E0 00000000 00000000 00000000 00000000 0001F0F0 F0F10004 00001F9 01F90000

F L P (key)

F L P RDF CIDF

CA at RBA 0



CI 0 pointer in use.

CI 1 pointer in use.

Control Interval (CI) Reclaim (Prior to z/OS 1.12)

Control Interval (CI) Reclaim - Prior to z/OS 1.12

- ❑ VSAM allows existing records to be erased.
- ❑ When all records in a CI are erased, the CI is reclaimed by placing the CI pointer back on the free list in the index sequence set record.
- ❑ Current CI reclaim design holds data set level serialization.
- ❑ New CI splits can reuse the reclaimed CIs.
- ❑ **The last CI in the CA is not reclaimed even though all records in the CI are deleted.**
- ❑ The high key of the last CI is retained.
- ❑ Only record keys lower than the high key CI and greater than the high key of the previous CA can reuse the CA.
- ❑ CI Reclaims are not reported!

Example of a CI Reclaim — Erase Record 0001

DDS PRINT Output of INDEX Component:

*** TRACK(CCHH) 0001000D R0 DATA 0000000000000000

COUNT 0001000D01000200

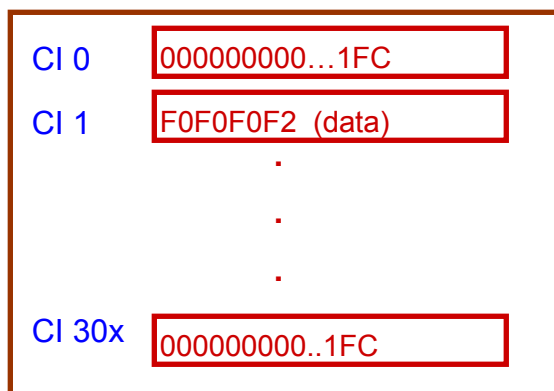
0000 01F90301 00000000 00000000 01000044 01E401E4 302F2E2D 2C2B2A29
0020 28272625 24232221 201F1E1D 1C1B1A19 18171615 14131211 100F0E0D 0C0B0A09
0040 08070605 04030200 00000000 00000000 00000000 00000000 00000000
0060 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0080 TO 01DF SAME AS ABOVE
01E0 00000000 00000000 00000000 00000000 0001F0F0 F0F10004 00001F9 01F90000

.9.....U.U.....
.....
.....
.....
.....0001.....

01E0 00000000 00000000 00000000 00000000 00000000 00000000 01001F9 01F90000

F L P RDF C1DF

CA at RBA 0



CI 0 pointer reclaimed.

Example Erase of Last Record in CA — Erase Record 0002

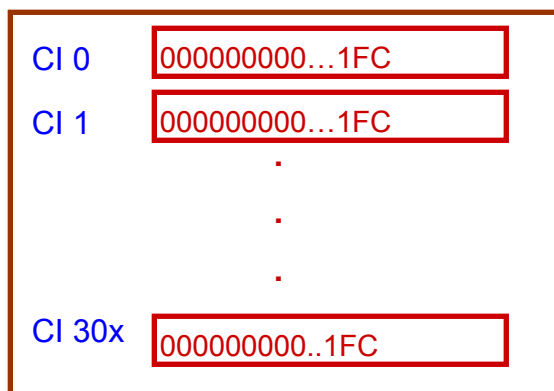
DDS PRINT Output of INDEX Component:

```
*** TRACK(CCHH) 0001000D    R0 DATA 0000000000000000
COUNT 0001000D01000200
0000 01F90301 00000000 00000000 01000044 01E401E4 302F2E2D 2C2B2A29 *9.....U.U.....*
0020 28272625 24232221 201F1E1D 1C1B1A19 18171615 14131211 100F0E0D 0C0B0A09 *.....*
0040 08070605 04030200 00000000 00000000 00000000 00000000 00000000 *.....*
0060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0080 TO 01DF SAME AS ABOVE
01E0 00000000 00000000 00000000 00000000 00000000 00000000 01001F9 01F90000 *.....*
```

01E0 00000000 00000000 00000000 00000000 00000000 00000000 01001F9 01F90000

F L P RDF CIDF

CA at RBA 0



← CI 1 not reclaimed!

Control Area (CA) Split (Prior to z/OS 1.12)

Control Area (CA) Split - Prior to z/OS 1.12

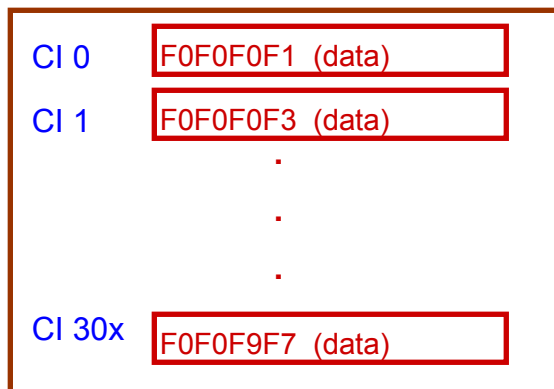
- ❑ When all the CIs in a CA are used, the next insert will result in a CI split which will trigger a CA split.
- ❑ A CA split will obtain a new CA (preformat it) and move half of the CIs from the old CA into the new CA.
- ❑ CA split may trigger an extend (EOV) if there are no more free CAs in the extent.
- ❑ A CA split will add a new index sequence set record and may also split a higher level index record. It may add a new index level if required.

Example of a Full CA — Inserted Odd Keys 0001-0097

DDS PRINT Output of INDEX Component:

```
*** TRACK(CCHH) 0001000D    R0 DATA 0000000000000000
COUNT 0001000D01000200
0000 01F90301 00000000 00000000 01000019 012401DB 00000000 00000000 *.9.....*
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0040 TO 011F SAME AS ABOVE
0120 000030F5 03012FF3 03012EF9 F102022D 03002CF7 03012BF5 03012A00 1FF8F302 *.5...3...91.....7...5....83.*
0140 0229F8F1 02022803 0027F703 0126F503 0125F303 0124F7F1 02022300 1FF60201 *.81.....7...5...3...71....6..*
0160 22F70301 21F50301 20F30301 1FF6F102 021E0300 1DF70301 1C001FF5 F502021B *.7...5...3...61.....7....55...*
0180 F303011A F5F10202 19030018 F7030117 F5030116 F3030115 001FF4F1 02021403 *.3...51.....7...5...3....41...*
01A0 0013F703 0112F503 0111F303 0110F3F1 02020F03 000E001E F2F70202 0DF50301 *.7...5...3...31.....27...5..*
01C0 0CF30301 0BF2F102 020A0300 09F70301 08F50301 070021F0 F0F1F300 0406F1F1 *.3...21.....7...5....0013...11*
01E0 02020503 0004F703 0103F503 0102F303 0101F0F0 F0F10004 000001F9 01F90000 *.7...5...3...0001.....9.9..*
```

CA at RBA 0



All CIs in use

Example of a CA Split — Insert Record 2

RBA 400 CI 2

```
0000 01F90507 00000000 00000000 00000000
      02000018 01E901F4 00000000 00000000

0020 00000000 00000000 00000000 00000000
      00000000 00000000 00000000 00000000

0040 TO 01DF SAME AS ABOVE

01E0 00000000 00000000 00000000 0001000B
      F0F0F4F7 00040000 000001F9 01F90000
```

RBA 0 CI 0

```
0000 01F90301 00000000 00000200 00000000 01000030 018001E7
      302F2E2D 2C2B2A29

0020 28272625 24232221 201F1E1D 1C1B1A19 00000000

0160 00000000 00000000 00000000 00000000 00000000 00000000
      00000000 000000F7

0180 030117F5 030116F3 03011500 0FF4F102 02140300 13F70301
      12000EF3 F5020211

01E0 030010F0 F0F0F500 0402F303 0101F203 0118F0F0 F0F10004
      000001F9 01F90000
```

RBA 200 CI 1

```
0000 01F90301 00006200 00000000 00000000 01000030
      017D01EB 302F2E2D 2C2B2A29

0020 28272625 24232221 201F1E1D 1C1B1A19 00000000
      00000000 00000000 00000000

0170 00000000 00000000 00000018

01E0 04F50301 030011F0 F0F5F300 0402F5F1 020201F0
      F0F40003 000001F9 01F90000
```

CA at RBA 0

CI 0	F0F0F0F1 (data)
CI 1	F0F0F0F3 (data)
..	
CI 17x	F0F0F4F7 (data)
CI 18x	F0F0F0F2 (data)
..	
CI 30x	00000000...1FC

← Record 2

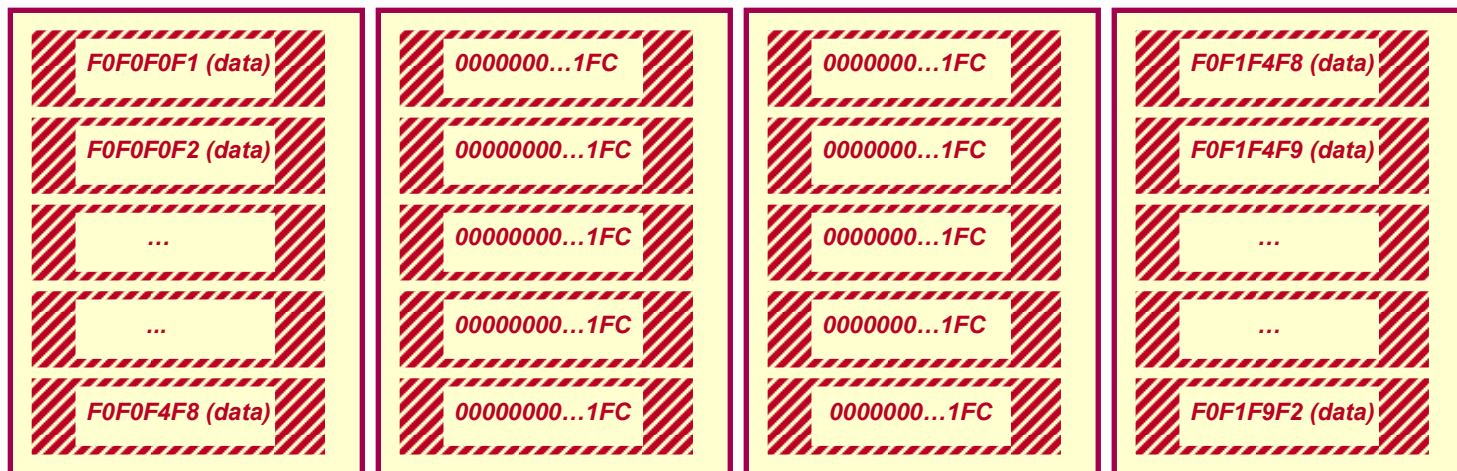
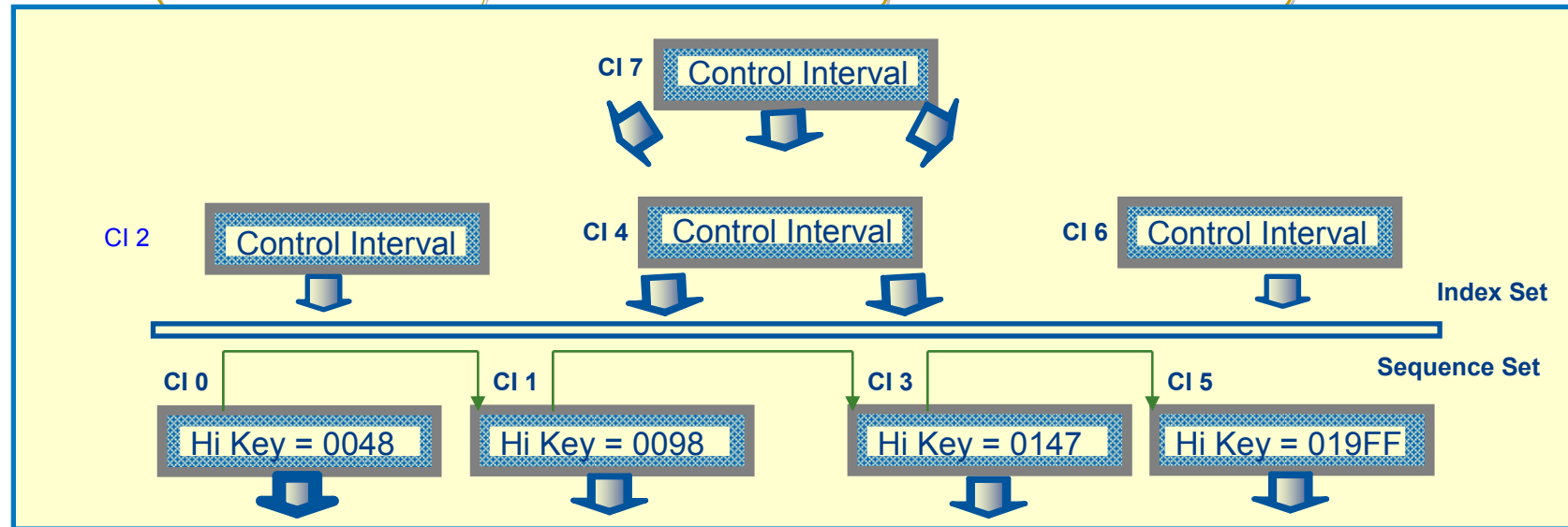
CA at RBA 6200x

CI 0	F0F0F4F9 (data)
CI 1	F0F0F5F1 (data)
..	
CI 18x	F0F0F9F7 (data)
..	
CI 30x	00000000...1FC

CI/CA Split and CI Reclaim (Prior to z/OS 1.12)

Example Of CI/CA Splits and CI Reclaims

(Inserted keys 0001 – 0192, erased 49-147)



CA 0

CA 1

CA 2

CA 3

"Fragmentation"



Index Component

Data Component

KSDS Fragmentation

KSDS Fragmentation

- ❑ Occurs in Key Sequence Data Sets (KSDSs).
- ❑ Occurs when erasing all the records in a CA.
- ❑ Due to a deficiency in VSAM KSDS space management.
- ❑ May be temporary or permanent.
- ❑ Can be reclaimed after the fact by reorganizing the data set:
 - ❑ Closing the KSDS
 - ❑ Unloading the KSDS to a backup file
 - ❑ Deleting and redefining the KSDS
 - ❑ Reloading the file from the backup file
 - ❑ Reopening the KSDS
- ❑ Can be prevented with Control Area Reclaim in z/OS 1.12.

Issues Related to KSDS Fragmentation

- ❑ Sequential access must read each sequence set index record and at least the last CI in the CA:
 - ❑ VSAM reads the last unclaimed CI to see if there are any records in it.
 - ❑ Results in two extra i/o's per empty CA.
- ❑ Empty CIs and their associated index records remain as valid buffers consuming your buffer pool.
- ❑ Data set uses more disk space than necessary to contain the records.
 - ❑ Results in possible outages due to extend failures.
- ❑ Extra index levels may be created as the data set grows.
 - ❑ Results in extra i/o's for each extra index level for direct access.
- ❑ Requires the data set to be taken offline to reorganize the space.

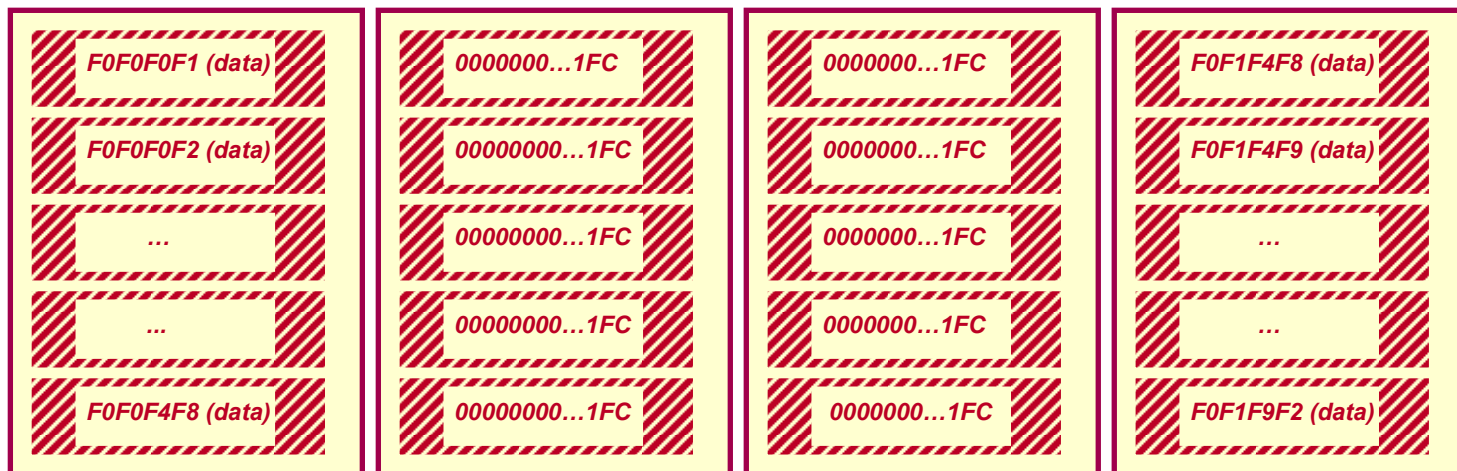
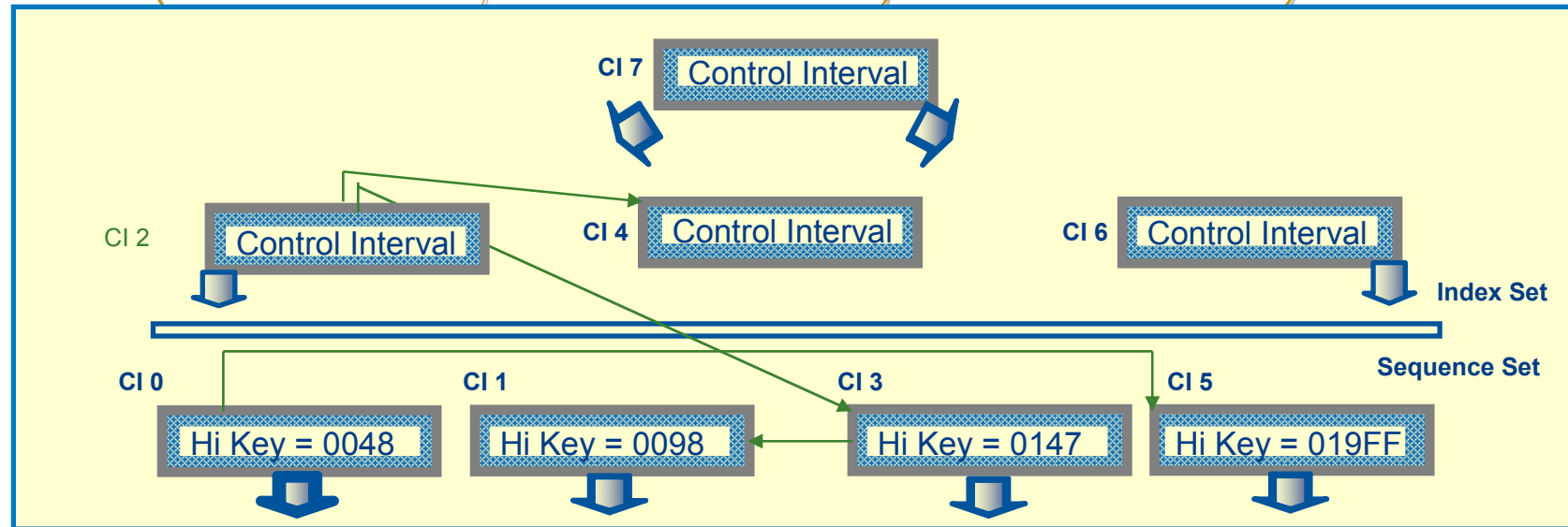
VSAM KSDS Space Management (z/OS 1.12)

VSAM KSDS Space Management (z/OS 1.12)

- ❑ New support applies to ALL KSDSs
 - ❑ Non SMS and SMS.
 - ❑ Catalogs.
 - ❑ AIXs.
 - ❑ Accessed via NSR, LSR, or RLS.
- ❑ Control Interval (CI) Split – no change.
- ❑ **Control Interval (CI) Reclaim** – VSAM will reclaim the last empty data CI in the sequence set record and trigger a CA Reclaim.
- ❑ **Control Area (CA) Reclaim** – VSAM will reclaim the empty sequence set index record and any empty higher level index records associated with empty sequence set record, by placing the index records on a free chain.
- ❑ **Control Area (CA) Split** – VSAM will check the free chain for reclaimed CAs before attempting to obtain a new CA. If a free CA exists it will be reused for the CA split.

CI/CA Reclaim/Reuse (z/OS 1.12)

Example Of CA Reclaim (CAs 1 and 2) (Inserted keys 0001 – 0192, erased 49-147)



CA 0

CA 1

CA 2

CA 3

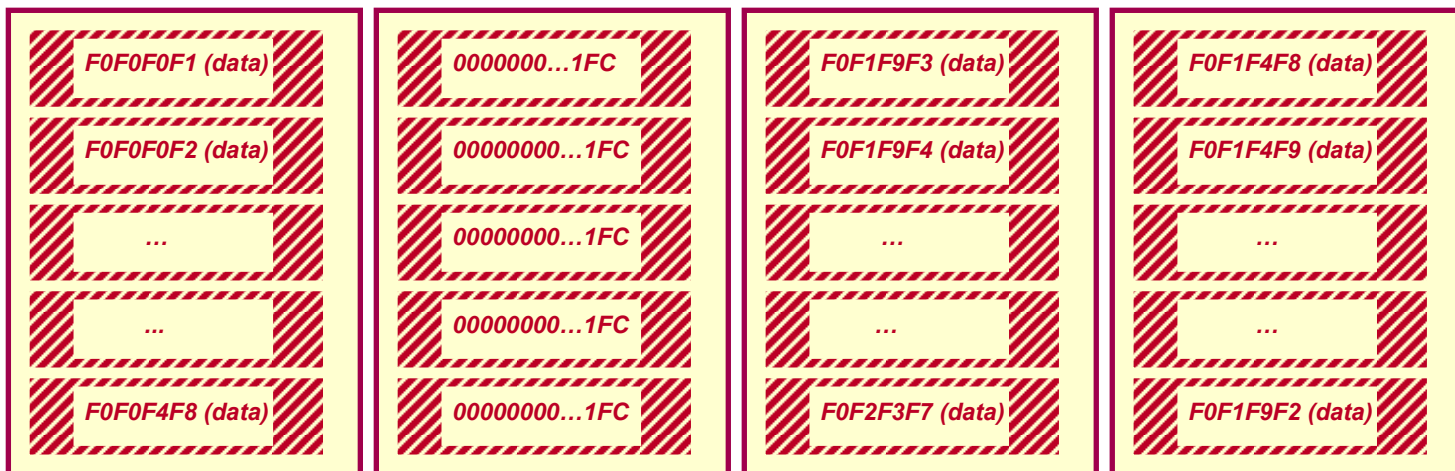
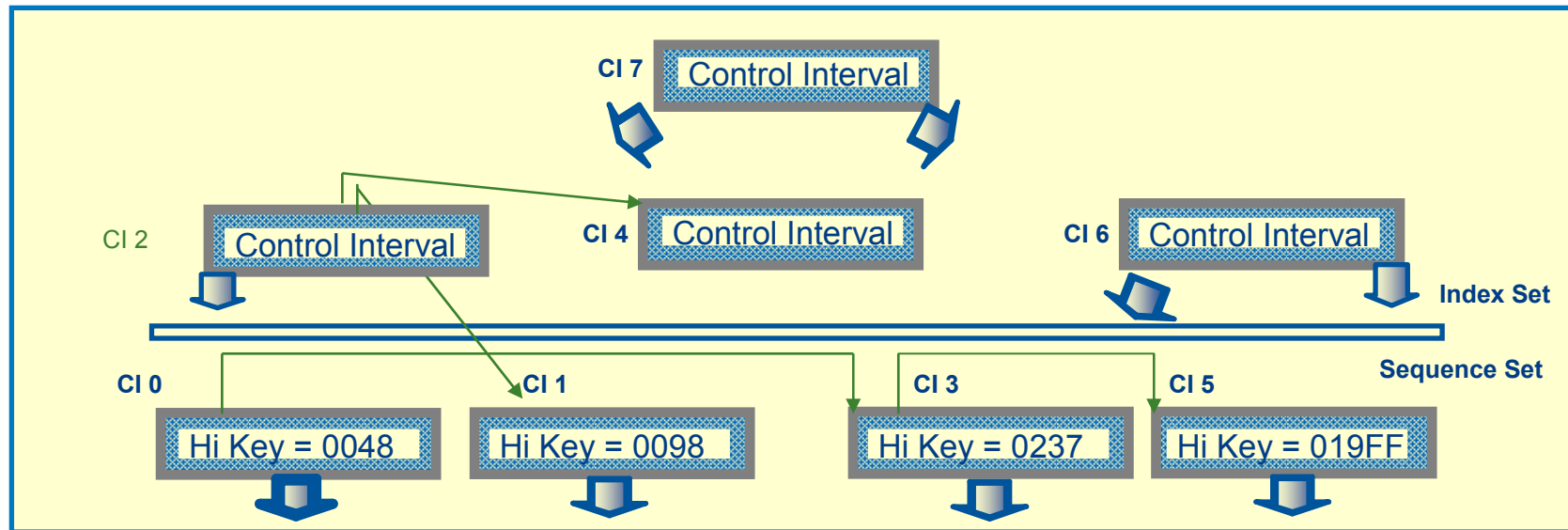
Note: Index CI 2, Data CA 0, and High Key Data CA are never reclaimed.



Index Component

Data Component

Example Of a CA Reuse (CA 2) - (Insert 0193-0237)



Enabling CA Reclaim (z/OS 1.12)

Enabling CA Reclaim

- ❑ Toleration maintenance.
- ❑ CA Reclaim parameters.
- ❑ Recommendations.

Enabling CA Reclaim – Toleration APARs

- ❑ Apply the following APARs on down level systems prior to enabling CA Reclaim on z/OS 1.12:
 - ❑ OA25108,OA32711
 - ❑ OA26256
 - ❑ OA26466
 - ❑ OA27557
 - ❑ OA33397
 - ❑ OA29142
 - ❑ OA37503

CA Reclaim - Parameters

- ❑ CA Reclaim can be enabled/disabled on a system level or on a data set level:
 - ❑ In z/OS 1.12 CA Reclaim is disabled by default on a system level basis.
 - ❑ In z/OS 1.12 CA Reclaim is enabled by default for individual KSDSs.

CA Reclaim – System Level Parameter

- ❑ New parameter in SYS1.PARMLIB(IGDSMSxx):
 - ❑ **CA_RECLAIM** (NONE/DATACLAS)
 - ❑ NONE is the default and indicates CA Reclaim is not active on this system.
 - ❑ **DATACLAS** indicates CA Reclaim is active at a system level and each data set's DATACLAS parm will dictate if CA Reclaim is active for the data set.
 - ❑ SETSMS **CA_RECLAIM()** or SET SMS=xx will dynamically set the CA Reclaim option on the system.

CA Reclaim – Data Set Level Parameters

- ❑ New DATACLAS parameter:
 - ❑ [CA_Reclaim\(YES/NO\)](#). Indicates if the file is enabled for CA Reclaim.
 - ❑ DATACLAS parm can be used with nonSMS data sets.
- ❑ New IDCAMS ALTER command:
 - ❑ [RECLAIMCA/NORECLAIMCA](#). Alters data set level CA Reclaim attribute.

CA Reclaim – Recommendations

- ❑ Enable CA Reclaim in your test environments:
 - ❑ Ensure test files have reclaimed CAs.
 - ❑ Test OEM products which access the test files with the reclaimed CAs.
- ❑ Prior to removing regular reorg jobs:
 - ❑ Install z/OS 1.12 on all LPARs.
 - ❑ Enable CA Reclaim on all LPARs.
 - ❑ Run one last reorg to remove any previously fragmented CAs, or run IDCAMS EXAMINE (from z/OS 1.12) to determine if unclaimed CAs exist.
- ❑ Refer to info apar II14640 for a Q&A type discussion on enabling CA Reclaim.

CA Reclaim Reporting and Recovery (z/OS 1.12)

CA Reclaim –Reporting (LISTCAT)

- ❑ **LISTCAT** will indicate:
 - ❑ If CA Reclaim is active for this data set.
 - ❑ Number of CAs reclaimed.
 - ❑ Number of CAs reused.
- ❑ **CAs reclaimed** are listed in the index component statistics under the “Records Deleted”.
- ❑ **CAs reused** are listed in the index component statistics under the “Records Inserted”.
- ❑ **CA Reclaims and reuses** will update the CA split count in the data component.

LISTCAT - CA Reclaim Status

SMSDATA

STORAGECLASS ---SXPXXS02 MANAGEMENTCLASS---(NULL)

DATACLASS -----KSCR000Y LBACKUP ---0000.000.0000

CA-RECLAIM-----**(YES)**

BWO STATUS-----00000000 BWO TIMESTAMP---00000 00:00:00.0

BWO----- (NULL)

LISTCAT – CAs Splits

DATA ----- KSDS.DATA

STATISTICS

REC-TOTAL-----93	SPLITS-CI-----0
REC-DELETED-----99	SPLITS-CA-----15

- SPLITS-CA include non add to end CA splits, reclaims and reuses.

LISTCAT – CAs Reclaimed/Reused

INDEX ----- KSDS.INDEX

STATISTICS

REC-TOTAL-----1359	SPLITS-CI-----0	EXCPS-----63
REC-DELETED-----10	SPLITS-CA-----0	EXTENTS-----1
REC-INSERTED-----5	FREESPACE-%CI-----10	SYSTEM-

TIMESTAMP:

REC-UPDATED-----0	FREESPACE-%CA-----10
-------------------	----------------------

X'C629B829F0A88201'

REC-RETRIEVED-----4077	FREESPC-----0
------------------------	---------------

- REC-DELETED = CAs reclaimed REC-INSERTED = CAs reused

CA Reclaim – Reporting SMF 64 Record

- 3 SMF64DAU FIXED(32); /* NUMBER OF TIMES A
CA WAS REUSED FROM
THE RECLAIM CHAIN*/

CA Reclaim – Reporting (EXAMINE)

- ❑ EXAMINE DATATEST:

- ❑ IDC01728I FOUND *nnnnnn* EMPTY CONTROL AREAS THAT HAVE NOT BEEN RECLAIMED.
- ❑ Use IDC01728I to determine existing fragmentation.

- ❑ EXAMINE INDEXEST:

- ❑ IDC11768I [CI SPLIT | [CA RECLAIM](#)] IN PROGRESS
- ❑ IDC01729I INDEX STRUCTURE NOT STANDARD -- CA RECLAIM COULD NOT BE DONE.

- ❑ EXAMINE DATATEST/INDEXTEST

- ❑ IDC11778I CA RECLAIM IN PROGRESS.
- ❑ IDC11724I DATA COMPONENT CA NOT KNOWN TO SEQUENCE SET.

CA Reclaim – Recovery VERIFY RECOVER

- New parameter for IDCAMS VERIFY:
 - VERIFY RECOVER. This keyword will recover interrupted CA Reclaim activity.
- The purpose of this new parameter is resolve temporary errors introduced by an interrupted CA Reclaim/Reuse prior to running IDCAMS EXAMINE.

CA Reclaim Performance Benchmarks

CA Reclaim – Benchmarks

- ❑ DFSMS Performance Benchmark
 - ❑ Boundary conditions.
- ❑ CICS Performance Benchmark: “**CICS & VSAM Control Area (CA) Reclaim**”

CA Reclaim Benchmarks - DFSMS

- Boundary Condition – Best Case:
 - Define and load a KSDS with 291285 records.
 - Delete all 291285 records.
 - Run EXAMINE DATATEST and INDEXTEST (tested separately)
 - EXAMINE uses VSAM NSR CINV access.

- DATATEST Results:

VSAM V1R12 CA_Reclaim AMS - Data	EXCP	CONN	Total CPU	CLOCK	SERV
Disable	414K	39100	0.158	2.912	9080K
Enable	2786	346	0.001	0.022	65238
Variance	-99.33%	-99.12%	-99.37%	-99.24%	-99.28%

- INDEXTEST Results:

VSAM V1R12 CA_Reclaim AMS - Index	EXCP	CONN	Total CPU	CLOCK	SERV
Disable	583	146	0.020	0.041	1451K
Enable	583	141	0.000	0.007	27562
Variance	0.00%	-3.42%	-100.00%	-82.93%	-98.10%

CA Reclaim Benchmarks - DFSMS

■ Boundary Condition – Worst Case:

- Define the worst possible KSDS (one record per CI, large key, few index CIs/CA)
- Insert and delete the same 500 keys 1000 times.
- RLS Results:

VSAM RLS CA Reclaim	EXCP	CONN	Total CPU	CLOCK	SERV
Disabled V1R12	3572K	159K	3.824	34.134	32433K
Enabled V1R12	4421K	172K	4.830	44.254	86919K
Variance	+23.77%	+8.18%	+26.31%	+29.65%	+168.00%

- NSR Results:

VSAM V1R12 CA_Reclaim	EXCP	CONN	Total CPU *	CLOCK	SERV
Disable	4145K	444K	1.706	31.667	19546K
Enable	5546K	612K	2.222	42.606	61208K
Variance	+33.80%	+37.84%	+30.25%	+34.54%	+213.15%

CA Reclaim Benchmarks - CICS

- CICS Performance Benchmark titled: “**CICS & VSAM Control Area (CA) Reclaim**”
- Authors: Arndt Eade, John Burgess, Hursley, UK Lab
- Ran 4 Tests comparing the benefits of CA Reclaim.

CA Reclaim Benchmarks – CICS (cont.)

Test # 1

The first test run involved running the STARTBR, 20 x READNEXT, ENDBR from a single transaction immediately after the file was Opened. This meant that none of the index or data buffers resided in the LSRPOOLS. DATASET1 originally contained records 0 – 9999999. Records 1000000 to 8000000 were then deleted. The STARTBR was positioned at record 999999 and then the first READNEXT issued.

No CA Reclaim:

DATASET1 results	Browses	20
Index EXCPs		374
Data EXCPs		372
Average CPU used		0.013909 seconds (13.9ms)
Average response time		0.335471 seconds (335.471 ms)

CA Reclaim:

DATASET2 results	Browses	20
Index EXCPs		1
Data EXCPs		2
Average CPU used		0.000526 seconds (.526 ms)
Average response time		0.001849 seconds (1.849 ms)

CA Reclaim Benchmarks – CICS (cont.)

Test #2

Similar to Test #1, but LSRPOOL is populated

Trans	Ave CPU	Ave Disp	#tasks	Av FCCount	Ave FC Browse	Ave Response
No CA Reclaim	0.000319	0.000342	19000	22	20	0.005611 (.56ms)
CA Reclaim	0.000087	0.000104	18583	22	20	0.001616 (.16ms)

Test #3

Regression test. Fragmented CAs not accessed. No change in performance.

CA Reclaim Benchmarks – CICS (cont.)

Test #4

Transactions insert, delete, read. Without CA Reclaim response time gradually increased.

Test ran for 5 hours with 14 SMF samples:

CR4A – No CA Reclaim

CR4B - CA Reclaim

	AVG Resp time		AVG CPUtime		AVG FC Wait Time		Number of tasks in sample Interval	
	CR4A	CR4B	CR4A	CR4B	CR4A	CR4B	CR4A	CR4B
1	0.002776	0.002574	0.000436	0.000435	0.002285	0.002085	602	602
2	0.002583	0.002347	0.000285	0.000247	0.002256	0.002059	3129	3130
3	0.002676	0.002311	0.000338	0.000243	0.002294	0.002027	2025	2025
4	0.002686	0.002346	0.000373	0.000247	0.002251	0.002057	2260	2260
5	0.002713	0.002359	0.000432	0.000243	0.002237	0.002075	2835	2836
6	0.002986	0.002396	0.000466	0.000232	0.002469	0.002124	3242	3243
7	0.002804	0.002404	0.000548	0.000238	0.002446	0.002124	3872	3873
8	0.002993	0.002434	0.000786	0.000283	0.002451	0.002105	8020	8021
9	0.002961	0.002335	0.000731	0.000240	0.002448	0.002053	5544	5544
10	0.002983	0.002456	0.000815	0.000245	0.003514	0.002167	1980	1980
11	0.210098	0.002901	0.007078	0.000290	0.202747	0.002565	1155	1155
12	0.377178	0.003932	0.012076	0.000370	0.364685	0.003508	337	338
13	0.373705	0.003945	0.012098	0.000374	0.361158	0.003519	291	292
14	0.408276	0.003783	0.012378	0.000384	0.395451	0.003348	203	204

Dataset without CA Reclaim grew 350% while dataset with CA Reclaim has not grown in size.

CA Reclaim Benchmarks – CICS (cont)

- ❑ **Conclusions:** CA RECLAIM will result in:
 - ❑ Lower and more consistent CPU and I/O activity when reading and browsing a VSAM file.
 - ❑ Where READ, DELETE and INSERT activity is occurring, the CPU and I/O costs will be lower and the rate at which the dataset increases in size will be significantly lower.

CA Reclaim “State of the Union”

CA Reclaim “State of the Union”

- ❑ Majority of the IBM z/OS customers are:
 - ❑ at z/OS 1.11 and in the process of rolling z/OS 1.13 into production.
 - ❑ at z/OS 1.12 and have finished the roll out into production.
- ❑ z/OS 1.12 customers have:
 - ❑ initiated CA Reclaim planning and migration.
 - ❑ enabled CA Reclaim in pre-production sysplexes.
 - ❑ have conducted full scale production tests in DR site tests.
 - ❑ have CA Reclaim enabled in production.

CA Reclaim “State of the Union” (cont.)

CA Reclaim APARs:

- OA35288 V0 ALTER NORECLAIMCA DOES NOT FUNCTION FULLY ON GENERIC REQUESTS (closed)
- OA36019 V0 INCORRECT VALUE USED IN CA RECLAIM RECOVERY PROCESSING (closed)
- OA36033 V0 NEW FUNCTION - CA RECLAIM CONTROL BLOCK BUILT DYNAMICALLY (closed)
- OA36891 V0 ABEND0C4 PIC4 IN IDA019RY (UA55391) DURING CA RECLAIM (closed)
- OA37082 V0 CHANGES TO EXAMINE IN SUPPORT OF OA36891
- OA37503 V0 CA RECLAIM FDBK 0108006D BROKEN INDEX IN MIXED RELEASE (closed)
- OA37519 V0 RECLAIMING HIGH KEY ENTRY RETURNING ERROR CODE 8. (closed)
- OA37635 V0 WITH CA RECLAIM ENABLED, FREESPACE EXCEEDS HI ALLOCATED SPACE (closed)
- OA37073 V0 ABEND0C4 PIC4 IN IDA019RY (UA55391) DURING CA RECLAIM (closed)
- OA37596 V0 CAS ABEND0C4 IDA019RX+730 /UA60512 LEVEL, REG A=ZERO (closed)
- OA38349 V0 HANG CAN OCCUR WHEN REUSE OF RECLAIMED CA'S OCCURS WITH (closed)

*HIPER

QUESTIONS ???