Telcordia Technologies

e-business Powered by IMS

IMS the world depends on it

HALDB User Experiences at Telcordia

Session C03

Steve Nathan
stephen.nathan@telcordia.com

An SAIC Company

RETURN TO INDEX

Disclaimer

- The purpose of this presentation is to provide a technical perspective of Telcordia's experience using IMS and HALDB.
- Although this document addresses certain IBM products, no endorsement of IBM or its products is expressed, and none should be inferred.
- Telcordia also makes no recommendation regarding the use or purchase of IMS products, any other IBM products, or any similar or comparable products.
- Telcordia does not endorse any products or suppliers. Telcordia does not recommend the purchase or use of any products by the participants.
- Each participant should evaluate this material and the products himself/herself.

Telcordia Technologies

Acknowledgements

- This presentation was made possible through the efforts of Vern Watts, Harley Beier, Claudia Ho, Rich Lewis, Dean Meltz and the other members of the HALDB, documentation and DBRC teams at the IMS Development Labs who worked very hard to make HALDB a useful product
- Thanks also to the IMS Level 2 team for all of their support

Telcordia Technologies

Presentation Outline

- Introduction – Telcordia
- HALDB Databases
- HALDB Partition Selection Exit
- HALDB Partitions
- HALDB Secondary Indexes and Logical Relationships
- HALDB Self-Healing Pointers
- HALDB and DBRC
- HALDB Naming Conventions
- HALDB Partition Initialization
- HALDB Load
- HALDB Utilities
- HALDB Partition Processing

Telcordia Technologies

Introduction - Telcordia

- Telcordia develops large IMS applications which among other things provision the telephone network
- The applications run in production at our clients – the regional telephone companies
- Our databases are not normally registered to DBRC in our development/test/training environments

Telcordia Technologies

Introduction – Telcordia

- One large application uses DEDB’s to partition the telephone data for availability
  - Data is partitioned by telephone exchange
  - The DEDB’s were developed before HALDB’s existed
- We had to migrate some additional function and data to this application from UNIX
  - We needed the same availability and partitioning
- We decided to use this as our opportunity to begin using HALDB’s
  - 62 UNIX files were converted to 7 HALDB’s
- Working with IBM we found many opportunities to enhance HALDB’s and their documentation

Telcordia Technologies

HALDB Databases

- HALDB – High Availability Large Data Base
- A new type of IMS Full Function (as opposed to Fast Path) partitioned database available in IMS 7.1
- Up to 1001 partitions per database
  - The architecture supports more
  - IBM could not easily test beyond 1001
- Up to 10 data set groups per partition
- Up to 4GB per data set (OSAM or VSAM)
- Approximately 40 Terabytes of data
  - Over 20,000 3390's
  - Over 6,000 bytes for each person on earth

Telcordia Technologies

HALDB Databases

- HALDB supports Partitioned HDAM (PHDAM) and Partitioned HIDAM (PHIDAM) database types
- HALDB supports OSAM and VSAM access methods
- HALDB supports Secondary Indexes
  - The Secondary Indexes are also partitioned
- HALDB supports some Logical Relationships

Telcordia Technologies

HALDB Databases

- The database defined by the DBD and in the IMS SYSGEN is the “Master Database”
- PHIDAM Primary Index definitions are automatic
  - There is no entry in the IMS SYSGEN
  - There is no DBD
  - There are no LCHILD statements in the PHIDAM database
- This means the Primary Index can not be read as a standalone database
  - Some applications did do this

Telcordia Technologies

HALDB Databases

- Determining which partition a root segment goes to is determined in one of two ways
  1. A high key can be defined for each partition with the HALDB Partition Definition Utility or batch DBRC definition
  2. A Partition Selection Exit (PSE) can determine the partition
     - The PSE is also invoked to determine the Next Partition for sequential access
       - Data does not have to be processed in Partition order
       - Not all partitions have to be processed
       - You can add a new Partition and logically process it in the middle
     - The PSE may also be invoked to determine the First Partition

Telcordia Technologies

HALDB Databases

- A “hybrid” database type is also supported
  - PHIDAM database with a Partition Selection Exit (PSE)
  - The PSE determines the partition for a database record in the same manner as a DEDB randomizer
  - Within the partition the roots are indexed and in key sequence
  - We used this “hybrid” for all of our HALDB databases
    - The HALDB’s are partitioned exactly the same as our DEDB’s
  - The data can easily be accessed sequentially
  - We do not have to allow nearly as much free space as with PHDAM

Telcordia Technologies

HALDB Databases

- The hierarchic structure of an IMS database is not changed
- An IMS database record (one root and all of its dependent segments) will only be in one partition
- Applications access HALDB databases with standard IMS calls
  - Availability status codes in the DBPCB are for the master database at schedule time (NA or NU)
  - A BA status code may be received for an IMS call due to a partition being unavailable

Telcordia Technologies

HALDB Partition Selection Exit

- The HALDB Partition Selection Exit is invoked for:
  - Structure Initialization
  - Structure Termination
  - Structure Modification
  - Select First Partition
  - Select Next Partition
  - Select Target Partition
- Passed 4 pre-chained save areas
- Passed a 512-byte reentrant work area
- Has a field which can be valued at Structure Initialization and passed to the other invocations
  - Allows a user table to be built and processed and freed

Telcordia Technologies

HALDB Partition Selection Exit

- In our DEDB randomizers if we do not like the key we return RC 4 in register 15
  - The application receives an FM status code
  - This capability was added to IMS Full Function randomizers in IMS 4.1
- When HALDB's were first designed a return code of 4 from the PSE would result in a pseudo-abend of the application
- We convinced IBM to change this to an FM status code
  - APAR PQ37561 (UQ47760)

Telcordia Technologies

HALDB Partition Selection Exit

- In our DEDB randomizers if we do not like the key we also WTO an error message describing what was wrong with the key
  - We also wanted to do this in the PSE
- The first time we tried it we abended the control region
  - The PSE had been invoked in cross-memory mode and SVC’s (e.g. WTO) are not allowed
- We had to add code to see if we were in cross-memory and if so use the Branch Entry WTO

HALDB Partition Selection Exit

```
         LA    R7,WTOMSG              SET ERROR MESSAGE TEXT
         BAL   R10,DOWTO              GO DO WTO

DOWTO    DS    0H
         LA     R5,PECUSER            ADDRESS OF 512-BYTE WORK AREA
         USING WTOD,R5                TELL ASSEMBLER
         MVC    WTOMSGE,WTOMSGL       MOVE MESSAGE MASK
         MVC    WTOTEXT,0(R7)         MOVE MESSAGE TEXT
         EPAR   R0                    PRIMARY ASID
         ESAR   R1                    SECONDARY ASID
         CLR    R0,R1                 PRIMARY = SECONDARY?
         BNE    DOWTOX                   NO - IN XM MODE
         WTO    MF=(E,WTOMSGE)        WTO TRACE MESSAGE
         BR     R10                   RETURN TO CALLER
DOWTOX   DS    0H
         WTO    MF=(E,WTOMSGE),LINKAGE=BRANCH
         BR     R10                   RETURN TO CALLER
         DROP   R5                    TELL ASSEMBLER
         EJECT
```

Telcordia Technologies

HALDB Partition Selection Exit

```
WTOMSGL  WTO   ‘                                             ',   X
               ROUTCDE=(11),DESC=(7),MF=L

WTOD     DSECT
WTOMSGE  DS   0CL54                  REENTRANT WTO MESSAGE
         DS    CL2                   TEXT LENGTH
         DS    CL2                   MCSFLAGS
WTOTEXT  DS    CL50
```

Telcordia Technologies

HALDB Partition Selection Exit

- Our application should never cross a partition boundary while processing
  - All processing for a unit of work is always within one partition
- We wanted to use the PSE to enforce this
  - It would have taken a great deal of code and dummy IMS segments to do this in the application
  - Whenever the PSE was called for the next partition it would say there was not another partition and the application would get a GB status code
  - However we had to process all the partitions for utility access
- In the PSE we needed to know which application was asking for the next partition

Telcordia Technologies

HALDB Partition Selection Exit

- The main control block passed to the PSE is the Partition Exit Communication Area (DFSPECA)
- There is no DSECT for this in the IMS Macro Library
  - It is only documented in the sample PSE exit (DFSPSE00)
  - I am hoping to get this fixed
  - This also applies to the other two PSE DSECTS
    - Partition Definition Area Prefix (DFSPDA)
    - Partition Definition Area Entry (DFSPDAE)

Telcordia Technologies

HALDB Partition Selection Exit

- The third field in the DFSPECA is coded as:
  - DS A RESERVED
- This field is really the PST address
  - I am hoping to get this documented
- Using the PST address you can determine which program is running and the environment in which it is running

HALDB Partition Selection Exit

```
         L     R8,PECPST          ADDRESS OF PST
         USING PST,R8             TELL ASSEMBLER
         L     R9,PSTSCDAD        ADDRESS OF SCD
         USING SCD,R9             TELL ASSEMBLER
         TM    SCDRGTYP,SCDRGBAT  BATCH OR UTILITY REGION?
         BNO   NRC12                 NO - NO NEXT PARTITION
         L     R10,SCDBPARM       ADDRESS OF PXPARMS
         USING DFSPRPX0,R10       TELL ASSEMBLER
         TM    RCTYP1,RCDLI       DLI REGION?
         BO    CHECKPGM              YES - CHECK PROGRAM
         TM    RCTYP1,RCPRE       DBB REGION?
         BNO   PSENEXT2              NO - GET NEXT PARTITION
CHECKPGM DS    0H
         CLC   RCPGM(2),=C‘xx'    OUR APPLICATION?
         BE    NRC12                 YES - NO NEXT PARTITION
         DROP  R8                 TELL ASSEMBLER
         DROP  R9                 TELL ASSEMBLER
         DROP  R10                TELL ASSEMBLER
*
PSENEXT2 DS    0H
*        FIND NEXT PARTITION
```

Telcordia Technologies

HALDB Partitions

- Each HALDB partition is independent
  - Allocation
  - Authorization
  - Reorganization
  - Recovery
  - Some commands

Telcordia Technologies

HALDB Partitions

- Each partition has a name
- There are many places where you have to refer to the Partition name where you would normally use a real database name
  - IMS commands
  - DBRC commands
  - Utility control cards

Telcordia Technologies

HALDB Partitions

- Commands can be issued against the Master database or against the partition
  - /START
  - /STOP
  - /DBD
  - /DBR
  - /DIS
  - /LOCK
  - /UNLOCK
- /STARTing a HALDB database does not start the partitions

Telcordia Technologies

HALDB Partitions

- The documentation is not clear on when the master database name should be used and when the partition name can/should be used
  - IMS documentation (Dean Meltz) has been extremely responsive in fixing these as I have found them
- The latest version of *The Complete IMS HALDB Guide* (Redbook SG-24-6945) has a detailed list of DBRC commands and which “database” name to use
  - This book is a “must” for HALDB users
  - Thanks to Jouko Jantti, Rich Lewis and Pat Schroeck for their instant response to my questions and suggestions

Telcordia Technologies

HALDB Secondary Indexes and Logical Relationships

- Partitioned Secondary Indexes (PSINDEX) are supported
  - We did not use these for our first implementation
  - We are looking to use them in the future
- Logical relationships are supported
  - We do not use any logical relationships
- “Self-healing pointers”
  - Reorganization of a partition does not require changes or reorganization of secondary indexes or logically related databases

Telcordia Technologies

Partitioned Secondary Indexes

- Secondary indexes for HALDB databases must also be partitioned – PSINDEX
- The partitioning is independent of the partitioning of the primary database
  - There can be a different number of partitions
  - One PSINDEX partition can contain records pointing to multiple primary database partitions
    - This could affect a timestamp recovery of a primary partition
    - A Partition Selection Exit on the PSINDEX database could force the partitioning to be the same but it would be very hard to do

Telcordia Technologies

Partitioned Secondary Indexes

- Secondary indexes for HALDB databases are created during the Initial Load of the target database
  - Work files are not created
  - No Prefix Resolution
  - No HISAM Unload/Reload
  - Segments are inserted randomly
    - This can greatly affect Initial Load performance!!!
- Secondary indexes for HALDB databases can be added using Secondary Index Utilities

Telcordia Technologies

Partitioned Secondary Indexes

- A PSINDEX segment is larger than the same INDEX segment for a non-HALDB secondary index
  - Contains 28-byte Extended Pointer Set instead of 4-byte RBA
  - Contains the target root key
  - /SX field is 8 bytes instead of 4
  - If you convert to HALDB from Full Function and you have programs which read the Secondary Index as a standalone database they may have to be changed
    - Telcordia does a lot of this

Telcordia Technologies

Self-Healing Pointers

- Each partition has a unique partition ID (PID)
  - If a partition is deleted in DBRC the partition ID will never be reused
  - You may have to take this into account in your Partition Selection Exit
  - APAR PQ48421 (IMS 7.1) or PQ73858 (IMS 8.1) allows a partition to be “disabled” instead of being deleted
- Each partition has a reorganization number
  - Stored in the partition
  - Incremented for each reorganization
- Each segment has a unique ID when created
  - Indirect List Entry Key (ILK)
  - 8 bytes
  - Stored in the segment prefix
  - Never changes for the life of the segment

Telcordia Technologies

Self-Healing Pointers

- Each partition has an Indirect List Data Set (ILDS) associated with it
  - This is a VSAM KSDS
- For each target segment (Secondary Index or Logical Relationship) a record called an Indirect List Entry (ILE) is stored in the ILDS during reorganization reload
  - The key of the ILE in the ILDS is the ILK
- There are no records in the ILDS after Initial Load
  - There are no records until the first reorganization

Telcordia Technologies

Self-Healing Pointers

- Secondary Indexes and logical relationships use an Extended Pointer Set (EPS) for finding the target segment
  - Partition Number
  - Reorganization Number
  - ILK
  - RBA Pointer
- Makes each segment bigger
- NOT updated during reorganizations

Telcordia Technologies

Self-Healing Pointers

- When accessing a secondary index target segment (or logical relationship) if the reorganization numbers in the pointer segment and the target database are the same then use the RBA pointer to access the segment directly
- If the reorganization numbers are not the same use the ILK to find the ILE in the ILDS
  - This has the current RBA pointer
  - If the PCB has update intent and the database access is update or exclusive then update the reorganization number and RBA in the EPS in the buffer but do not set the “Altered Buffer” flag
    - Self-healing pointers - if something else alters the buffer
    - I prefer a utility to mass heal all of the pointers

Self-Healing Pointers

- If a database has no Secondary Indexes and no Logical Relationships then it NEVER needs to use the ILDS – but IMS still requires it
- We opened an incident with IBM and got partial relief
- With IMS 7.1 APAR PQ61206 (UQ75705) IMS will not allocate the ILDS in the DLISAS region if it is not needed
  - This could result in the savings of 100's of allocated data sets
  - It still needs to be defined to VSAM
  - It will still be allocated by batch jobs
  - I am hoping to eliminate it altogether in the future

Telcordia Technologies

Self-Healing Pointers

- Database segment prefixes are larger for HALDB databases
- Database segments add the 8-byte ILK
  - 4-byte RBA
  - 2-byte Partition ID
  - 2-byte Reorganization Number
- All dependent segments will have a 4-byte Physical Parent Pointer

Telcordia Technologies

Self-Healing Pointers

- Secondary Index segment prefixes are larger for HALDB PSINDEX databases
- The 4-byte RBA pointer is replaced by a 28-byte Extended Pointer Set (EPS)
- The Root Key of the indexed segment is also stored in the segment prefix
- The /SX field in the sub-sequence field increases from a 4-byte RBA to an 8-byte ILK.

Telcordia Technologies

HALDB and DBRC

- HALDB databases MUST be defined to DBRC
  - Production
  - Training
  - Test
  - Development
- The database hierarchy (datasets, segments, fields, pointer options, secondary indexes, logical relationships) are defined in the DBD
- The partitions are defined in DBRC – not in the DBD
  - There are no AREA statements in the DBD

Telcordia Technologies

HALDB and DBRC

- This will be no fun for your development and test organizations
  - They will have to learn more about DBRC than they ever wanted to
    - Or your DBA's and SYSPROG's will be busier than they ever wanted to be
  - Some of the commands they may need to become familiar with:
    - CHANGE.DBDS DBD(part) DDN(part) ICOFF
    - CHANGE.DBDS DBD(part) DDN(part) NORECOV
    - CHANGE.DB DBD(part) NOBACK SSID(jobname)
    - CHANGE.DB DBD(part) UNAUTH SSID(jobname)
    - CHANGE.SUBSYS SSID(jobname) STARTRCV
    - CHANGE.SUBSYS SSID(jobname) ENDRECOV
    - DELETE.SUBSYS SSID(jobname)

Telcordia Technologies

HALDB and DBRC

- When HALDB was first announced the only way to define the partitions in DBRC was via the HALDB Partition Definition Utility
- This is a time-consuming ISPF application
- You would have had to go through several hundred ISPF panels to define 50 partitions for each of our 8 HALDB’s
- We told IBM right from the start that HALDB’s would not be usable at our customer sites without batch DBRC support for HALDB

Telcordia Technologies

HALDB and DBRC

- There is now batch DBRC support for HALDB
- For IMS 7.1 support was added via PQ35893 (UQ49705) for INIT.DB and INIT.PART so you could at least define the partitions with batch DBRC
  - There was also limited CHANGE.DB support
- IMS 8.1 added support for CHANGE.DB, CHANGE.PART, DELETE.DB and DELETE.PART for HALDB databases and partitions
  - This was propagated to IMS 7.1 with PQ59888 (UQ69393)

Telcordia Technologies

HALDB Naming Conventions

- DBD names are 8 characters
- DDNAMES are 7 characters + an IMS assigned suffix
  - xxxxxxxA – xxxxxxxJ for data set groups 1-10
  - xxxxxxxX for the primary index
  - xxxxxxxL for the ILDS data set
- Data set names are a user defined (37-byte maximum) prefix plus an IMS assigned 7 byte suffix - .A00001, .B00001, .X00001, .L00001

HALDB Naming Conventions

- If you display the partition in DBRC you will see data set ID's for the data sets
  - xxxxxxxA – DSID=0001
  - xxxxxxxL – DSID=0003
  - xxxxxxxX – DSID=0005
  - xxxxxxxB – DSID=0006
  - xxxxxxxC – DSID=0007
  - etc.
- DSID's 0002 and 0004 are used for the index components of the ILDS and Primary Index for I/O error purposes

Telcordia Technologies

HALDB Naming Conventions

- You have to plan ahead for naming conventions to make DDNAMES and data set names unique
  - Strongly consider using the DBD name or Partition name (or both) as a middle data set node
- These naming conventions also force the same high-level node for OSAM and VSAM data sets
  - Some organizations have standards that require different high-level nodes

HALDB Naming Conventions

- Database data set allocation is based on the data set names in DBRC
  - The user prefix defined in DBRC + the 7-byte IMS suffix
- There are no DD CARDS
- There are no DFSMDA dynamic allocation members

HALDB Partition Initialization

- HALDB database partitions are initialized in one of two ways
  - HALDB Partition Data Set Initialization Utility (DFSUPNT0)
    - Sounds great
    - Not very user friendly
    - Use with caution
      - Hopefully will be enhanced in the future
  - IMS Database Prereorganization Utility (DFSURPR0)
    - DBIL=Master DBD name

Telcordia Technologies

HALDB Partition Initialization

- When the HALDB partitions are first defined to DBRC each partition has a status of PINIT
  - Partition Initialization is required
- When the IMS Prereorg Utility is run for a database it will initialize all partitions in a PINIT status
  - This means all partitions have to be allocated
  - This may not be what is wanted for development or test or training databases

Telcordia Technologies

HALDB Partition Initialization

- Our HALDB DBRC definition procedure is run once per database and has three steps
  - Define the database
    - INIT.DB DBD(xxxxxxxx)
  - Define all of the HALDB partitions
    - INIT.PART
  - Set all of the HALDB partitions to NOPINIT
    - CHANGE.DB DBD(xxxxxxxx) NOPINIT

Telcordia Technologies

HALDB Partition Initialization

- Our HALDB initialization procedure is run once per partition and has three steps
  - Step 1
    - OSAM delete/define the partition
    - VSAM delete/define the Primary Index
    - VSAM delete/define the ILDS
  - Step 2
    - DBRC utility to set the partition to PINIT
  - Step 3
    - IMS Prereorganization Utility
      - This will initialize only the PINIT partition

Telcordia Technologies

HALDB Partition Initialization

- Our HALDB initialization procedure is prone to error
  - We have had many different scenarios where the jobs failed or running two jobs simultaneously caused errors
    - Identifying the error was not easy
    - Recovering from the error was not easy
- IBM has hinted that this area will be improved in the future
  - We really need to be able to tell IMS to initialize a specific partition for a specific HALDB database

Telcordia Technologies

HALDB Partition Initialization

- After the HALDB partition has been initialized the Image Copy Needed flag is on
- The partition may have to be Image Copied in its initialized state
  - More later
- DBRC GENJCL.IC can be done with the Master database name and will generate Image Copy steps for all partitions
- DBRC GENJCL.IC can also be done by partition

Telcordia Technologies

HALDB Database Load

- The manual states that HALDB's are loaded with a PROCOPT=L/LS PCB
- This is NOT required
- After the partition has been initialized it can be loaded with PROCOPT=L/LS
- If the partition has been initialized and Image Copied it can be loaded with either PROCOPT=L/LS or PROCOPT=A/I
- For large amounts of data PROCOPT=L/LS will be much more efficient
  - Remember the Secondary Index performance impact

Telcordia Technologies

HALDB Database Load

- When we first loaded our PHIDAM databases in load mode (PROCOPT=LS) we got an 0C4 in DFSDLOC0
  - This was resolved by PQ73193 (PQ73731 for IMS 8.1)
- This was followed by an 0C4 in DFSDDLE0
  - APAR PQ73700 was opened but will be closed as documentation
  - We were using PARM=DBB to load the PHIDAM database
  - But the block builder does not have enough information at ACBGEN time to build the blocks correctly
  - It will be documented that loading HALDB databases in Load Mode (PROCOPT=L/LS) will require PARM=DLI

Telcordia Technologies

HALDB Image Copy

- Image Copy for HALDB is supported by all current IMS Image Copy products
- Image Copy is by partition
  - DD cards must be included in the JCL
    - May change in IMS V9
- GENJCL.IC DBD(masterdb) generates JCL for all data set groups for all HALDB partitions
- GENJCL.IC DBD(partname) generates JCL for all data set groups for one HALDB partition
- GENJCL.IC DBD(partname) DDN(dsname) generates JCL for one data set group for one HALDB partition

Telcordia Technologies

HALDB Recovery

- Recovery for HALDB is supported by all current IMS Recovery products
- Recovery is by partition
  - DD cards must be included in the JCL
    - May change in IMS V9
- DBRC GENJCL support is the same as for Image Copy
- PHIDAM Primary Indexes and the ILDS are not recovered (or Image Copied)
  - They are rebuilt using the DFSPREC0 utility
  - DBRC support for this requires GENJCL.USER

Telcordia Technologies

HALDB Reorganization

- Reorganization for HALDB is supported by some current IMS Full Function Reorganization products
  - Standard IMS Unload/Reload
  - IBM IMS HP Unload and HP Load
  - BMC
  - Neon
- Reorganization is by partition
  - DD cards are not included in the JCL
- PHIDAM Primary Index is rebuilt
- ILDS is updated

HALDB Reorganization Reload

- HALDB Reorganization Reload is easier than Standard Full Function reload
- The partitions do not have to be deleted and defined
  - OSAM data sets are overwritten
  - VSAM data sets use the REUSE option
- If you do delete/define to change or move the data sets it is not more work
  - You do not have to set the partition to PINIT
  - You do not have to reinitialize the partition with the Prereorg utility
  - Initialization is only required for initial load and migration reload
    - Every partition has to be initialized once

Telcordia Technologies

HALDB Reorganization Reload

- HALDB Reorganization Reload is easier than Standard Full Function reload
- Secondary indexes and logically related databases DO NOT have to be reorganized
  - No work data set
  - No Prefix Resolution
  - No HISAM unload/reload
  - No Prefix Update
  - The Self-healing pointers take care of this

Telcordia Technologies

HALDB Reorganization Reload

- When using the HD reload utility during database migration (MIGRATE=YES or MIGRATX=YES during HD Unload) be sure to specify the ILDSMULTI keyword
  - IMS V7 PQ36991
  - IMS V8 PQ54227
- This will write the ILDS records to a dataspace during reload
  - They are then sorted and inserted using VSAM load mode

Telcordia Technologies

Reorganizing HALDB PSINDEX

- HALDB Secondary Index databases (PSINDEX) are reorganized using the HD unload/reload utilities
  - DFSURGU0 – HD Unload
  - DFSURGL0 – HD Reload
- The HISAM unload/reload utilities do not support PSINDEX
  - DFSURUL0 – HISAM Unload
  - DFSURRL0 – HISAM Reload

Telcordia Technologies

HALDB Pointer Checker – Free Space Analysis

- Pointer Checker/Free Space Analysis is supported by some products
  - IBM HP Pointer Checker
  - BMC Pointer Checker Plus
  - Neon iCheck

Telcordia Technologies

HALDB Database Buffer Pools

- HALDB database data sets are assigned to OSAM and VSAM buffer pools using the DFSVSMxx member just like Full Function databases
- If the Master database name is used all partitions are assigned to the same subpool
- Individual partitions may be assigned to their own subpools

Telcordia Technologies

HALDB Partition Processing

- When HALDB's were first designed there was no way to run a batch or BMP job against just one partition or a list of partitions
  - This was available for DEDB's using the DFSCTL control cards
- We asked for the same processing capability for HALDB

Telcordia Technologies

HALDB Partition Processing

- The first attempt to fix this was via PQ57313 (UQ65876)
  - PQ58600 (UQ67984) for IMS 8.1
  - It introduced the DFSHALDB DD card to input the parameters
  - You were able to restrict all calls for a DBPCB to a single HALDB partition
  - Unfortunately you had to specify the PCB as a relative PCB number

HALDB Partition Processing

- The second attempt to fix this was via PQ65486 (UQ73331)
  - PQ65489 (UQ73332) for IMS 8.1
  - This allowed the specification of a PCB label instead of the relative PCB number
  - You are still restricted to one HALDB partition
- I hope to get a list of partitions some time in the future

Telcordia Technologies

Conclusion

- HALDB's offer support for partitioning Full Function databases for size and availability
- IBM will offer enhancements to HALDB's in the future
  - Online reorganization
- As with anything new – there is a learning curve

Telcordia Technologies

RETURN TO INDEX