IBM Software Group

HALDB – test and production realities

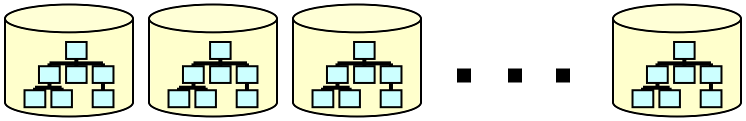
Janet LeBlanc
Silicon Valley Lab
leblancj@ca.ibm.com

e business on demand software

Agenda

- HALDB Test Environments
  - Creating Test Databases & DBRC entries
  - DBRC for every programmer
  - One DBRC for all test environments
- Initial Loads HALDB databases
- Applications recognizing partitions
- Maintaining HALDB DBs
- Finding and resolving problems
- Migration to HALDB

HALDB Overview



- Databases are partitioned
  - Up to 1001 partitions per database
  - Partitions have up to 10 data set groups
  - Partitions may be allocated, authorized, and reorganized independently
- Database Types
  - Partitioned HDAM (PHDAM)
  - Partitioned HIDAM (PHIDAM)
  - Partitioned Secondary Index (PSINDEX)
- Self healing pointers
  - Reorganization of partition does not require changes to secondary indexes or logically related databases which point to it

Test Databases

- **Non-HALDB test databases**
  - Often, not registered in RECONs
  - Each programmer may have one or more versions of a database

- **All HALDB databases are registered in RECONs**
  - Multiple versions of a database must be defined in different RECONs
  - DBRC does not allow multiple databases with the same name
  - Multiple test versions of a database require multiple RECONs
  - Plan your batch test environments

Defining Test Environments

- **Use the same DBD as production**
  - DBD does not include partition or data set information
  - Place in test DBDLIB and ACBLIB

- **Create test partition definitions**
  - Define partitions for test environment

    or
  - Use Partition Definition Utility EXPORT and IMPORT functions
  - Moves partition definitions between RECONs
  - They may be modified after IMPORT
  - Data set name prefix, RAA, etc.

HALDB and DBRC

- **Transport a DB to a different IMS**
  - Unload/reload
  - Application load
  - Image copy recovery
  - DFDSS

HALDB and DBRC

- **Transport a DB to a different IMS**
  - Via unload/reload
    - Redefine partition definition in target RECON
      - Partition definition may be different
    - Reset ILK required
      - If DB copy is permanent
- HALDB TOOLKIT:
      - Create a copy of the partition definition
      - Use ILK reset utility for unload

IHCHALDB

```
COPYDBRC DBD(ddddd)
INCLIND(Y)
INCLIC(Y)
TODD(OUTDBRC) )
```

IHCHALDB

```
RESETILK DBD(ddddd)
```

HALDB and DBRC

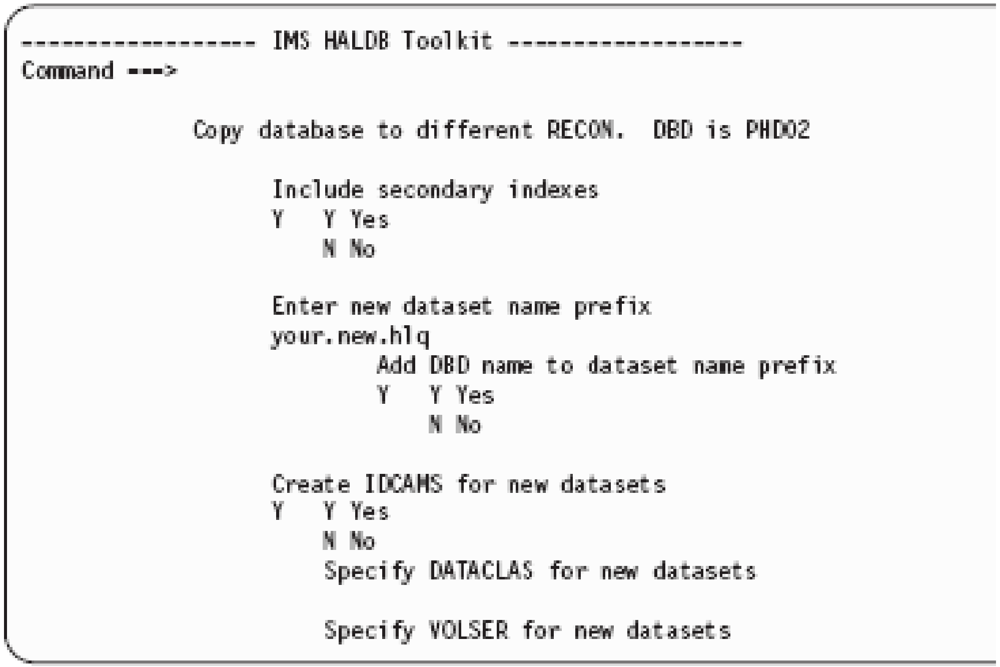
- **Transport** via application program
  - Uses PSB
  - Similar to unload/reload
  - No ILK problems
  - TOOLKIT:
    - Create a copy of the partition definition
    - Use HALDB load support to ease PSINDEX performance

IHCHALDB

COPYDBRC DBD(*ddddd*)
INCLIND(Y)
INCLIC(Y)
TODD(OUTDBRC))

Copy via image copy

- Partition definition must be identical
  - Target DBRC must have deleted partitions
  - IC DBRC definition must be transported
  - Change of DSNPREFX
  - Allocation of new target datasets
- Recover using target RECON



```
------------------ IMS HALDB Toolkit ------------------
Command --->

             Copy database to different RECON.  DBD is PHD02

                   Include secondary indexes
                   Y   Y Yes
                       N No

                   Enter new dataset name prefix
                   your.new.hlq
                           Add DBD name to dataset name prefix
                           Y   Y Yes
                               N No

                   Create IDCAMS for new datasets
                   Y   Y Yes
                       N No
                       Specify DATACLAS for new datasets

                       Specify VOLSER for new datasets
```

TOOLKIT:

- Create the partition definition for the target RECON
- Copy the image copy definition to the target RECON
- Create the IDCAMS statements to allocate the new files

Copy via DFDSS

- Via DFDSS
  - Similar to image copy
  - Target RECON must reflect deleted partitions
  - Database must be /DBRed

- TOOLKIT:
  - Create the partition definition for the target RECON
  - Create the DFDSS commands.

HALDB and DBRC

```
//ALLO EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE my.test.RECON1
DELETE my.test.RECON2
DELETE my.test.RECON3
SET MAXCC=0
DEFINE CLUSTER(NAME(my.test.RECON1) …
DEFINE CLUSTER(NAME(my.test.RECON2) ….
DEFINE CLUSTER(NAME(my.test.RECON3) ….
```

```
//CRE EXEC PGM=IHCHALDB, REGION=60M
//STEPLIB           DD DISP=SHR,DSN=IHC320.SIHCLOAD
//                  DD DISP=SHR,DSN=IMS910.SDFSRESL
//DFSRESLB DD DISP=SHR,DSN=IMS910.SDFSRESL
//IMS DD DISP=SHR,DSN=my.DBDLIB
//RECON1            DD DISP=SHR,DSN=SOURCE.RECON1
//RECON2            DD DISP=SHR,DSN=SOURCE.RECON2
//RECON3            DD DISP=SHR,DSN=SOURCE.RECON3
//MSGPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//IHCSYSIN DD *
COPYDBRC DBD(nnnnn) –
DSNPREF(new.hlq) –
DSNDBD(DBD) - INCLIND(YES) –
TODD(DBRCOUT)
/*
//DBRCOUT DD DISP=(,PASS),SPACE=(TRK,(1,1)),UNIT=SYSALLDA
```

```
//DBRC EXEC PGM=DSPURX00,REGION=50M,COND=(4,LE)
//STEPLIB DD DISP=SHR,DSN=IMS910.SDFSRESL
//RECON1 DD DISP=SHR,DSN=my.test.RECON1
//RECON2 DD DISP=SHR,DSN=my.test.RECON2
//RECON3 DD DISP=SHR,DSN=my.test.RECON3
//DFSRESLB DD DISP=SHR,DSN=IMS910.SDFSRESL
//IMS DD DISP=SHR,DSN=your.DBDLIB
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
INIT.RECON NOFORCER CATDS TAPEUNIT(3480)
//        DD DISP=(OLD,DELETE),DSN=*.CRE.DBRCOUT
```

- **Test Databases**
  - RECON for everybody
    - Batch testing or BTS testing
    - Required when using different database datasets
    - DBRC knowledge required
  - TOOLKIT:
    - Clone partition definition
      - Exchange DSNPREFX
    - Rebuild RECON each time

HALDB and DBRC

- **Test Databases**
  - Using only one RECON
    - Good enough for testing online
      - Need to deal with DB status in DBRC
  - Better solution required

HALDB and DBRC

- **Simulating DBRC=N**
  - Using one RECON only
    - RECON holds HALDB definition
      - Partition names/numbers, key ranges, file attributes
    - DSNPREFX is substituted
      - Batch/BTS applications
      - Rules can be defined to create DSNPREFX
        - Via ISPF interface
        - Via batch update utility
    - Status flags in DBRC are ignored
      - You can process a HALDB partition when IC needed is on

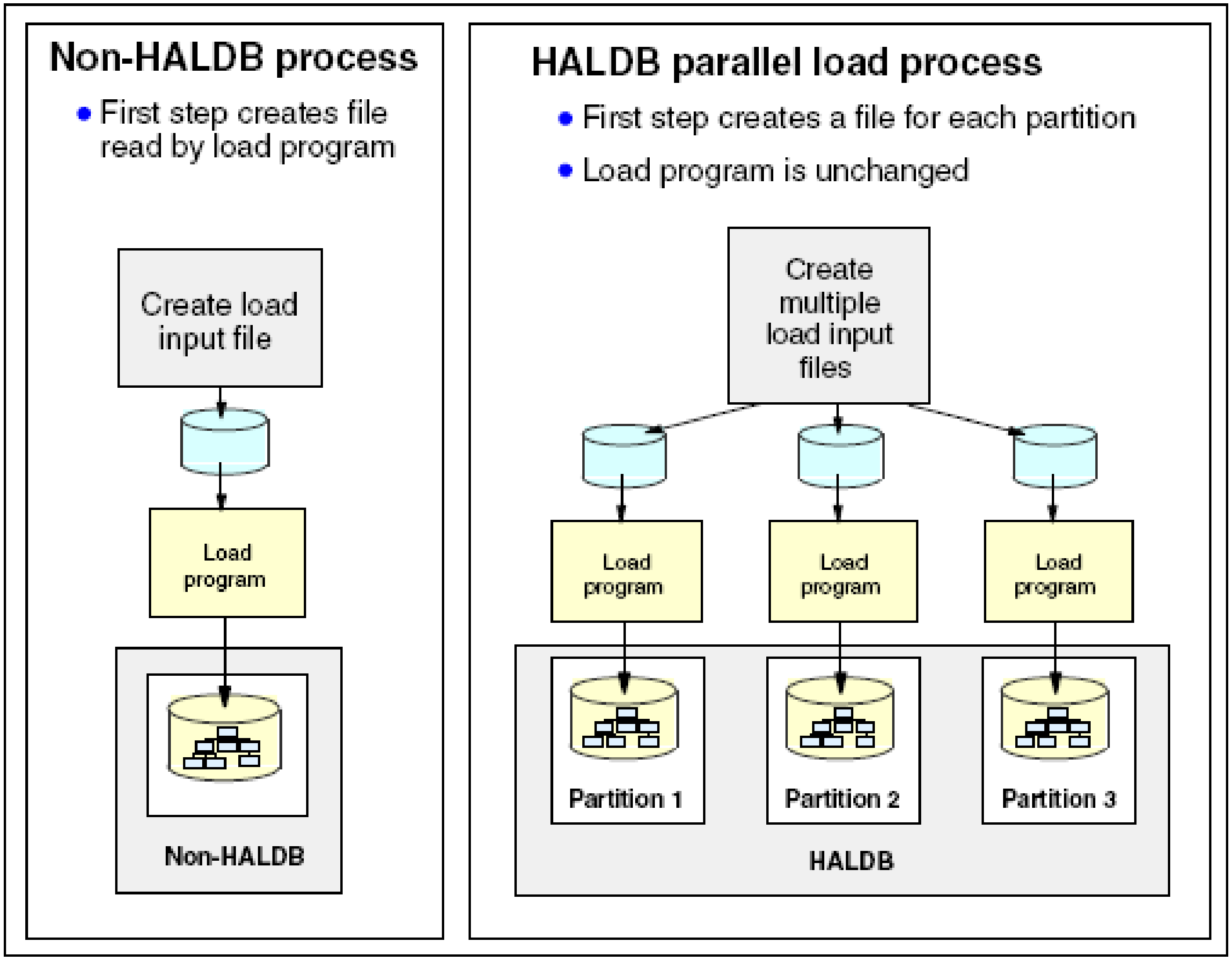
```
------------------------       HALDB Toolkit       ------------------------
Command ===>

                         SYSTEM Definition for IVP1

 System description.....: IBM test

 System status
           Y  Intercept active
 System settings
           Y  DSNPREFX is substituted
           Y  DSNPREFX is created by DSNRULES
                     Y  Rules are on SYSTEM level
                        Rules are on USER level
                        Rules are on DBD level

 Dataset name (DSNPREFX) creation rules:
              SYS.......: IMSP
              HLQ.......: DBSMS.CKIHC
              Rule......: HLQ  DBD                     (HLQ,SYS,USER,DBD,PART)
              Result....: DBSMS.CKIHC.dbd.A00001
```

HALDB and DBRC

- **Backup HALDB definitions**
  - Physical description is kept in DBRC
  - DELETE.DB “kills” the database
    - No clue of physical file definition
    - Even available ICs are of no use
  - TOOLKIT:
    - Utility to recreate DBRC partition definition from current
    - Should be used each time the partitions change

Applications & HALDB

Parallel Loads



Non-HALDB process

- First step creates file read by load program

HALDB parallel load process

- First step creates a file for each partition
- Load program is unchanged

Splitting an unload file

```
//S2         EXEC PGM=IHCHALDB,
//                  REGION=80M
//STEPLIB      DD DISP=SHR,DSN=IHC320.SIHCLOAD
//             DD DISP=SHR,DSN=IMS910.SDFSRESL
//RECON1       DD DISP=SHR,DSN=your.RECON1
//RECON2       DD DISP=SHR,DSN=your.RECON2
//RECON3       DD DISP=SHR,DSN=your.RECON3
//IMS          DD DISP=SHR,DSN=your.DBDLIB
//TRACE        DD SYSOUT=*
//SYSPRINT     DD SYSOUT=*
//SYSABEND     DD SYSOUT=*
//MSGPRINT     DD SYSOUT=*
//IHCSYSIN     DD *
   RUN       PGM(IHCUSPLT) –
             DBD(nnnnn)
//DFSUINPT     DD DISP=SHR,DSN=your.hd.unload.dataset
//PART01Z      DD DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,(50,20)),
//                  DSN=&&PART01
//PART02Z      DD DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,(50,20)),
//                  DSN=&&PART02
```

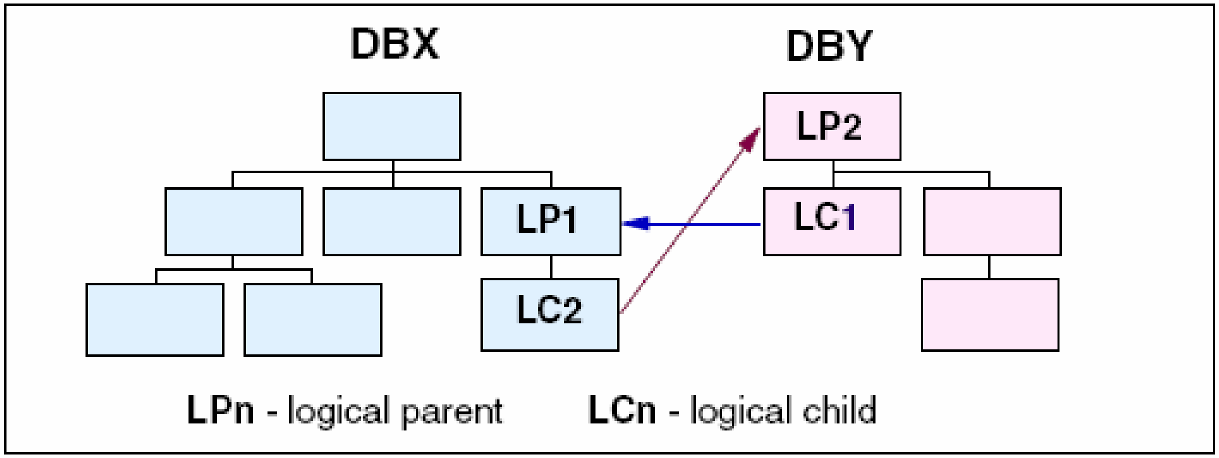
Program Loads with Secondary Indexes

- Initial load of a database with secondary indexes also creates the secondary indexes
- Index partitions must be initialized prior the load
- Bufferpools must include the buffers for the indexes
- Initial loads do not create entries in the ILDS
- Building of secondary indexes during the load can be turned off with a parameter in DFSVSAMP DD

```
DFSVSAMP   DD *
OPTIONS,BLDSNDX=NO
VSRBF=4096,500
IOBF=(8192,200)
/*
```

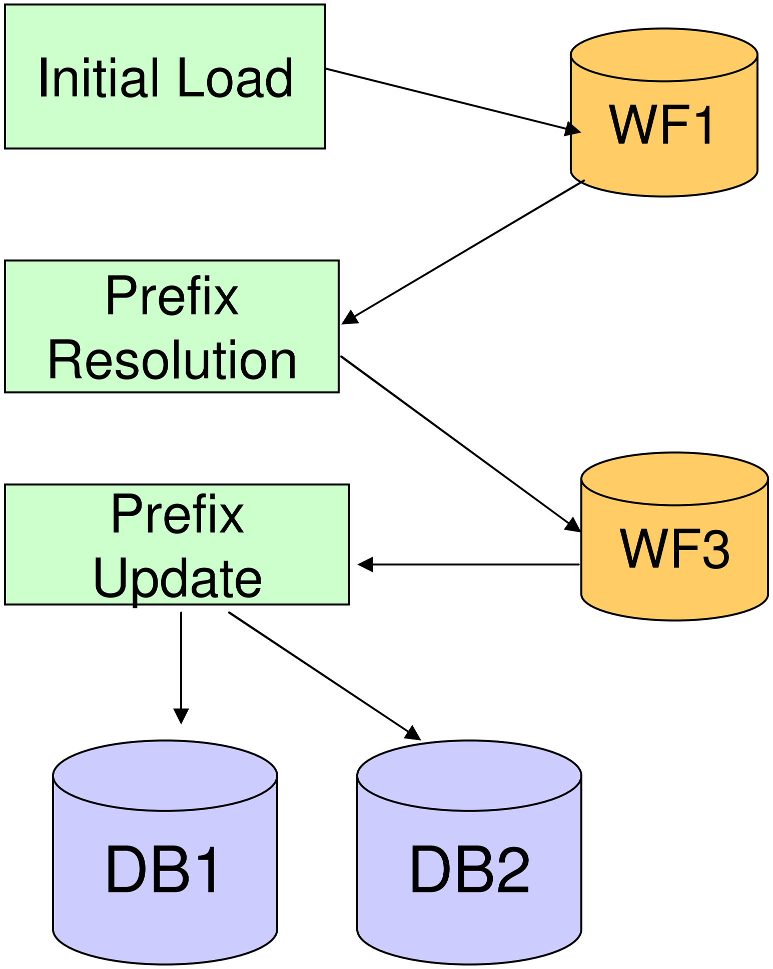
Program Loads with Logical Relationships

- Execute the Prereorganization utility to initialize the partitions in DBX.
- Execute the Prereorganization utility to initialize the partitions in DBY.
- Execute the initial load application program for DBX. Do not load logical children LC2. The program does not create a work file.
- Execute the initial load application program for DBY. Do not load logical children LC1. The program does not create a work file.
- Execute an update application program for a logical database. Insert either logical children LC1 or LC2, but not both. The insertion of either of the paired segments will create its paired logical child.

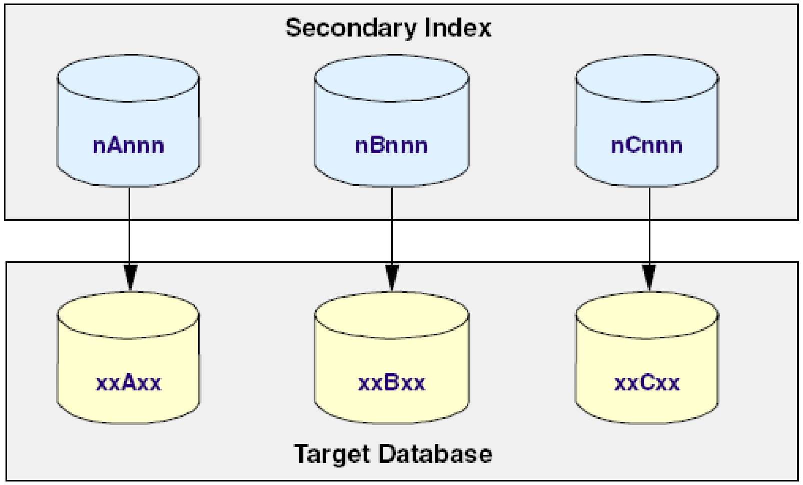


HALDB Toolkit - Application Support

- **Loading a logical related database**
  - Tool "delays" the inserts
    - Are written to WF1 file
    - Are inserted after load has completed
      - Reuse the prefix update step.
    - IC needed turned on at end
    - No logtape registered



Applications & HALDB Partitioning



- Selective Partition Processing

```
DFSHALDB DD *
HALDB PCB=(4,POHIDKA)
HALDB PCB=(PCBNUM2,POHIDJA)
```

```
DFSHALDB DD *
HALDB PCB=(3,PVHDJ5A,NUM=4)
HALDB PCB=(PCBNUM7,PVHDJ5B,NUM=3)
```

- Issue – you need to know which partition a key belongs to – it can change

Application Support

Application Programs

- **DFSHALDB statement generator:**
  - Provide starting key and ending key
  - In subsequent application step, you specify:

```
//S2          EXEC PGM=IHCHALDB,TIME=(1,20),
//                 REGION=80M
//STEPLIB          DD DISP=SHR,DSN=IHC320.SIHCL
//                 DD DISP=SHR,DSN=IMS910.SDFSRESL
//RECON1           DD DISP=SHR,DSN=IMSPLXA.IMS91.RECON1
//RECON2           DD DISP=SHR,DSN=IMSPLXA.IMS91.RECON2
//RECON3           DD DISP=SHR,DSN=IMSPLXA.IMS91.RECON3
//IMS              DD DISP=SHR,DSN=IMSA.DBDLIB
//$$DYNTR          DD *
//TRACE            DD SYSOUT=*
//SYSPRINT         DD SYSOUT=*
//SYSABEND         DD SYSOUT=*
//MSGPRINT         DD SYSOUT=*
//IHCSYSIN         DD *
    RUN            PGM(IHCUDFSH) DBD(PHDO2) PCB(2)
//KEYS             DD *
08001000
98008000
//DFSHALDB         DD UNIT=SYDSA,DISP=(,PASS),
//                      DSN=&&HAL,SPACE=(TRK,1)
```

low key

high key

```
//DFSHALDB DD DSN=&&HAL,DISP=(OLD,DELETE)
```

Application Support

Application Programs

- **Partition Selection API**
  - Application provides root key
  - API returns partition name and number

Selection call:

CALL IHCUAPI using

(″SEL″,HANDLE,RETCODE,RSNCODE,*yourkey*,PARTNAME,PARTNUM)

The caller provides *yourkey*.

The API returns PARTNAME and PARTNUM.

IBM Software Group

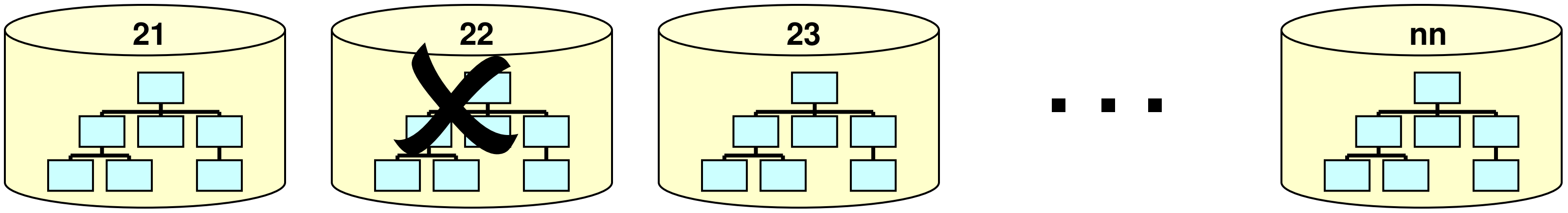
Maintaining HALDBs Databases - MAINTAIN functions

business on demand software

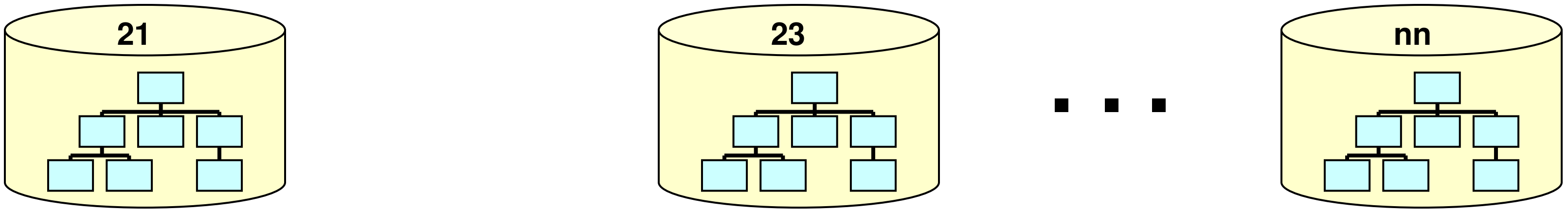
Production Maintenance

- Deleting a single partition
- Merging two DBDs into one DBD
- Changing partition boundaries
- Healing Index Pointers
- Reorganizing a PSINDEX
- Maintaining databases online

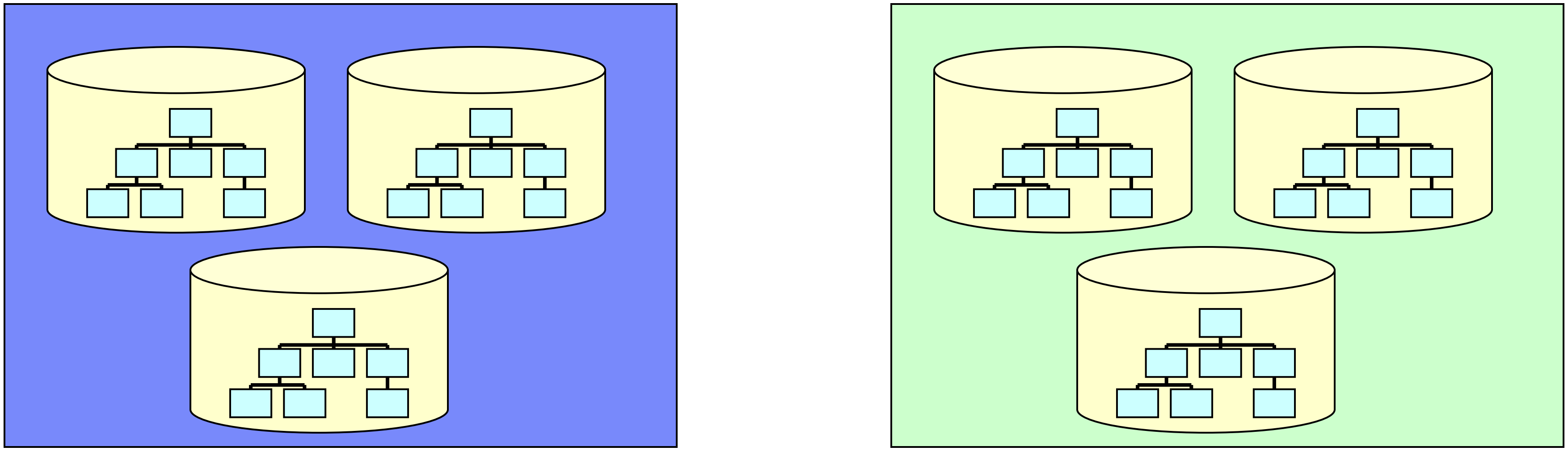
Deleting a partition



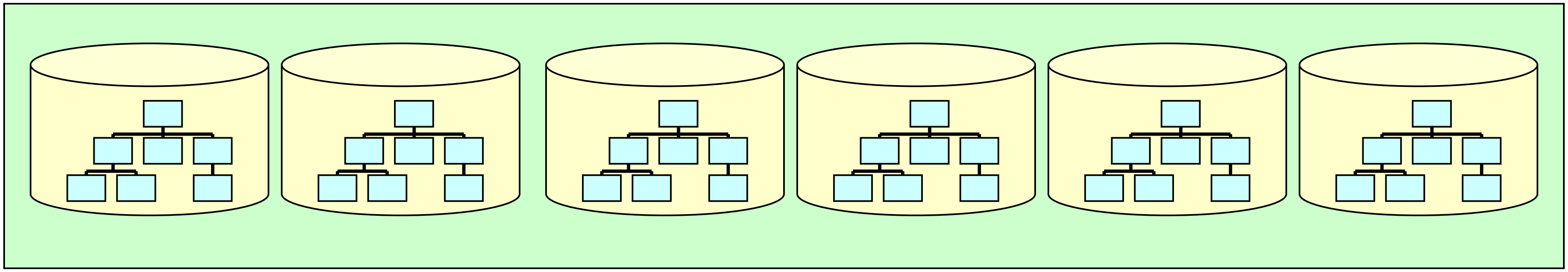
```
//IHCSYSIN DD *
   REORGIND DBD(PHDO2) DROP(PHDO22)
/*
```



Merging 2 DBDs into one DBD



```
//IHCSYSIN DD *
   HALDBCOPY FROMDBD(PHDPAYA) TODBD(PHDPAYRL)
/*
```



MAINTAIN – change partitions

Sample Scenarios	**MAINTAIN parameters**
Consolidate two partitions into one using PARTNUM	MAINTAIN DBD(*xxxx*)PARTLIST(part1, part) PARTNUM(1)
Consolidate all partitions into one using one key (highest key)	MAINTAIN DBD(*xxxx*)PARTITION(*) KEYS(keysin)
Split one partition into two using PARTNUM	MAINTAIN DBD(*xxxx*)PARTITION(part1)PARTNUM(2)
Split one large partition into 2048 size partitions using PARTSIZE	MAINTAIN DBD(*xxxx*) PARTITION(part2)PARTSIZE(2048)
Split one partition into three using your own key boundaries	MAINTAIN DBD(*xxxx*)PARTITION(part1) KEYS(KEYSIN)
Reorganize one partition without any changes	MAINTAIN DBD(*xxxx*)PARTITION(part3) PARTNUM(1)

MAINTAIN – Online

- The ONLINE parameter specifies whether or not to perform online maintenance The default value is N.

```
ONLINE  (Y|N|YES\NO)
```

Database Handling

- **Heal secondary index pointer**
  - EPS pointer not “healed” after REORG
  - Instant healing when required
  - Costly if many during online time

```
//IHCSYSIN DD *
    PTRHEAL DBD(PHDO2)
/*
```

Reorganizing a PSINDEX

- The following tasks can be performed by this utility:
  - Reclaim space
  - Reclaim split indexes
  - Remove deleted index records
  - Remove index records for a dropped primary partition

```
//IHCSYSIN DD *
    REORGIND DBD(PHDO2)
/*
```

IBM Software Group

Production Health

e business on demand software

HALDB Overview - Pointers

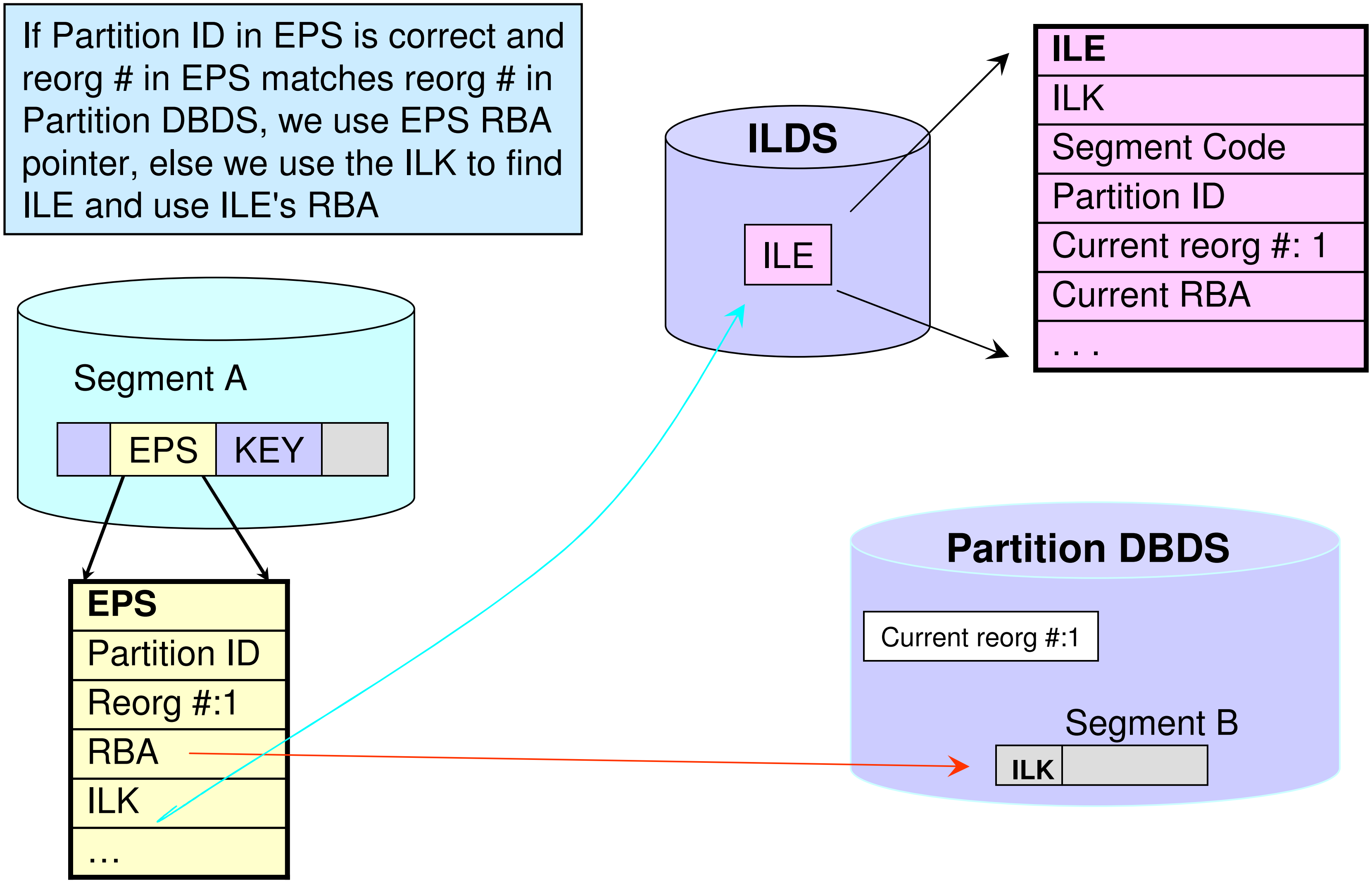
- Logical child segments and secondary index segments include:
  - Key of target
    - Key of target's root for secondary indexes
    - Logical Parent's concatenated key for logical relationships
    - Used to determine partition in which target resides
  - Extended Pointer Set (EPS)
    - RBA of target when last known
    - Reorganization number of target partition when RBA pointer was accurate
      - Used to determine if RBA is still accurate
    - Indirect list key (key of Indirect List Entry) for target segment
      - Used when RBA pointer is not still accurate
- Other pointers are unchanged from non-HALDB databases

HALDB Overview - ILDS

- **An Indirect List Data Set (ILDS) is associated with each partition**
  - One KSDS per partition
- **ILDS contains accurate RBA pointer to each secondary index or logical relationship target**
  - Entries are created or updated by reorganization
  - Key of entry is ILK (indirect list key) associated with target segment
  - Segments which are not targets of sec. ind. or log. rel. do not have entries
- **ILDS is used when reorganization number in logical relationship or secondary index pointer is "out of date"**
  - "Out of date" when reorg. number does not match partition's reorg. number
  - "Out of date" indicates that the pointer has not been healed since last reorg.

HALDB Overview – Using the EPS and ILE

If Partition ID in EPS is correct and reorg # in EPS matches reorg # in Partition DBDS, we use EPS RBA pointer, else we use the ILK to find ILE and use ILE's RBA



Health Checker

- **Threshold parameter driven**
  - Are mostly percentage parameter
  - Give a low and high value
    - Low: warn about an upcoming problem
    - High: problem is now more urgent
- **Allow other options to be turned off/on**
  - ILK verification
    - On occasion only
  - Root extract
    - Record layout is described in the manual

Health Checker

- **Example**:

```
CHECK  DBD(PHDO2) -
 MINROOT(500) -            (avoid warnings)
 VERIFYILK(Y) -            (to verify the ILK)
 VERIFYILE(Y) -            (to verify the ILE)
 KEYS(KEYSOUT) -           (extract root key)
 4GBWARN(3000,3500) -      (in MB)
 EXTENT(20,30)   -
 SYNONYM(10,20)  -
 ROOTOVFL(10,20)  -
 ROOTHOME(10,20)  -
 SPLIT(5,20)     -
 PTRHEAL(50,80)  -
 AVGFSE(5,12)    -         (is a number)
 RAPUSE(40,80)   -
 SPACE(300,2500)           (in MB)
```

Health Checker

- **Runs very fast**
  - Up to 8 datasets at the same time
  - Verification of ILK and ILE can be turned off

- **Provides decision support on partition maintenance**
  - Combining/splitting/rearranging
  - Adjusting randomizing parameter for partitions

Health Checker

- **How to resolve**
  - Partition too big
    - Split partition or rearrange with others
  - Pointer healing
    - Run “heal” utility or rebuild indexes.
  - Missing ILEs
    - Rebuild indexes
    - Reorg partition(s)
    - Rebuild ILDS

Health Checker

- **How to resolve**
  - Invalid or duplicate ILKs
    - Use ILK reset tool
    - Reload database with PROCOPT=L
      - Delay index build
  - Root in wrong partition
    - Talk with the application people
      - Could be invalid
    - Reorg the entire database

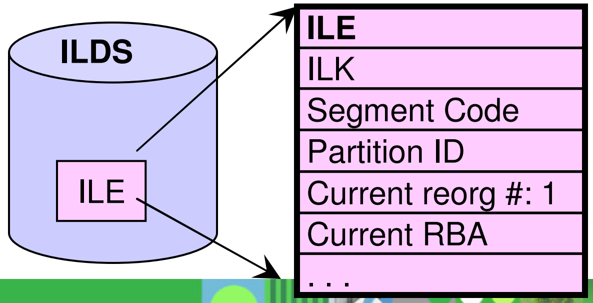
ILK Reset

- Resolves conflicting ILKs
- Fixes:
  - Invalid ILKs
  - Duplicate ILKs
- Run when HALDB Analyzer detects invalid ILKs

- STEPS:
  1. Run ILK Reset utility
  2. Reload database using unload file
  3. Rebuild secondary indexes

```
//S2 EXEC PGM=IHCHALDB,
//              REGION=50M
//STEPLIB       DD DISP=SHR,DSN=IHC320.SIHCLOAD
//              DD DISP=SHR,DSN=IMS910.SDFSRESL
//DFSRESLB      DD DISP=SHR,DSN=IMS910.SDFSRESL
//RECON1        DD DISP=SHR,DSN=RECON1
//RECON2        DD DISP=SHR,DSN=RECON2
//RECON3        DD DISP=SHR,DSN=RECON3
//IMS           DD DISP=SHR,DSN=DBDLIB
//TRACE         DD SYSOUT=*
//MSGPRINT      DD SYSOUT=*
//SYSUDUMP      DD SYSOUT=*
//IHCSYSIN      DD *
    RESETILK    DBD(ddddd)
//DFSURGU1      DD UNIT=SYSDA,DISP=(,PASS),
//              DSN=&&UNL,SPACE=(CYL,(1,1))
```

Unload File

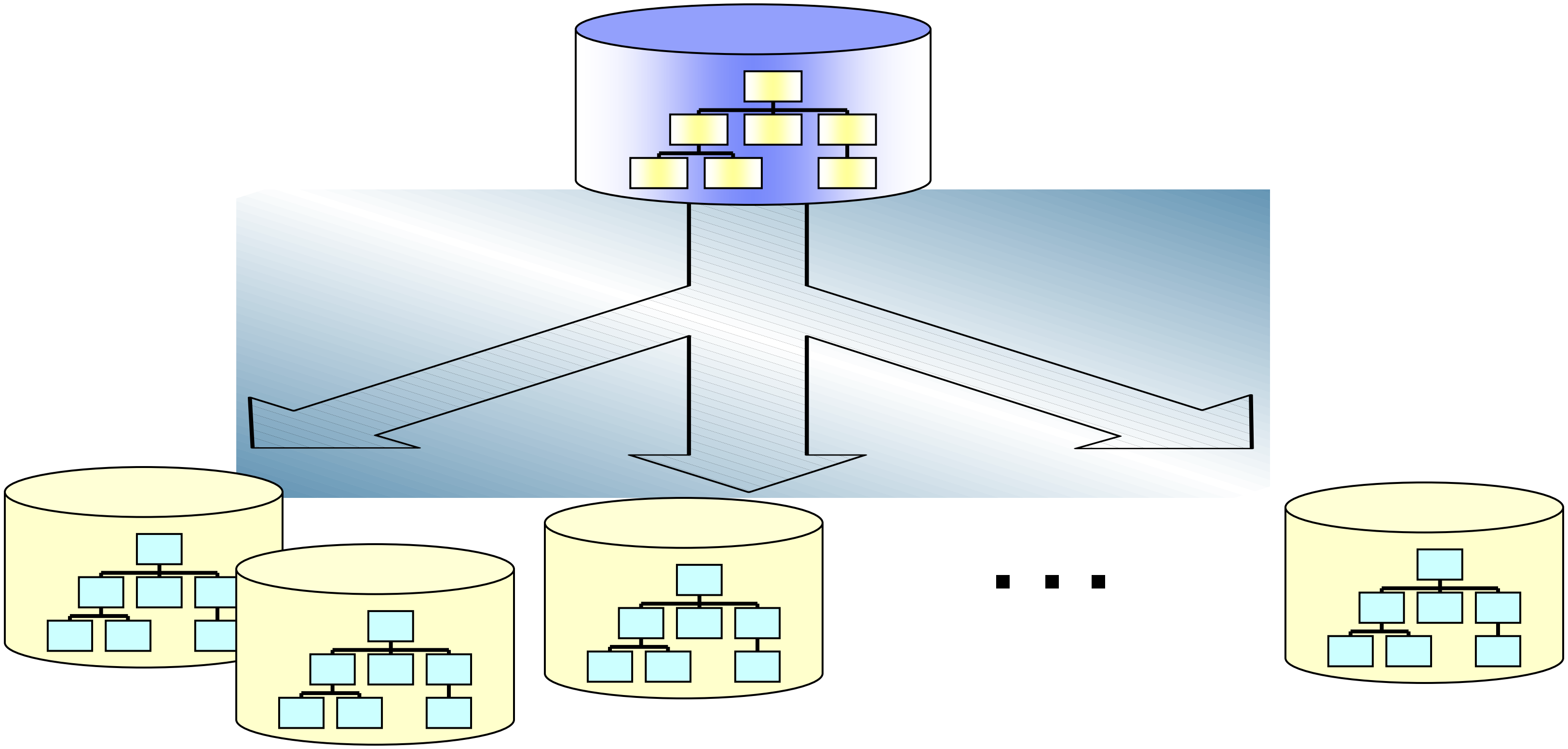


ILDS

ILE

ILE
ILK
Segment Code
Partition ID
Current reorg #: 1
Current RBA
. . .

Migration to HALDB



Converting to HALDB

- **Utility does it all**
  - Find partition boundaries
  - Convert the DBD
  - Do the DBRC work
  - Calculate and allocate the new files
  - Run the utilities
  - Can handle SHISAM, HISAM, DEDBs, PDB/PDF
- **DB can be open for update**
  - Requires ORF (IMS Online Reorg Facility) to be installed
  - Downtime is short

CONVERT

- **SYSIN command**

```
CONVERT  DBD(dbdname)           Primary DBD name
     DBDPATT(*****…)             to create a partition name
     DSNPREF(your.hlq)           appended with DBD name
     PARTSIZE(2048)|PARTNUM(nn)|KEYS(ddname)
```

- **Takeover concept**
  - No renames, all dataset names are new
  - Activate only if successful
  - Use of temporary RECON
  - RECON notification delayed
  - Activate changed DBD

CONVERT

- **Other parameter**
  - Back to 1 DSG
  - VSAM to OSAM
  - HDAM randomizing parameter
  - SMS constructs

- **Backup options**
  - DBD
  - DBRC
  - Dataset allocation

IMS HALDB Toolkit for z/OS

Application Support

- Dynamic DFSHALDB Statement Build
- Partition Selection API

Test Environment Support

DBRC Handling

- Cloning DBRC Definitions
- Copy HALDB definitions to different RECONs
- Backup DBRC definitions

HALDB Maintenance

- Reorganizing a PSINDEX
- Consolidate or split partitions
- Heal Index Pointer
- Load a Single Partition
- Delete a Single Partition
- Merge HALDBs
- Add empty partition to end of HALDB

HALDB Analyzer

- Analyze HALDB Constructs
- Extract Root Keys

Convert to HALDB

- ISPF Interface
- Single step batch
- Integration with ORF allows near online conversion
- Testing of partition selection exits

System Utilities

- Split Unload File
- ILK Rebuild
- ACBLIB report
- Create DBD Source

Q & A

With Janet LeBlanc and Christian Koeppen

More Information

- HALDB Toolkit:
http://www.ibm.com/software/data/db2imstools/imstools/ims-haldb-toolkit/index.html

- IMS Tools website:
http://www.ibm.com/software/data/db2imstools/products/ims-tools.html

- Documentation:
http://www.ibm.com/software/data/db2imstools/imstools-library.html