E44

IMS Database Reorganization Yesterday, Today and Tomorrow

Hélène Lyon - IBM EMEA IMS Tools team
helene.lyon@fr.ibm.com

St. Louis, MO **Sept. 30 - Oct. 3, 2002**

RETURN TO INDEX

Agenda

- **Introduction**
- **IMS Databases Overview**
- **The Reorganization Process**
  - Using standard IMS Utilities
  - Using IBM Tools

IMS Database Reorganization: Yesterday, today and tomorrow

The process of performing an IMS Database Reorganization is continuing to change as batch windows shrink and you move closer to 7 X 24 operations. The evolution of the tools/utilities available from IBM are reflective of this change. This presentation will put in perspective the different solutions to perform Full Function database reorganizations with currently available, recently announced and potential future tools/utilities.

What is a Database ?

- **A collection of interrelated data items organized in a form for easy retrieval**
  - The collection of data is stored in a computer system
  - The retrieval is done by application programs
  - Each item of data only needs to be stored once
    - Shared among the programs and users
- **An IMS database is organized as a hierarchy**
  - A database is a group of related database records (DBRs)
  - A database record is a single hierarchy of related segments
    - Data at lower levels depends on data at higher levels for its context
  - A segment is a group of related fields
  - A field is a single piece of data
    - It can be used as a key for ordering the segments
    - It can be used as a qualifier for searching
    - It may only have meaning to the applications

Why Reorganization is needed?

- **A Requirement in the Database Management Area!**
  - Changing Physical Storage
    - Database physical disorganization due to insert/detete processses and high update activity
    - More I/Os needed to get/store the information in the database
    - Response time degradation
  - Changing Database Structure
    - Change in database design

- **A Part of the Tuning and Monitoring Process!**
  - Database Manager Component
    - Reduction of physical I/Os by good use of buffers
  - Applications accessing databases
    - Response time
    - Amount of database calls
  - Many Other Things to do!
    - Advices in redbooks
      - *IMS Version 7 Performance Monitoring and Tuning Update, SG24-6404*

Reorganizing and Restructuring

Can take a long time!

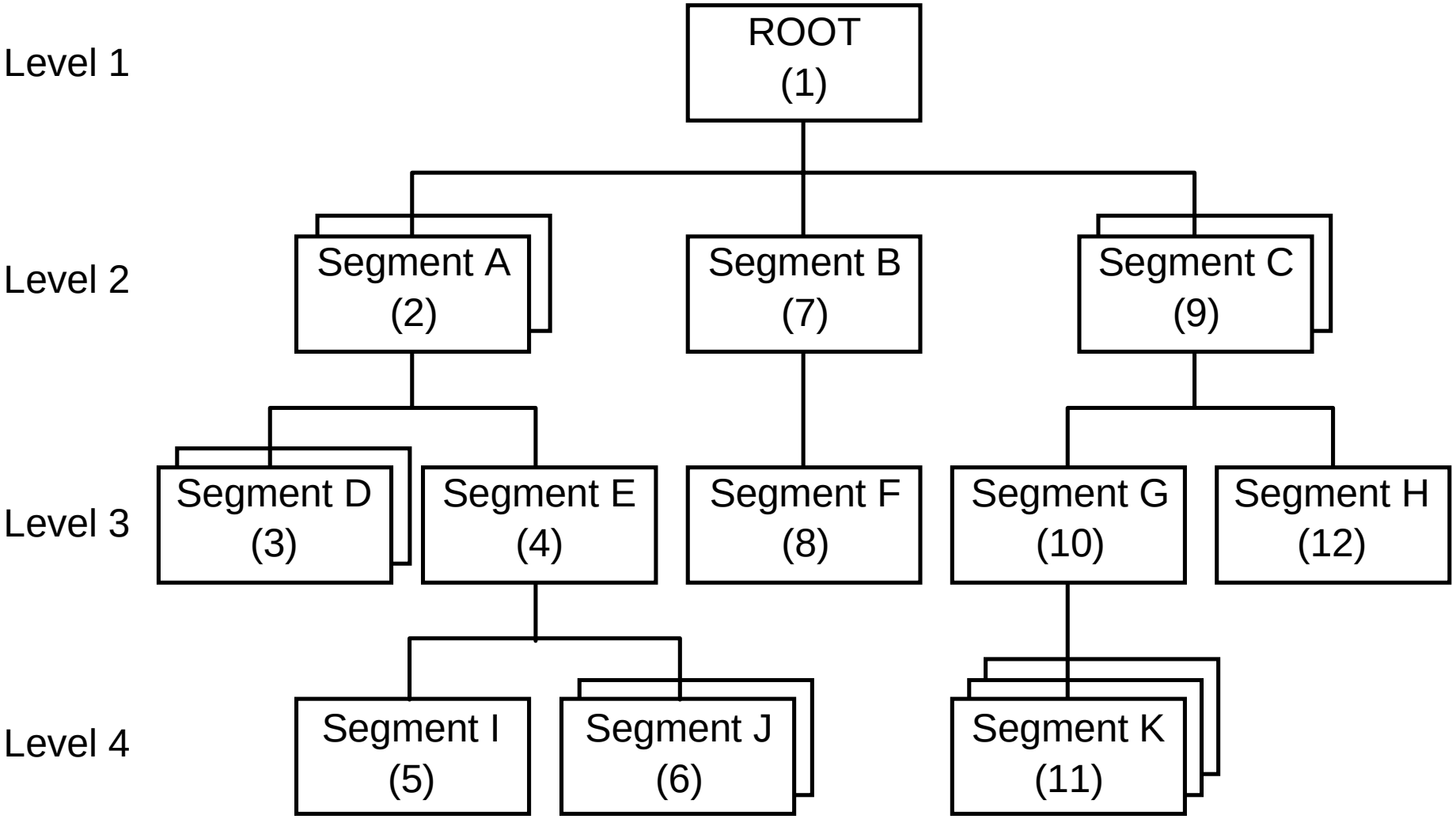
In designing your database, remember to plan for maintenance!

How often to Reorganize?
How quickly a database becomes disorganized and how bad the performance becomes?
Can we expect space shortage again?

IMS Database Overview

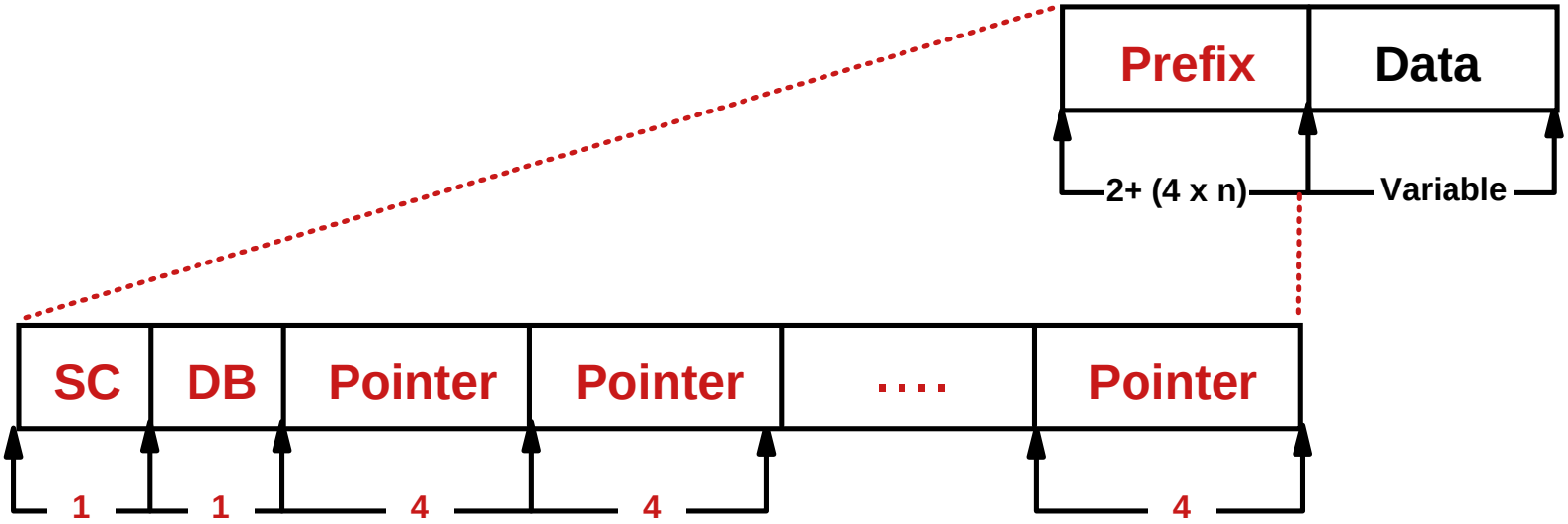
- Detailed in the Hand-Out
- Not shown during the presentation

IMS Database Overview



IMS Database Overview ...

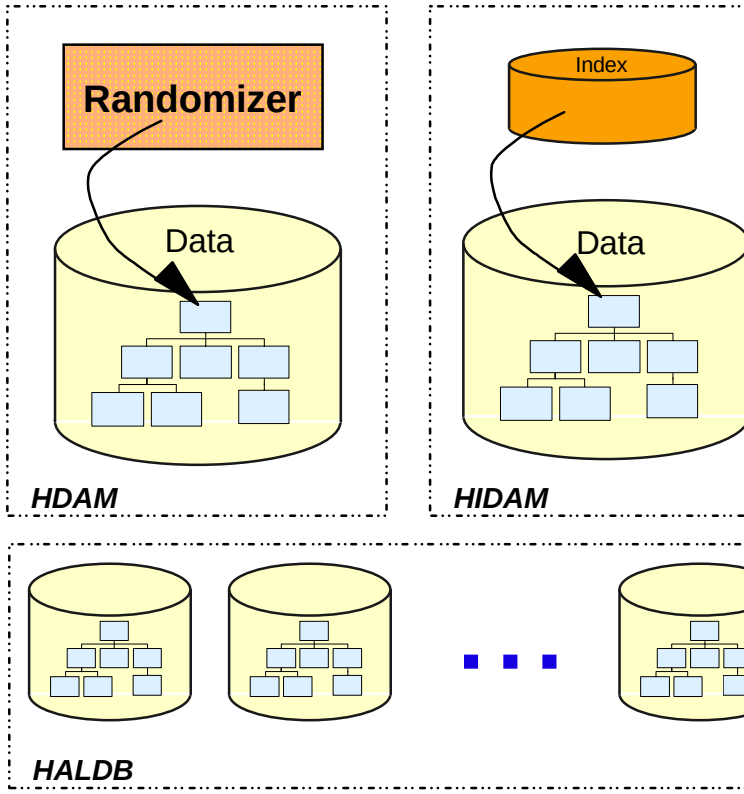
- **Segments are stored with a prefix and a data portion**
  - Prefix Portion
    - Used only by IMS
    - SC = segment code, 1 byte
    - DB = delete byte, 1 byte
    - 0 to n pointers, 4 bytes each
  - Data Portion
    - What the application program sees



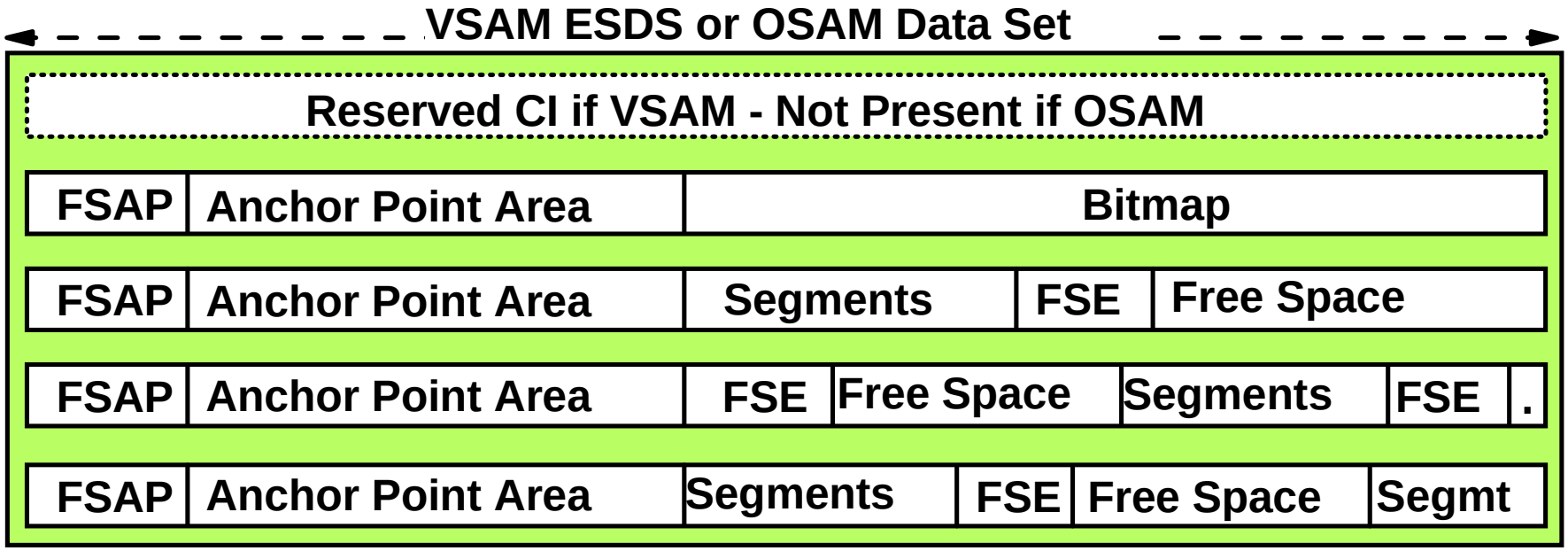
Direct Organization

- **Physical storage is independent of hierarchic sequence**
  - Pointers are used to maintain segment relationships
    - Pointers are in the segment prefix
    - Segments can be stored 'anywhere'
    - Segments are not physically moved
  - Space from deleted segments can be reused

- **Direct Database Types**
  - Hierarchic Direct Access Method (HDAM)
    - Randomizing module for direct access to root
    - No sequential access in the root order
  - Hierarchic Indexed Direct Access Method (HIDAM)
    - Access to the root using an index
    - Sequential access in the root order possible
  - And the 2 HALDB Types
    - Partitioned HDAM
    - Partitioned HIDAM



HD Storage



- **All HD data is in a single VSAM ESDS or OSAM data set**
- **One Logical Record (LR) per CI or block**
  - Logical record length = block size for OSAM
  - Logical record length = block size -7 for VSAM
- **All segments are stored as an even number of bytes**

HD Storage ...

Bitmap

- One bit per block or CI
  - First bit corresponds to the bitmap itself
- 1 = enough space to store the LONGEST segment in the database
- 0 = not enough space for the LONGEST segment
- If bitmap has N bits, block or CI N + 1 is a new bitmap

Anchor Point Area

- Contains one or more 4-byte Root Anchor Points (RAP)
  - In HIDAM: 1 RAP if the root has PTF or HF pointer
  - In HDAM: RMNAME parameter specifies number of RAPs
- Each RAP contains the address of a root segment or 0

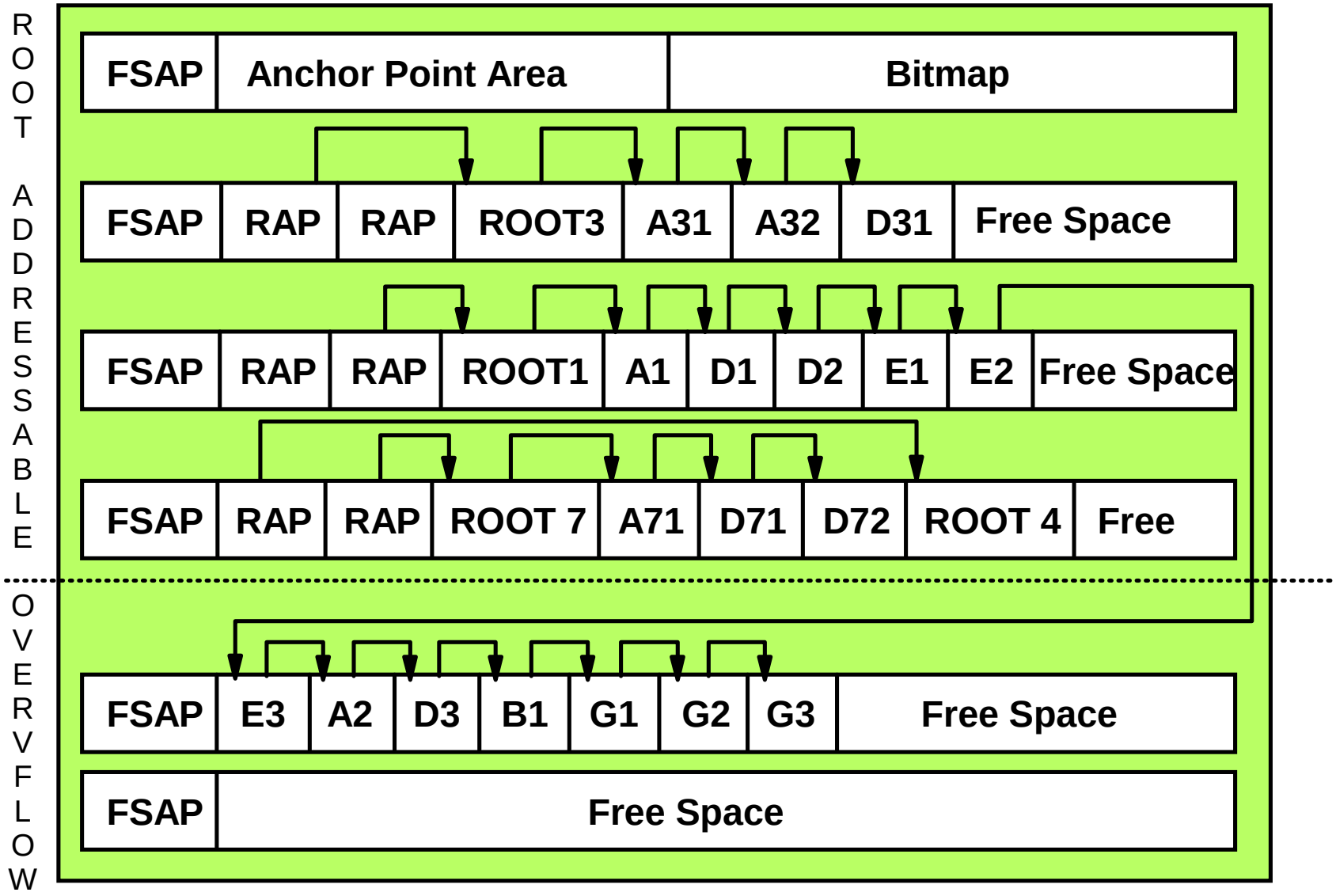
Free Space Anchor Point (FSAP)

- Two 2-byte fields
  - First the offset in bytes to first FSE
  - Second is a flag indicating if this block is a bitmap
    - *0 = this is not a bitmap*

Free Space Element

- First 2 bytes are offset, in bytes, to next FSE
  - Zero if this is the last FSE in the block or CI
- Second 2 bytes are length of free space, including FSE
  - No FSE is created if free space is less than 8 bytes long
- Last 4 bytes is the task ID of the program that freed the space
  - Allows a program to free and reuse the same space without contention
  - Useful in determining who free the space

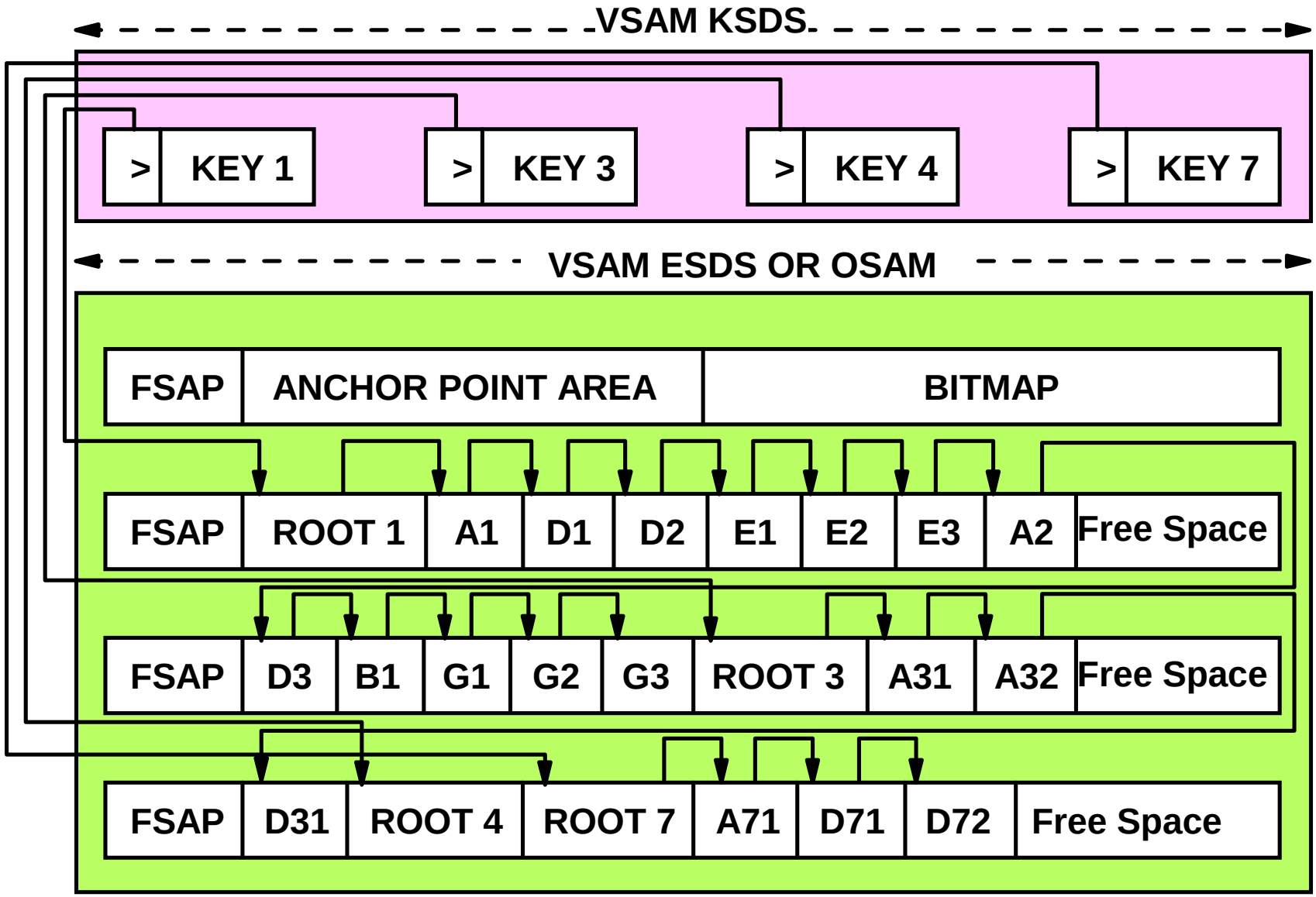
HDAM Storage



HDAM Storage ...

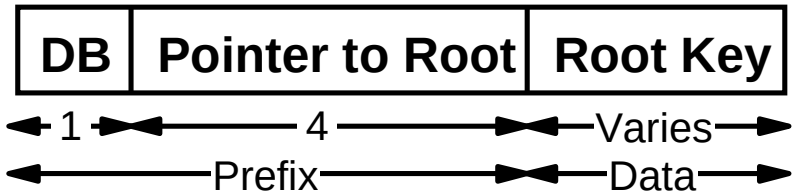
- **Root Addressable Area (RAA)**
  - Number of blocks or CIs defined in RMNAME parameter
  - Primary storage area for roots and dependents
    - Number of dependents at initial load is limited by RMNAME
    - Insert until specified bytes limit would be exceeded
  - All RAPs are in the RAA
  - Location is determined by Randomizer specified in RMNAME
    - Randomizer input is the root segment's key
    - Randomizer output is a block number and RAP number
    - Keys that randomize to same block and RAP are synonyms
    - Synonyms are chained using PTF pointers
    - Chain is ascending key sequence or by insert rules
- **Overflow Area**
  - For segments that do not fit in the RAA
  - No RAPs are present in the overflow area

HIDAM Storage



HIDAM Storage ...

- **Data Component**
  - A VSAM ESDS or OSAM data set
  - No RAA or Overflow portions
  - Database records are stored in key sequence
  - Roots must have unique keys
  - Segments in hierarchic sequence
  - You can specify that free space be left after loading
    - A percentage in each block or CI
    - Every Nth block or CI
- **Index Component**
  - VSAM KSDS
  - The index is a root-only database
  - One index segment for each database root



HALDB (High Availability Large Database)

- **Capacity**
  - Databases are partitioned
    - Up to 1001 partitions per database
    - Partitions have up to 10 data set groups
  - Each partition can be size of non-partitioned database

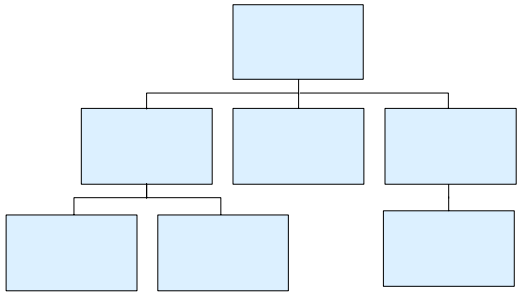
Up to 10,010 data sets per database!

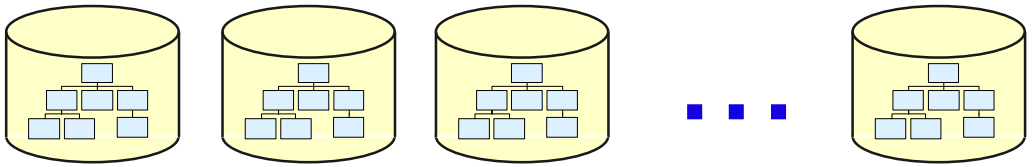
Greater than 40 terabytes

- **Availability**
  - Partition independence
    - Allocation, authorization, reorganization, and recovery are by partition
  - Self healing pointers
    - Reorganization of partition does not require changes to secondary indexes or logically related databases which point to it
- **Manageability**
  - Smaller partitions are easier to manage
- **Usability**
  - Partition definition is via an ISPF Partition Definition Utility

HALDB (High Availability Large Database) ....

- **HALDB supports new database types**
  - PHDAM - partitioned HDAM
  - PHIDAM - called partitioned HIDAM
    - The primary index is automatically defined and partitioned
  - PSINDEX - partitioned secondary index
  - Logical relationships and secondary indexing are supported
  - OSAM and VSAM are supported
- **HALDB definition is via**
  - new ISPF Partition Definition Utility to define partition
    - registers HALDB db with DBRC
  - DBDGEN to define logical database structure
  - IMS System Definition for database name
- **Partition selection methods**
  - by key
  - by user exit routine - Partition Selection Exit routine

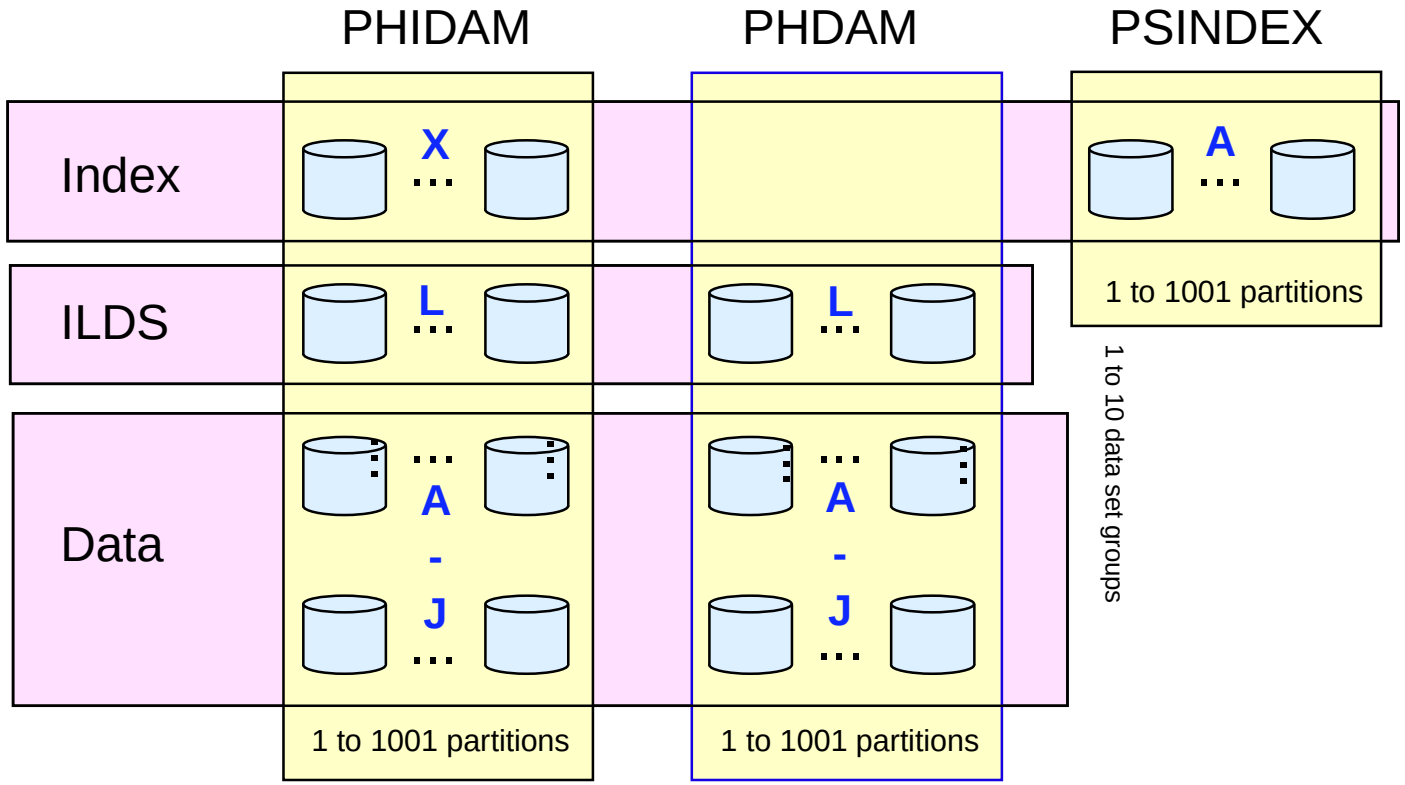




HALDB New Pointer Scheme - Indirect Pointers

- **HALDB uses direct pointers and indirect pointers**
- **New indirect pointer scheme introduced**
  - Self healing pointer
  - Handles data reorganizations of records pointing to same or other partitions
    direct pointers in segment prefix automatically updated from the ILE when required after reorganizations
- **Indirect List Data Set (ILDS)**
  - contains Indirect List Entries (ILEs) created for segments involved in inter-record pointing
- **Benefit:**
  - Reorganizations of partition do not require changes to indexes or logically related databases
  - Prefix resolution and prefix update utilities not required

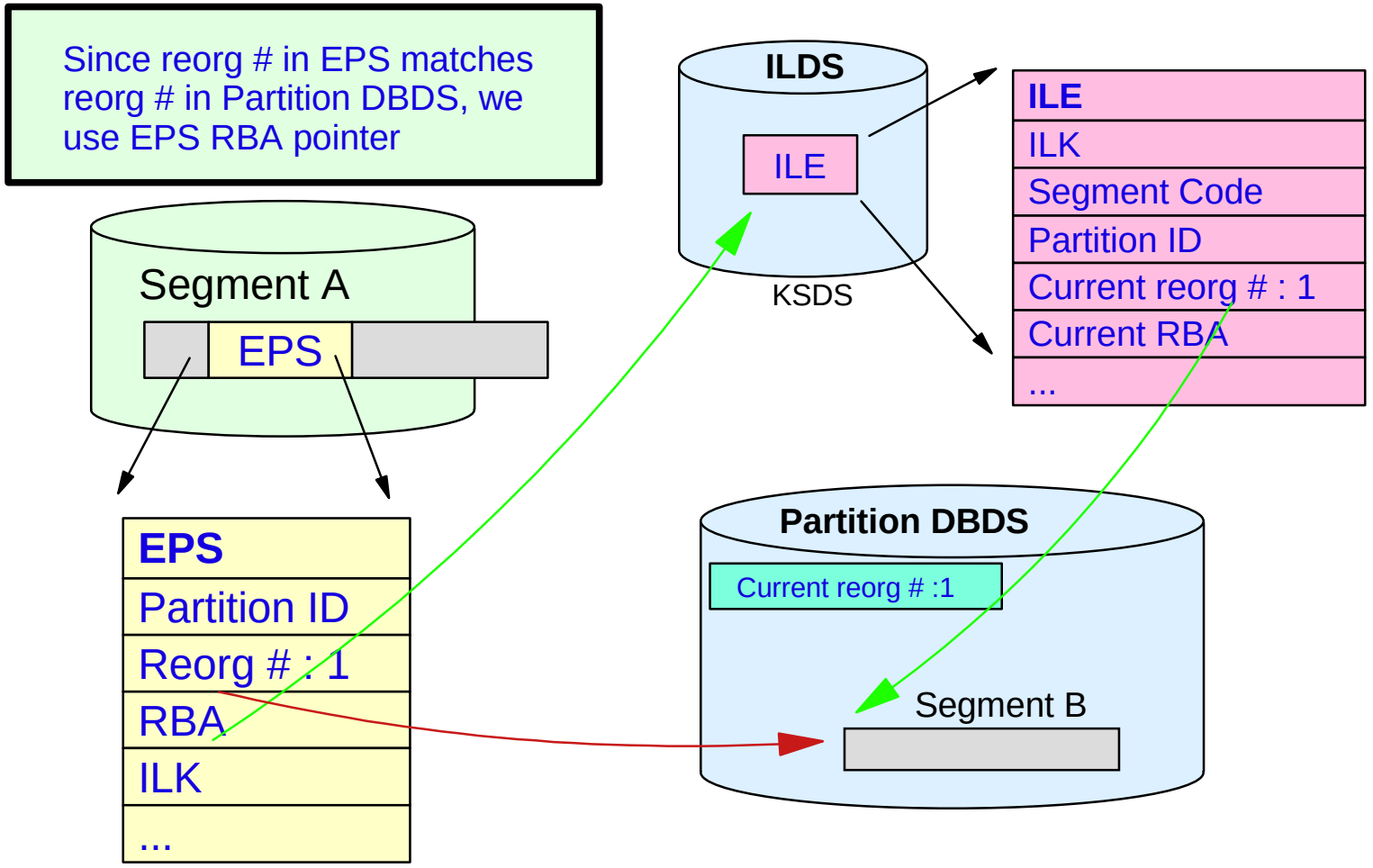
HALDB Database Data Sets



- **The data sets in a partition have generated data set names and DDNAMEs.**
  - X - PHIDAM index
  - L - ILDS
  - A through J - Data data sets
  - A - PSINDEX

HALDB - Self-Healing Pointers

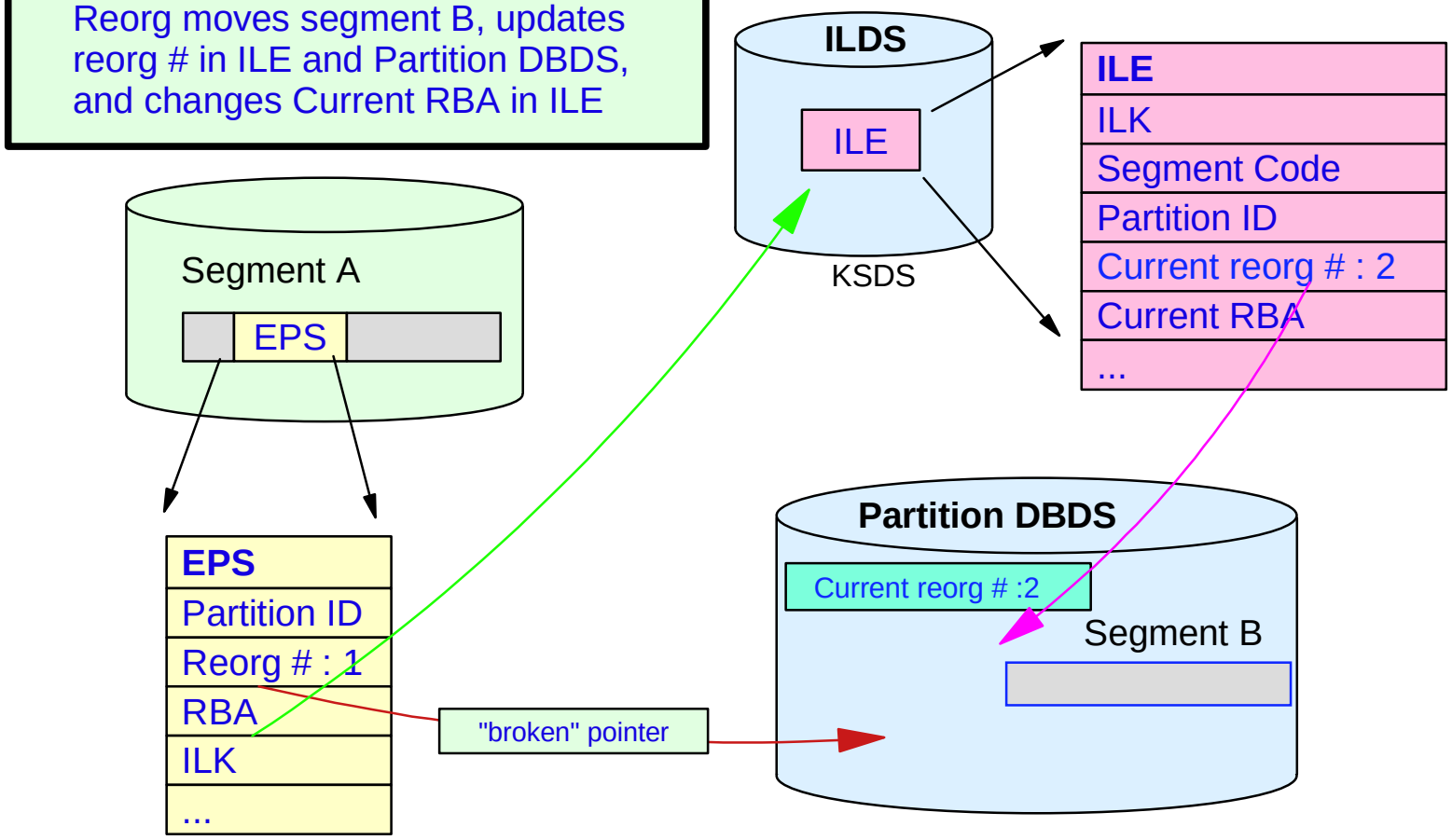
Using an Extended Pointer Set (EPS)



HALDB - Self-Healing Pointers ...

After reorganization of Partition

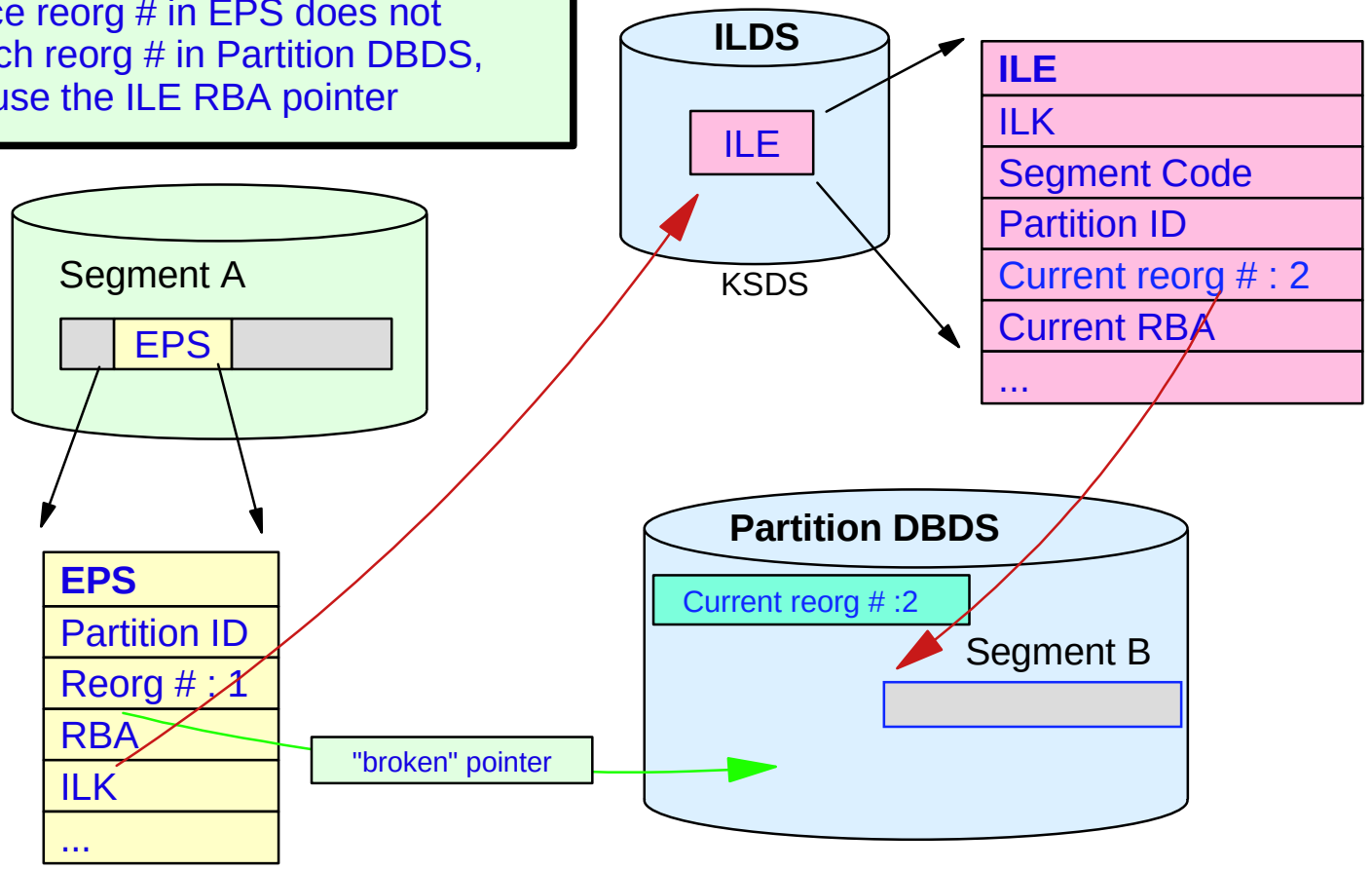
Reorg moves segment B, updates reorg # in ILE and Partition DBDS, and changes Current RBA in ILE



HALDB - Self-Healing Pointers ...

Using the EPS after the reorganization

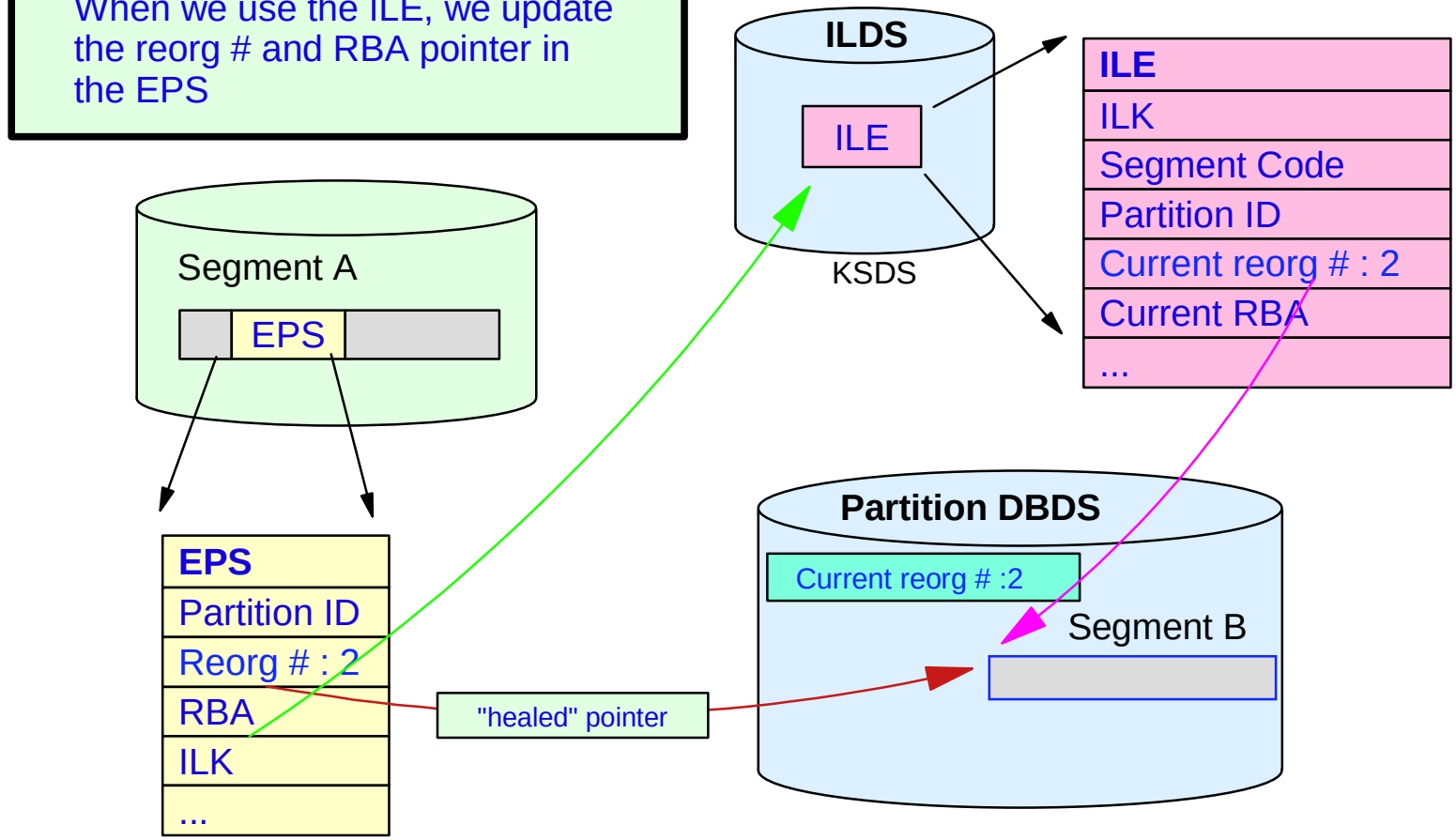
Since reorg # in EPS does not match reorg # in Partition DBDS, we use the ILE RBA pointer



HALDB - Self-Healing Pointers ...

"Healing" the EPS

When we use the ILE, we update the reorg # and RBA pointer in the EPS



Processing HD Databases

- **Delete**
  - The segment and all of its dependents are removed
  - FSE is used to indicate the space is free
    - Create a new FSE and update the FSAP/FSE Chain
    - Update length field of preceding FSE
  - Segment points are updated
- **Replace**
  - No change in length or fixed-length
    - Overwrite old segment with updated segment
  - Shorter segment
    - Space previously occupied is freed
    - FSE created if at least 8 bytes shorters
  - Longer segment
    - If enough adjacent free space, store in original location
    - If no space available, HD space search algorithm used
- **Insert**
  - Store in the Most Desirable Block (MDB)
    - HDAM root MDB
      - *The one which is selected by the randomizer*
      - *The one containing its previous synonym*
    - HIDAM root MDB
      - *If no backward pointer, same as the next higher key root*
      - *If backward pointer, same as the next lower key root*
    - Dependents
      - *If Physical, same as parent or previous twin*
      - *If Hierarchic, same as previous segment in hierarchy*
  - Second most desirable block (2nd MDB)
    - Nth Block or CI left free during loading
      - *If in buffer pool or bitmap shows space available*
    - Specified by FRSPC parameter
      - *If not specified, then no second MDB*

Processing HD Databases ...

- **HD Space Search Algorithm**
  - In the MDB
    - This will be in the buffer pool.
  - In the 2nd MDB
  - Any block in the buffer pool on the same cylinder
  - Any block on the same track
    - If the bitmap shows space available
  - Any block on the same cylinder
    - If the bitmap shows space available
  - Any block in the buffer pool within +/- SCAN cylinders
  - Any block within +/- SCAN cylinders
    - If the bitmap shows space available
  - Any block at the end of the data set is in the pool
  - Any block at the end of the data set
    - If the bitmap shows space available
    - Extend the data set if necessary
  - Any block where the bitmap shows space

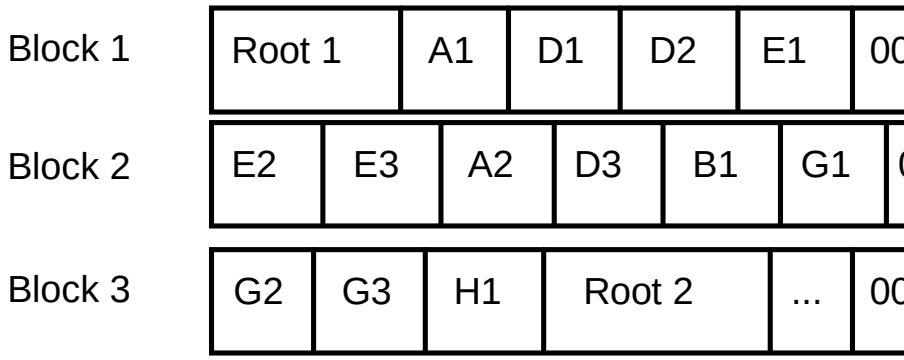
Sequential Organization

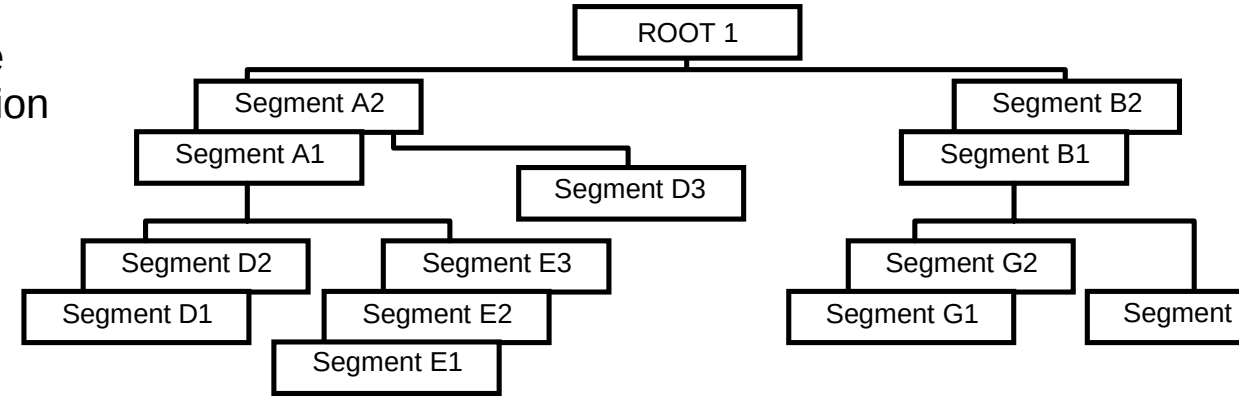
- **The data is physically stored in hierarchic sequence**
  - Database records are stored in a root key sequence
    - If no root key, they are stored as presented
  - Segments in a record are stored in hierarchic sequence

- **Sequential Database Types**
  - Hierarchic Sequential Access Method (HSAM)
  - Simple Hierarchic Sequential Access Method (SHSAM)
    - Root-only HSAM
  - Hierarchic Indexed Sequential Access Method (HISAM)
  - Simple Hierarchic Indexed Sequential Access Method (SHISAM)
    - Root-only HISAM using VSAM
  - Generalized Sequential Access Method (GSAM)
    - No hierarchy, no database records, no segments
  - And the HALDB PSINDEX
    - Partitioned Secondary Index

Sequential Organization - HSAM

- **HSAM**
  - Tape or DASD
  - BSAM or QSAM
    - QSAM if online or PROCOPT=GS
  - Fixed-Length, Unblocked format
    - RECFM=F
    - logical record length=physical block size
  - Cannot Delete or Replace
    - Update by rewriting the database
  - Insert allowed when loading the database
  - Restrictions
    - No pointers in prefix - SC and DB only
      - *Delete byte is not used*
    - No multiple data set groups (MSDG)
    - No logical relationships or secondary indices
    - No variable length segments
    - No edit/compression or data capture
    - No logging, recovery, or reorganization

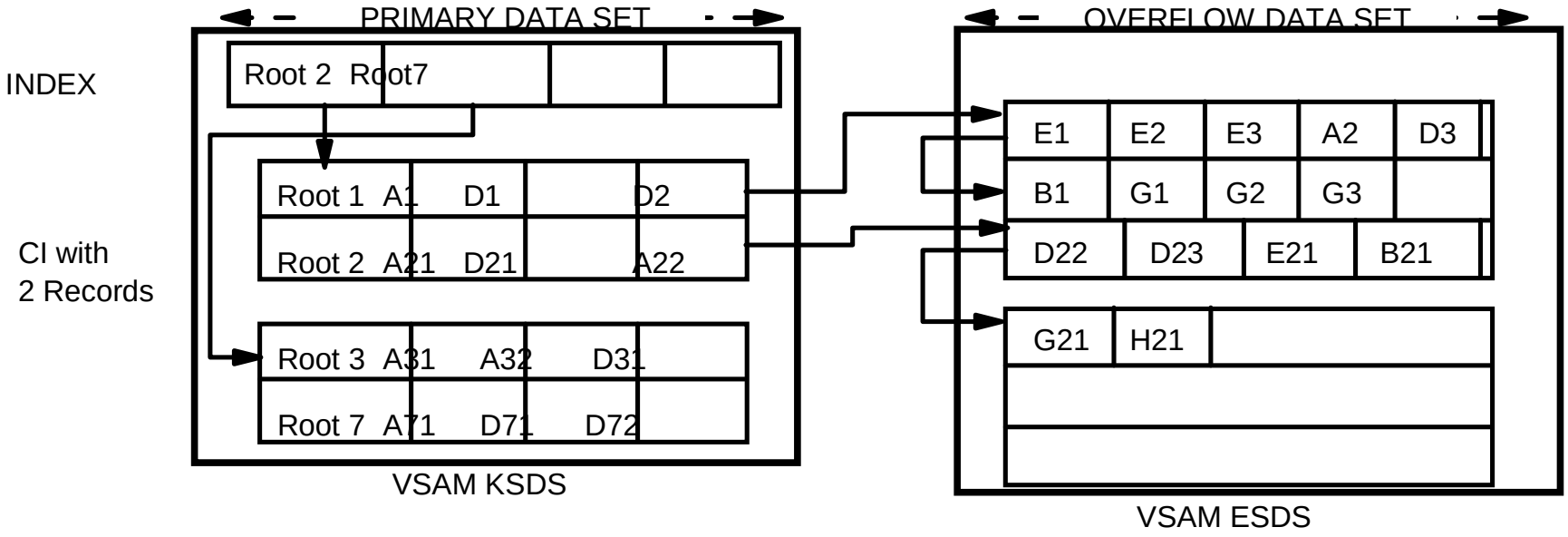




Sequential Organization - HISAM

HISAM

- DASD and VSAM only
  - KSDS for the primary data set, ESDS for the overflow data set
- Each root must have a unique key
- A database record is stored as 1 record in the primary data set and 0 to N records in the overflow data set
- HISAM works better when
  - Applications randomly access the records and then read the segments sequentially
  - Most of the database records are the same size
  - Relatively few dependents per root
  - Very low insert/delete activity



PSINDEX Database

- **The Secondary Index Databases of HALDB databases!**
- **Creation of a HALDB Secondary Index**
  - With the initial load of their target database
    - Work files are not created
    - Prefix Resolution, HISAM Unload and Reload Utilities are not used
- **Addition of a secondary index**
  - No utilities to add a secondary index
  - Must be done by Initial load of the target database or by the use of ***IMS Index Builder***
- **Miscellaneous**
  - The reorganization of the target database does not affect a secondary index
  - A secondary index has no ILDS
  - A partition selection exit may be used
  - Shared secondary indexes are not supported
  - Secondary indexes must have unique keys
  - The target root key is stored in index segment

IMS Database Reorganization Process

IMS DB Reorganization - The First Need

Reorganizing and Restructuring

- Changing Physical Storage
  - Database physical disorganization due to insert/detete processses and high update activity
  - Segments in a database record are stored across too many CIs or blocks
- Changing Database Structure
  - Change any physical characteristics: size of data sets, volume, ...
  - Change the DL/I access method (VSAM or OSAM) or type of databases
  - Change in HDAM RAA
  - Add or delete segment types

Physical Reorganization

- To reclaim and consolidate free space
- To rebuild the free-space
  - as specified in DATASET macro for HDAM, HIDAM
- To optimize the grouping of the root segment and its dependent segments
  - into one block, or into adjacent blocks if they do not fit into a single block

Proactive Approach

- Regular monitoring of database
  - .... Using IMS HP Pointer Checker
- History of the collected monitoring information
- Only ONE change at a time

IMS DB Reorganization - The Process

From Simple to Very Complex

- Use of logical relationship, or/and secondary indices
  - To reflect the new physical position of the segments in the reorganized DB
- Use of HALDB
  - To simplify the reorganization process!

The Simplest Reorganization Process

- DB Unload
  - Sequential file provided as output
- Delete / define physical data set
  - only necessary if you have multiple extents or volumes, or are using VSAM without HALDB
- DB Reload

Additional Steps

- DB Backup
- DBDGEN / ACBGEN
- Preparation of logical relationships (LR) and/or secondary indices (SI) rebuilt
  - Collect some prefix information
- Rebuild of logical relationships (LR) and/or secondary indices (SI) connections
  - Collect more prefix information
  - Update pointer information located in segment's prefix
- DBRC Notification
- Image Copy
  - New base for forward recovery

Remember

In the Reorged DB, location of segments has changed!

Without HALDB

If LR or SI used, pointer update needed to reflect new segment location

With HALDB

Self-Healing Pointer

Shorten the reorg time to your window

IMS DB Reorganization - Utilities Classification

- **Reorganization Without UCF**
  - Combination of utilities
  - Utility list depends on the type of database and whether it uses LR or SI

- **Partial Reorganization**
  - Old way of doing reorg
  - When only parts of database need to be reorganized
    - To reduce the amount of time it takes to do a total reorganization into smaller pieces
  - Database Surveyor utility
    - To determine which parts of your database to reorganize
  - Partial Database Reorganization utility
    - To reorganize a group of DBRs
      - *with continuous relative block numbers with HDAM*
      - *with continuous key values with HIDAM*

- **Reorganization Using UCF (DFSUCF00)**
  - Utility Control Facility (UCF) used as a controller, determining which of the various reorganization utilities need to be executed and then getting them executed.
    - Simpler JCL
    - Reduction in the number of decisions operations people must make
  - Advantages
    - Restart Capability
    - WTOR available to stop the job, as well as to enter certain options.
    - User exits capability
    - Protection against some operational problems
  - Not often used

Reorganization Process for Sequential DB

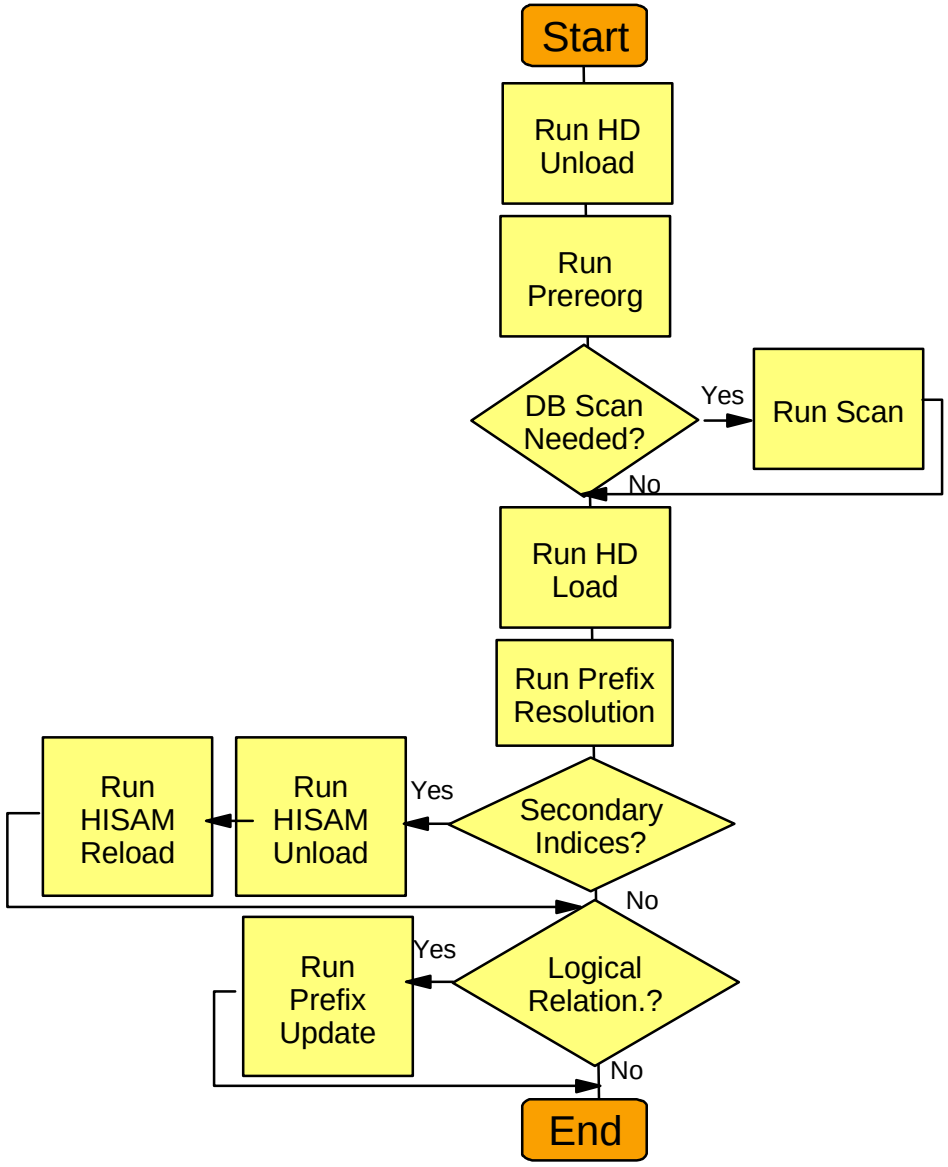
- **For HSAM, SHSAM databases**
  - Reorganization using user-written Program
    - Program to read the old database and then create a new database
  - The IMS standard utilities cannot be used.

- **For HISAM databases**
  - Standard IMS HISAM Utilities
    - HISAM Reorganization Unload (DFSURUL0)
    - HISAM Reorganization ReLoad (DFSURRL0)
    - Restrictions
      - *No structural changes allowed*
      - *No support of LR and SI*
  - Standard IMS HD Utilities
    - HD Reorganization Unload (DFSURGU0)
    - HD Reorganization Reload utility (DFSURGL0)

- **For SHISAM databases**
  - Standard IMS HISAM Utilities not supported
  - Standard IMS HD Utilities
    - HD Reorganization Unload (DFSURGU0)
    - HD Reorganization Reload utility (DFSURGL0)
  - VSAM Repro

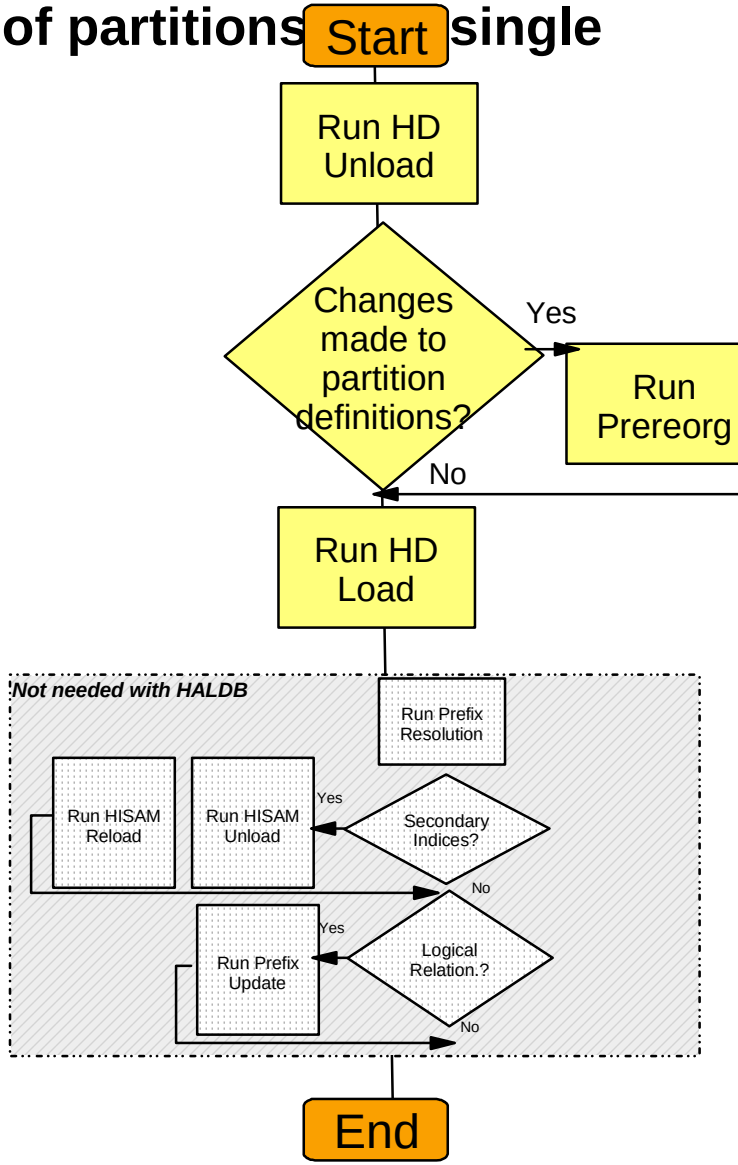
Reorg. Process for Direct DB - Non HALDB

- **Standard IMS Utilities**
  - HD Reorganization Unload (DFSURGU0)
  - HD Reorganization Reload utility (DFSURGL0)
  - Database Prereorganization Utility (DFSURPR0)
  - Database Scan Utility (DFSURGS0)
    Used to scan DBs that are not reorganized but are involved in LR with DBs that are being reorganized
  - Database Prefix Resolution Utility (DFSURG10)
  - Database Prefix Update Utility (DFSURGP0)
  - HISAM Reorganization Unload (DFSURUL0)
    For unload of HIDAM primary index database or secondary index database
  - HISAM Reorganization Reload (DFSURRL0)
    For reload of HIDAM index database



Reorganization Process for HALDB

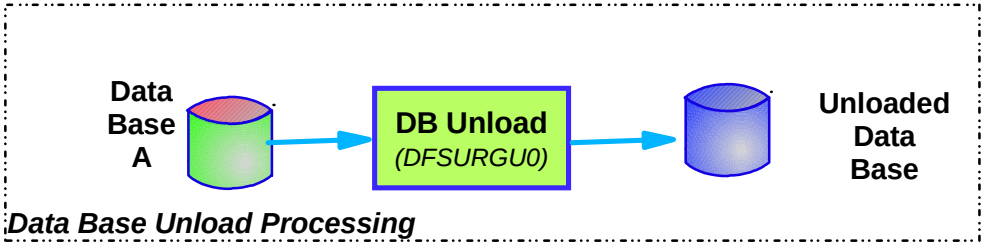
- **Reorganization of the entire database, or a range of partitions single partition**
  - Reorg partitions in parallel
    Create enough partitions to meet your requirement
- **Standard IMS HD Utilities**
  - HD Reorganization Unload (DFSURGU0)
    For PHDAM, PHIDAM and PSINDEX
  - Database Prereorganization Utility (DFSURPR0)
    To initialize the changed partitions
    Define of partition data set and ILDS needed for new partitions
  - HD Reorganization Reload utility (DFSURGL0)
    For PHDAM, PHIDAM and PSINDEX
- **No rebuild of secondary indexes**
  - Prefix Resolution, HISAM Unload, HISAM Reload, or Index Builder are not required
- **No updates to logical relationships**
  - DB Scan, Prefix Resolution, and Prefix Update are not required



IMS Database Unload and Reload Utility

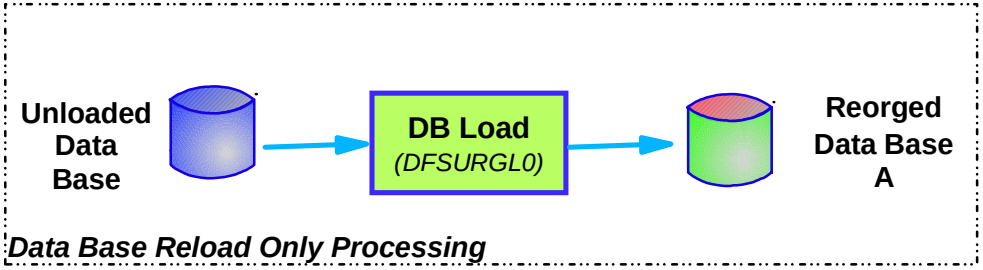
Database Unload

- Support for HDAM, PHDAM, HIDAM, PHIDAM, PSINDEX, or HISAM
- Standard IMS Utility: DFSURGU0
  can benefit of OSAM Sequential buffering



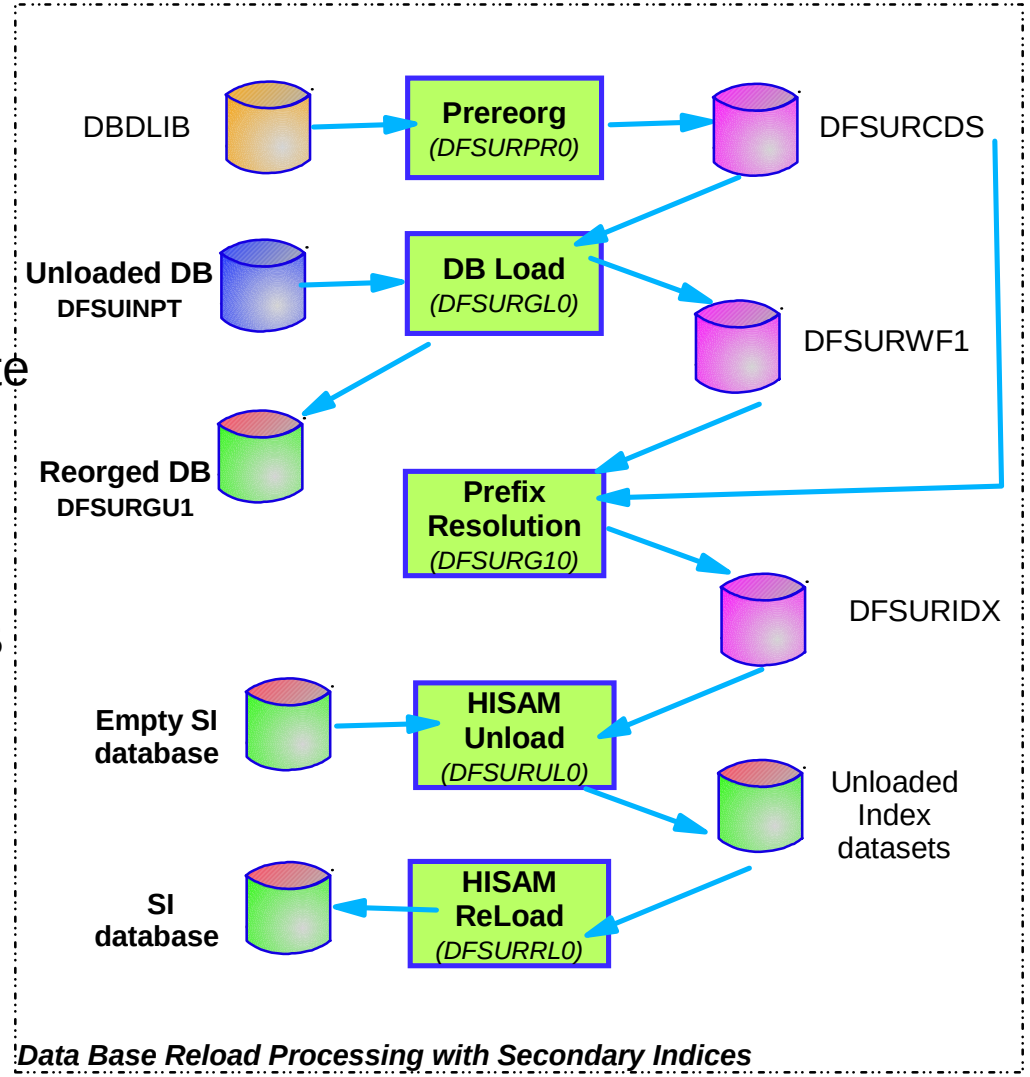
Database Reload

- 4 types of reload
  Reload only
  Reload with secondary indices (SI) only
  Reload with logical relationship (LR) only
  Reload with both LR and SI
- For HIDAM database
  Primary index is reloaded automatically when the main DB is reloaded



IMS Database Reload with Secondary Indices

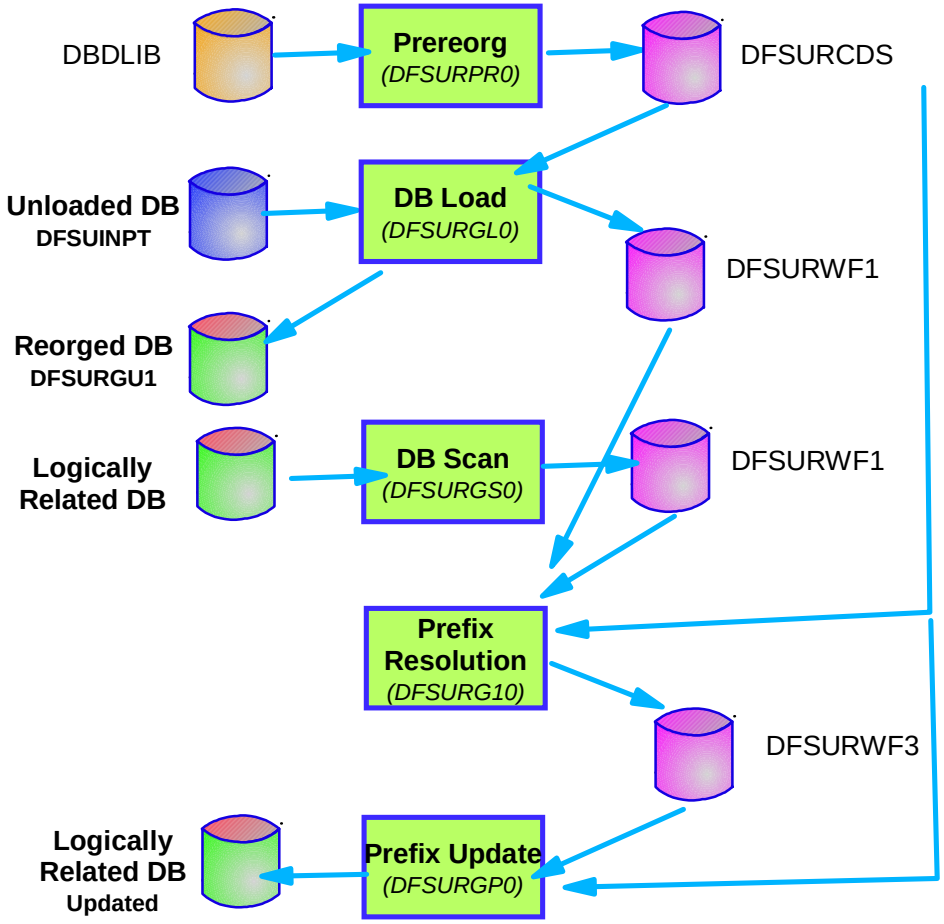
- **Prereorg utility**
  - To define which DBs are involved in SI
  - DFSURCDS control data set created
    - What information to collect in DFSURWF1?
    - What pointers need to be resolved later?
- **Prefix Resolution utility**
  - Information needed to create or to update a SI database
- **HISAM Unload and Reload utilities**
  - To recreate or build a new SI database
  - To merge/replace a SI in a shared SI DB
  - To extract a SI from a shared SI DB



Data Base Reload Processing with Secondary Indices

Reload with Logical Relationships

- **Prereorg utility**
  - To define which DBs are involved in LR
  - DFSURCDS control data set created
- **Database Scan Utility**
  - Used to scan DBs that are not reorganized but are involved in LR with DBs that are being reorganized
- **Prefix Resolution utility**
  - Accumulation/sort of the information needed to resolve LR
- **Prefix Update utility**
  - To update the prefix of each segment affected by the reorg of the dB



Data Sharing Considerations

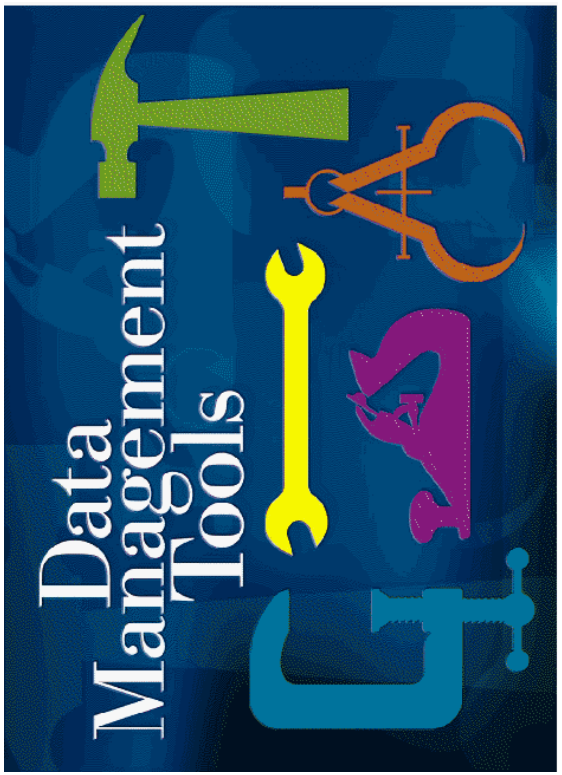
- **Before reorganizing a shared database**
  - No other subsystems should be authorized
  - Commands to prevent further authorizations except for reorganization and recovery utilities.

    Global /DBRECOVERY command for the database to be reorganized

    CHANGE.DB command with the NOAUTH keyword to manually update the RECON data set

    After the reorganization is complete, manually update the RECON data set by issuing the CHANGE.DB command with the AUTH keyword for the database that was just reorganized.
- **Recommendation: Ensure that recovery utilities do not run during the reorganization.**

IMS Database Reorganization Process using IBM Tools



IMS DB Reorganization - The Second Need

- **All IMS DB users have a need to reorganize DBs on a regular basis**
  - Pressure for increasing availability

- **The Needs**
  - First Need is to get utilities to perform the DB Reorg
    - Reliable ones
    - Basic support, serial processing
  - **Second Need** is to get Faster Utilities and Parallel Processing
    - To reduce this process to fit in the ever schrinking Batch Window
    - To get read access to the DB while it is being reorganized
  - Third Need is to get Online Utilities
    - To get Update access to the DB while it is being reorganized
    - Non disruptive process

- **The Answer**
  - Additional IBM Data Management Tools

Approaches for reorganizing IMS FF and HALDB database

IMS Base Utilities
IMS High Performance Utilities
IMS Parallel Reorganization

Future
IMS Online Reorganization
IMS HALDB Online Reorg.

IMS DB Reorg. - IBM Tools available Today

- **IMS High Performance Unload**
  - Replaces/improves the standard IMS HD Reorg Unload utility
- **IMS High Performance Load**
  - Replaces/improves the standard IMS HD Reorg Load utility
- **IMS High Performance Prefix Resolution**
  - Only needed when using logical relationships
  - Not needed with the new type of IMS databases available with IMS V7 (HALDB)
- **IMS Index Builder**
  - Only needed when using secondary indices, or HIDAM primary indices
  - Avoids taking image copies of indices
    For FF and HALDB databases
  - Recovers damaged indices without running the full reorg process
- **IMS Parallel Reorg V2**
  - A MUST!
  - Infrastructure to operate IMS HP Unload, IMS HP Reload and IMS Index Builder in parallel, allowing a significant reduction in the reorganization time.

IMS High Performance Unload - 2 Components

HP Unload

- Unloads HDAM, HIDAM, HISAM and SHISAM databases
  - With IMS V7, support of the new HALDB database types PHDAM, PHIDAM
- Provides an IMS application program interface
  - Assembler, COBOL, or PL/1
- Contains a high-performance DB retrieval engine
  - The *HSSR Engine*
  - High speed buffering technique similar to OSAM SB
  - Different from DL/I Buffering

> 80% reduction in EXCPs
> 75% reduction in CPU tim
(over base IMS unload)

Sequential Subset Randomizer

- Creates randomizing module for fast sequential processing of record subset
- Allows physical clustering of database records in same subset

IMS HP Unload in the Reorganization Process

- **Database unload utilities**
  - Unload a database in compressed or uncompressed format
  - Unload a broken HIDAM, HDAM, or HISAM database
    - Skip invalid HD pointers or invalid records
  - Create unloaded sequential data sets for use by application programs
  - Use dynamic allocation of DB datasets
- Easy to read output reports and statistics
  - DB statistics report or Randomizing statistics report
- Pre-tuned with defaults that should be adequate for most databases
- **Full support for HALDB**
  - Unload one, several or all partitions
  - Migration/fallback support
  - New function for user exit FABHEXTR
    - Specify that n records are to be unloaded from each HALDB Partition
      - *PARTEXTR control statement*

IMS HP Unload in the Reorganization Process ...

- **Similar capabilities for the two utilities**

- **Only in FABHURG1**
  - Unload into *CS format
  - Migration unload to HALDB and fallback unload from HALDB
  - Creation of a Database Extract
    - Using FABHEXTR user exit routine while running FABHURG1 utility
    - Reload performed with the standard IMS HD Reload utility DFSURGL0
- **Only in FABHFSU**
  - Parallel Scan Facility (PSF) mode
    - Produce multiple unload datasets
    - Concurrent run
  - Unload into HSAM format

- **IBM recommendations**
  - Use FABHURG1 unload utility if you want to
    - Unload the database in one of the five unload formats provided by FABHURG1.
    - Extract a part of the database by using FABHEXTR exit routine.
    - **Migrate to HALDB**
  - Use FABHFSU unload utility if you want to
    - Edit segment data or discard some segments on the basis of criteria you chose
    - Use the Parallel Scan facility function

IMS High Performance Load - 2 Components

HP Load Utility

- Performance replacement of IMS HD Reorganization Load utility (DFSURGL0)
- Complement to IMS High Performance Unload Tool
  - compressed/uncompressed input in various formats
  - dynamic allocation of DB datasets
- Full support for HALDB
  - With IMS V7, new HALDB database types PHDAM, PHIDAM
- Self Optimization

> 71% reduction in elapse time
> 92% reduction in CPU time
(over base IMS load utility)

HDAM Physical Sequence Sort for Reload (PSSR) Utility

- Previously in DBTools SMU
- Sorts the unloaded database data set before reload
  - Used with HALDB, when changing partition boundaries during reorg
- Avoids “cascading” on Reload

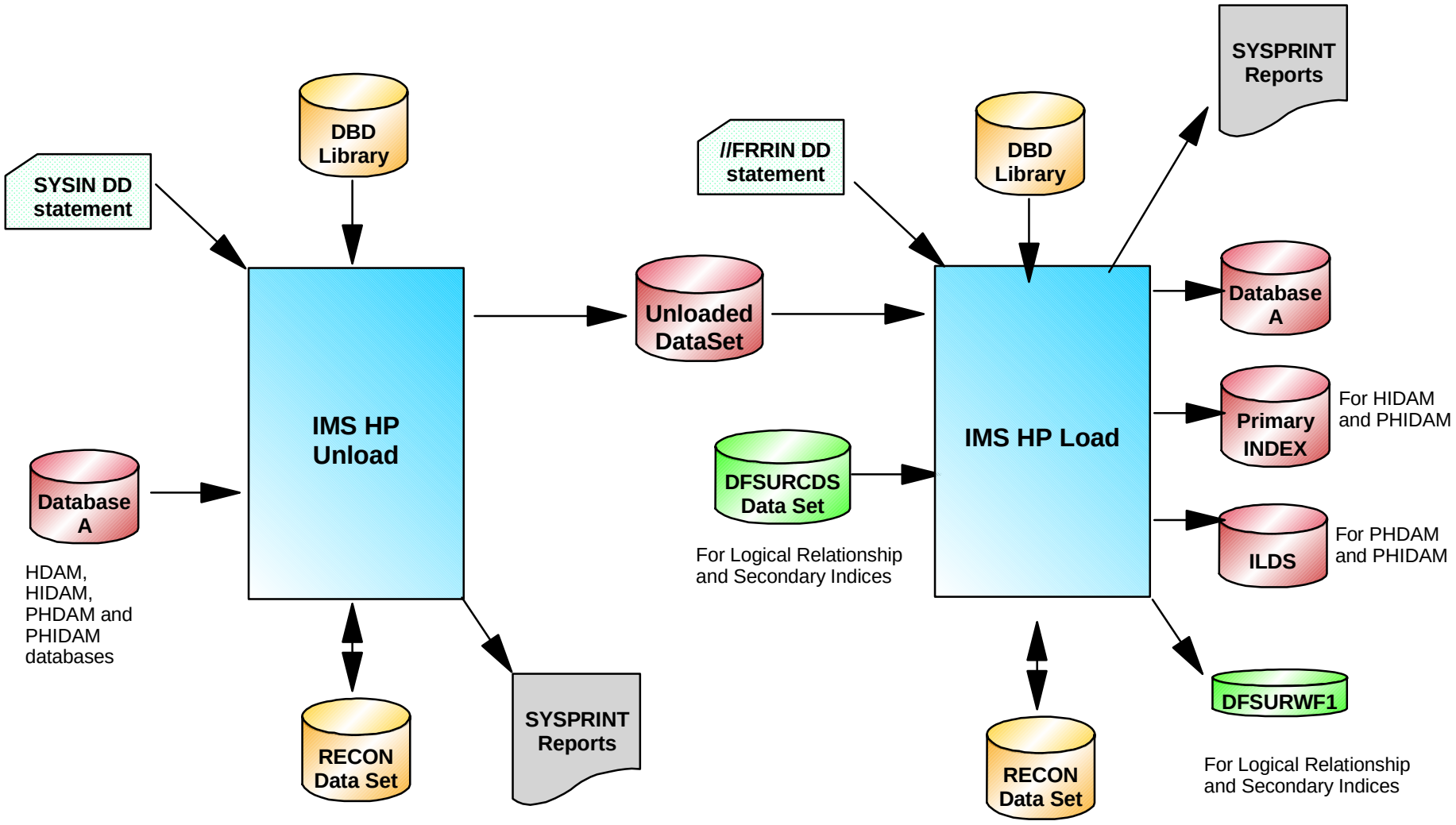
IMS HP Load in the Reorganization Process

- **Support of IMS HDAM, HIDAM, PHDAM and PHIDAM databases**
  - From an HD unloaded data set created by:
    - IMS High Performance Unload product
    - IMS HD Reorganization Unload utility (DFSURGU0)
  - Supports reloading of compressed segments without calling compression routine
  - Initializes empty HDAM and HIDAM databases
    - *DFSUINPT DD Dummy statement*
- **Support of HISAM and SHISAM with PQ53316**
  - DB Reload
  - DB Initialisation
- **Support of logical relationship or secondary indexes**
  - Creates a DFSURWF1 data set that can be used by:
    - Index Builder (IB) product
    - IMS High Performance Prefix Resolution
    - IMS Prefix Resolution utility (DFSURG10)
- **Support of HALDB**
  - Automatically initializes HALDB partition data set before reload
  - Provides a performance replacement for IMS Partition Initialization Utility
  - Creates ILDS
  - When changing partition boundaries during reorg of HALDB
    - PSSR may be used prior load to sort segments by RAP within partitions for PHDAM.

IMS HP Load in the Reorganization Process ...

- **User exit facility for additional processing of each segment**
  - Assembler, COBOL or PL/1
  - edit segment
  - delete segment (or this and subsequent segments of DB record)
  - force segment into overflow (HDAM or PHDAM)
  - force segment to start new block (HIDAM or PHIDAM)
- **Statistics reports to aid in tuning the database**
  - Reporting on space use and segment pointer statistics
- **Same Functions than IMS HD Reorganization Unload utility (DFSURGU0) with the following restrictions**
  - HDAM DBDs must specify nonzero max relative block nbr in the RMNAME operand of the DBD macro.
  - HP Load does not run under Utility Control Facility(UCF).
  - HP Load may use more DASD than IMS HD Reorganization Reload utility.
    Because of the difference between the free space search algorithms

IMS HP Unload and Load Data Flow

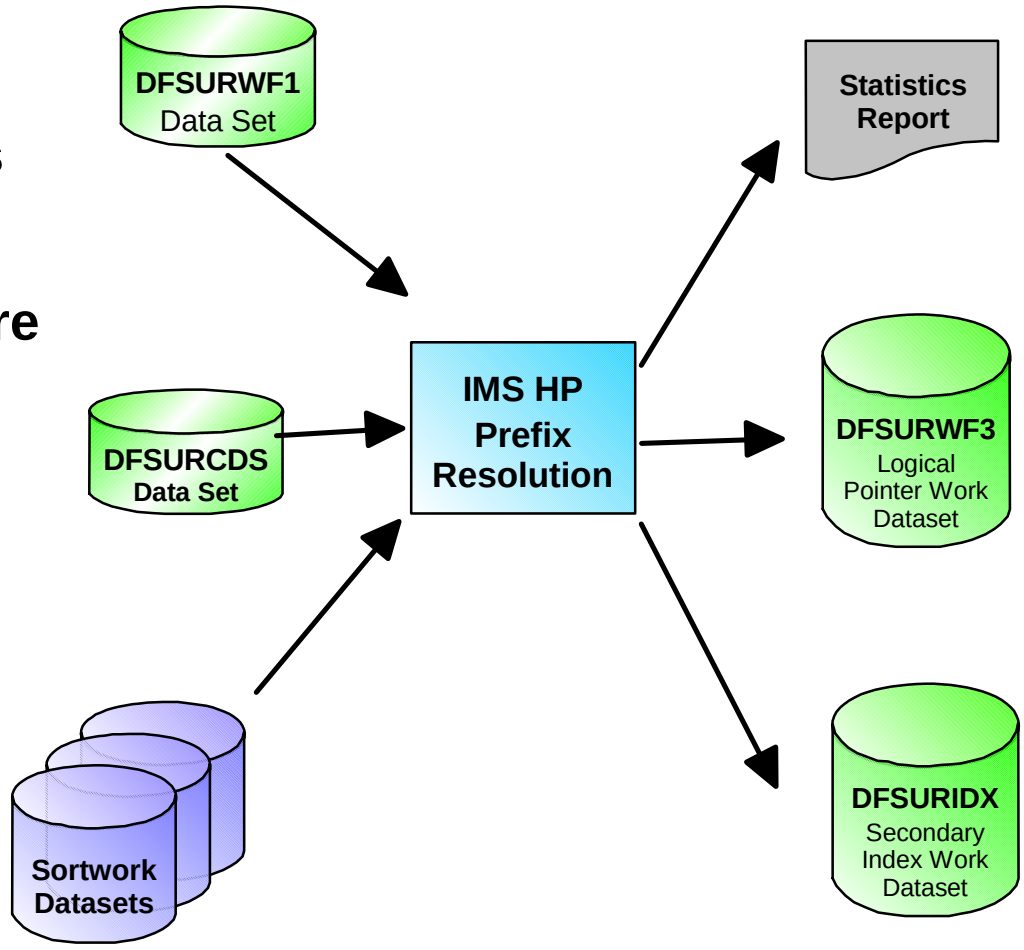


IMS High Performance Prefix Resolution

- **Replacement product for the IMS Database Prefix Resolution utility (DFSURG10)**
  - Reduction of elapsed time
  - Reduction in tape handling and DASD allocation
- **Creates a data set containing the information needed to resolve the logical relationship pointers and to create secondary index databases**
  - DFSURWF3 used as input to IMS Prefix Update utility (DFSURGP0)
  - DFSURIDX used as input to IMS HISAM Reorganization Unload utility (DFSURUL0)
- **Needs to be run after reorganization of Logically Related databases**
- **Eliminates the intermediate Works File 2 (WF2) data set**
  - **With** the use of IBM BatchPipes with Version 1
  - **Without** the use of IBM BatchPipes with Version 2
    HIPPRPIPE Data Transfer Service
- **Creates Statistical Reports**
  - Diagnostics and Summary of Logical Parents Without Logical Children
  - Statistics and Distribution of Logical Parents Based on the Number of Their Logical Children

IMS HP Prefix Resolution ...

- **Reduction in reorganization time when logical relationships or secondary indices are used**
- **Easy to use - no control statements required**
- **Output reports and statistics that are familiar to DBAs**



IMS Index Builder - Major Functions

- **Support for Full-Function non-partitioned DB and HALDB**
- **Creating New Secondary Indexes**
- **Rebuilding Secondary Indexes**
  - Using as input output from initial load or reload after a reorg (DFSURWF1)
  - Using as input a DL/I scan of the IMS database
  - Using as input the output from prefix resolution (DFSURIDX)
- **Creating Input for IMS HP Prefix Resolution**
  - Split Function
- **Initializing Empty Secondary Indexes**
- **Rebuilding a HIDAM Primary Index**
- **Using IMS IB for Recovery of Secondary Indices**
  - Change in the GENJCL.RECOV skeleton

To rebuild indices after reorganization process

To recover damaged secondary indices and avoid full reorganization process

To avoid taking IC of secondary indices

To minimize elapsed time

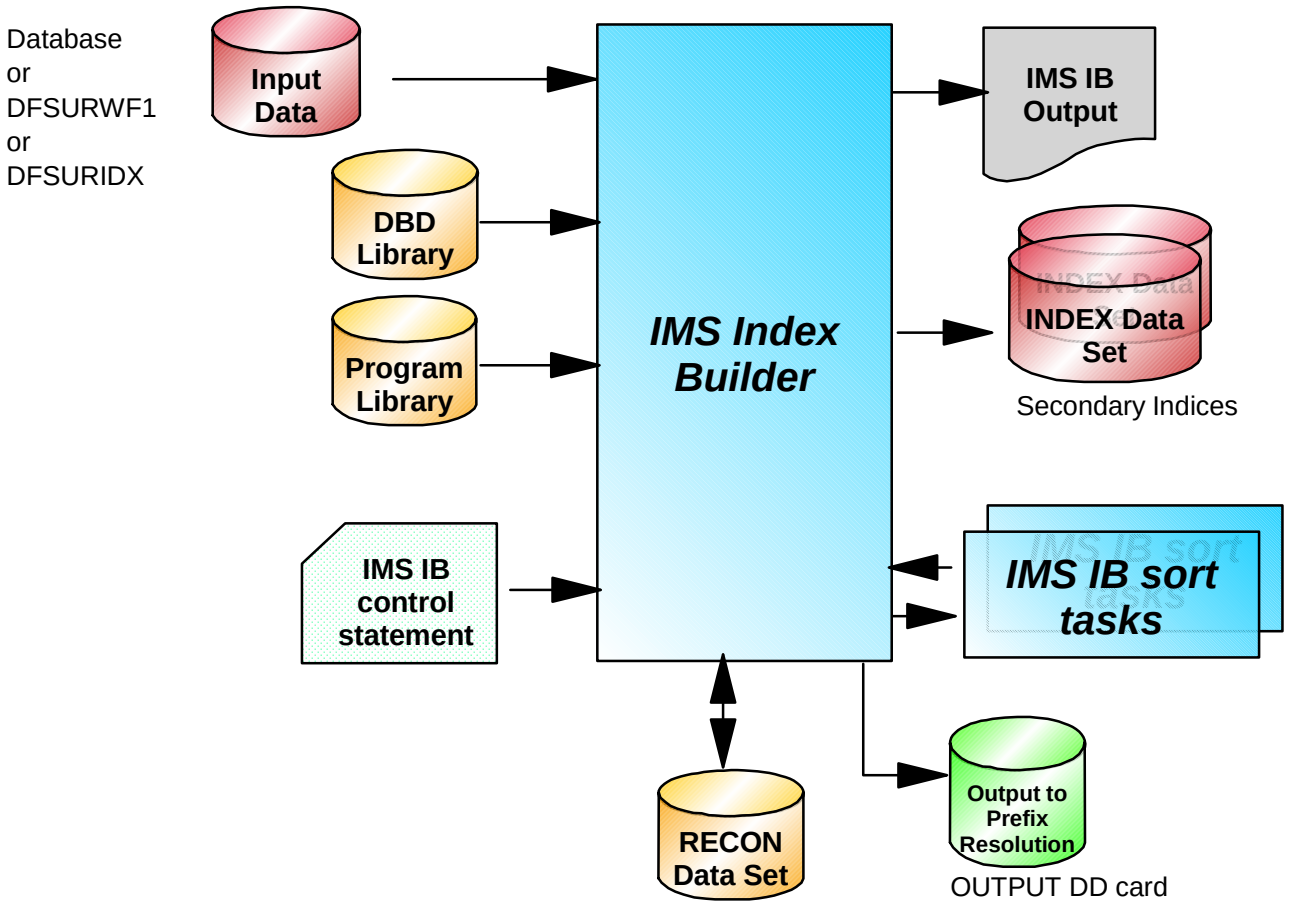
....

IMS Index Builder - Performance

- **IMS Index Builder achieves high performance by using parallel processing and by overlapping processing steps**
  - Uses a separate task for each secondary index
    - SORTs for all indexes performed in parallel
    - LOADs of all indexes are in parallel
  - Each sort uses fixed length records and is thus as fast as possible
  - Records are passed to SORT immediately when they are received from IPR Reload
    - so overlapping the input phase with the SORT phase
  - Sorting is done in the address space

IMS Index Builder in the Reorganization Process

- **Rebuilding Secondary Indices**
- **Dynamic allocation recommended**

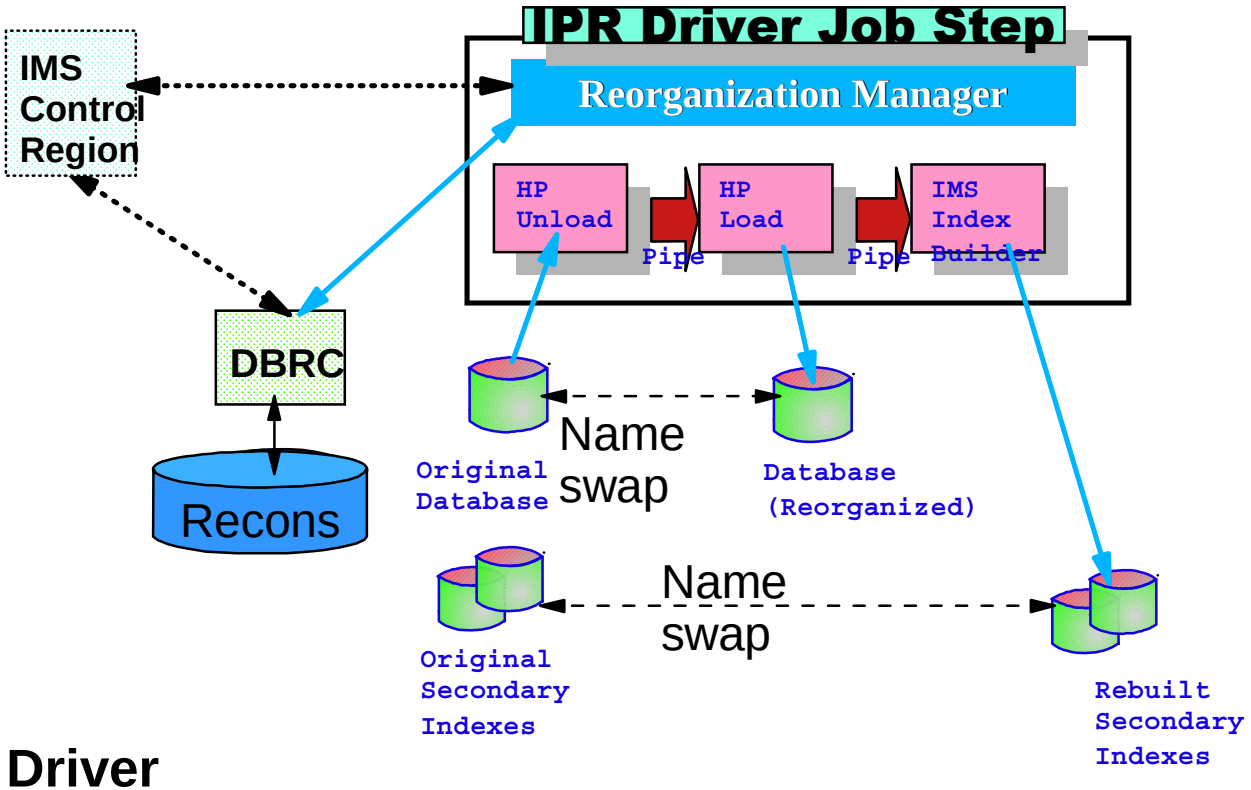


IMS Parallel Reorganization V2

- **Parallel (concurrent) use of *HP Unload*, *HP Load*, and optionally, *IMS Index Builder***
  - Reorganization of a full function DB and its secondary indexes
  - Data is reorganized into a shadow database (and shadow indexes)
    name swapping done at end
  - Support of HDAM, HIDAM
- **Very High Speed alternative to serial use of unload and reload tools/utilities**
  - Support of HDAM, HIDAM, HALDB and (S)HISAM
- **Appropriate for normal reorganization purposes**
  - reclaiming fragmented free space
  - bringing related segments close to each other
- **None structural changes to the DBD are allowed**
  - eg. adding a new field, making sequence field bigger, etc.

IMS Parallel Reorganization provides the infrastructure to operate IMS HP Unload, IMS HP Reload and IMS Index Builder in parallel (or independently), allowing a significant reduction in the reorganization time.

IMS Parallel Reorg. - Parallel Processing



- **Performed by the IPR Driver**
  - Dynamic allocation of Input and Output database datasets
  - Full DBRC support, including DBRC notification processing
  - Automated name swapping for input and output database data sets
    Provides both manual and automated post processing capabilities to rename the input and output database data sets and notify DBRC.
  - Automated IMS Command Processing
    /DBR or /DBD command

IPR Driver Components

- **HP Unload, HP Reload and IMS Index Builder Tools**

- **Data Transfer Service between tasks - IPRPIPE**
  - High performance data transfer service between tasks running in the IPR Driver address space
  - Between the HP Unload task and the HP Load task for transferring unloaded data
  - Between the HP Load task and the IMS Index Builder task for transferring DFSURWF1 secondary index data
    DFSURWF1 dataset will still be used for logical relationship data

- **Reorganization Manager - provides following functions**
  - Manages the overall process
  - Dynamic allocation of database data sets
  - Allowing valid DBD change during reorganisation
  - Automated IMS command processing during reorganisation
  - Automated data set name swapping
  - Automated DBRC notification processing after the name swapping

IPR Driver Components - Reorg. Manager

- **Dynamic Allocation**
  - available for input and shadow datasets
    - input name from DFSMDA
  - shadow name generated from input name
    - requires input dataset name be 42 characters or less
    - shadow DSN created by adding suffix (".x" where x is any letter except "T")
  - If using DBRC, dataset name is checked for consistency between DFSMDA and DBRC
  - Enables the DB reorg job to be parameter driven rather than JCL driven, leading to simpler and fewer sets of JCL
- **Automated IMS Command Processing**
  - Requested using IMSCMD=YES input parameter
  - Reorganisation Manager will issue
    - /DBDUMP DATABASE dbdname GLOBAL NOFEOV
    - **or**
    - /DBR DATABASE dbdname GLOBAL NOFEOV
  - Commands are issued through an E-MCS console for an online IMS subsystem
  - Alternatively, you can request a WTOR be issued prior to start of reorg
    - operators responsible for any necessary DB commands
    - AUTHFAIL=WTOR parameter

IPR Driver Components - Reorg. Manager ...

Post-Reorg Processing

- Occurs only if NAMESWAP=YES option specified
- Not supported for database with Logical Relationships
- 2 functions
  - Automated Data Set Name Swapping
  - Automated DBRC Notification

Automated Data Set Name Swapping

- Not available when non-HALDB DB has logical relationships
  - User reponsibility after all necessary utilities have been completed
- Reorganisation Manager uses standard IDCAMS services
- Requires input dataset name is 42 characters or less
  - input dataset name temporarily suffixed with ".T"
- Requires DBRC be active

Automated DBRC Notification

- Only if the data set name swapping succeeds
- DBRC NOTIFY.REORG command issued for the database and indexes
  - creates the REORG records and turns the IC NEEDED flags on
- Option to issue the DBRC NOTIFY.UIC commands to turn the IC NEEDED flag OFF for database and indexes that are registered as NONRECOV in DBRC

IPR Driver Components - Benefits

- **Simplified reorganization process**
  - Single job step, with full DBRC support
- **Very high speed reorganization**
  - Unload, Reload and Index Builder each have fewer I/Os
  - All three run in parallel
- **Enhanced data availability**
  - DB available for Read-Only for most of the reorganization process
  - If reorganization fails, original DB is still available for normal use
    - No need to run recovery
    - Re-run the reorg when ready
- **Enhanced function**
  - All the features of stand-alone HP Unload and HP Load are available
    - eg. user exits, SEARCH=, statistical reports, bypassing errors, etc

IPR Parallel Reorg JCL & Control Parameters

```
//*---------------------------------*
//* IMS HIGH PERFORMANCE UNLOAD  *
//*---------------------------------*
// EXEC FABHULU,MBR=FABHURG1,DBD=DBSAMP01
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD DSN=IMSDB.HDUNLD,DISP=(,CATLG),UNIT=TAPE
//SYSIN DD *
DECN
PROGMON=10000
/*
.........................................

//*---------------------------------*
//* IMS HIGH PERFORMANCE LOAD    *
//*---------------------------------*
//RELOAD EXEC PGM=DFSRRC00,
       PARM='ULU,DFSURGL0,DBSAMP01,,,,,,,,,,Y,N'
//STEPLIB  DD DSN=HPS.SHPSLMD0,DISP=SHR
//         DD DSN=IMSESA.RESLIB,DISP=SHR
//DFSRESLB DD DSN=IMSESA.RESLIB,DISP=SHR
//IMS      DD DSN=IMSESA.DBDLIB,DISP=SHR
//DFSUINPT DD DSN=IMSDB.HDUNLD,DISP=(OLD,PASS)
//DFSURWF1 DD DISP=(NEW,PASS),DSN=IMSDB.DFSURWF1,UNIT=TAPE
//DFS
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//DFSVSAMP DD *
512,4
/*
//FRRIN DD *
DATASPACE=YES
PROGMON=10000
/*
//*---------------------------------*
//* IMS INDEX BUILDER            *
//*---------------------------------*
//IMSIB EXEC PGM=IIUSTART
//STEPLIB DD DISP=SHR,DSN=IIU.V2R1M3.SIIULMOD
//        DD DISP=SHR,DSN=IMSESA.RESLIB
//DFSRESLB DD DISP=SHR,DSN=IMSESA.RESLIB
//IMS DD DISP=SHR,DSN=IMSESA.DBDLIB
//DFSURWF1 DD DISP=OLD,DSN=IMSDB.DFSURWF1,BUFNO=99
//IIUIN DD *
PROC BLD_SECONDARY,TSTDATA,ALL
INPUT DFSURWF1
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//IIUSTAT DD SYSOUT=*
//IIUSNAP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
```

```
//REORG EXEC PGM=HPSGMAIN,PARM='DBD=DBSAMP01,DBRC=Y'
//STEPLIB      DD DISP=SHR,DSN=HPS.SHPSLMD
//             DD DISP=SHR,DSN=IMSESA.RESLIB
//IMS DD DISP=SHR,DSN=IMSESA.DBDLIB
//HPSOUT DD SYSOUT=*
//IIUPRINT DD SYSOUT=*
//HPSOUT2 DD SYSOUT=*
//IIUSOUT DD SYSOUT=*
//HPSIN DD *
(REORG)
IMSCMD=YES
NAMESWAP=YES
INDEXBLD=YES
(UNLOAD)
PROGMON=10000
DBSTATS=YES
(RELOAD)
DATASPACE=YES
PROGMON=10000
(INDEXBLD)
MAXTASKS=5
/*
```

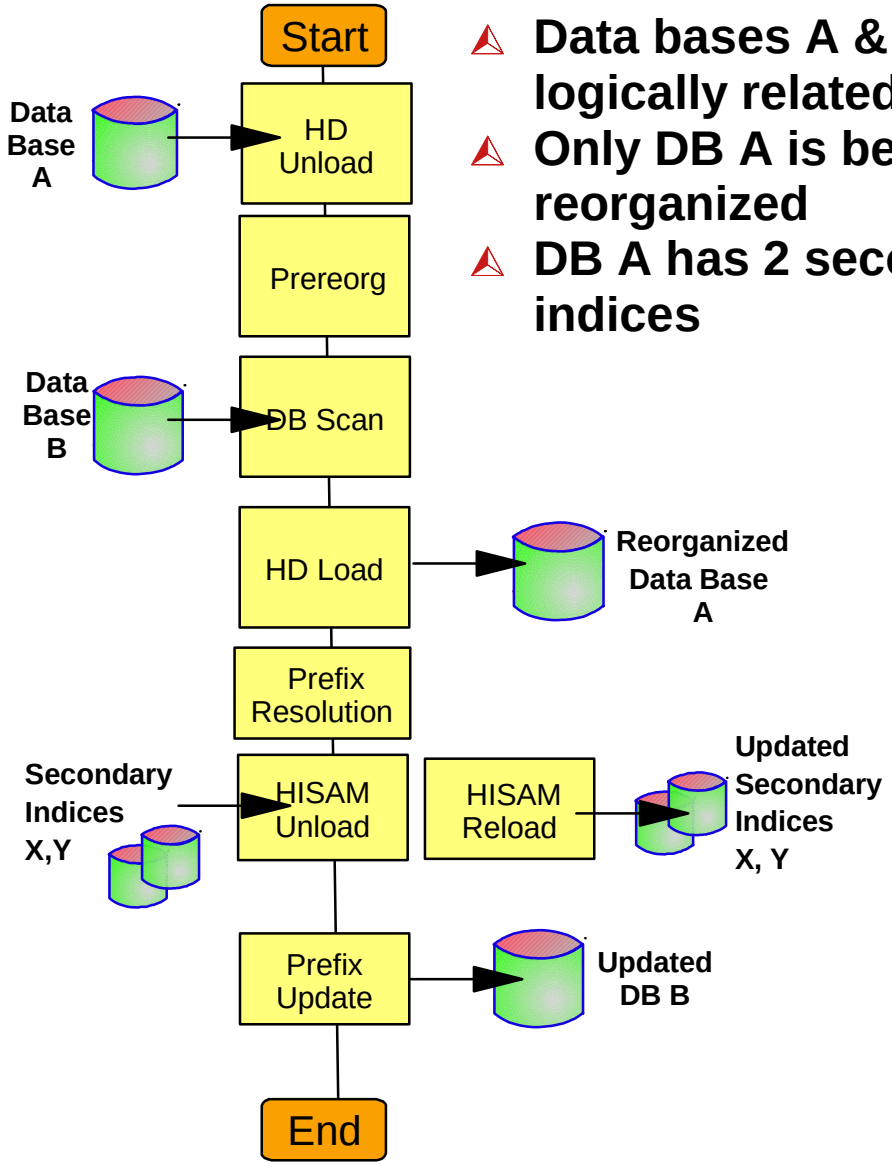
Reorganisation Manager parameters

DECOMPRESS=N is default

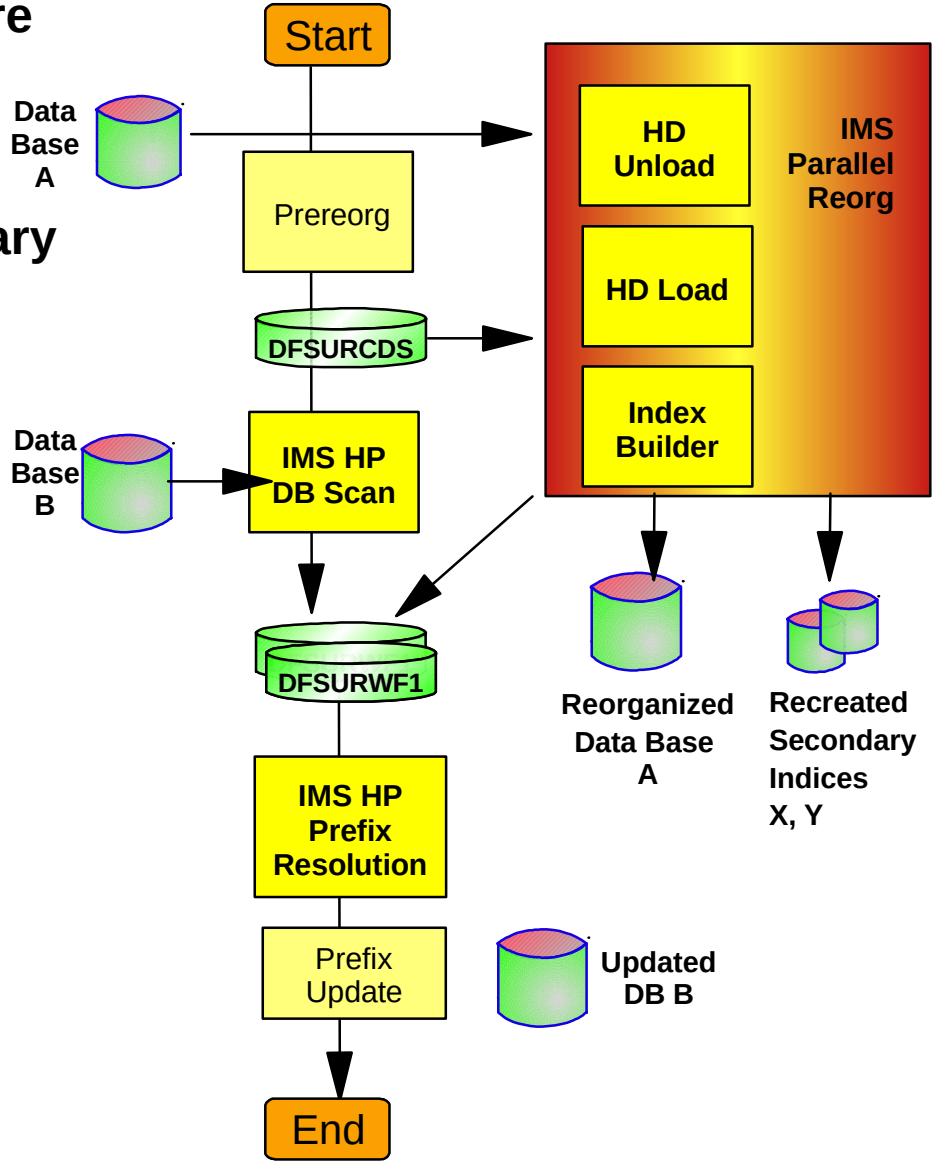
Reload Parameters are same as for HP Load

IPR chooses its own parameters for index builder

Example of Complex Reorganization Process



- Data bases A & B are logically related
- Only DB A is being reorganized
- DB A has 2 secondary indices



IMS Database Reorganization Process

Conclusion

How to get a Well-Organized Database?

- **Provide enough embedded free space at database load or reorganization time.**
  - Free space used to insert new segments near their related segments
  - No DB size limit with HALDB!
- **Select an appropriate database reorganization frequency**
- **Use efficient HDAM and PHDAM randomizing modules and randomizing parameters**

The Best Reorg is No Reorg!

Choose the amount of free space based on the growth and performance characteristics of your database.

For new databases, use a large number (40 to 60%), and increase or decrease this value as needed.

It is a good idea to schedule a reorganization for the database before the performance suffer, maybe when the reusable free space is less than 10%.

Bibliography

- **IMS V7 Administration Database**
  - Chapter 14. Tuning Your Database
- **IMS V7 Utilities for the IMS DB and IMS TM**
  - Part1 - Definition, Migration, Initialization, and Reorganization Utilities
  - Part 6 - Utility Control Facility
- **IMS HP Unload**
  - License Product ID: 5655-E06
  - User's Guide: SC27-0936
- **IMS HP Unload**
  - License Product ID: 5655-E07
  - User's Guide: SC27-0938
- **IMS HP Prefix Resolution**
  - License Product ID: 5655-E08
  - User's Guide: SC27-1590
- **IMS Index Builder**
  - License Product ID: 5655-E24
  - User's Guide: SC27-0930
- **IMS Parallel Reorg**
  - License Product ID: 5655-F74
  - User's Guide: SC27-1180