

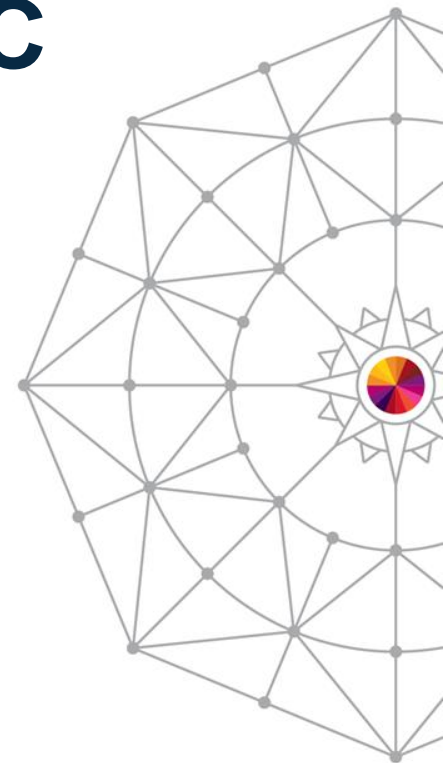
# IMS 13 Database and DBRC Enhancements

*Nancy Stein*  
*IBM – zGrowth Team / IMS*

*Wednesday, August 6<sup>th</sup>, 2014*  
*Session #16176*  
*Room 402*



#SHAREorg



# IMS 13 Database and DBRC Enhancements

- Database Versioning
- HALDB Alter
- DEDB Alter
- DBRC Enhancements

# IMS 13 Database Versioning

# Database Versioning

- IMS 13 allows application programs to use different versions of the same physical database
  - Multiple views of the physical data are maintained in the IMS catalog
  - Application programs can use different views of the same physical IMS database
- Benefit
  - Customers can support multiple versions of an IMS database
  - Physical database structure can be changed without having to modify all the existing application programs using the database

# Database Versioning Overview

- Provides the ability to assign user-defined version identifiers to different versions of an IMS database structure
- Enables structural changes to a database while providing multiple views of the physical IMS data to various applications
- Applications referencing a new physical database structure can be brought online without affecting applications that use previous database structures
- Applications not requiring sensitivity to the new physical database structure can continue to access the database without any modifications or recompilation

**Can be used in conjunction with IMS 13 Database ALTER function.**

# Database Versioning Overview (cont'd)

- Database Versioning supports the following database types
  - DEDB
  - HDAM
  - HIDAM
  - PHDAM
  - PHIDAM
- Database Versioning supports the following structure changes to a database
  - For all supported database types
    - Increasing the length of a segment
    - Adding a new field to undefined space at the end of a segment
  - For Full-Function and HALDB database types only
    - Adding a new field that defines an alternative mapping of bytes in a segment

# Database Versioning Overview (cont'd)

- Database Versioning requires IMS catalog enablement
  - DBD version definitions must be stored in the IMS catalog
    - Populate catalog with DBD version definitions
- All systems in an IMSplex must be running IMS 13
  - DBRC MINVERS value of “13.1” required

# Database Versioning Overview (cont'd)

- Before Database Versioning is enabled for a database
  - IMS continues to only recognize the current active database definition
- Version numbers must be maintained in incremental values
  - Specified on the DBD
- Applications programs can specify a desired database version
  - System default setting
  - PSB setting
  - PCB in the PSB
  - DL/I INIT call



# Database Versioning Implementation

- DFSDFxxx PROCLIB: new keywords in DATABASE section
  - DBVERSION= new keyword to enable database versioning
  - DBLEVEL= new keyword to indicate default DB version to be used
- Database and Program Generation Statements
  - DBD: DBVER= database version number
  - PCB: DBVER= database version number
  - PSBGEN: DBLEVEL= overrides the DBLEVEL= specified in the DFSDFxxx PROCLIB member
- “INIT VERSION” DL/I Call
  - Program can set a specific version of a specific database
  - Overrides all other version number specifications for a database
  - Must be issued before issuing a DL/I call to the database

# Implementation – DFSDFxxx PROCLIB Member

- New keywords added to DFSDFxxx DATABASE section
  - DBVERSION = Y | N
    - Enables database versioning
    - Database versioning is disabled by default
  - DBLEVEL = CURR | BASE
    - Ignored when DBVERSION=N
    - CURR (default)
      - *IMS returns data from all databases using the current DBD version, which is the current physical level, unless a specific version is specified*
    - BASE
      - *IMS returns data from all databases using the lowest DBD version number retrieved from IMS Catalog, unless a specific version is specified*

# Implementation – DBD Generation Statement

- New parameter added to DBD Statement
  - DBVER=*n*
    - Specifies a DBD version number to be associated with a database structure change
    - Supports ACCESS types DEDB, HDAM, HIDAM, PHDAM & PHIDAM
    - Numeric values from 1 – 2147483647 (2 Gigs)

```

>>-DBD--NAME= (dbname1)--,ACCESS=- (---+--+HDAM---+--+OSAM--+--+)-+-----+----->
|      |      |      |      |      |      |      |      |      |      |      |      |
| -HIDAM-- |      |      |      |      |      |      |      |      |      |      |
| -PHDAM-- |      |      |      |      |      |      |      |      |      |      |
| -PHIDAM- |      |      |      |      |      |      |      |      |      |      |
|          |      |      |      |      |      |      |      |      |      |      |
|-----DEDB-----|      |      |      |      |      |      |      |      |      |
|          |      |      |      |      |      |      |      |      |      |      |
|          |      |      |      |      |      |      |      |      |      |      |
...

```

# Implementation – PCB Statement in PSB

- New parameter added to PCB Statement
  - DBVER=*n*
    - Specifies the version of the DBD to use when accessing the database
    - Must match a defined DBD version number stored in the IMS catalog
    - If multiple PCBs within a PSB refer to the same database, each PCB must specify the same DBD version number
    - Numeric values from 0 – 2147483647 (2 Gigs)
    - If not specified, the DBD version used depends on DBLEVEL= parm in the PSBGEN statement or the DFSDFxxx PROCLIB member

```

>>+-----+--PCB--TYPE=DB--+-,DBDNAME=--+--name-----+-----+----->
   | -label-----|                | -,NAME=-----|                | -,DBVER=n-|
...

```

# Implementation – Program Generation Statement

- New parameter added to PSBGEN Statement
- DBLEVEL=CURR | BASE
  - Specifies the default database version level returned to programs using this PSB and not requesting specific database versions
    - *DBLEVEL=CURR*
      - *PCBs within the PSB will access the database using the current physical structure*
    - *DBLEVEL=BASE*
      - *PCBs within the PSB will use the lowest base version in then IMS catalog*
  - Overrides the default setting for DBLEVEL specified in DFSDFXxx PROCLIB member

```

>>-PSBGEN--PSBNAME=-name--+-----+----->
      |                                     |
      |-,DBLEVEL=-+-CURR-+-|
      |   -BASE-   |
  
```

...

# Implementation – DL/I INIT Call

- New VERSION function added to the INIT call interface
  - Application program can specify the database name(s) and the DBD version(s) to be used when making DL/I calls to the database(s)
  - Specified version number must match a version number defined on a DBD for the named database and stored in the IMS catalog
  - Takes precedence over all other version number specifications and defaults (ie. PCB statement, PSBGEN statement, DFSDFxxx)
  - An INIT call used to version a database has to be executed prior to the first DL/I call for the database, but not before first “GU” call to IOPCB
  - Can only issue one INIT VERSION call for a specific database within an application

# Implementation – DL/I INIT Call (cont'd)

- I/O Area contains the “VERSION” function and database parms
  - “VERSION(*dbnameA=version#, ..., dbnameZ=version#, ...*)”
    - *dbname*: specifies a physical database name
    - *version*: specifies a DBD version number to be used when accessing the database
    - separate sub-parameters with a comma
    - no duplicates allowed

DLI call:

```
>>-INIT---+ i/o_pcb +---+ i/o_area -----><
          | -aib -----|
```

i/o area:

```
          .---.-----
          |                                     |
          v                                     |
>>-VERSION(--- dbname=version +---)-----><
```

# Segment Versioning Example

## “CIF” Customer Account Segment

Account Number	Member Name	Balance	Credit Limit

- Database Versioning enabled with DBLEVEL=BASE
- **BASE** version of the customer account segment
- Segment is fixed length
- No space remaining for additional fields



# Field Added to Existing DB Segment

## “CIF” Customer Account Segment

Account Number	Member Name	Balance	Credit Limit	Reward Points

- New version of the customer account segment defined in DBD
  - Expands the length of the segment
  - New field called **Reward Points** defined in expanded space
  - CIF DBD defined as DBVER=201401
- Existing applications are not updated
  - Existing applications do not have to know the new field exists
  - Existing applications do not see or update the new field

# Segment READ/REPLACE Behavior

- Application scheduled using the **BASE** version of the database
  - Read a “CIF” Customer Account segment
  - Update the Balance to 200,000 with a REPL call
    - Only the credit balance field is changed
    - Reward Points field and others are left unchanged

## Customer Account segment before replace

Account Number	Member Name	Balance	Credit Limit	Reward Points
555555	Vern Watts	100,000	500,000	50,000

## Customer Account segment after replace

Account Number	Member Name	Balance	Credit Limit	Reward Points
555555	Vern Watts	<b>200,000</b>	500,000	50,000

# Segment READ/REPLACE Behavior

- **Application scheduled using the 201401 version of the database**
  - Read a “CIF” Customer Account segment
  - Update the Balance to 250,000 and Reward Points to 100,000 with a REPL call
    - Both the Balance and the Reward Points fields are changed
    - Others fields are left unchanged

## Customer Account segment before replace

Account Number	Member Name	Balance	Credit Limit	Reward Points
555555	Vern Watts	200,000	500,000	50,000

## Customer Account segment after replace

Account Number	Member Name	Balance	Credit Limit	Reward Points
555555	Vern Watts	<b>250,000</b>	500,000	<b>100,000</b>

# Database Version Determination

- If database versioning is enabled, the database version used to return the IMS data to a program is determined as follows:
  - ✓ **DFSDFxxx DATABASE Section DBLEVEL=** parameter
  - ✓ **PSBGEN Statement DBLEVEL=** parameter
  - ✓ **PCB Statement DBVER=** parameter
  - ✓ **DL/I "INIT VERSION"** call

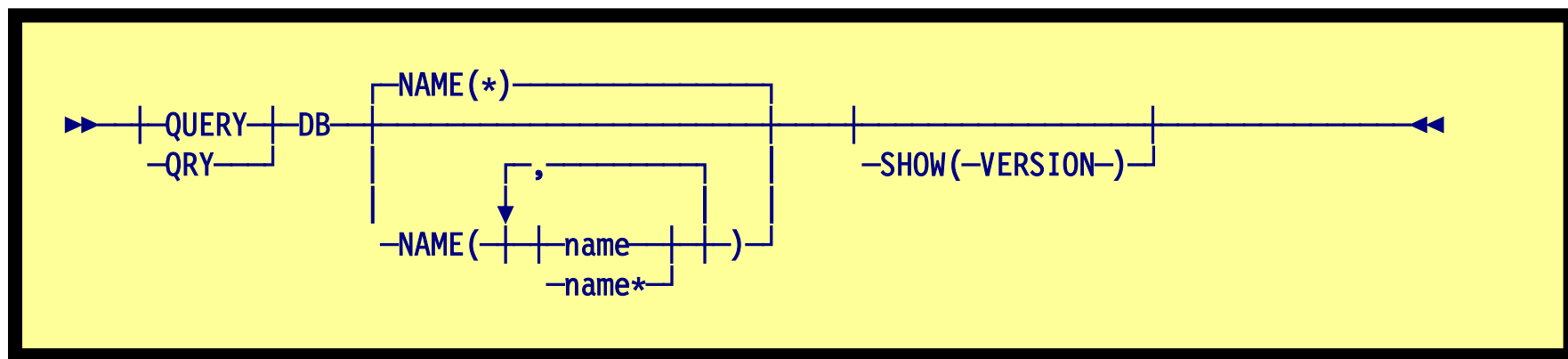
**DL/I INIT VERSION call takes precedence over all other database version number specifications and defaults**

# Database Versioning Operational Considerations

- When an application program requires a database version not currently the active version:
  - IMS will go to the Catalog to get the information for the requested DBD version
  - IMS will compare the Catalog info for the DBD to the currently active DBD information and determine the differences
  - If the IMS Catalog is not enabled, an “NA” status code is returned because the catalog database is not available
- When database versioning is enabled and the application program does not request a version:
  - By default IMS will retrieve the database data at the current physical level
  - Unless the DFSDfxxx default behavior is overridden with the DBLEVEL=BASE parameter

# Enhanced Type-2 QUERY Command

- QUERY DB
  - New SHOW(VERSION) option
  - Returns the version number of a database that is currently active in the online IMS system
    - For HALDB Master – response shows master and each partition
    - For HALDB Partition – response shows only the requested partition
    - For FP DEDB – response show only the DEDB
  - Cannot specify the VERSION filter with other SHOW filters



# Enhanced Type-2 QUERY Command

- QUERY DB Output
  - Output headers for successful commands

QUERY DB NAME(*dbname*) SHOW(VERSION)

```
DBName          PartName      MbrName      CC      TYPE          VERSION
-----          -
```

- Output headers for commands with errors

QUERY DB NAME(*dbname*) SHOW(VERSION)

```
DBName          PartName      MbrName      CC      CCText          TYPE          VERSION
-----          -
```

Completion Code	Completion Code Text	Meaning
0		Command completed successfully for the resource.
10	Resources not found	<ul style="list-style-type: none"> <li>• The resource name is unknown to the client that is processing the request. The resource name might have been typed in error or the resource might not be active at this time. Confirm the correct spelling of the resource name specified on the command.</li> <li>• For full function databases that have not been opened or initialized.</li> <li>• If the command is used to query the version number of a HALDB partition, the output will list a response line for just that partition. If the partition is disconnected from the HALDB master, possibly due to a DBR of the database, then the output will show a completion code of `10` in the CC column to indicate no resource found.</li> </ul>
195	Unsupported DB type	Database versioning for this database access type is not supported.

# Database Versioning Migration Considerations

- Populate the IMS catalog with ACBLIB definitions (DBD/PSB)
- Enable the IMS catalog
  - Before Database Versioning is enabled for a database, IMS continues to only recognize the current physical database definition
- All IMS systems in an IMSplex must be running IMS 13
- Enable IMS Database Versioning
  - Specify the new parameter to enable a default database versioning
    - Use the default DBLEVEL=CURR setting in DFSDFxxx so all applications access databases at the latest, physical DB version
    - Set the DBLEVEL=BASE parameter in DFSDFxxx so all applications access databases at the oldest, lowest DB version
  - Use the PSBGEN DBLEVEL parm or the PSB PCB DBVER= parm if an application needs to use a version of a database different from the system default level
  - Use DLI INIT VERSION call is needed for dynamic version switching



# Database Versioning Summary

- IMS 13 allows application programs to use different versions of the same physical database
  - Multiple views of the physical data are maintained in the IMS catalog
  - Application programs can use different views of the same physical IMS database
- Benefit
  - Customers can support multiple versions of an IMS database
  - Physical database structure can be changed without having to modify all the existing application programs using the database

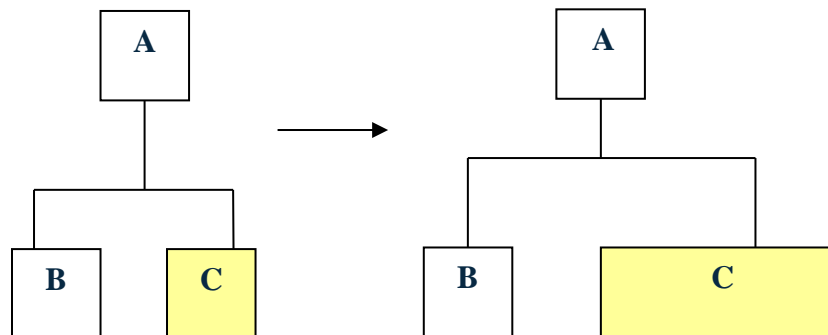
# IMS V13 HALDB Alter

# HALDB Alter

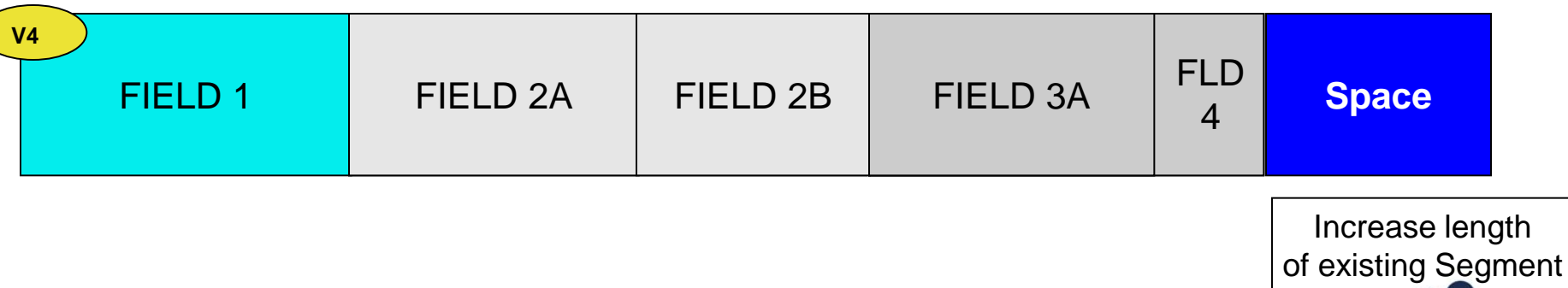
- IMS 13 provides ability to make structural changes to a HALDB database without a database outage
  - Structural changes can be made to DB segment definitions
  - HALDB Online Reorganization is used to apply the structural changes to the online database
  - Online Change process is used to activate the new ACBLIB member(s) in the online IMS system
- Benefit
  - Eliminate a database outage when structural changes must be made to segment definitions in a DBD
  - Improved online availability of HALDB databases

# HALDB Alter Overview

- Structural changes can be made to online HALDB segments
  - PHDAM
  - PHIDAM
- Types of structural changes
  - Add a new field to space at the end of an existing segment
  - Increase the length of an existing segment
  - Define new fields to remap fields and space in an existing segment



# HALDB Alter Overview (cont'd)



# HALDB Alter Overview (cont'd)

- Modify the DBD source definition(s) and run gen(s)
- Online Reorganization (OLR) to ALTER the online database from the current structure to the new structure
  - DB unload / reload not required
  - Requires Type-2 INITIATE OLREORG
    - /INITIATE OLREORG does not support altering a HALDB
- Online Change (OLC) to activate the changed ACBLIB members in the IMS online system
- Application programs can start using the new database structure

# HALDB Alter Preparation

- Modify the DBD source code
  - Define new fields in space at the end of segment(s)
    - Specify new FIELD statements
  - Define new fields to remap existing fields & space in segment(s)
    - Specify new FIELD statements
  - Increase the length of segment(s)
    - Specify new length in the BYTES= parameter of SEGM statement
- Run DBDGEN
  - Modified DBD source used as input
- Run PSBGEN(s) (if needed)
  - Modified PSB source used as input
- Run ACBGEN
  - Create a new member in a **staging** ACBLIB
- Modify or code new application programs to use new fields

# HALDB Alter Preparation - DBRC

- CHANGE.PART Command
  - Use to set attributes for a HALDB partition before it is altered
    - New ALTERSZ keyword to set block/CI size of output partition data sets
    - New NOALTRSZ keyword to clear block/CI size of output partition data sets
  - Sizes cannot be changed if the HALDB is currently being altered

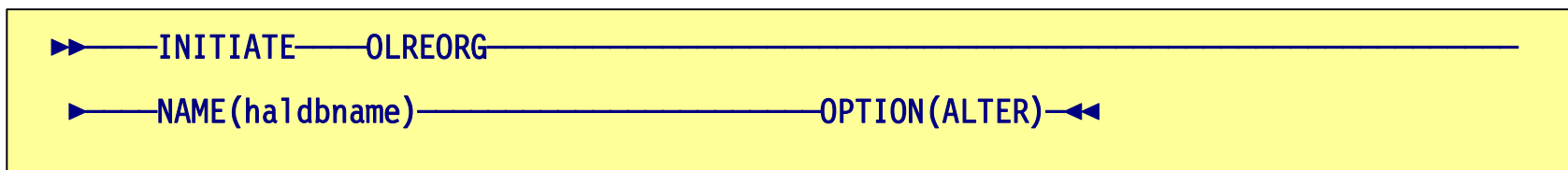
```

▶▶ CHANGE.PART—DBD (name)—PART (name)—
      |-----|
      |ALTERSZ (nnnnn)|
      |NOALTRSZ      |
      |-----|
      ▶
  
```



# HALDB OLR Alter Command

- **INITIATE OLREORG**
  - Use Online Reorg to ALTER the structure of a HALDB database
    - Applies structural changes to the online HALDB database
    - All database partitions are included in the reorg ALTER process



Keyword	Function
NAME()	<ul style="list-style-type: none"> <li>• specifies the name of a HALDB master</li> <li>• only one HALDB master database name can be specified</li> <li>• you cannot use the wildcard character (*)</li> </ul>
OPTION()	<ul style="list-style-type: none"> <li>• Specifies the different options that will affect how HALDB online reorganization performs (all options are valid)</li> </ul>
ALTER	<ul style="list-style-type: none"> <li>• specifies that the command will initiate the database structural alter processing</li> </ul>

# HALDB Alter Online Process

- Issue Type-2 Initiate Online Reorganization command

**INITIATE OLREORG NAME(masterdb) OPTION(ALTER)**

- Initiates the dynamic structural change processing for all the partitions of this HALDB
- Only one HALDB master database name can be specified per INIT command
- IMS reads the current DBD version from the active ACBLIB
- IMS reads the new DBD definition from a staging ACBLIB
  - Staging ACBLIB is dynamically allocated
  - Member info is used to construct the OLR output data set
  - Output data sets for all the HALDB partitions are built
- If necessary, 10 TCBs will be scheduled concurrently for ALTER

# HALDB Alter Online Process

- Application programs accessing the existing database, using the current version of the DBD, continue running
- Internal tables are built to represent the changes between the active/input DMB and the staged/output DMB
- Type-2 TERM OLREORG or Type-1 /TERM command is allowed while database structure change is in progress
- Type-2 INIT OLR with OPTION(ALTER) command will restart the HALDB ALTER structure process where it left off

# HALDB Alter Online Process (cont'd)

**For application programs accessing the altered database:  
Until OLC is completed, IMS reads the database using  
the unaltered DBD and returns the unaltered segment structure**

- Stop access to the altered HALDB database
- Use Type-2 UPDATE or Type-1 /DBR command
- Do not use UPDATE START(QUIESCE)
- Issue Online Change commands to complete the alter process for the changed ACBLIB member(s)
  - Member Online Change is recommended
    - Reads directly from the staging ACBLIB
    - Can process specific ACBLIB member(s) requiring activation

**INITIATE OLC TYPE(ACBMBR) NAME(acbmember)**

# HALDB Alter Online Process (cont'd)

- Start access to the altered HALDB database

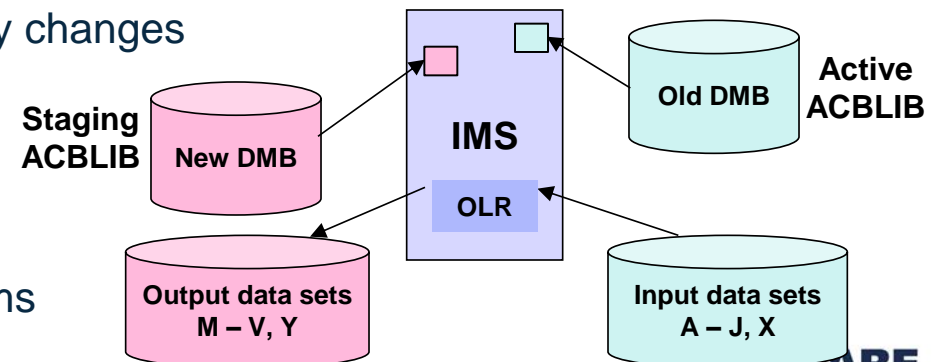
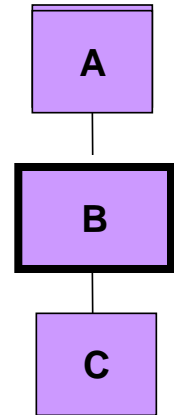
**New HALDB database structure can now be used by modified or new application programs needing the new segment fields**

- Implement new or modified IMS application programs

# HALDB Alter Example

- Update DBD source for MASTER database
  - Increases the size of segment B from 30 bytes to 40 bytes
- Run DBDGEN
- Run ACBGEN(s) into a staging ACBLIB
- Make updates to affected application programs / create new programs
- DBRC CHANGE.PART ALTERSIZE() to alter BLK or CI size, if necessary
- Issue INIT OLREORG NAME(MASTER) OPTION(ALTER)
  - IMS allocates Staging ACBLIB
  - IMS builds input DMB control blocks using the Active ACBLIB
  - IMS builds output DMB control blocks using the Staging ACBLIB
  - IMS allocates the output database data sets
- Backup Active ACBLIB members affected by changes
- Stop MASTER DB access
- Perform Online Change
- Start MASTER DB access
- Implement new/updated application programs

MASTER DB

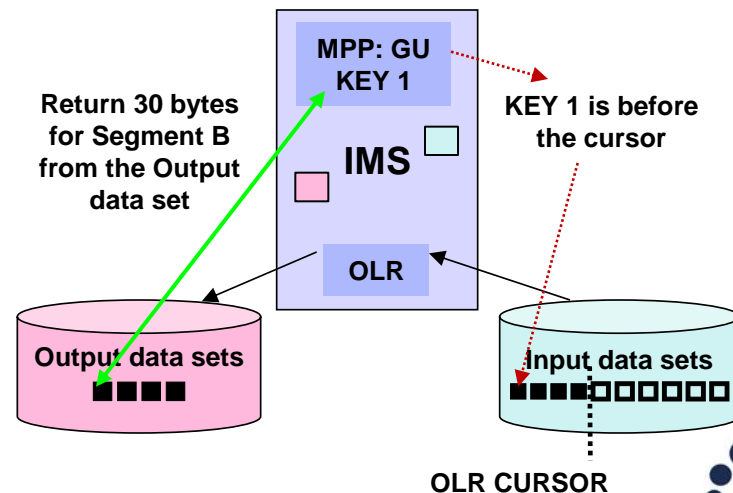


# HALDB Alter Example (cont'd)

- As part of building the input and output DMB control blocks, IMS compares the segments and fields and creates a table of the deltas
- When ALTER is in progress, an application program reads “MASTER” looking for segment “B” with KEY1
- If the segment key is before the OLR cursor
  - IMS reads segment B from the output data set whose size is 40 bytes
  - IMS checks the delta table before returning the segment to the application
  - Based on the table, IMS **returns 30 bytes** of data to the application

Delta Table

Name	Input	Output
B	30	40



Until ALTER and OLC are complete, IMS reads the segment using the old DMB and returns the unaltered segment structure.

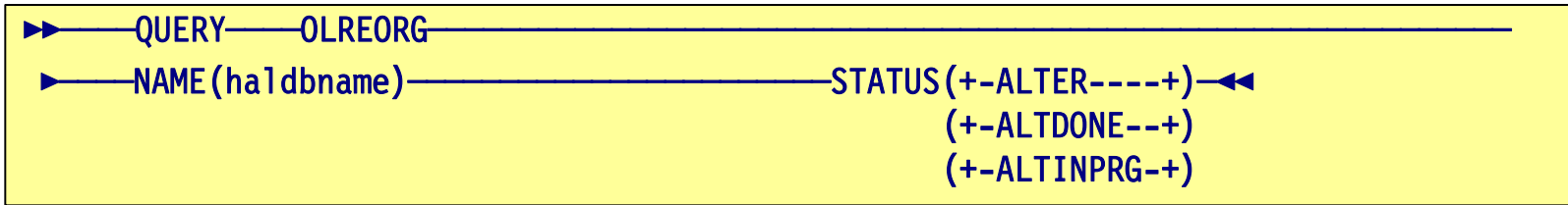
# HALDB Alter Segment Field Fill Values

- For fields added at the end of a segment
  - If DBD field definition is TYPE 'X'
    - Fill with x'00'
  - If DBD field definition is TYPE 'P'
    - Fill low order byte with x'0C' and other bytes with x'00'
  - If DBD field definition is TYPE 'C'
    - Fill with x'40'
- When space without field definition(s) added to a segment
  - Fill with x'00'



# HALDB Alter Type-2 QUERY Command

- QUERY OLREORG
  - Use QUERY to check on ALTER processing for a HALDB database



Keyword	Function
NAME() <i>optional</i>	<ul style="list-style-type: none"> <li>• NAME(*) is the default, to query all defined HALDB partitions</li> <li>• specifies either one or more partition names or the name of a HALDB master</li> <li>• wildcard character (*) is not allowed, except as NAME(*)</li> </ul>
STATUS()	<ul style="list-style-type: none"> <li>• allows you to display the online reorganizations that possess a specified status</li> </ul>
ALTER ALTDONE ALTINPRG	<ul style="list-style-type: none"> <li>• specifies type of output for the partitions of the HALDB master, identified on the NAME() parameter:                             <ul style="list-style-type: none"> <li>- status of processing for all partitions of a HALDB being altered</li> <li>- partitions for which alter processing is completed</li> <li>- partitions currently undergoing a structural change</li> </ul> </li> </ul>

# HALDB Alter Operational Considerations

- All IMS data sharing systems must be running IMS 13
  - DBRC MINVERS value of “13.1” required
- Type-2 command environment required to initiate ALTER
  - Requires the Common Service Layer (CSL)
    - Structured Call Interface (SCI)
    - Operations Manager (OM)
  - Type-1 /INITIATE OLREORG command is not supported
- OLR processing is done for all partitions in a HALDB database
- Member OLC is recommended to bring specific ACBLIB(s) online
- Combine HALDB ALTER with new DB versioning

# IMS 13 DEDB Alter

# DEDB Alter

- IMS 13 adds ability to dynamically modify a DEDB Area while the Area remains online
  - Alter the physical attributes of a DEDB Area
    - SIZE, UOW, or ROOT specifications
  - Change the randomizer used for a DEDB Area
- Benefits
  - Improved DEDB Area availability
    - Make definitional modifications without taking the Area offline
  - Improved management of DEDBs
    - Provide flexibility in implementing Area changes into the system
    - Eliminate system down time for DEDB definition changes

# DEDDB Alter Overview

- A new DEDDB Alter utility is provided to dynamically modify a DEDDB database
  - Runs as a standard Fast Path IFP utility
  - DEDDB Areas remain online during utility execution
- A 2-stage randomizer must be used
  - Enables areas to be processed individually
- DEDDB alter does not support DEDDB databases in VSO or SVSO mode
  - Must be unloaded with /VUNLOAD before executing Alter utility
- DEDDB areas to be altered must be registered to DBRC
- Requires DBRC MINVERS “13.1” value for all IMS subsystems that share the DEDDB Area

# Fast Path IFP Utility for DEDB Alter

- TYPE ALTER
  - Invokes the DEDB Alter utility
- ALTERAREA area\_name | REPLRAND
  - ALTERAREA area\_name
    - Modify the active DEDB AREA statement
      - SIZE, UOW, ROOT, RMNAME
  - REPLRAND
    - Modify the active DEDB DBD statement
      - RMNAME
- Prep work needed for ALTERAREA and REPLRAND
  - Modify the DEDB DBD (ie. AREA or RMNAME parameters)
  - Assemble and bind any new randomizer (if being changed)
  - Run the DBDGEN utility to create new DEDB DBD definitions
  - Run the ACBGEN utility for PSBs that reference the changed DEDB DBD definitions to the staging ACBLIB
  - Allocate and register to DBRC “Shadow Area” and “Shadow IC” data sets

# Run DEDB Alter Utility - ALTERAREA Function



- ALTERAREA area\_name
  - Change SIZE, UOW and ROOT values to increase the size of DEDB Area while DEDB Area remains online
    - SIZE - Change the CI size of an Area dataset in a DEDB database
    - UOW and ROOT - Change the Root Addressable and Independent Overflow parts of an Area in a DEDB database
  - During DEDB Alter execution
    - Migrate *Active* Area to *Shadow* Area with new space definitions
  - After DEDB Alter completes
    - *Shadow* Area data set promoted to *Active* Area data set
    - Old *Active* Area data set demoted to *Shadow* Area data set
  - Only one active DEDB Area can be Altered at a time
    - Concurrent DEDB Alter for an Area in same DEDB database is not supported
    - Concurrent DEDB Alter for an Area in another DEDB database is supported
  - ALTERAREA does not support DEDB databases with SDEPs
  - ALTERAREA supports randomizer change while altering an *Active* Area



# Run DEDB Alter Utility - REPLRAND Function



- REPLRAND
  - Change Randomizer in RMNAME of DEDB DBD while DEDB online
    - Existing Area randomizer must be a 2-stage randomizer
    - New Area randomizer must be 2-stage randomizer
    - New randomizer name must be different from original name for the Area
  - During DEDB Alter execution
    - Read the active DEDB Area with the old randomizer
    - Migrate *Active Area* to *Target Area* with new randomizer
  - After DEDB Alter completes
    - New randomizer replaces old randomizer
    - All Areas in DEDB database use new randomizer
  - Only one active DEDB Area can be changed at a time
    - Concurrent DEDB alter for another Area in same DEDB is not supported
    - Concurrent DEDB Alter for an Area in another DEDB database is supported
  - REPLRAND supports DEDB databases with or without SDEPs





# IMS 13 DBRC Enhancements

# DBRC Coexistence with IMS 13

- DBRC Coexistence SPEs
  - IMS 11      APAR PM53134 / PTF UK80026
  - IMS 12      APAR PM53139 / PTF UK80027
- Upgrade RECON data set to the IMS Version 13
  - Issue a CHANGE.RECON UPGRADE command
- MINVERS value must be set to the lowest level of IMS that uses or shares the RECON data sets
  - IMS 11      “11.1”
  - IMS12      “12.1”
  - IMS 13      “13.1”

# DBRC Support for HALDB Alter

- INIT.PART command enhanced with new functionality for the BLOCKSIZE keyword
  - Use this keyword to define Control Interval (CI) sizes for VSAM data sets
- CHANGE.PART command enhanced with a new keyword, ALTERSIZE
  - Used before HALDB alter to specify new partition block sizes or CI sizes
- CHANGE.DB command enhanced with two new keywords (ALTER or NOALTER)
  - Used to change the HALDB partition status to indicate it is being altered
  - Status cannot be changed if the partition is authorized
- The NOTIFY.REORG command is enhanced with an ALTER keyword
  - Used to indicate the online reorganization process is altering a HALDB
- An "ALTER" indicator is added to the reorganization (REORG) record

# DBRC Support for HALDB Alter (cont'd)

- DB (TYPE=HALDB) record is changed
  - Two counter fields are added to indicate the number of partitions to be altered and the number of partitions that have been altered
- DB (TYPE=PART) record is changed
  - Two new flags added to indicate the partition is being altered and to indicate the alter process has completed
  - An array of new OSAM block sizes / VSAM CI sizes added and used during the HALDB alter process
  - Record size increased by 80 bytes when the HALDB alter process is active and the offset of a new array of OSAM block sizes or VSAM CI sizes is not zero
- Several DBRC API requests are enhanced to provide the new information stored in the RECON data sets

# DBRC Support for DEDB Alter

- INIT.ADS command is enhanced with new SHADOW and SHADOW IC keywords
  - Use to register shadow area data sets and shadow image copy data sets to DBRC
- DBDS (TYPE=FP) record is enhanced
  - Two counters are added to show the number of available shadow data sets and the number of shadow data sets that are registered to DBRC
  - A list of the shadow area data sets is added
- DB (TYPE=DEDB) record is enhanced
  - New counter field indicates DEDB alter is active



 Find us on  
**Facebook**  
[facebook.com/IMSFans](http://facebook.com/IMSFans)

  
[twitter.com/IBM\\_IMS](http://twitter.com/IBM_IMS)  
[Twitter.com/IMSTWIT](http://Twitter.com/IMSTWIT)

**Linked in**  
[linkd.in/IBM\\_IMS](http://linkd.in/IBM_IMS)

  
[imsmadesimple.tumblr.com](http://imsmadesimple.tumblr.com)

  
**IMSLISTSERV.BMC.COM**  
[imslistserv.bmc.com](http://imslistserv.bmc.com)

  
[ibm.co/IMS\\_dWBlog](http://ibm.co/IMS_dWBlog)

  
[ibm.com/vrm/newsletter/11069](http://ibm.com/vrm/newsletter/11069)

  
[www.slideshare.net/ibmims](http://www.slideshare.net/ibmims)

  
[ibm.co/IMS\\_RUGs](http://ibm.co/IMS_RUGs)

  
[bit.ly/IMS\\_YouTube](http://bit.ly/IMS_YouTube)  


# Acknowledgements and Disclaimers



**Availability.** References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

© **Copyright IBM Corporation 2014. All rights reserved.**

***U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.***

IBM, the IBM logo, ibm.com, zOS, IMS, DB2, CICS, InfoSphere, WebSphere, RACF, System z, zEC12, zBC12, z196, z10EC, zBX, zIIP, zAAP, ICF, and SoftwareXcel are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

Other company, product, or service names may be trademarks or service marks of others.



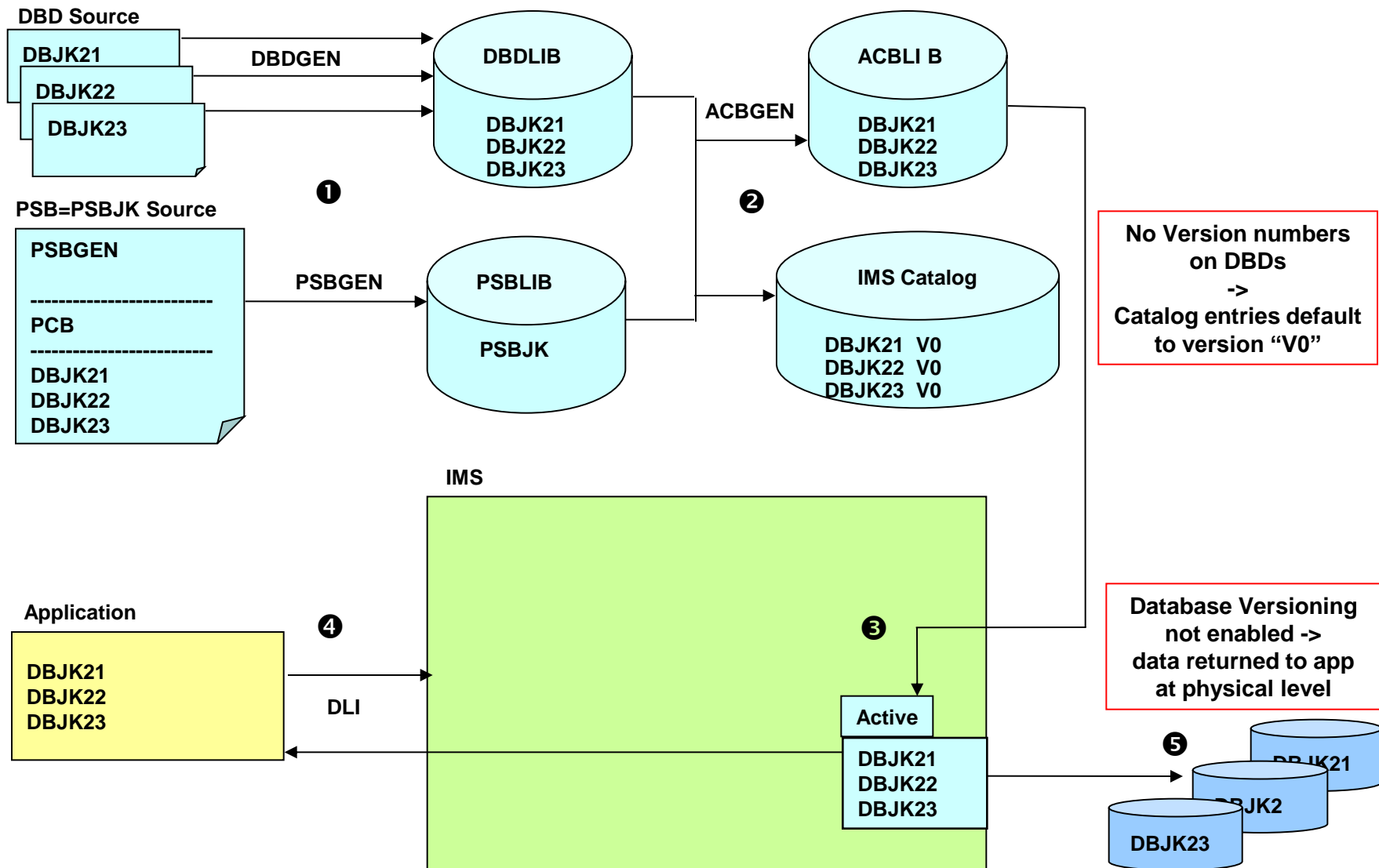
# ADDITIONAL SLIDES:

Slides removed from presentation  
for timing purposes

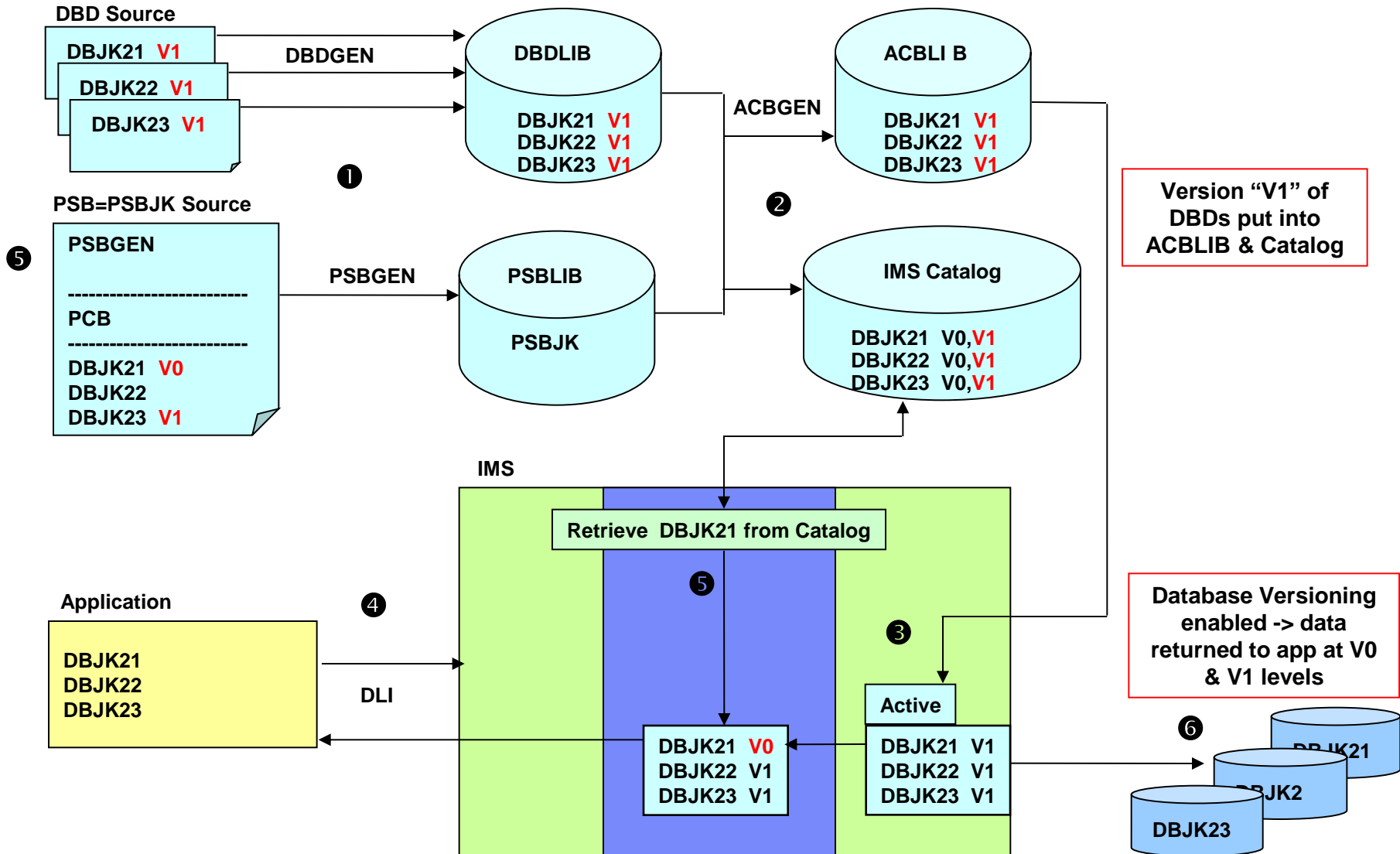


# Database Versioning

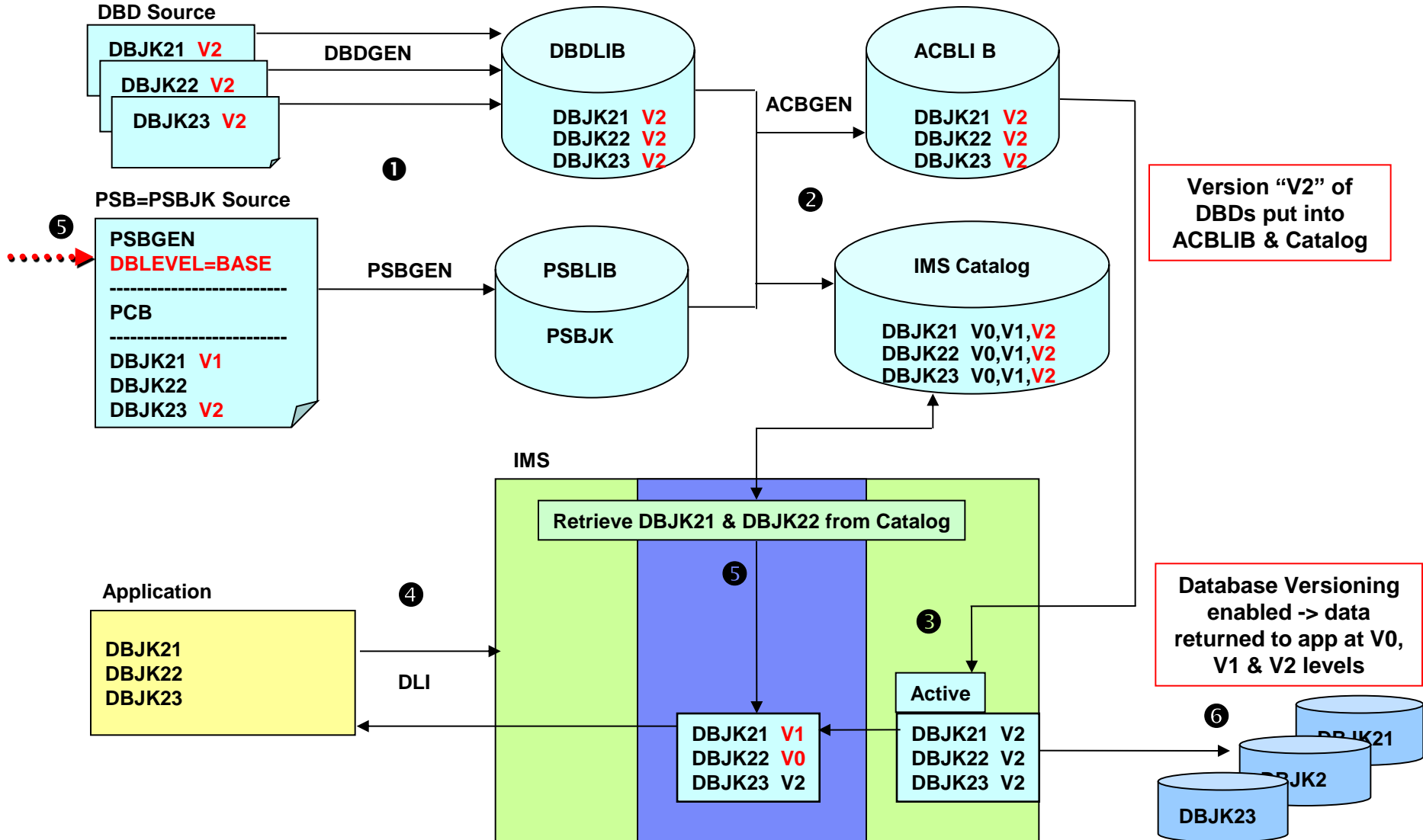
# Non-Versioning Process Flow



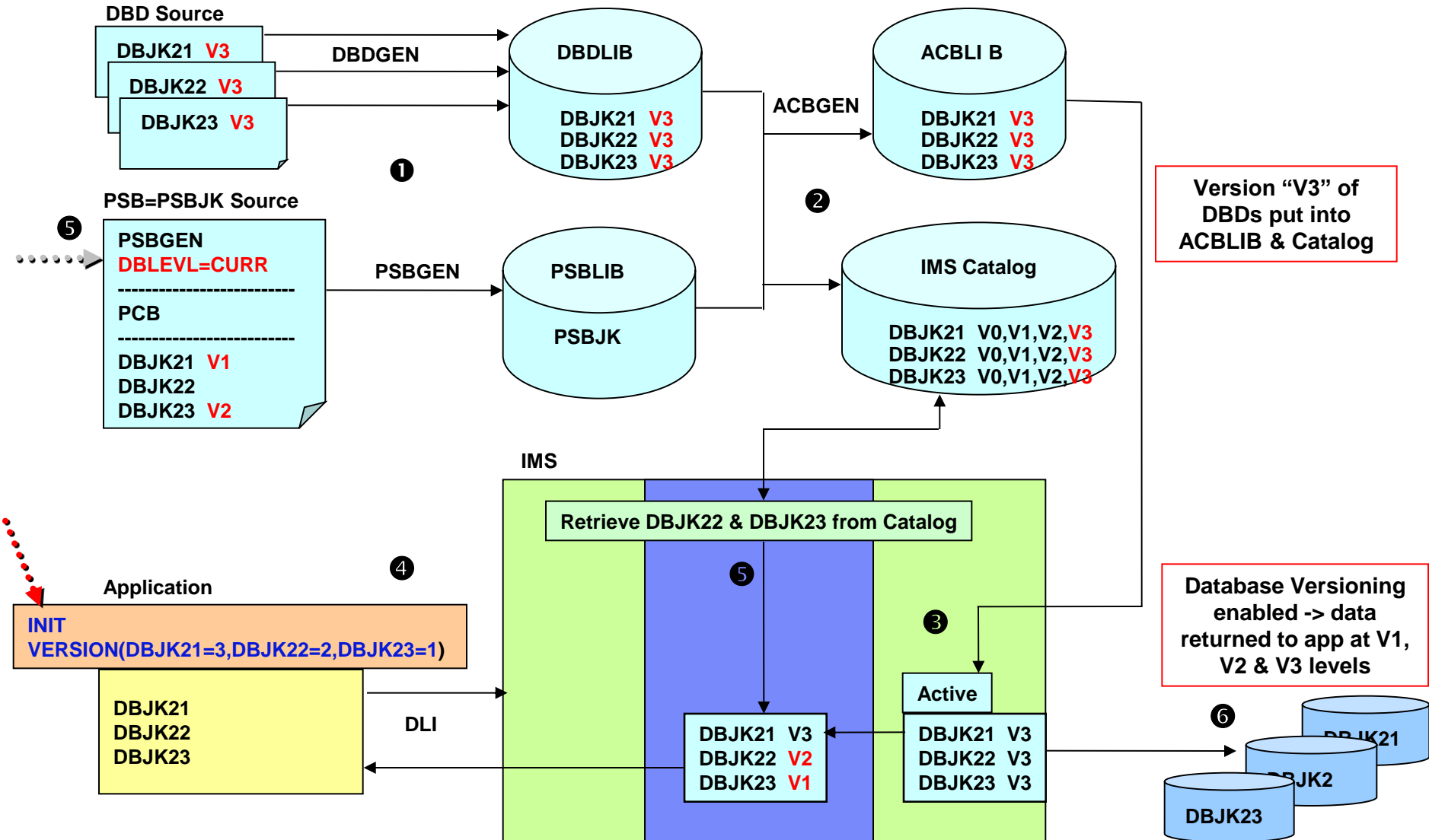
# Versioning Process Flow #1 - DBLEVEL=CURR (default)



# Versioning Process Flow #2 - DBLEVEL=CURR (default)



# Versioning Process Flow #3 - DBLEVEL=CURR (default)



## ***Database Versioning Errors***

### ■ U3303 PseudoAbend

- A DL/I call to access a specific version of the database data failed
  - Database Versioning enabled
    - invalid database version number was specified
    - requested database version cannot be found in the IMS catalog
    - current database structure contains a change that is not supported by versioning
    - storage error occurred while building the internal blocks required to satisfy a request for a version of a database other than the current version
  - Database Versioning not enabled
- DFS3303I message issued to console before the U3303 abend
- If INIT STATUS GROUPA was issued prior to DL/I call
  - U3303 pseudoabend is suppressed and a “BA” status code is returned

## Database Versioning Error

- DFS3549E
  - Application program attempted to access a prior version of a database but IMS cannot build the internal control blocks required to access prior versions of the database
    - latest version of the database contains a change in the database definition that is not supported by database versioning:
      - exit routine changed
      - number of segments changed
      - insert rule changed
      - delete rule changed
      - segment code changed
      - field length changed
      - segment length is truncated
      - segment changed from fixed-length to variable-length or vice versa
      - field was deleted, moved to another segment, or its name was changed
      - key length of the field changed
      - value of the TYPE keyword on the FIELD statement changed

**CHANGE NOT SUPPORTED BY DATABASE VERSIONING: RS=*rsnc* PST=*pstno*  
PSB=*psbname* DATABASE=*dbname* VERSION=*vernum* SEGMENT=*segmname***

**CHANGE NOT SUPPORTED BY DATABASE VERSIONING: RS=*rsnc* PST=*pstno*  
PSB=*psbname* DATABASE=*dbname* VERSION=*vernum* SEGMENT=*segmname*  
FIELD=*f1d\_name***

# Database Versioning Error Messages

## Enhanced IMS Messages

Message	Description
<b>DFS3303I</b>	<p>New conditions:</p> <p><b>INVDBVER</b> An invalid number for a full-function database was specified on a PCB or an INIT VERSION call that was issued by the application program. The specified database version number must be equal to or less than version number of the current database that is active in the IMS system. Also, a database version cannot be specified on a PCB if database versioning is not enabled. Database versioning is enabled by specifying DBVERSION=Y in the database section of the DFSDFxxx PROCLIB member.</p> <p><b>INVDBCHG</b> The current database structure of a full-function database contains a change that is not supported by database versioning. Prior versions of the database are incompatible with the current version and can not be accessed. The changes that are supported by database versioning are:</p> <ul style="list-style-type: none"> <li>• Increasing the size of a segment</li> <li>• Adding new fields without changes made to existing fields</li> </ul> <p><b>NOCATALG</b> The IMS catalog is not enabled. Database versioning requires the IMS catalog.</p> <p><b>NOSTORAG</b> A storage error occurred while building the internal blocks that are required to satisfy a request for a version of a full-function database other than the current version.</p> <p><b>NOVERFND</b> The requested version of a full-function database cannot be found in the IMS catalog</p>
<b>DFS3549E</b>	<p>Database change not supported by versioning:</p> <p>An application program attempted to access a prior version of a High Availability Large Database (HALDB), but IMS cannot build the internal blocks that are required to access prior versions of the database, because the latest version of the database contains a change in the database definition (DBD) that is not supported by database versioning. Application programs cannot access any prior version of the database, unless the application programs are changed or the unsupported change is removed from the database.</p>



# Database Versioning Error Messages

- New IMS Messages

Message	Description
<b>DFS0006E</b>	An error was detected while attempting to load the data management block (DMB) of an altered database.
<b>DFS0123E</b>	An application program attempted to access a prior version of a Fast Path data entry database (DEDB), but IMS cannot build the internal blocks that are required to access prior versions of the database, because the latest version of the database contains a change in the database definition (DBD) that is not supported by database versioning
<b>DBD180</b>	The DBVER operand is specified on a DBD statement for a database access type that does not support database versioning.
<b>DBD181</b>	The value on the DBVER operand in the DBD statement is not valid.
<b>PCB540</b>	The value on the DBVER operand of the PCB statement was not valid.
<b>PGEN259</b>	The value on the DBLEVEL operand of the PSBGEN statement was not valid

# HALDB Alter

## ***HALDB Alter Operational Considerations***

- Enhanced QUERY OLREORG command
- New OLR return / reason codes, and completion codes for ALTER
- New “SF” status code
- New DFS and DSP messages
  - DFS1849E DFS3197I DFS3198I DFS3436E DFS3547E
  - DSP0174E DSP0175E DSP1097E
- Changed DFS messages
  - DFS047A DFS2991E
- Several new DBRC commands & modified LIST outputs

## HALDB Alter Type-2 QUERY Command

### ■ QUERY OLREORG

- Use QUERY to check on ALTER processing for a HALDB database

```

▶▶———QUERY———OLREORG———
▶———NAME(haldbname)———STATUS(+ALTER----+)◀◀
                                   (+ALTDONE---+)
                                   (+ALTINPRG-+)
  
```

Keyword	Function
NAME() <i>optional</i>	<ul style="list-style-type: none"> <li>• NAME(*) is the default, to query all defined HALDB partitions</li> <li>• specifies either one or more partition names or the name of a HALDB master</li> <li>• wildcard character (*) is not allowed, except as NAME(*)</li> </ul>
STATUS()	<ul style="list-style-type: none"> <li>• allows you to display the online reorganizations that possess a specified status</li> </ul>
ALTER ALTDONE ALTINPRG	<ul style="list-style-type: none"> <li>• specifies type of output for the partitions of the HALDB master, identified on the NAME() parameter:             <ul style="list-style-type: none"> <li>- status of processing for all partitions of a HALDB being altered</li> <li>- partitions for which alter processing is completed</li> <li>- partitions currently undergoing a structural change</li> </ul> </li> </ul>

# HALDB Alter Return, Reason and Completion Codes

## INITIATE OLREORG Return and Reason Codes

Return Code	Reason Code	Meaning
X'00000000'	X'00000000'	Command completed successfully.
X'00000010'	X'00004520'	Another OLR Alter in progress.

## INITIATE OLREORG Completion Codes

Completion Code	Completion Code Text	Meaning
0		Command completed successfully
1E1	OLR ITASK creation failed	OLR internal ITASK can not be created
1E2	Incorrect HALDB version detected	The ddir version for the database is different than the number recorded in the RECON DB record for the HALDB
1E3	Partition queued for OLR	The partition is being queued for HALDB alter processing
1E4	HALDB alter pending for Online Change	An alter request against the same HALDB was done, but an Online Change has not yet done for that HALDB
1E5	No DB structure change detected	An alter request is made but there are no database structure changes made
1E6	Insufficient CI/Block size detected	During alter processing, the CI/Block size of the database data set is smaller than the database largest segment size
1E7	Unsupported DBD changes detected	An alter request is made but the type of DBD changes made are not supported
(many more)	...	...
1EC	Logical database error	During alter processing, an error related to logical relationship on the altered DBD is detected

## ***HALDB Alter New “SF” Status Code***

### ■ Explanation

- An application program that has field-level sensitivity attempted to read a database segment that was altered in length by the ALTER option of the HALDB online reorganization function.
  - Until the ACB members for the altered database are activated by the online change function, the altered segment cannot be accessed by application programs that have field-level sensitivity.

### ■ System action

- IMS returns this status code and continues to run normally

### ■ System programmer response

- Use the member online change function to activate the ACB members for the altered database and rerun the application program

## HALDB Alter DBRC LIST Command Output

- LIST output for a HALDB Master Database contains new info
  - “ALTER COUNT” indicates number of partitions that are to be altered
  - “ALTER COMPLETE COUNT” indicates number of partitions for which the alter process has completed

```

DB
DBD=DBOHIDK5                      DMB#=3          CHANGE#=3      TYPE=HALDB
SHARE LEVEL=3                      GSGNAME=**NULL**
DBRCVGRP=**NULL**
PSNAME=**NULL**  DBORG=PHIDAM      DSORG=OSAM     CURRENT PARTITION ID=00001
FLAGS:                               COUNTERS:
RECOVERABLE                        =YES           PARTITIONS                      =4
ONLINE REORG CAPABLE               =YES           DATA SET GROUP MEMBERS        =2
                                   ALTER COUNT                      =4
                                   ALTER COMPLETE COUNT              =2
  
```

-Number of partitions to be altered  
 -Number of partitions altered

## HALDB Alter DBRC LIST Command Output (cont'd)

- LIST output for a HALDB Database Partition contains new info
  - “ALTER BLOCK SIZE” lists new block sizes to be used by the alter process
  - “ALTER IN PROGRESS” indicates whether the alter process has started
  - “PARTITION ALTERED” indicates whether the alter process has completed

only for OSAM

### OSAM BLOCK SIZE:

A = 4096

B = 4096

### ALTER BLOCK SIZE:

A = 0

B = 8192

'0' indicates no  
BLKSIZE change

### FLAGS:

BACKOUT NEEDED =OFF  
 READ ONLY =OFF  
 PROHIBIT AUTHORIZATION=OFF

TRACKING SUSPENDED =NO  
 OFR REQUIRED =NO  
 PARTITION INIT NEEDED =NO  
 OLREORG CURSOR ACTIVE =NO  
 PARTITION DISABLED =NO  
 ONLINE REORG CAPABLE =YES  
 REORG INTENT =NO  
 QUIESCE IN PROGRESS =NO  
 QUIESCE HELD =NO  
**ALTER IN PROGRESS =NO**  
**PARTITION ALTERED =NO**

### COUNTERS:

RECOVERY NEEDED COUNT =0  
 IMAGE COPY NEEDED COUNT =0  
 AUTHORIZED SUBSYSTEMS =1  
 HELD AUTHORIZATION STATE=3  
 EEQE COUNT =0  
 RECEIVE REQUIRED COUNT =0  
 OLR ACTIVE HARD COUNT =0  
 OLR INACTIVE HARD COUNT =0

Partition values



# HALDB Alter DBRC LIST Command Output (cont'd)

DB

```

DBD=POHIDKA  MASTER DB=DBOHIDK5  IRLMID=NULL  CHANGE#=3  TYPE=PART
USID=0000000002  AUTHORIZED USID=0000000002  HARD USID=0000000002
RECEIVE USID=0000000002  RECEIVE NEEDED USID=0000000000
DSN PREFIX=IMSTESTS.DBOHIDK5  PARTITION ID=00001
PREVIOUS PARTITION=NULL**  NEXT PARTITION=NULL**
OLRIMSID=NULL**  ACTIVE DBDS=M-V
REORG#=00000
ONLINE REORG STATISTICS:
  OLR BYTES MOVED = 5576000
  OLR SEGMENTS MOVED = 16000
  OLR ROOT SEGMENTS MOVED = 4000

FREE SPACE:
  FREE BLOCK FREQ FACTOR=0  FREE SPACE PERCENTAGE=50

PARTITION HIGH KEY/STRING (CHAR):  (LENGTH=5 )
  . . . . .
PARTITION HIGH KEY/STRING (HEX):
  FFFFFFFF

OSAM BLOCK SIZE:
  A = 4096
  B = 4096
ALTER BLOCK SIZE:
  A = 0
  B = 8192

FLAGS:
  BACKOUT NEEDED      =OFF
  READ ONLY           =OFF
  PROHIBIT AUTHORIZATION=OFF

  TRACKING SUSPENDED  =NO
  OFR REQUIRED          =NO
  PARTITION INIT NEEDED =NO
  OLREORG CURSOR ACTIVE =NO
  PARTITION DISABLED  =NO
  ONLINE REORG CAPABLE =YES
  REORG INTENT        =NO
  QUIESCE IN PROGRESS =NO
  QUIESCE HELD       =NO
  ALTER IN PROGRESS   =NO
  PARTITION ALTERED   =NO

COUNTERS:
  RECOVERY NEEDED COUNT =0
  IMAGE COPY NEEDED COUNT =0
  AUTHORIZED SUBSYSTEMS =1
  HELD AUTHORIZATION STATE=3
  EEQE COUNT            =0
  RECEIVE REQUIRED COUNT =0
  OLR ACTIVE HARD COUNT =0
  OLR INACTIVE HARD COUNT =0

```

## HALDB Alter DBRC LIST Command Output (cont'd)

- LIST output for a REORG contains new info
  - REORG record now indicates whether the HALDB partition DBDS was altered during the online reorganization

```
REORG
  RUN      = 12.063 12:19:55.476258      *      USID = 0000000002
  REORG#   = 00005                                ALTER
  STOP     = 12.063 12:20:13.210119      ONLINE  RECOV = NO
```

**Partition was altered**

## ***HALDB Alter DBRC API Query Output***

- API query request output has changed
  - DSPAPQHP – HALDB Partition output block
    - Now returns the array of new Block / CI sizes to be used by the OLR alter process
    - Will only exist if the alter process is still in progress
  - DSPAPQHB – HALDB output block
    - Two new counters:
      - the total number of partitions being altered
      - the number of partitions for which the alter process has completed
  - DSPAPQRR – DBDS reorganization output block
    - New flag which indicates whether an online reorganization altered the HALDB

## ***HALDB Alter DBRC RECON Records***

- Content of several DBRC records have changed
  - DSPDBHRC – Database record (DB)
    - no size increase
  - DSPPTNRC – Partition record (PART)
    - size increases only when ALTER is running
  - DSPRRGRC – Reorganization record (REORG)
    - no size increase

**Remember to reassemble any programs using the changed DBRC control blocks !**

## ***HALDB Alter DBRC Command Changes***

### ■ CHANGE.DB

#### – ALTER | NOALTER

- New, optional keywords
- Specifies whether the HALDB partition is in the process of being altered.
- Cannot be changed if the partition is authorized

### ■ NOTIFY.REORG

#### – ALTER

- New, optional keyword
- Specifies that OLR **altered** the database structure
- Indicates that OLR **is altering** the database structure
- ALTER can only be specified with ONLINE

**Only use these commands in cases of a failure where RECONs need to be cleaned up !**

# ***HALDB Alter Security Considerations***

- **Security Considerations**

- New ALTER form of Online Reorg can change a database structure
  - INITIATE OLREORG command should be secured
- RACF attributes for Type-2 INIT command

IMS Command	Command Keyword	RACF access authority	Resource name
INIT	OLREORG	UPDATE	IMS.plxname.INIT.OLREORG

## ***HALDB Alter Performance Considerations***

- Performance Characteristics
  - Reorganizing a HALDB with the ALTER option
    - Performance should be same as reorganizing a HALDB without the ALTER option

## ***HALDB Alter Summary***

- **IMS 13 provides ability to make structural changes to a HALDB database without a database outage**
  - Structural changes can be made to DB segment definitions
  - Online Reorganization is used to apply the structural changes to the online HALDB database
  - Online Change process is used to activate the new ACBLIB member(s) in the online IMS system
- **Benefit**
  - Eliminate a database outage when structural changes must be made to segment definitions in a DBD
  - Improved online availability of HALDB databases



# Fast Path DEDB Alter

## ***DEDB Alter***

- **IMS 13 adds ability to dynamically change DEDB specifications**
  - Users can dynamically change UOW, SIZE, ROOT, Randomizer while DEDB is online
  - New DEDB Alter utility is used for DEDB changes
  - DRD is not required for DEDB Alter
  - Supports VSO Areas if /VUNLOAD is done before DEDB Alter is executed
  
- **Benefits**
  - Improved management of DEDB definitions
    - Eliminate system down time for modifications to DEDB definitions
    - Improve data availability since changes are done while DEDB is online

## ***DEDB Alter Utility***

- DEDB Alter utility supports two functions
  - ALTERAREA area\_name
    - Allow DBD parms (UOW, ROOT, SIZE, RMNAME) to change values
  - REPLRAND
    - Allow DBD parm (RMNAME) to change Randomizer name

## ***Preparation for DEDB Alter***

- DEDB Alter function changes:
  - ALTERAREA area\_name
    - Modify the active DEDB AREA statement (SIZE, UOW, ROOT)
    - Modify the active DBD statement (RMNAME)
  - REPLRAND
    - Modify the active DEDB DBD statement (RMNAME)
  
- IMS Gens needed for ALTERAREA, REPLRAND:
  - Run the DBDGEN utility to create new DEDB DBD definitions
  - Run the ACBGEN utility for all PSBs that reference changed DEDB DBD
    - New ACBs are added to staging ACBLIB data sets

## ***Preparation for DEDB Alter***

- ACBLIB staging library needs dynamic allocation member
  - Create a DFSMDA member for the ACBLIB staging library
    - If one does not already exist
  - Sample JCL:

```
DFSMDA TYPE=INITIAL
DFSMDA TYPE=IMSACB,DSNAME=STAGING.LIBRARY
DFSMDA TYPE=FINAL
```

- New DEDB Alter Datasharing Group Name
  - Defined in <SECTION=FASTPATH> in DFSDFxxx Proclib Member
  - ALTERGRP=nnnnn (Prefixed by DBFnnnnn)
  - Used for datasharing communications between datasharing partners

## Preparation for DEDB Alter

- DEDB Alter commits new DBD from Staging ACBLIB Library into Active ACBLIB
  - For IMS datasharing, need to know if:
    - Each IMS is **sharing same** ACBLIB (ACBSHR=Y)
    - Each IMS has **its own** ACBLIB (ACBSHR=N)
  - If Common Service Layer is **not used**
    - ACBSHR=Y|N is under <SECTION=FASTPATH> in DFSDFxxx
  - If Common Service Layer **is used**
    - ACBSHR=Y|N uses following precedence:
      - 1<sup>st</sup> DFSCGxxx PROCLIB member
      - 2<sup>nd</sup> <SECTION=COMMON\_SERVICE\_LAYER> in DFSDFxxx
      - 3<sup>rd</sup> <SECTION=FASTPATH> in DFSDFxxx
  - DEDB Alter uses ACBSHR setting from local IMS system
    - All IMS datasharing systems must have same ACBSHR setting

## ***Preparation for DEDB Alter***

- Allocate Shadow Area data sets
  - Shadow Area data sets used for migrating existing data from Active Areas
  - Single Area Data Sets (SADS)
    - Single Shadow data set is required
  - Multiple Area Data Sets (MADS)
    - 2 to 7 Shadow data sets are required
  - SADS can become MADS and MADS can become SADS after DEDB Alter
    - Depends on the number of allocated Shadow Area data sets
  - Shadow data sets are only for DEDB Alter utility
- Allocate Shadow IC data sets:
  - Created while DEDB Alter migrates data to Shadow Area data sets
- Shadow Area data sets + Shadow Image Copy data sets  $\leq 7$

## Preparation for DEDB Alter

- Register Shadow data sets to DBRC (INIT.ADS)

```
>>-INIT.ADS--ADDN(shadowname)--ADSN(shadowname)--AREA(name)--DBD(name)--->
```

```
.-UNAVAIL-----.
```

```
>-----+-----+-----+-----+-----+-----+-----><
  `-'-AVAIL-----'                '-SHADOW-----+-----'
                                     '-IC-----'
```



## ***Preparation for DEDB Alter***

- Format Shadow data sets with DEDB Area Init Util (DBFUMIN0)
  - Shadow Area data sets and Shadow IC data sets formatted
  - Two new control cards: ACTIVE | SHADOW
    - ACTIVE = Format Area data sets for DEDB Area
    - SHADOW = Format Shadow Area and Shadow IC data sets
  - Formats Active or Shadow data sets in one execution, but not both
  - When DBRC=Y, formats both Shadow Area and Shadow IC in one execution
- New ACB from Staging ACBLIB used for formatting
- After utility completes, flags are set in RECON:
  - Shadow Area data sets are marked “SHADOW AVAIL”
  - Shadow IC Area data sets are marked “SHADOW IC AVAIL”
- Shadow data sets must be formatted before DEDB Alter utility runs

## ***DEDB Alter Utility Control Statements***

- TYPE ALTER
  - Invoke DEDB Alter utility
  
- ALTERAREA area\_name | REPLRAND
  
- UNKEYSEG NONE | ALL | ISRTFILA
  - NONE: Do not allow un-keyed segments in DEDB
  - ALL: Allow un-keyed segments in DEDB
  - ISRTFILA: Allow un-keyed segments in DEDB if insert rule is FIRST or LAST
  
- TIMEOUT timeout\_value (1-999) | 15 seconds
  - Number of seconds for DEDB Alter DL/I activity to be quiesced:

## ***DEDB Alter Utility Control Statements (continued)***

- **RETRY NO | YES | retry\_value (1-99)**
  - NO: Do not retry after TIMEOUT value expires.
  - YES: Retry after TIMEOUT value until the utility completes successfully
  - retry\_value: Number of utility retries after TIMEOUT value occurs
  
- **RETRYWAIT retrywait\_value (1-999) | 60**
  - If RETRY YES or RETRY retry\_value
    - Number of seconds to wait before retrying the commit process
  
- **GO**
  - Execute the DEDB Alter utility

## DEDB Alter Utility Execution (ALTERAREA)

- Sample JCL: DEDB Alter Utility (ALTERAREA)

```
//ALTAREA JOB ...
//FPUTIL PROC SOUT=A,RGN=1M,
//          DBD=,REST=00,DIRCA=002,
//          PRLD=,IMSID=,AGN=,SSM=,ALTID=
//FPU EXEC PGM=DFSRR00,REGION=&RGN,
//          PARM=(IFP,&DBD,DBF#FPU0,&REST,00,,1,
//          &DIRCA,&PRLD,0,,,,&IMSID,&AGN,&SSM,,
//          &ALTID)
//STEPLIB DD DSN=IMS.CRESLIB,DISP=SHR
//PROCLIB DD DSN=IMSVS.PROCLIB,DISP=SHR
//SYSPRINT DD SYSOUT=&SOUT
//SYSUDUMP DD SYSOUT=&SOUT,...
//S0 EXEC FPUTIL,RGN=1M,DBD=DEDBJN21,REST=00,IMSID=IMS1
//SYSIN DD *
TYPE ALTER
ALTERAREA DB21AR0
RETRY NO
TIMEOUT 30
GO
/*
```

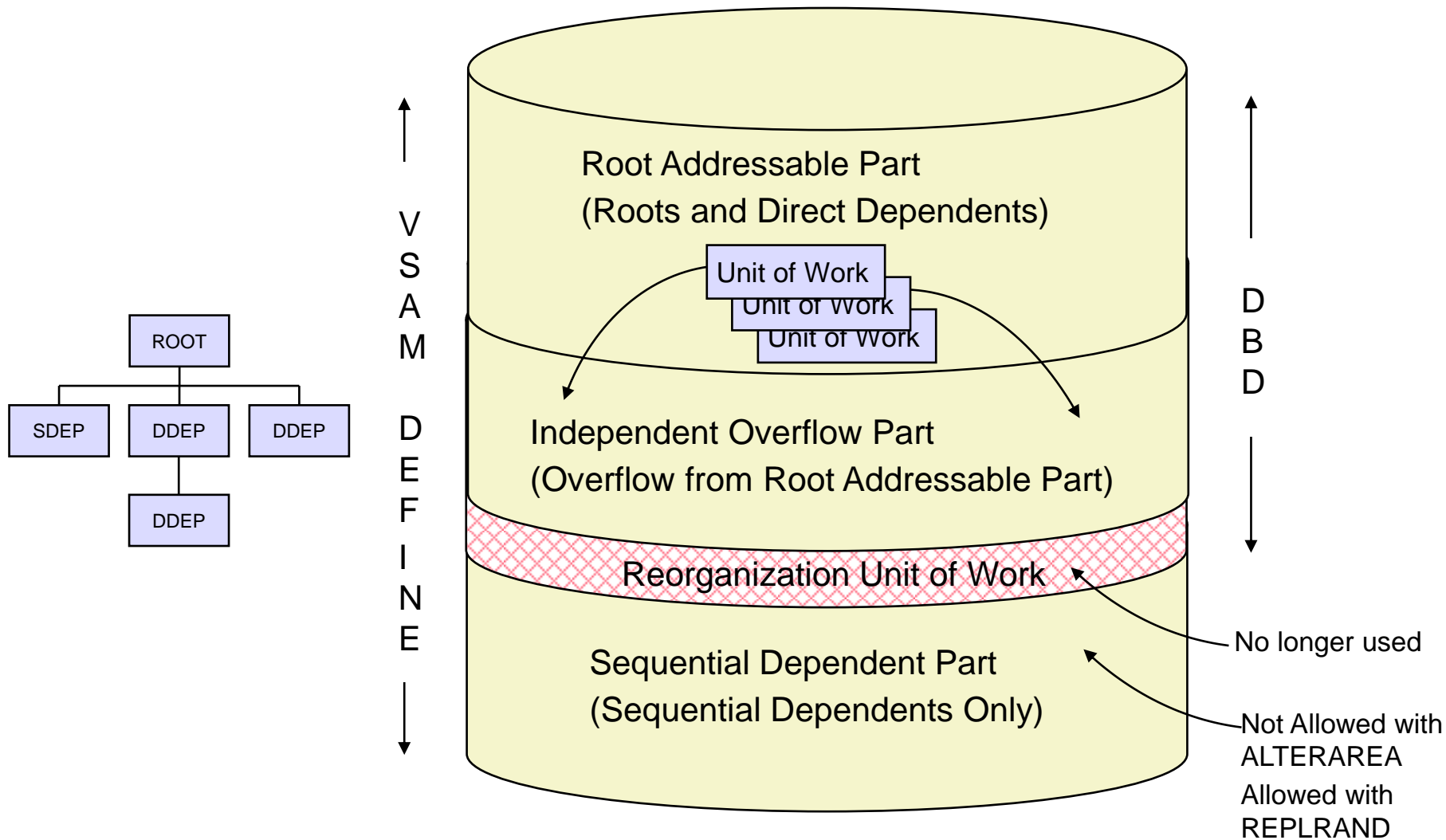
## ***DEDB Alter (ALTERAREA Function)***

- ALTERAREA area\_name
  - Changes UOW, SIZE, ROOT on DEDB DBD statement while DEDB online
    - SIZE
      - Change the CI size of an Area in a DEDB database
    - UOW and ROOT
      - Change the Root Addressable and Independent Overflow parts of DEDB area
  - Changes Randomizer in RMNAME DEDB DBD statement while DEDB online
    - During DEDB Alter, read active DEDB Area with old randomizer
      - Migrate Active Area to Target area with new randomizer
    - After DEDB Alter, new randomizer replaces old randomizer
      - All Areas in DEDB database use new randomizer
      - New name must be 2-stage randomizer
      - New name must be different than original name active for DEDB area
  - Only one active DEDB Area can change at a time
  - ALTERAREA does not support DEDB databases with SDEPs
    - Can replace randomizer using REPLRAND function for DEDBs with SDEPs

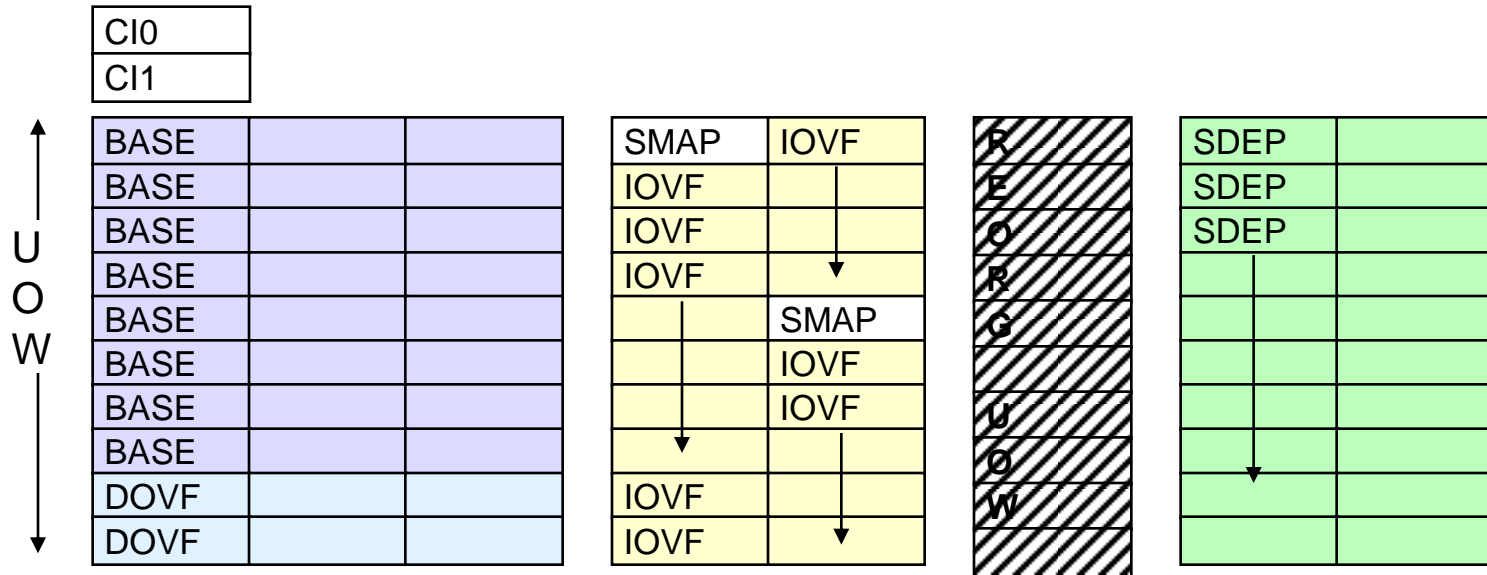
## ***Relationship of UOW, ROOT and SIZE Parameters***

- **UOW=(number1,overflow1)**
  - Number of Control Intervals (CI) in a UOW
    - Number1 = number of Control Intervals (CI) in a UOW
    - Overflow1 = number of Control Intervals in overflow section of UOW
  
- **ROOT=(number2,overflow2)**
  - Space allocated to Root Addressable Part and Independent Overflow
    - Number2 = Space (in UOWs) for Root Addressable Part and Independent Overflow
    - Overflow2 = Space (in UOWs) for Independent Overflow
  
- **SIZE=value**
  - Control Interval (CI) size (in Bytes)
    - 512 bytes, 1 KB, 2KB, 4KB, 8KB, 16KB, 20KB, 24KB, 28KB
    - SIZE value must match the CI size defined to VSAM

# DEDB Area Structure



# Area Terminology



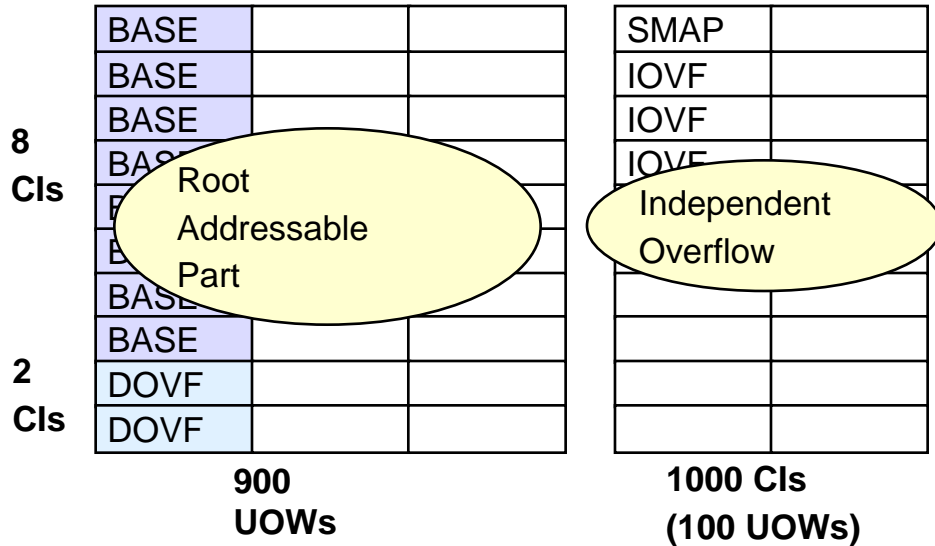
- CI0            Control Record
- CI1            Contains Area Control Block (DMAC) and Error Queue Elements (EQEs)
- UOW          Unit of Work (BASE and DOVF CIs)
- BASE          Only CIs with RAPs (also called RAP CIs)
- DOVF          Dependent Overflow (for UOW only)
- IOVF          Independent Overflow (when DOVF is full)
- SMAP          Space Map (monitors free space in IOVF)
- REORG        Reorganization Unit of Work (no longer used, but still allocated in ADS)
- SDEP          Sequential Dependents



## ***Examples of Increasing DEDB Area size***

- Expand the DEDB Area using ***same*** Control Interval (CI) size
  - Increase UOW= to put more CIs in a UOW
  - Increase ROOT= to allocate more space for Root Addressable Part and Independent Overflow
  
- Expand the DEDB Area using ***different*** Control Interval (CI) size
  - Increase SIZE= to create a larger CI size
  - Increase UOW= to put more CIs in a UOW
  - Increase ROOT= to allocate more space for Root Addressable Part and Independent Overflow

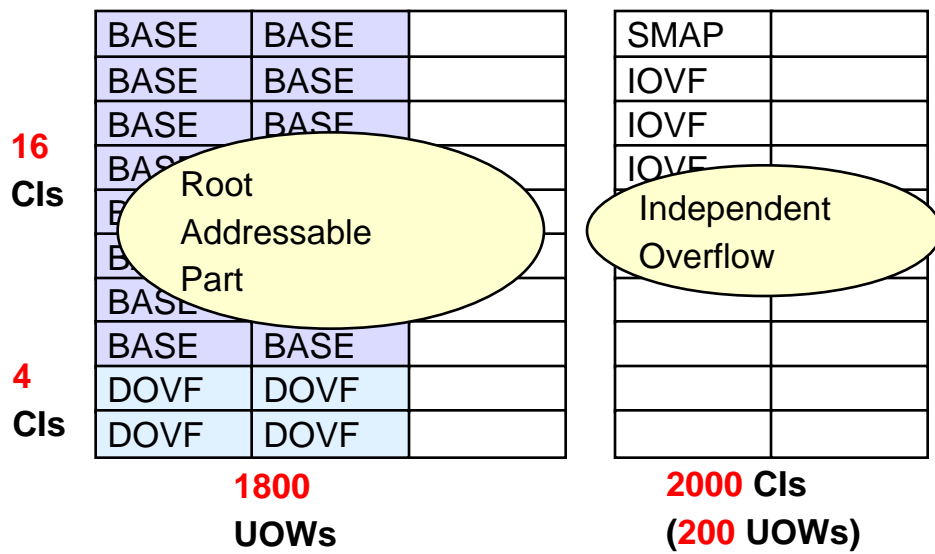
# DEDB Alter Example (ALTERAREA)



```

DBD      NAME=FPDEDB,ACCESS=DEDB,RMNAME=DEDBRAND
AREA     DD1=AREA1,...
AREA     DD1=AREA2,SIZE=4096,UOW=(10,2),ROOT=(1000,100)
AREA     DD1=AREA3,...
    
```

**ALTERAREA**  
 UOW = (20,4)  
 ROOT = (2000,200)



```

DBD      NAME=FPDEDB,ACCESS=DEDB,RMNAME=DEDBRAND
AREA     DD1=AREA1,...
AREA     DD1=AREA2,SIZE=4096,UOW=(20,4),ROOT=(2000,200)
AREA     DD1=AREA3,...
    
```

## ***DEDB Alter Utility Execution (ALTERAREA)***

- Complete preparation steps:
  - Alter DEDB DBD, run DBDGEN, run ACBGEN
  - Allocate Shadow Area and Shadow IC data sets, Register and Format
- Execute DEDB Alter Utility (ALTERAREA)
  - A. Data is migrated from active DEDB Area to Shadow Area and Shadow IC
    - Uses new UOW, ROOT and SIZE parms in staging ACBLIB
    - Use current randomizer to read Active DEDB Area
    - Use new randomizer (if changing) to insert to Shadow Area data set
  - B. Commits new randomizer and/or UOW, ROOT and SIZE changes
    1. Active DEDB Area is quiesced and DL/I calls are suspended
    2. Shadow Area data set is synchronized with Active DEDB Area
    3. Changed ACB in the Staging ACBLIB is moved to the Active ACBLIB
      - New randomizer replaces existing randomizer (if changed)
    4. Shadow Area data set becomes new DEDB Area data set
      - Original DEDB Area data set is preserved
    5. Un-Quiesce DEDB Area and resume suspended DL/I calls

## ***Post-DEDB Alter Utility Execution***

- If DEDB Alter is Successful
  - Shadow Area data set is promoted to Active Area data set
    - Old active area data set is demoted to SHADOW area data set
  - Shadow IC data set is promoted to user Image Copy
    - Registered in DBRC as “user image copy”
      - Counted as user image copy in GENMAX value
    - Not a standard image copy
      - Instead, it is an image of Active Area data set
    - To recover the Area with this image copy
      - (1) Notify DBRC that Area was restored
        - NOTIFY.RECOV DBD(name) AREA(name) RCVTIME(time\_stamp)
        - time\_stamp = Time when SHADOW IC was created by DEDB Alter utility
      - (2) Issue GENJCL.RECOV with no image copy
        - GENJCL.RECOV DBD(name) AREA(name) USEAREA

## ***Post-DEDB Alter Utility Execution***

- If DEDB Alter is Unsuccessful
  - Active Area data sets remain active and accessible to IMS systems
  - If Shadow Area data sets are marked as AVAIL:
    - Shadow Area data sets have not been written to
      - Can be used in subsequent DEDB Alter utility executions
  - If Shadow Area data sets are marked as UNAVAIL:
    - Shadow Area data sets have been written to
      - Shadow Area data set must be:
        - Re-allocated
        - Re-formatted with DBFUMIN0

## DEDB Alter Utility Execution (REPLRAND)

- Sample JCL: DEDB Alter Utility (REPLRAND)

```
//ALTAREA  JOB  ...
//FPUTIL   PROC  SOUT=A,RGN=1M,
//          DBD=,REST=00,DIRCA=002,
//          PRLD=,IMSID=,AGN=,SSM=,ALTID=
//FPU      EXEC  PGM=DFSRRC00,REGION=&RGN,
//          PARM=(IFP,&DBD,DBF#FPU0,&REST,00,,1,
//          &DIRCA,&PRLD,0,,,,&IMSID,&AGN,&SSM,,
//          &ALTID)
//STEPLIB  DD  DSN=IMS.CRESLIB,DISP=SHR
//PROCLIB  DD  DSN=IMSVS.PROCLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT,...
//S0      EXEC  FPUTIL,RGN=1M,DBD=DEDBJN21,REST=00,IMSID=IMS1
//SYSIN    DD  *
TYPE      ALTER
REPLRAND
RETRY     NO
TIMEOUT   30
GO
/*
```

## ***DEDB Alter (REPLRAND Function)***

- REPLRAND
  - Changes Randomizer in RMNAME DEDB DBD statement while DEDB online
    - New name must be 2-stage randomizer
    - Existing randomizer must be a 2-stage randomizer
    - New name must be different than original name active for DEDB area
    - After DEDB Alter, new randomizer replaces old randomizer
      - All Areas in DEDB database use new randomizer
  - Supports DEDB database with or without SDEPs

## ***DEDB Alter Utility Execution (REPLRAND)***

- Complete preparation steps:
  - Assemble and linkedit new randomizer
  - Modify DEDB DBD with new randomizer, run DBDGEN, run ACBGEN
- Execute DEDB Alter Utility (REPLRAND)
  - A. Load the new randomizer from IMS SDFSRESL STEPLIB concatenation
  - B. Load the new ACB from the Staging ACBLIB.
  - C. Commits new randomizer change
    1. Active DEDB database is quiesced causing DL/I calls are suspended
    2. Changed ACB in the Staging ACBLIB is moved to the Active ACBLIB
    3. DEDB database is un-quiesced and suspended DL/I calls are resumed



## ***Post-DEDB Alter Utility Execution***

- If DEDB Alter is Successful
  - New Randomizer replaces existing randomizer
- If DEDB Alter is Unsuccessful
  - Existing Randomizer remains in effect

## ***Prerequisites***

- Software requirements
  - All IMS data sharing systems need to be at IMS 13
  
- Hardware requirements
  - Same as IMS 13
  
- Tooling
  - None

# Restrictions

- Software restrictions
  - None
  
- Environment restrictions
  - The randomizer must be a 2-stage randomizer
    - RMNAME=(randomizer\_name,2)
    - Logic must act like a 2-stage randomizer
  - DEDB Areas must be registered to DBRC
  - Supports ACBSHR=Y for sharing IMS system if it is an:
    - Active system for XRF
    - Active system for FDBR
  - MINVERS = 13.1

## ***DEDB Alter Summary***

- **IMS 13 adds ability to dynamically change DEDB specifications**
  - Users can dynamically change UOW, SIZE, ROOT and Randomizer while DEDB is online
  - New DEDB Alter utility is used to make changes
  
- **Benefits**
  - Improved management of DEDB definitions
    - Eliminate system down time for modifications to DEDB definitions
    - Improve data availability since changes are done while DEDB is online

# DBRC Migration and Coexistence

## ***Supported Migrations and Coexistence***

- **IMS 11 to IMS 13**
  - Apply DBRC coexistence SPE APAR PM53134 to IMS 11
    - PTF UK80026
  - Allow IMS 11 to understand IMS 13 RECON records
- **IMS 12 to IMS 13**
  - Apply DBRC coexistence SPE APAR PM53139 to IMS 12
    - PTF UK80027
  - Allow IMS 12 to understand IMS 13 RECON records

## RECON Listings

- "COEXISTENCE LEVEL" in subsystem record listing
  - Added by IMS 10
  - May be used to determine if subsystems would cause an upgrade failure

```

SSYS
SSID=IMS1      LOG START=12.267 12:45:47.2
SSTYPE=ONLINE  ABNORMAL TERM=OFF  RECOVERY STARTED=NO  BACKUP=N
TRACKED=NO     TRACKER TERM=OFF   SHARING COVERED DBS=NO
IRLMID=**NULL**  IRLM STATUS=NORMAL      GSGNAME=**NULL**
COEXISTENCE LEVEL=13.1

AUTHORIZED DATA BASES/AREAS=4      VERSION=11.1  XRF CAPABLE=NO
                                     ENCODED
  -DBD-      -AREA-      -LEVEL-      -ACCESS INTENT-      -STATE-
PDHDOKA     **NULL**      0            UPDATE              6
PDHDOKB     **NULL**      0            UPDATE              6
PDHDOKC     **NULL**      0            UPDATE              6
PDHDOKD     **NULL**      0            UPDATE              6

```

- In this example the subsystem is at 11.1 but has the 13.1 coexistence maintenance applied

## ***CHANGE.RECON UPGRADE (IMS V11 to V13 Only)***

- Upgrade reads all database records to ensure that the high-order bit is on in all DMB numbers
  - If high-order bit is not on (should not occur)
    - High order bit is turned on if database is not authorized
      - Message issued:  
DSP1235W THE INTERNAL REPRESENTATION OF THE DMB NUMBER  
FOR DATABASE xxxxxxxxx IS INCORRECT
    - High order bit is not turned on if database is authorized
      - Message issued:  
DSP1236E THE INTERNAL REPRESENTATION OF THE DMB NUMBER  
FOR DATABASE xxxxxxxxx COULD NOT BE CORRECTED BECAUSE  
THE DATABASE IS AUTHORIZED
        - This condition causes upgrade to fail



## ***CHANGE.RECON UPGRADE CHECKUP***

- CHECKUP keyword verifies RECON Upgrade will work
  - New in IMS 12
  - Upgrade is not done
  - May be used to check if upgrade would be successful
    - Only reads records which could stop upgrade
  - Messages are issued indicating whether upgrade would be successful
    - DSP1238I RECON UPGRADE CHECKUP IS BEGINNING
    - DSP1239I RECON UPGRADE CHECKUP COMPLETED WITH NO ERRORS FOUND
      - RC=0
      - RC=4
        - When DSP1235W issued (found high-order bit off in the DMB number and the database or area is not authorized)
    - DSP1240E RECON UPGRADE CHECKUP COMPLETED AND FOUND ERROR RC=12
      - DSP1236E issued (found high-order bit off in the DMB number and the database or area is authorized)

## ***RECON Upgrade***

- RECONs are upgraded after IMS 13 is installed
  - Upgrade must use the IMS 13 DBRC utility (DSPURX00)
- Two RECONs and a spare must be available for concurrent upgrade
- Only two RECONs are required if there is no subsystem record
  - One RECON is okay in testing environment with no subsystem records
- **CHANGE.RECON UPGRADE**
  - May be executed while subsystems are running
    - Upgrade fails if there is a subsystem record for an IMS 11 or IMS 12 subsystem without the DBRC coexistence SPE
      - Some utilities do not create subsystem records
        - They are not protected by the check for subsystem records
        - If they are running without the SPE, unpredictable results may occur
        - Examples: Change Accumulation, Log Archive, DSPURX00, HALDB Partition Definition Utility (PDU), some DBRC API applications
  - May be invoked using the DBRC API

## ***RECON Upgrade***

- DMB Table record is added if it does not exist (IMS 11 to 13 Only)
- Some RECON records are larger in IMS 13 than IMS 11 or 12
  - Upgrade from IMS 11 or IMS 12 does not increase the size of the RECONs
- Recommendation for upgrades from IMS 11
  - Ensure that RECONs have room for the additional DMB Table record if it did not exist prior to the upgrade
    - May require availability of secondary extents

## ***RECON Upgrade***

- Upgrade processing from IMS 11 to IMS 13
  - Reads SSYS records to check for DBRC SPE
  - Reads all database records
    - Turns on high-order bit if it is not on and database is not authorized
    - Fails if high-order bit is not on and database is authorized
  - Builds DMB table record if the DMB number in the RECON header is greater than 0
  - Updates RECON header record
    - Sets version indicator and MINVERS value
  - Updates RECON header extension record
    - Sets version indicator
  - After COPY1 is upgraded, it is copied to COPY2

## ***RECON Upgrade***

- Upgrade processing from IMS 12 to IMS 13
  - Reads SSYS records to check for DBRC SPE
  - Updates RECON header record
    - Sets version indicator and MINVERS value
  - Updates RECON header extension record
    - Sets version indicator
  - After COPY1 is upgraded, it is copied to COPY2

## ***RECON Upgrade***

- Parallel RECON Access processing
  - RECON activity is quiesced
  - RECONS are closed and reopened in LSR (Local Shared Resources) mode
  - Records are upgraded
  - COPY1 is copied to COPY2
  - RECONS are reopened in PRA mode
  - Quiesce is ended

# MINVERS

- IMS 13 MINVERS valid values
  - '11.1', '12.1', and '13.1'
- Upgrade of RECONS
  - MINVERS('10.1') changed to MINVERS('11.1')
  - MINVERS('11.1') remains MINVERS('11.1')
  - MINVERS('12.1') remains MINVERS('12.1')
- MINVERS 12.1 is required for XCF use by APPC synchronous conversations and OTMA CM1 (send-then-commit)
- MINVERS 13.1 is required for:
  - IMS 13 Synchronous Program-to-Program Switch in Shared Queues Env
  - IMS 13 HALDB Alter
  - IMS 13 DEDB Alter
    - Note: HALDB Alter and DEDB Alter cannot be active if lowering the MINVERS value from 13.1

## ***MINVERS Special Consideration for '9.1'***

- **IMS 11 MINVERS valid value is '9.1'**
  - DBRC Precision Timestamp only when MINVERS='10.1' or later
- **Two issues could occur**
  1. If using CA data sets
    - Image copies could have same timestamp (1/10 second precision)
    - Recommendation:
      - After upgrading RECON to MINVERS='11.1' or above
      - Create image copies prior to running Change Accum
  2. If you have Jobs that reference the RECON
    - Ensure they can handle the increased precision



## ***Log Archive (DFSUARC0) Region Size***

- After RECONs are upgraded to IMS 13, then IMS 12 or IMS 11 Log Archive jobs will use additional memory
  - Both the IMS 13 and either IMS 11 or IMS 12 versions of RECON records are kept in memory
    - DBRC converts the records to IMS 11 or IMS 12 for processing by IMS 11 or IMS 12
  
- Recommendation
  - Use REGION=0M for IMS 11 and IMS 12 archive jobs

## ***DBRC Migration Steps***

1. Install IMS 11 or IMS 12 DBRC Migration/Coexistence SPEs
2. Install IMS 13 DBRC Type 4 SVC
  - The IMS 13 Type 4 SVC may be used with IMS 11 or IMS 12
3. Upgrade RECONs using the IMS 13 SDFSRESL library
4. Begin using IMS 13
5. Discontinue all use of IMS 11 and IMS 12
6. CHANGE.RECON MINVERS('13.1')