B78

IMS V9 DB and DBRC Enhancements

Rich Lewis

IMS
Technical Conference

Sept. 27-30, 2004

Orlando, FL

RETURN TO INDEX

DB and DBRC Enhancements

- Database Enhancements
  - Image Copy Large Tape Block Sizes
  - HALDB Online Reorganization
  - HALDB Specific Partition Initialization
  - IRLM 2.2
  - Multi-Area Structures for DEDB SVSO
  - Fast Path Area Open/Close Enhancements
  - XML DB
  - IMSplex Database Commands
  - PS EDI
  - Improved Message with Database Abends
- DBRC Enhancements
  - Command Authorization for /RMxxxx commands
  - More Than 32K Database Registrations
  - HALDB Dynamic Allocation by GENJCL.IC
  - DBRC Application Programming Interface (API)

HALDB Online Reorganization and XML DB are covered in greater detail in separate presentations.

Large Image Copy Block Size on Tape

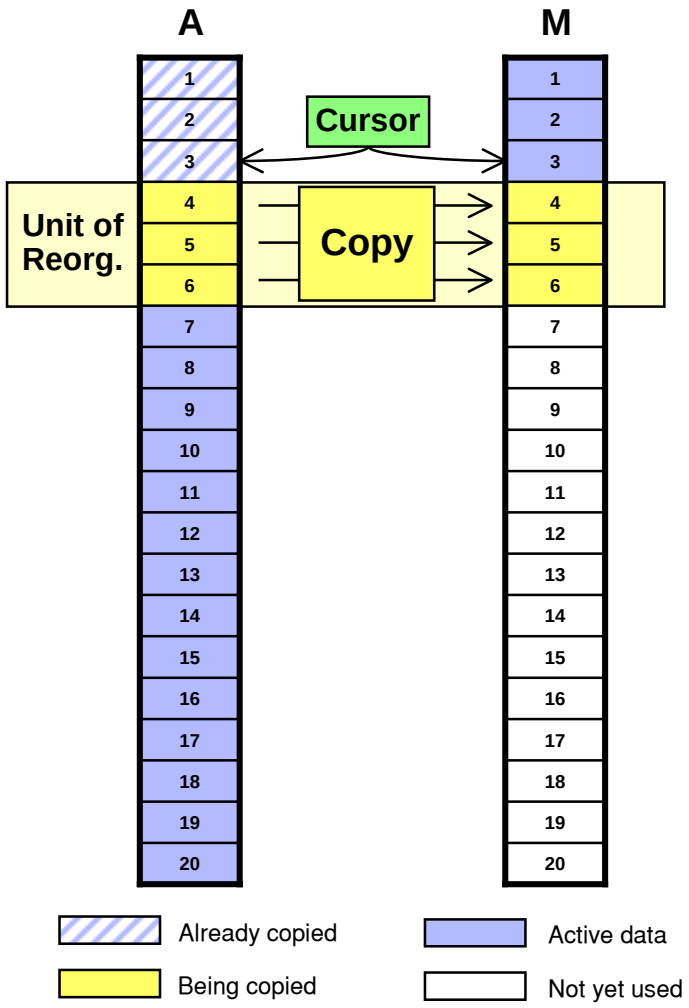
- Tape block sizes > 32K allowed for image copies
  - Requires supporting hardware
  - Invoked when BLKSIZE on Image Copy JCL DD statement
    - Not specified (system determined block size)
    - >32K
  - Supported by:
    - Image Copy utility (DFSUDMP0)
    - Database Recovery utility (DFSURDB0)
    - Database Recovery Facility tool (DRF)
  - Not supported by:
    - Image Copy 2 utility (DFSUDMT0)
    - Online Image Copy utility (DFSUICP0)
- Benefits
  - Improved performance and greater tape capacity

HALDB Online Reorganization

- Absolutely no outage for reorganzations
  - Applications access all of the data during reorganizations without restriction
- Online reorganization technique
  - Writes new data sets
    - Dynamically allocates output data sets (optional)
    - Deletes input data sets when reorganization completes (optional)
    - Duplicate data sets
      - Only for partitions being reorganized
      - Only during the reorganization of the partitions
- Supports data sharing
  - Other IMS systems may read and update the partitions while they are reorged
  - Reorganization may be done in any data sharing IMS system

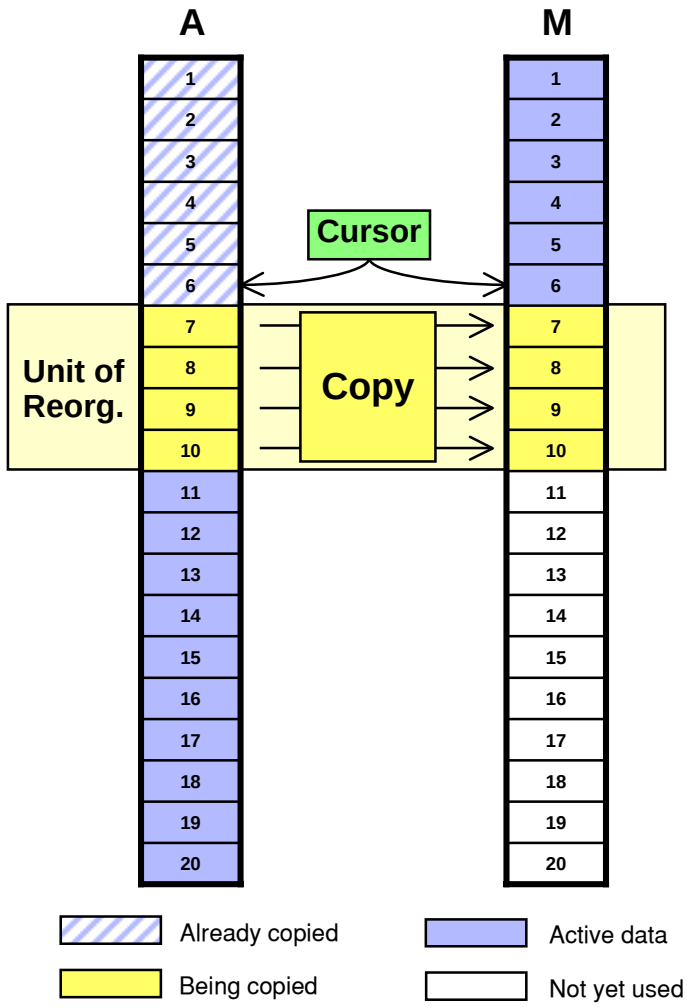
HALDB Online Reorganization

- Two active sets of data sets
  - Both are used during the reorg.
  - Records are copied to new data set
- 'Unit of Reorg'
  - Set of records being copied are one time
  - Records are locked during copy
    - Number of records in UOR is dynamically adjusted
    - Algorithm limits time taken, bytes copied, and locks held at any time
  - Cursor determines which data set contains active record



HALDB Online Reorganization

- Data set used is based on cursor value
  - Cursor on Record 6
  - Access Record 5:
    - Access from M data set
  - Access Record 14:
    - Access from A data set
  - Access Record 9:
    - Wait for lock,
      - then access from M data set



HALDB Specific Partition Initialization

- Background
  - HALDB partitions must be initialized before they may be used
  - Partition initialization makes database data sets non-empty
    - They can be opened for update
  - Partition initialization is required before:
    - Initial load (PROCOPT=L)
    - Migration reload
      - HD Reload using output of unload of non-HALDB
    - Using empty partition
      - New partition with no data
  - 'Partition initialization required' status is kept in RECONs
    - Cannot authorize partition with this flag on
  - Initialization is done either by Partition Initialization utility or the Database Prereorganization utility

HALDB Specific Partition Initialization

- Previous capabilities
  - Initialize partitions for a list of databases
    - Initializes only partitions with 'partition initialization needed' flag on

    or
    - Initializes all partitions without regard to the 'part. init. needed' flag
      - Invoked with INITALL control statement in DFSOVRDS data set
- Added capabilities
  - Initializes a list of partitions
    - Specify each partition on a separate control statement
    - Partitions may be in different databases
    - Initializes partitions even when the 'part. init. needed' flag is not on
- Benefits
  - Improved flexibility
    - Especially important for testing
      - Eliminates need to turn on the 'part. init. needed' flag

IRLM 2.2

- 64-bit addressing support
  - Uses 64 bit addressing if z/Architecture and z/OS 1.3 or greater
    - Most lock control blocks are placed above the bar
      - Provides support for more concurrently held locks
      - Primarily required for DB2 systems with “abusive locking”
  - Uses 31-bit addressing if not z/Architecture or not z/OS 1.3 or greater
  - PC parameter is ignored
    - PC=YES is always used (even if PC=NO is specified)
      - Relieves potential memory constraint
      - PC=NO performance benefit was insignificant
    - MAXCSA= is ignored
      - Only applies when PC=NO is used
- Benefits
  - Greater concurrent lock capacity

IRLM 2.2

- IMS V9 includes IRLM 2.2
  - IRLM 2.1 is not shipped with IMS V9
- Compatibility
  - IRLM 2.1 may be used with IMS V7, V8, and V9
  - IRLM 2.2 may be used with IMS V7, V8, and V9
  - IRLM 2.1 and 2.2 may coexist in the same IMS data sharing group

Multi-Area Structures for DEDB Shared VSO

- IMS V9 allows multiple areas to share a CF structure
  - Previous releases required a CF structure for each shared area
  - Multi-area structures defined in DBRC
    - New MAS keyword for INIT.DBDS and CHANGE.DBDS
    - System-managed duplexing may be used
      - IMS-managed duplexing (CFSTR2) is not used
  - Areas assigned to the same structure in DBRC must use the same buffer pool
    - DEDBMAS statement in DFSVSMxx creates private buffer pools

      `DEDBMAS=(poolname,cisize,pri,sec,max,lkasid,StructureName)`
    - Default pool built for structure's areas if appropriate DEDBMAS statement is not present
- Benefits
  - Simplified structure management
  - Avoids potential z/OS limitation of 512 structures per sysplex

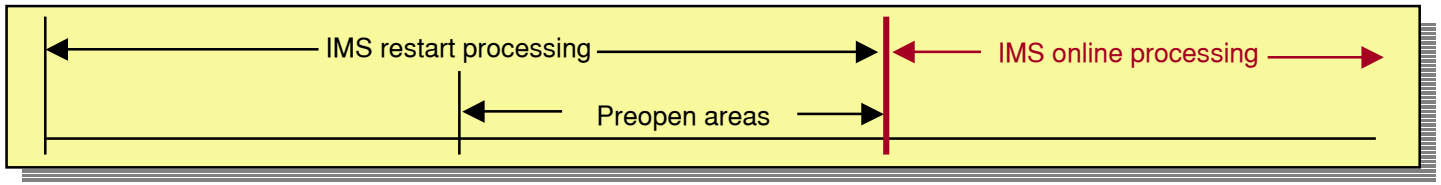
Fast Path Area Open and Close Enhancements

- New capabilities
  - Parallel open and close processing
    - Multiple TCBs are used for open and close
  - Async open processing with online activity
    - Does not delay IMS restart completion
  - Options to (re)open areas after /ERE completes
    - Open only those with PREOPEN specified (like previous releases)
    - Open all of those open at time of termination
    - Open both those with PREOPEN and those open at time of termination
  - Restart and reopen of areas after IRLM reconnect
- Benefits
  - Improved performance
  - Simplified operations

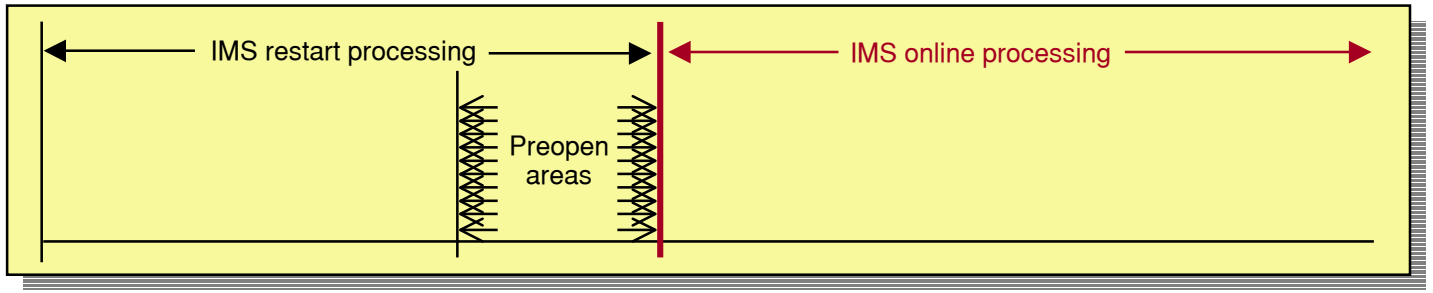
Fast Path Area Open and Close Enhancements

IMS Restart Processing for DEDB Areas

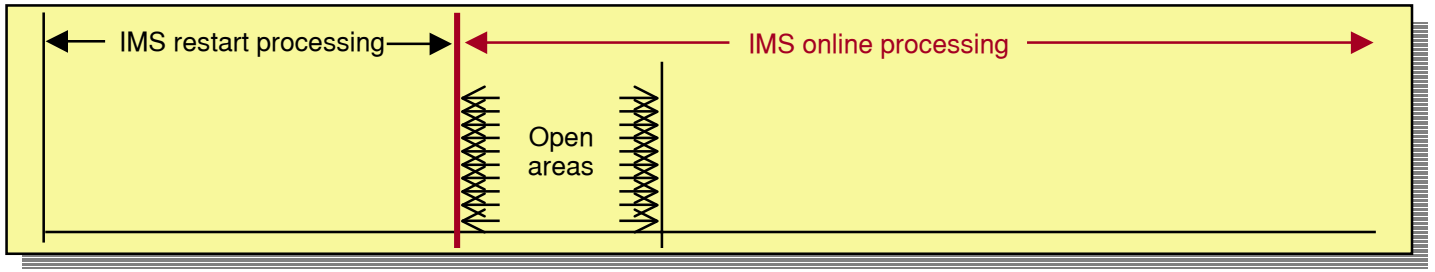
- Pre-IMS V9



- IMS V9 with option to open during restart



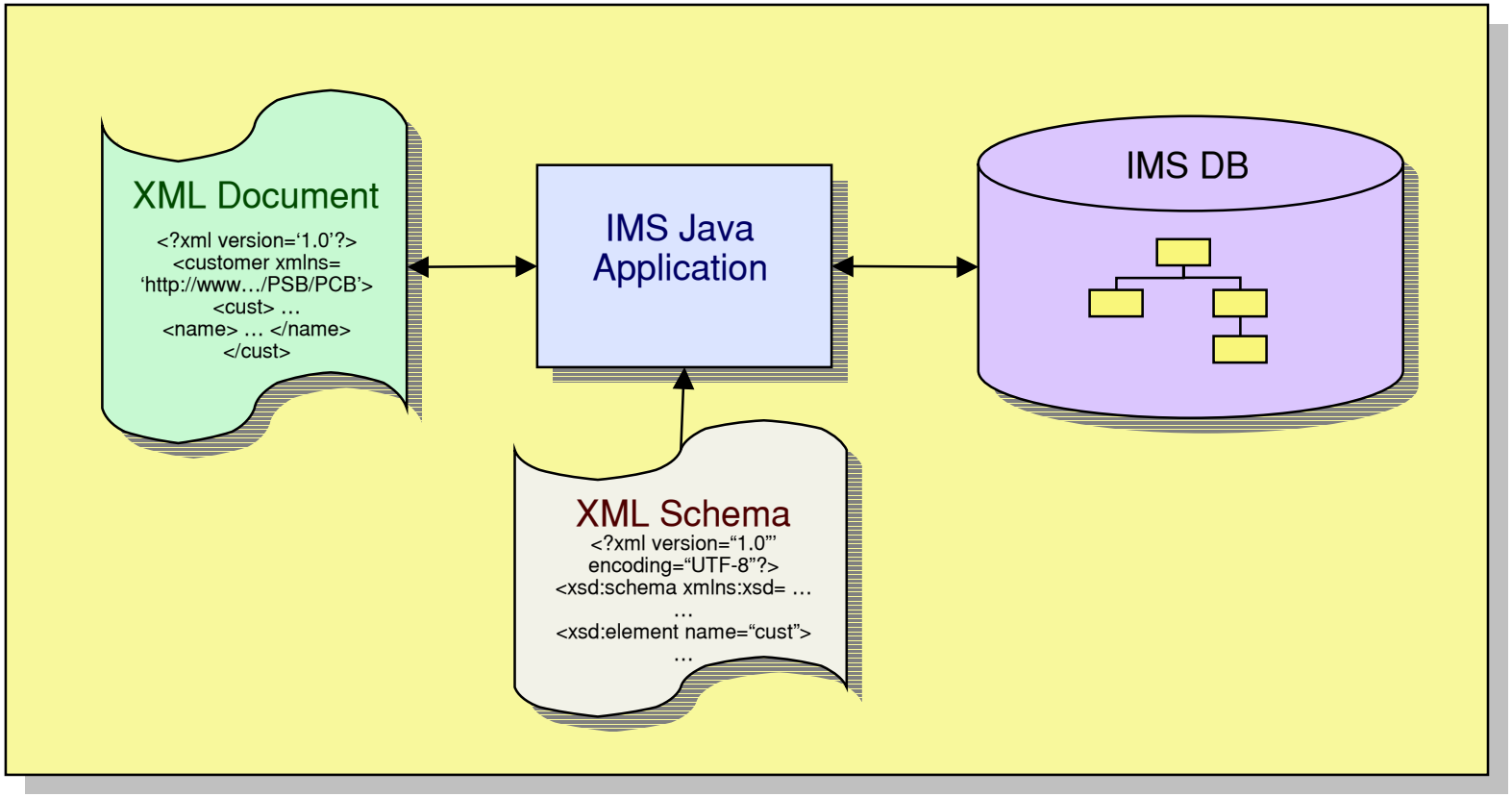
- IMS V9 with option to (re)open after restart completes



XML Database

- Storage and retrieval of XML documents in IMS databases
  - Composition of XML documents from existing IMS databases
  - Creation of IMS segments from XML documents (decomposition)
  - Intact storage of XML documents (without decomposition)
  - Tooling assistance to define metadata for mappings
  - IMS Java application programming support
- Benefits
  - Easy exchange of data between IMS databases and XML documents
    - Existing IMS databases may be used to create XML documents
    - Existing applications are unaffected

XML Database



IMSplex Database Commands

- IMSplex (type-2) database commands
  - Extends IMSplex commands to IMS database resources
  - New commands
    - QUERY or QRY verb
      - DB and AREA resources
    - UPDATE or UPD verb
      - DB, AREA, and DATAGRP resources
- Benefits
  - Consistent format with IMSplex commands introduced in IMS V8
  - Consolidated responses from multiple IMS systems
  - Use of wildcards with resource names

IMS V9 has new terminology for commands:

- Type-1 commands:
  - /DIS, /START, etc.
- Type-2 commands:
  - QRY, UPD, INIT, etc.

IMSplex (Type-2) Database Commands

- Examples of QUERY or QRY:

Type-2 Command	Equivalent Type-1 Command
QUERY DB(ABC)	/DIS DB ABC
QRY DB(*)	/DIS DB ALL
QRY DB(AB*)	No equivalent
QRY DB STATUS(ALLOCF)	/DIS DB ALLOCF
QRY AREA NAME(*) STATUS(STOPPED)	/DIS AREA STOPPED

IMSplex (Type-2) Database Commands

- Examples of UPDATE or UPD:

Type-2 Command	Equivalent Type-1 Command
UPDATE DB NAME(ABC) START(ACCESS)	/START DB ABC
UPD AREA NAME(XYZ) STOP(SCHD)	/STOP AREA XYZ
UPD DB NAME(ABC) STOP(ACCESS)	/DBR DB ABC NOFEOV
UPD DB NAME(ABC) STOP(UPDATES)	/DBD DB ABC
UPD DB NAME(ABC) START(ACCESS) SET(ACCTYP(READ)) OPTION(OPEN)	/START DB ABC ACCESS=RD OPEN
UPD DB NAME(AB*) START(ACCESS)	No equivalent

Disabling of z/OS DFSMS V1R5 PS EDI

- PS EDI
  - Physical Sequential Enhanced Data Integrity
  - PS EDI prevents concurrent opens of
    - Data sets with DSORG=PS (e.g. OSAM)
    - Allocated with DISP=SHR
    - Opened for output or update
  - New optional function in z/OS DFSMS V1R5
    - Invoked by IFGPSEDI member of SYS1.PARMLIB
      - WARN mode: only send message if rule violated
      - ENFORCE mode: prevent concurrent opens for update
      - Exception data sets may be listed
        - Concurrent opens for update allowed for them
    - Authorized programs may disable function for their data sets
      - IMS uses this technique to disable PS EDI for some data sets

Disabling of z/OS DFSMS V1R5 PS EDI

- PS EDI is not required for some IMS data sets:
  - Database data sets
    - DBRC and IRLM provide data integrity
  - OLDS, WADS, RDS, and MSDB dump data sets with XRF
    - Alternate must have write capability for takeover
  - PS EDI is disabled by IMS for these data sets
    - Bit in DCBE is set by IMS
    - APARs supplying this function:
      - PQ83940 for IMS V9
      - PQ83941 for IMS V8
      - PQ83942 for IMS V7
- PS EDI is excellent function for other IMS data sets
  - OLDS and WADS when not in XRF takeover

Improved Message with DB Abends

- Message identifies database involved in abend
  - Previously only issued for HALDB
    - Now issued for non-HALDB full function databases
      - Such as U085x (DFS554A is also issued)

```
DFS0832I ABEND Uwwww REASON CODE xxxx yyyyyyyy zzzzzzzz
```

`wwww` – abend code

`xxxx` – reason code

`yyyyyyyy` – '`PARTITION`' for HALDB
'`DATABASE`' for non-HALDB

`zzzzzzzz` – partition or database name or
'`NOTAPPLI`' if no DBD is available

DBRC Command Authorization for /RMxxxx

- IMS V8 added authorization for DBRC commands
  - Support for DBRC utility (DSPURX00) and HALDB Partition Definition Utility
  - Did not include /RMxxxx commands
- IMS V9 support
  - Support for /RMxxxx commands
    - RACF (or equivalent)
    - DBRC Command Authorization Exit routine (DSPDCAX0)
      - Optional
- Benefits
  - Extends consistent security to online DBRC commands

DBRC Command Authorization for /RMxxxx

- DBRC command authorization invocation
  - Same for utilities and /RMxxxx commands
  - Activated by
    - CHANGE.RECON CMDAUTH (SAF|EXIT|BOTH|NONE,safhlq)
  - Profiles can differ for different RECONs
    - Controlled by safhlq
  - Commands can be authorized at
    - Command verb level
      - For example, GENJCL command
    - Command verb + resource type level
      - For example, GENJCL.RECOV command
    - Command verb + resource type + resource name level
      - For example, GENJCL.RECOV DBD(ACCDB)

DBRC – More Than 32K Database Registrations

- Previous releases only allowed 32,767 registrations of databases
  - Deleting a database did not free a global DMB number in the RECONs
- IMS V9 allows more than 32,767 registrations of databases
  - Maximum databases registered at any time is still 32,767
- Benefit
  - Avoids potential problem for installations with many databases

DBRC – HALDB Dynamic Allocation by GENJCL.IC

- GENJCL.IC does not generate DD statements for HALDB data sets
  - Supplied ICJCL skeletal JCL member changed from IMS V8
  - Dynamic allocation is used
- Benefit
  - Especially useful for HALDB Online Reorganization users
    - IC utilities determine the active data set, then dynamically allocate it

DBRC Application Programming Interface (API)

- DBRC API
  - Allows users to write programs to read RECONs
    - Provides API to return information from all records
  - Sample program is provided

- Benefits
  - Supported method for retrieving RECON data
    - Easier than parsing LIST.RECON output
    - Easier than writing VSAM application to read RECONs
  - Release independent
    - Future releases will not require modifications to programs

DBRC Application Programming Interface (API)

- API Overview
  - Assembler language
    - Assembler macros are provided
  - Functions
    - Start the environment
      - Allocates and opens the RECONs
    - Query the RECONs
      - Acquires storage and places information in this storage
    - Release buffer storage
      - Releases storage acquired for queries
    - Stop the environment
      - Closes and deallocates RECONs

DBRC Application Programming Interface (API)

- Typical program structure
  - Include the API DSECTs and supply working storage
    - DSPAPQxx macros
      - Mapping macros for data returned
        - Typically, a macro for each RECON record type
  - Initialize the API (allocate and open RECONs)
  - Issue one or more query requests
    - Specifies data to be retrieved
  - Process data from queries
  - Release buffer storage
  - Terminate the DBRC API

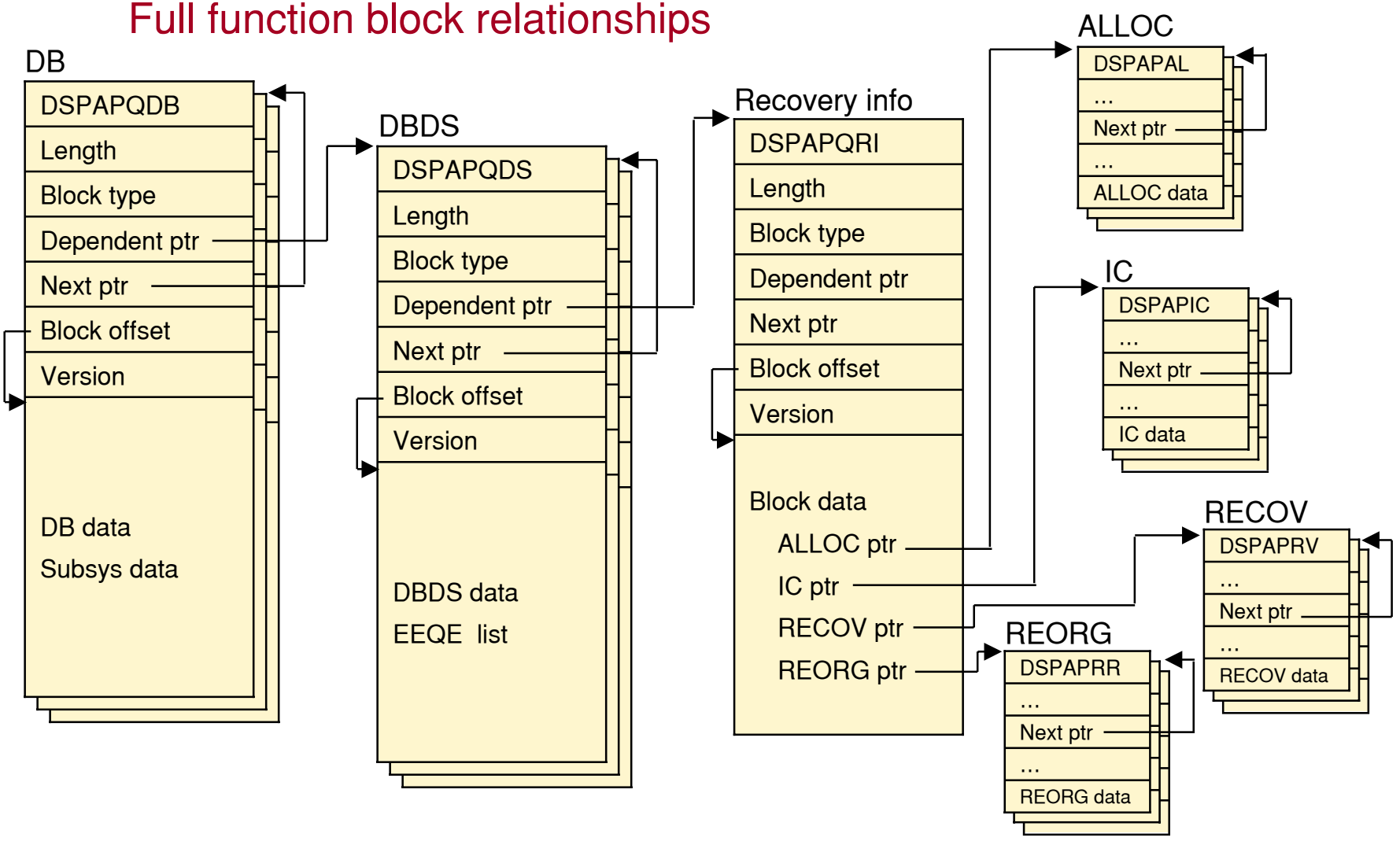
```
DSPAPI FUNC=DSECT
DSPAPQxx
DSPAPQxx

DSPAPI FUNC=STARTDBRC

DSPAPI FUNC=QUERY
Process results
DSPAPI FUNC=QUERY
Process results
DSPAPI FUNC=RELBUF

DSPAPI FUNC=QUERY
Process results
DSPAPI FUNC=RELBUF


DSPAPI FUNC=STOPDBRC
```

DBRC Application Programming Interface (API)

- Using queries
  - QUERY may ask for
    - Records by name (e.g. database name)
    - Records by list of names
    - First record (e.g. first database record)
    - Next record (e.g. next database record after specified database
    - Set of records (e.g. database record and its data set and image copy records)
  - QUERY returns blocks of data
    - Blocks are chained in RECON hierarchy
    - Blocks are mapped by DSPAPQxx macros
  - Application program follows pointers between blocks to access the data

DBRC Application Programming Interface (API)

Full function block relationships



DBRC Application Programming Interface (API)

- Summary
  - Assembler interface for users to access RECON data
  - Sample program is provided
- Benefits
  - Supported method for retrieving RECON data
  - Release independent
    - Future releases will not require modifications to programs

DB and DBRC Enhancements

- Database Enhancements
  - Image Copy Large Tape Block Sizes
  - HALDB Online Reorganization
  - HALDB Specific Partition Initialization
  - IRLM 2.2
  - Multi-Area Structures for DEDB SVSO
  - Fast Path Area Open/Close Enhancements
  - XML DB
  - IMSplex Database Commands
  - PS EDI
  - Improved Message with Database Abends
- DBRC Enhancements
  - Command Authorization for /RMxxxx commands
  - More Than 32K Database Registrations
  - HALDB Dynamic Allocation by GENJCL.IC
  - DBRC Application Programming Interface (API)