

IMS Catalog and IMS Open Database

Nancy Stein

IBM – IMS Advanced Technical Sales Support

Tuesday, March 11, 2014

Session #14779

www.SHARE.org



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, System z, zEC12, z196, z10EC, zBX, zIIP, zAAP, ICF, SAP, SoftwareXcel, zOS, IMS, InfoSphere, DB2, CICS, RACF and WebSphere, 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

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

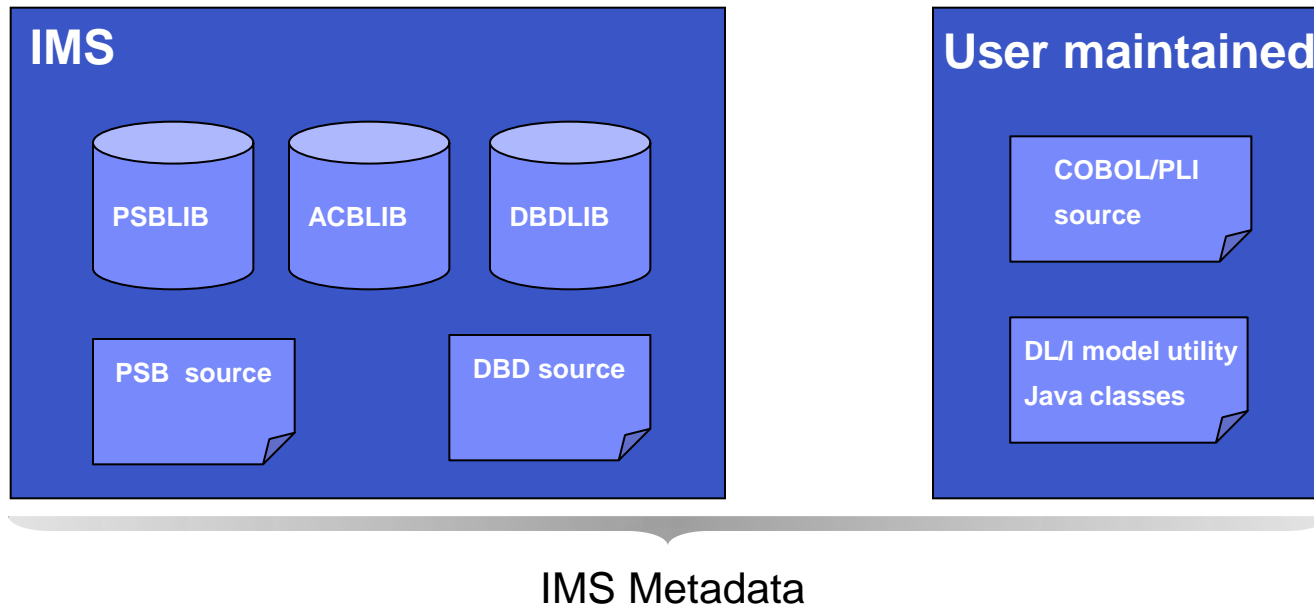


Agenda

- IMS Metadata
- IMS Open Database and Universal Drivers (V11)
- IMS Catalog for Metadata (V12)
- IMS Catalog Lifecycle
- IMS Catalog Database
- Enablement and Implementation of the IMS Catalog
- Application Programming with the IMS Catalog

IMS Metadata

- Databases partially defined in the IMS DBD
 - Only key/searchable fields needed by applications
 - Remaining segment data is not defined
- Remaining database segment definition is within Applications
 - COBOL COPYBOOKS and PL/I INCLUDEs map all the segment data
 - Applications can have different mappings for one segment



IMS Metadata



ord# INT	lastName CHAR	firstName CHAR	payment CHAR	type CHAR	date DATE
1011000100110	101100000101001001101	101101100100010100110	0101101010110110010101001010100101010	0110100010101010010111101010	0111010100100
0111010100100	011101011001001001001	011101110001100100100	0000111011101100101011101010000010101	01101110010000100111010010111	

*Defined in
DBD*

*Defined in
Appl Cpylibs
(IMS Java)*

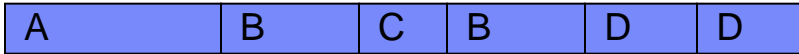
*Defined in
XML Schema:
Local file or
IMS Catalog*

```
<PurchaseOrder number="113246">
  <lastName> IBM      </lastName>
  <firstName> IMS     </firstName>
  <date>04/26/1968 </date>
  <payment type="MC"> 5414 2263 4895 1145 </payment>
```

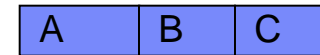


DBD + PSB = IMS DB Schema

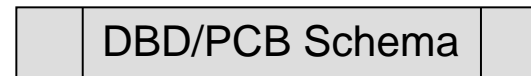
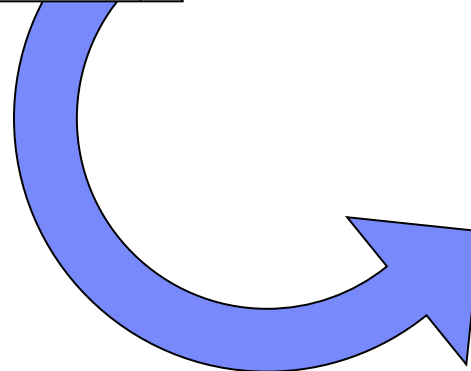
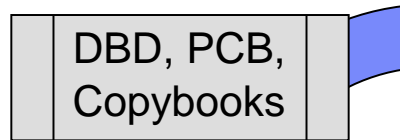
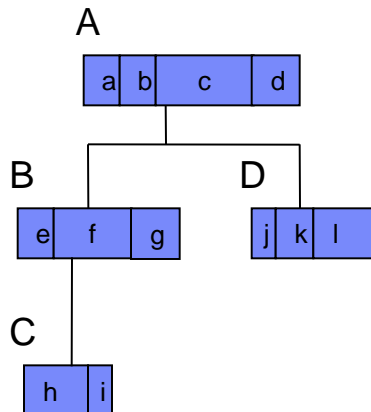
Physical Disk Data



Program I/O Area



IMS Hierarchy



Changes to DBD Statement



- **DBD**
 - ENCODING=
 - Specifies default encoding of all character data in the database defined by this DBD
 - Value can be overridden in individual segments or fields
- **SEGM**
 - EXTERNALNAME=
 - Optional alias for NAME= parameter used by Java application programs to refer to segment
- **FIELD**
 - CASENAME=
 - Name of the map case when alternative mappings are defined for fields in a segment
 - DATATYPE=
 - Specifies the external data type of the field
 - EXTERNALNAME=
 - Optional alias for the NAME= parameter used by Java application programs to refer to field
- **DFSMARSH**
 - Specifies the DATATYPE converter routine to transform binary data to external data format
- **DFSMAP**
 - Defines alternative field mappings in a segment
- **DFSCASE**
 - Defines a map case for a segment type that uses DFSMAP alternative field mapping
- **REMARKS=**
 - Optional user comments: 1- to 256-character field added to DBDGEN statement



Data Types



- Data types have multiple metadata elements
 - **Application data type**
 - Universal drivers use application data type to present data to clients
 - **Physical data type**
 - Universal drivers use physical data type to marshal/unmarshal data to and from the database
 - **Example**
 - Application data type is DECIMAL(10,2) [decimal with precision 10 and scale 2]
 - Physical data type is a signed packed decimal (AD community doesn't need to know this)
- New data type support
 - **Structs** (nested n levels with no constraint on element data types)
 - Accessed via SQL and DLI
 - **Arrays** (nested n levels with no constraint on element data types)
 - Accessed via SQL and DLI
 - **User-defined**
 - Name of UDT can be defined to the catalog and intended to be used at runtime by Universal drivers to marshal/unmarshal data
 - Can be part of a Struct or Array element



Changes to PSB PCB Statement



- **PCB**

- **EXTERNALNAME=**

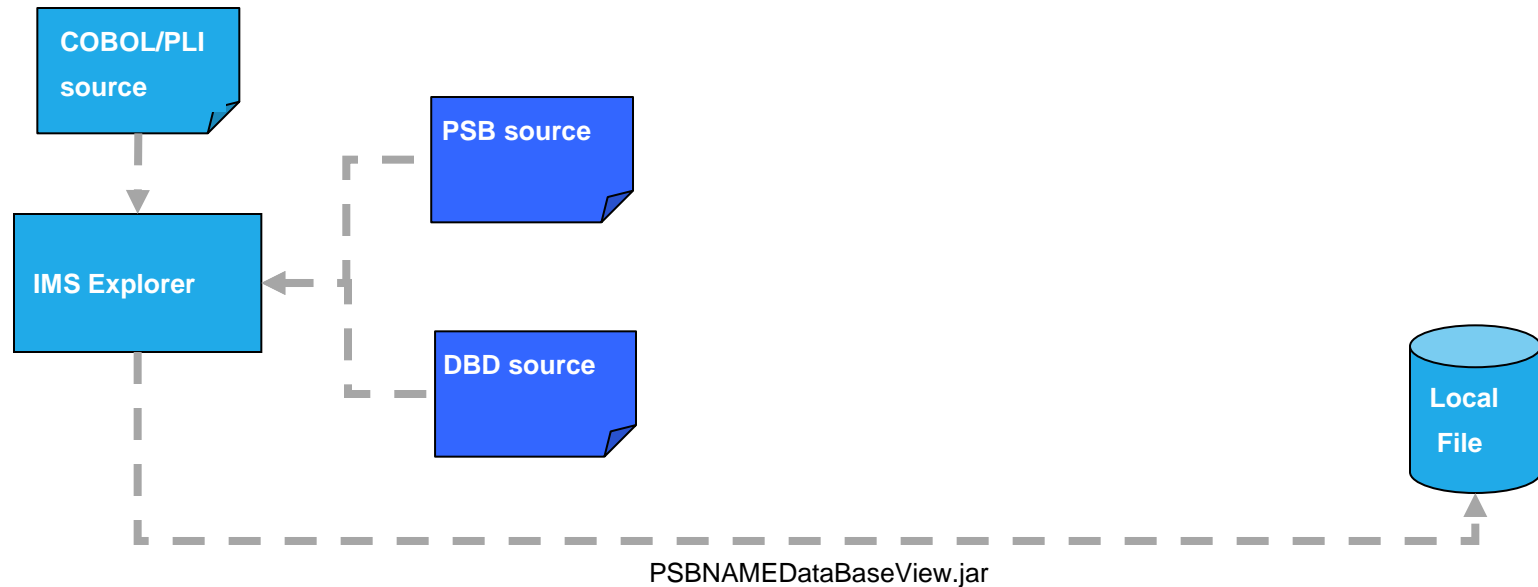
- Optional alias for the PCB label or the PCBNAME= parameter
- Java application programs use the external name to refer to the PCB

- **REMARKS=**

- Optional user comments: 1- to 256-character field added to PSBGEN statements



Local File for IMS Java Metadata



- IMS Explorer parses DBD, PSB and COBOL or PL/I source code
 - Generates PSBNAMEDataBaseView.java and .class files
 - Export as .jar file to file system where application is deployed
- Key points
 - Metadata deployed to multiple file locations
 - Metadata offline and not guaranteed to be current

IMS Open Database (IMS V11)

- Enables access to IMS databases anywhere in the Enterprise directly from local and distributed Java clients !
 - Clients use industry standard DRDA (Distributed Relational Database Architecture) to communicate with IMS Connect
 - Universal Drivers
 - Universal DB Resource Adapter (JEE)
 - Universal JDBC Driver
 - Universal DLI Driver
 - Roll Your Own

Opens the IMS Databases

Makes IMS DB Connectivity and

Application development simpler



IMS Connect becomes the TCP/IP gateway to IMS Transactions and IMS Data

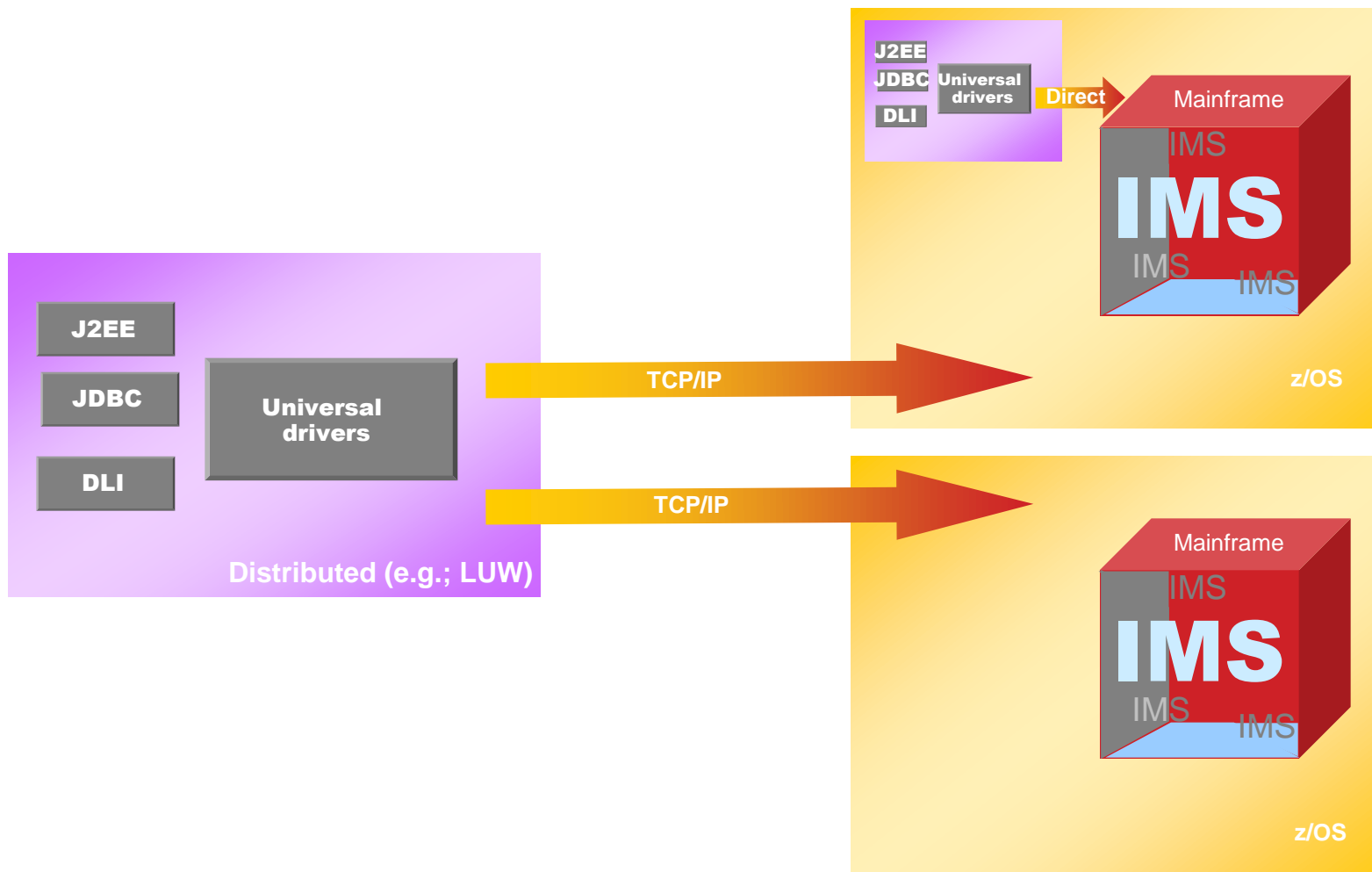
Solution Highlights – Connectivity



- Type 4 Driver Support
 - Universal drivers support Type 4 connectivity to IMS databases from TCP/IP enabled platforms and runtimes
 - WebSphere Application Server
 - Windows
 - zLinux
 - z/OS
 - Standalone Java SE
- Type 2 Driver Support
 - Universal drivers support Type 2 connectivity to IMS databases from z/OS runtimes
 - WebSphere Application Server for z/OS
 - IMS Java dependent regions (JMP & JBP)
 - CICS
 - All environments zAAP eligible (includes zAPP on zIIP)



IMS Open Database



Solution Highlights – Universal Drivers



- Three Universal drivers
 - Universal DB Resource Adapter
 - JEE deployment
 - Standard Java SE deployment
 - WebSphere Appl. Server
 - SQL Calls
 - Universal JDBC driver
 - Non-JEE and JEE deployment
 - Standard Java SE deployment
 - SQL Calls
 - Universal DLI driver
 - Non-JEE and JEE deployment
 - Standard Java SE deployment
 - DL/I call concepts (SSA, command codes)



Solution Highlights – JEE Deployment



- Universal DB Resource Adapter
 - JCA 1.5
 - XA transaction support
 - Manage multiple datasource connections in a single UOW
 - Local transaction support
 - Manage multiple datasource connections each in their own UOW
 - Connection pooling
 - Connections released for future use
 - Connection sharing
 - Multiple programming models available
 - JDBC (Universal JDBC driver incorporated)
 - CCI with SQL interactions
 - CCI with DLI interactions



Solution Highlights – JDBC



- Universal JDBC driver
 - Significant enhancements to classic JDBC offered in IMS 9 and IMS 10
 - Standardized SQL support for an SQL subset
 - Keys of parent segments are included in tables as foreign keys
 - XA transaction support (type 4)
 - Local transaction support (type 4)
 - Concurrency control
 - Control release of distributed locks
 - Updatable result set support
 - Batch support
 - Fetch multiple rows in a single network call
 - JDBC metadata discovery support
- Standard SQL and metadata discovery enables significant integration opportunities for IMS
 - Use metadata stored in the IMS Catalog
 - Enable JDBC tooling to work with IMS DBs



Solution Highlights – DLI



- Universal DLI driver
 - Java implementation of the DL/I API
 - Complete DL/I support for database access
 - All IMS command codes supported
 - Can mix usage of JDBC and DLI drivers in the same application
 - SQL cannot always express what DLI offers



Solution Highlights – Java Dependent Region



- Java dependent region resource adapter
 - Java framework for applications operating in an IMS container
 - Message queue processing
 - Program switching
 - Deferred and immediate
 - Transaction demarcation
 - GSAM support
 - Additional IMS call support necessary for IMS transactions
 - INQY
 - INIT
 - LOG
 - Etc
 - Shipped with Type 2 Universal drivers



Open Database Manager - ODBM

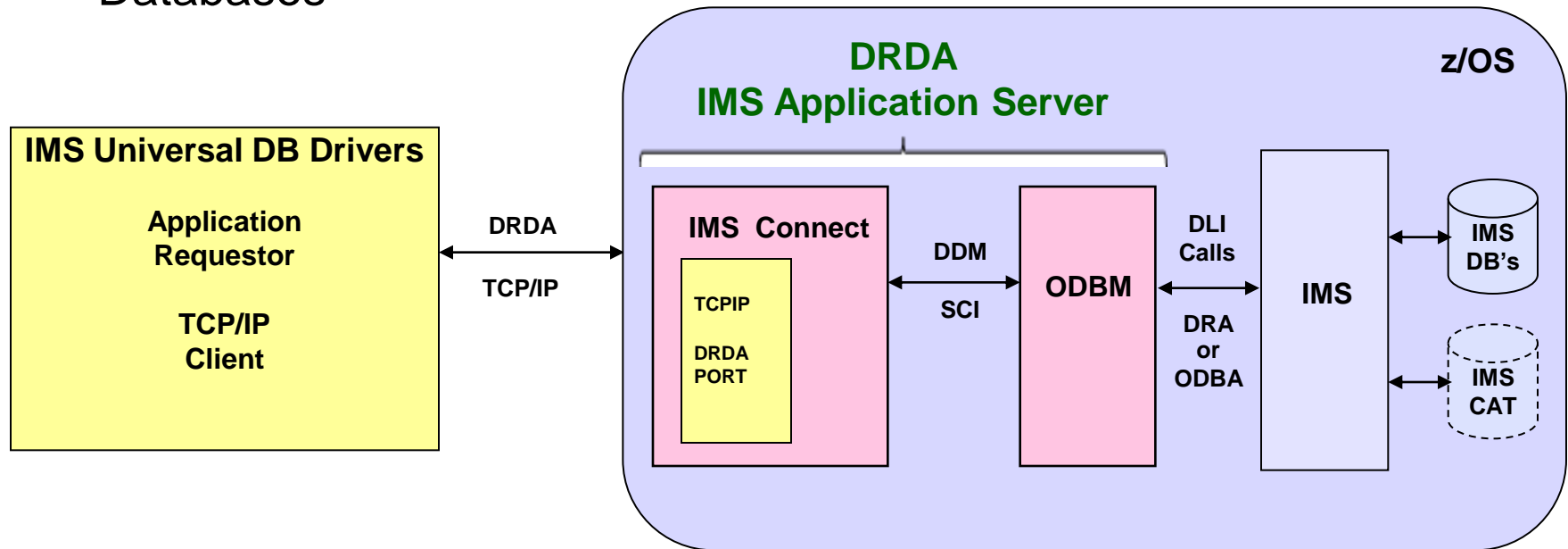


- z/OS address space
 - Common Service Layer component
- Functions
 - Receives database connection requests from IMS Connect
 - Translates incoming database requests from DDM protocol to DL/I calls
 - Manages ODBA or DRA interface to IMS
 - Sends/receives byte string to/from IMS
 - Translates responses into DDM protocol for the client
 - Send results set back to client
- Provides isolation of application containers from IMS
 - Abend of container during DL/I call does not cause IMS U0113 abend
- Distributed Syncpoint (global transaction) requires RRS on z/OS
- Use of RRS with ODBM is optional
 - If RRS=Y (default), ODBM uses ODBA interface
 - If RRS=N, ODBM uses DRA interface (like DBCTL interface)



Open Database Manager - ODBM

- IMS Connect and Open Database Manager work together as a DRDA server for IMS data
- Allows distributed access to IMS Databases via TCP/IP using the IMS Universal Database Drivers
- Enables use of SQL database query language for access to IMS Databases



Open Database Manager - ODBM



- ODBM Setup and Configuration
 - PROCLIB members
 - ODBM BPE Configuration
 - ODBM BPE exit list
 - ODBM Initialization member – CSLDIxxx
 - ODBM Configuration member – CSLDCxxx
 - CSLODBM procedure
 - ODBM execution parameters
- New type-2 commands
 - QUERY ODBM TYPE(option)
 - UPDATE ODBM START|STOP(TRACE)
 - UPDATE ODBM START|STOP(CONNECTION)
 - UPDATE ODBM TYPE(CONFIG)
- Four new ODBM exit routines

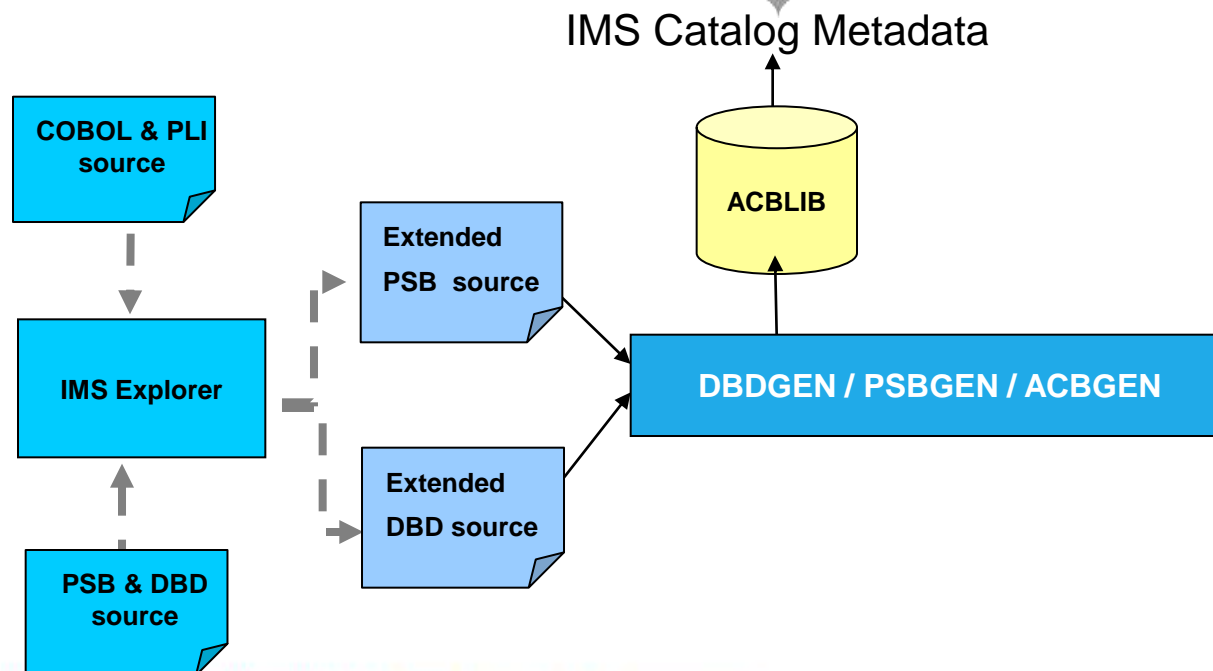


IMS Catalog for IMS Java Metadata (IMS V12)

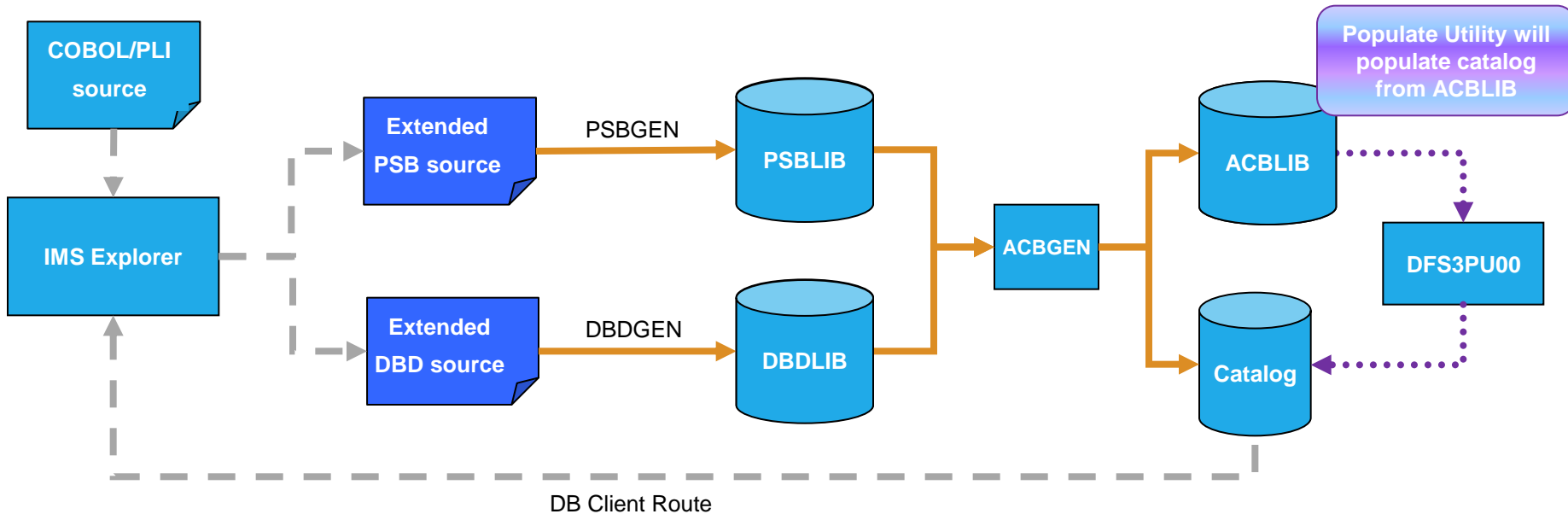


Contains IMS program & database resources, and relevant application metadata

Database Definitions	Application Program Specifications
<ul style="list-style-type: none">▪ Segments▪ Fields	<ul style="list-style-type: none">▪ Arrays and Structures▪ Field data types and data type conversion▪ Field redefines▪ Alternative Field maps for a segment



IMS Catalog Lifecycle



- ACBGEN will populate ACBLIB and catalog in same UOW
 - Populates ACBLIB with standard ACB info and extended info
 - Populates the catalog with extended info
- Key points
 - Trusted, online, IMS metadata stored in the IMS catalog
 - Only way to update catalog is via the Populate Utility or ACBGEN process
 - Extended info is acquired via the IMS Explorer
 - Extended info stored in ACBLIB members for recoverability

The IMS Catalog Database



- Contains metadata related to an IMS system's databases & programs
 - DBD and PSB and Application information
- IMS PHIDAM/OSAM HALDB database
 - Defined with 4 Data Set Groups (DSGs)
- One Secondary Index
- Unique feature → “Unregistered Catalog Database”
 - DBRC use is optional for the IMS Catalog database
 - Only HALDB that isn't required to be defined to DBRC
 - IMS can manage allocation/creation of catalog database
 - Done via parameters in the “CATALOG” section of DFSDFxxx PROCLIB member

The IMS Catalog Database



- IMS provides the DBD and PSB source code and object code for the Catalog database
- PHIDAM DBD name is DFSCD000
- PSINDEX DBD name is DFSCX000
 - Used to connect DBDs to PSBs that reference them
- PSBs provided to access the Catalog database
 - Initial load process - DFSCPL00
 - Used by the Catalog Populate Utility
 - Update access - DFSCP001
 - Used by the new ACBGEN Utility
 - Read access - DFSCP000 (COBOL/HLASM), DFSCP002 (PL/I), and DFSCP003 (PASCAL)
- Default catalog PCB is DFSCAT00
- ACBGEN required for all catalog PSBs

Let's look at the IMS Catalog database physical structure...

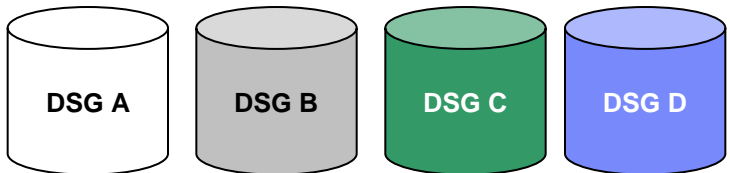
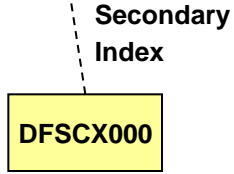
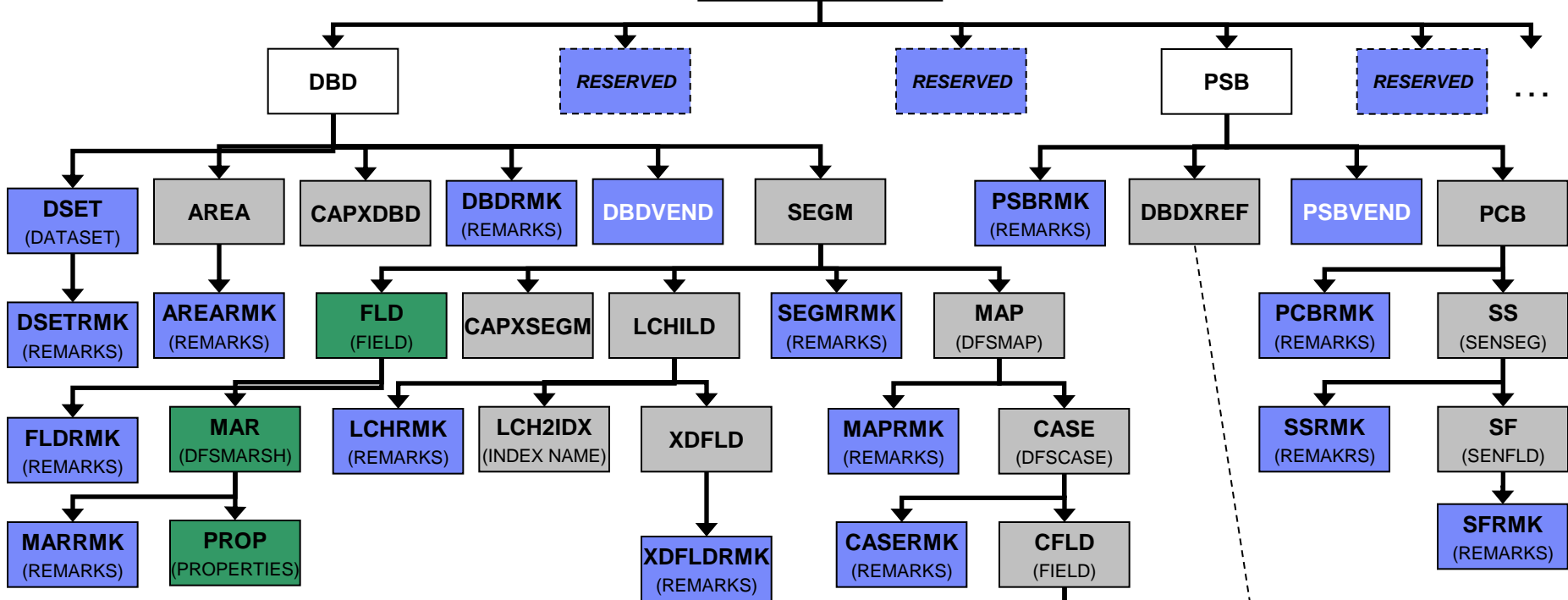


Physical Catalog Record Structure

• Key value: **DBD**bbbbbbdbdname1



• Key value: **PSB**bbbbbbpsbname1



Supported by Existing Database Utilities

- HALDB Index/ILDS Rebuild utility (DFSPREC0)
- HD Reorganization Unload utility (DFSURGU0)
- HD Reorganization Reload utility (DFSURGL0)
- Batch Backout utility (DFSBB000)
- Database Recovery utility (DFSURDB0)
- Database Image Copy utility (DFSUDMP0 / DFSUDMT0)
 - Only batch image copies for an unregistered Catalog

New Utilities for the IMS Catalog

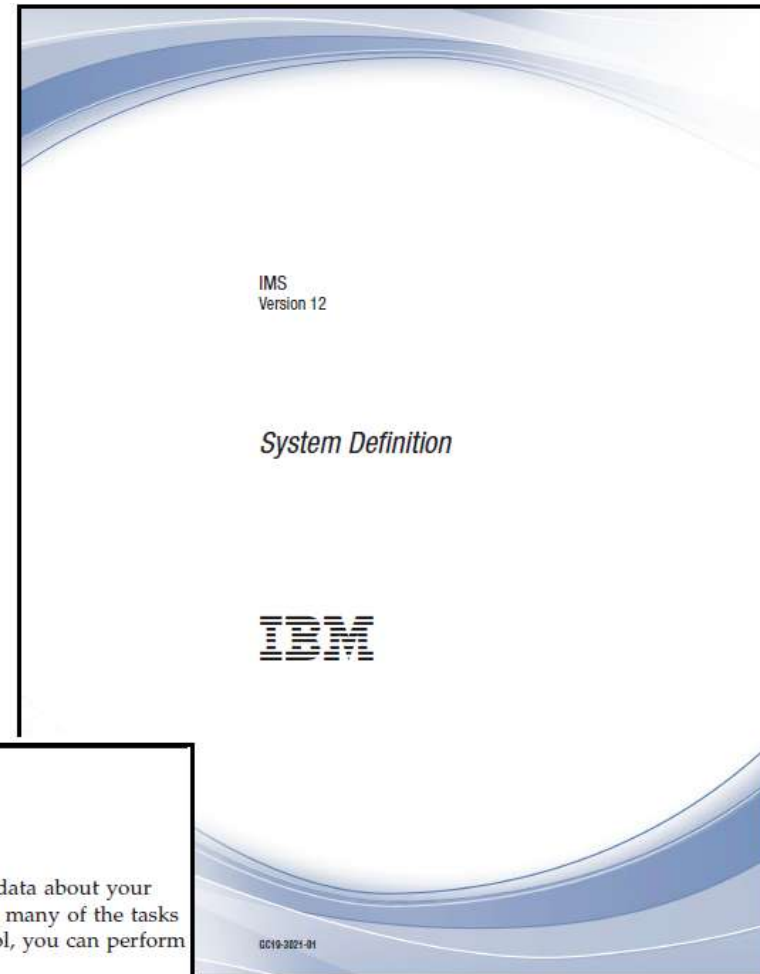


- **IMS Catalog Populate utility (DFS3PU00)**
 - Loads or inserts records into the IMS catalog
 - Estimates the size of the IMS catalog data sets
- **ACB Generation and Catalog Populate utility (DFS3UACB)**
 - Generates the ACBLIB members and loads the IMS catalog metadata in one unit of work
- **IMS Catalog Alias Names utility (DFS3ALI0)**
 - Defines a complete list of catalog aliases to IMS
- **IMS Catalog Copy utility (DFS3CCE0, DFS3CCI0)**
 - Copies IMS catalog database definitions and metadata between IMS systems
 - Migrate an IMS catalog database from a test to production environment
 - Copy a production IMS catalog database to another installation for system cloning or DR purposes (etc...)
- **IMS Catalog Partition Definition Data Set utility (DFS3UCD0)**
 - When DBRC is not used to manage the catalog database: creates the IMS catalog partition definition data set and populates it with information about the IMS catalog HALDB partitions
- **IMS Catalog Record Purge utility (DFS3PU10)**
 - Removes unnecessary versions of IMS catalog records from the IMS catalog database based on criteria that you specify



IMS Batch Processing with the Catalog

- Enabling the IMS catalog for IMS batch processing
 - Specify DFSDFxxx member on job EXEC parm
 - Requires JCL change to implement
- Exit Routine (DFS3CDX0)
 - Optional Catalog Definition user exit routine
 - Alternative to specifying DFSDFxxx member through job JCL
 - Available if users cannot or choose not to modify job JCL



Chapter 9. IMS catalog definition and tailoring

The IMS catalog is an optional system database that stores metadata about your databases and applications. As a system database, IMS performs many of the tasks required to define the catalog for you. If you need greater control, you can perform some of these tasks yourself.

The following topics provide the information you need to define and enable the IMS catalog.

IMS Catalog Enablement



- Add catalog DBDs and PSBs to your DBDLIB, PSBLIB & ACBLIB
 - Copy DBD and PSB object code from SDFSRESL to your DBDLIB and PSBLIB

```
//CPYCMEM EXEC PGM=IEBCOPY
//SDFSRESL DD DSN=SDFSRESL,DISP=SHR
//DBDLIB DD DSN=MYIMS.DBDLIB,DISP=OLD
//PSBLIB DD DSN=MYIMS.PSBLIB,DISP=OLD
//SYSIN DD *
        COPY OUTDD=DBDLIB,INDD=( (SDFSRESL,R) ),LIST=YES
        SELECT MEMBER=(DFSCD000,DFSCX000)
        COPY OUTDD=PSBLIB,INDD=( (SDFSRESL,R) ),LIST=YES
        SELECT MEMBER=(DFSCPL00,DFSCP000,DFSCP001,DFSCP002,DFSCP003)
```

- ACBGEN the catalog DBD and PSB resources into your ACBLIB

```
//CATABCB EXEC PGM=DFSRR00,PARM='UPB'
//STEPLIB DD DSN=SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=SDFSRESL,DISP=SHR
//IMS DD DSN=MYIMS.DBDLIB,DISP=SHR
// DD DSN=MYIMS.DBDLIB,DISP=SHR
//IMSACB DD DSN=IMS.ACBLIB,DISP=OLD
//SYSIN DD *
        BUILD PSB=(DFSCPL00)
        BUILD PSB=(DFSCP001)
        BUILD PSB=(DFSCP000)
```

N.B. The MODBLKS resources for the Catalog databases and programs do not need to be defined !

IMS Catalog Enablement

- Modify DFSDFxxx PROCLIB Member
 - New CATALOG section(s) for catalog related parameters
 - Single section format <SECTION=CATALOG>
 - Multiple section format <SECTION=CATALOG*imsid*>
 - Multiple IMS systems sharing one DFSDFxxx PROCLIB member
 - *imsid* suffix must be a four character IMS ID
 - CATALOG section parameters
 - CATALOG=N | Y
 - Catalog is disabled or enabled
 - If enabled, IMS automatically creates catalog DDIR & PDIRs at IMS startup
 - ALIAS=DFSC | xxxx (no default value)
 - Specifies any 1-4 alphanumeric value used as a Catalog database name prefix
 - Enables use of non-shared, Catalog databases within an IMSplex
 - Use in a data sharing environment where each IMS has its own Catalog database and all are registered in a single set of RECONS
 - At runtime, the alias Catalog database names are dynamically replaced with internal database names **DFSCD000** and **DFSCX000**
 - For standalone IMS system – use “**DFSC**” which is the standard Catalog database name prefix → **DFSCD000** and **DFSCX000**

IMS Catalog Enablement



- DFSDFxxx PROCLIB Member
 - CATALOG section parameters (continued)
 - Information used by Catalog Populate Utility to automatically allocate the Catalog database data sets
 - DATACLAS
 - Optional data class for SMS managed data sets
 - MGMTCLAS
 - Optional management class for SMS managed data sets
 - STORCLAS
 - Required storage class for SMS managed data sets
 - IXVOLSER
 - Volume serial number for primary and secondary catalog indices
 - Required for non-SMS managed data sets
 - SPACEALLOC
 - Free space % (0 to 9999) added to the IMS-computed size of the primary & secondary data set allocations
 - SMSVOLCT
 - Number of volumes (1-20) created by the Catalog Populate utility for SMS-managed



IMS Catalog Enablement



- DFSDFxxx PROCLIB Member
 - CATALOG section parameters (continued)
 - RETENTION=(MAX=2 | nnn) or RETENTION=(PERIOD=0 | nnn)
 - Specifies retention schedule for metadata in the IMS catalog
 - By default IMS keeps only two copies of the DBD or PSB in catalog
 - MAX=2 | nnn
 - Maximum number of versions of a DBD or PSB to be stored before they are replaced first-in first-out
 - PERIOD=0 | nnn
 - Maximum number of days a version of a DBD or PSB is to be stored before it can be replaced
 - Metadata versions older than the specified retention period are not automatically deleted, but available for removal when new version of metadata is added
 - Default value of “0” disables this feature



IMS Catalog Enablement



- Definition of the HALDB structure
 - Partitioning of the catalog is the users responsibility
 - Minimum of 1 partition is required
 - Last partition must be able to contain the highest-key PSB record
 - Catalog HALDB uses the high-key selection method
 - No use of Partition Selection Exit allowed
 - Catalog Database Definition
 - For systems that use DBRC
 - Catalog database can be defined to the RECONs with the DBRC utility and commands
 - For systems that do not use DBRC
 - Catalog database must be defined to the Catalog Partition Definition data set using the Catalog Partition Definition Data Set utility, DFS3UCD0
 - If an ALIAS is used in the CATALOG sections of the DFSDFxxx member, each alias Catalog database must be defined



IMS Catalog Enablement



- Using DBRC DSPURX00 utility and commands to define the Catalog database to the RECONS

```
//DEFCAT EXEC PGM=DSPURX00
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IMS DD DSN=IMS.DBDLIB,DISP=SHR
//SYSIN DD *
INIT.DB DBD(DFSCD000) TYPHALDB SHARELVL(3)
INIT.PART DBD(DFSCD000) PART(DFSCD01) -
  DSNPREFIX(dsnprefix.DFSCD000) -
  BLOCKSZE(4096) -
  KEYSTRNG(X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF')
INIT.DB DBD(DFSCX000) TYPHALDB SHARELVL(3)
INIT.PART DBD(DFSCX000) PART(DFSCX01) -
  DSNPREFIX(dsnprefix.DFSCX000) -
  KEYSTRNG(X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF') /*
```

- If using a catalog alias prefix, replace **DFSC** in the database and partition names for the catalog and the catalog secondary index with the four character ALIAS name prefix
- You might need to define multiple alias name databases to the RECONS



IMS Catalog Enablement



- Using the Catalog Partition Definition Data Set utility, DFS3UCD0, to define the Catalog database (for systems that do not use DBRC)

```
//S1 EXEC PGM=DFS3UCD0,REGION=0M
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSHDBSC DD DSN=...,DISP=
//SYSPRINT DD SYSOUT=*
//IMS DD DSN=IMS.DBDLIB,DISP=SHR
//SYSIN DD *
HALDB=(NAME=DFSCD000)
PART=(NAME=DFSCD000,PART=partitionname,
      DSNPREFIX=dsnprefix,
      KEYSTRNG=keystring)
HALDB=(NAME=DFSCX000)
PART=(NAME=DFSCX000,PART=partitionname,
      DSNPREFIX=dsnprefix,
      KEYSTHEX=FFFFFFFFFFFFFFFF) /*
```

Unregistered
Catalog database

- Catalog Partition Definition data set is populated with the information specified in the HALDB and PART control cards
 - RECON-like information for catalog database partition definition and structure
- The name DFSCD000 in the HALDB and PART statements contains the default catalog prefix **DFSC**. If your catalog uses an alias name prefix, substitute it in the JCL



IMS Catalog Enablement



- After Catalog database is defined in Catalog Partition Definition Data Set
 - Identify unregistered Catalog database names
 - UNREGCATLG parameter in the DATABASE section of the DFSDFxxx member

```
/* **** */
/* Database Section */
/* **** */
<SECTION=DATABASE>
UNREGCATLG=(DFSCD000,DFSCX000) /* Unregistered IMS catalog DB */
/* **** */
/* */
/* **** */
```

- If using an alias name prefix, replace **DFSC** in the UNREGCATLG database names with the four character alias name prefix
- Limitations of using an unregistered Catalog database
 - NO IMS Data Sharing support
 - NO OLR support
 - NO partition definition change support
 - User must rebuild catalog partitions
 - Manual recovery required for unregistered Catalog databases



IMS Catalog Enablement



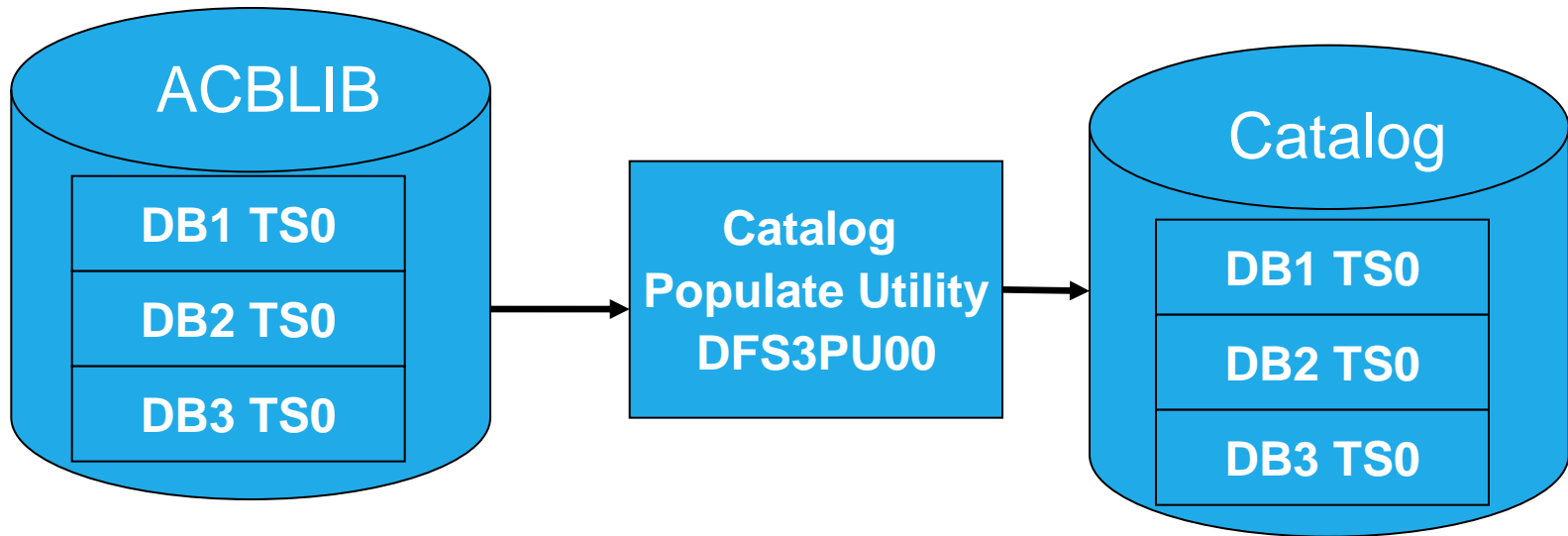
- After Catalog database is defined in Catalog Partition Definition Data Set
 - Create a new DFSMDA dynamic allocation member for the Catalog Partition Definition data set

```
//DYNALOC JOB
//*
//STEP EXEC IMSDALOC
//SYSIN DD *
    DFSMDA TYPE=INITIAL
    DFSMDA TYPE=CATDBDEF, DSNAME=dsn
    DFSMDA TYPE=FINAL
    END
/*
```

- **dsn** is the name of the Catalog Partition Definition data set
 - Catalog Partition Definition data set was allocated in the DFS3UCD0 utility on the DFSHDBSC DD
- Dynamically allocate Catalog Partition Definition data set in any IMS job step



IMS Catalog Members After Populate Utility



- Start with 3 members in an ACBLIB
- Run the Catalog Populate Utility, **DFS3PU00**
- Created catalog members will contain timestamp TS0
- This is the same timestamp from the ACB member

Catalog Populate Utility Report



CATALOG DFSCD000
PARTITION DFSCD01
NUMBER OF SEGMENTS INSERTED INTO THE CATALOG
INSERTED AVERAGE

SC	SEGMENT	SEGMENTS	DSG	PARENT	SEGS/PARENT
1	HEADER	4228	A		
2	DBD	2530	A	HEADER	0.6
3	CAPXDBD	7	D	DBD	0.0
5	DSET	2599	D	DBD	1.0
7	AREA	139	D	DBD	0.1
9	SEGM	16337	B	DBD	6.5
10	CAPXSEGM	1	D	SEGM	0.0
12	FLD	16426	C	SEGM	1.0
14	MAR	16426	C	FLD	1.0
17	LCHILD	2687	B	SEGM	0.2
20	XDFLD	134	B	LCHILD	0.0
33	PSB	1840	A	HEADER	0.4
35	PCB	9190	B	PSB	5.0
37	SS	75274	B	PCB	8.2
39	SF	1105	B	SS	0.0
41	DBDXREF	8886	D	PSB	4.8

SEGMENT within existing header duplicates not inserted

DBD	71	0
PSB	72	0

ESTIMATED SPACE REQUIREMENT TO HOLD INSERTED SEGMENTS

DSG	BLKSIZE	BLOCKS
A	4096	596
B	4096	9343
C	4096	8214
D	4096	236
DSG		RECORDS
L		8886
X		4230
SECONDARY INDEX		RECORDS
DFSCX000		8886

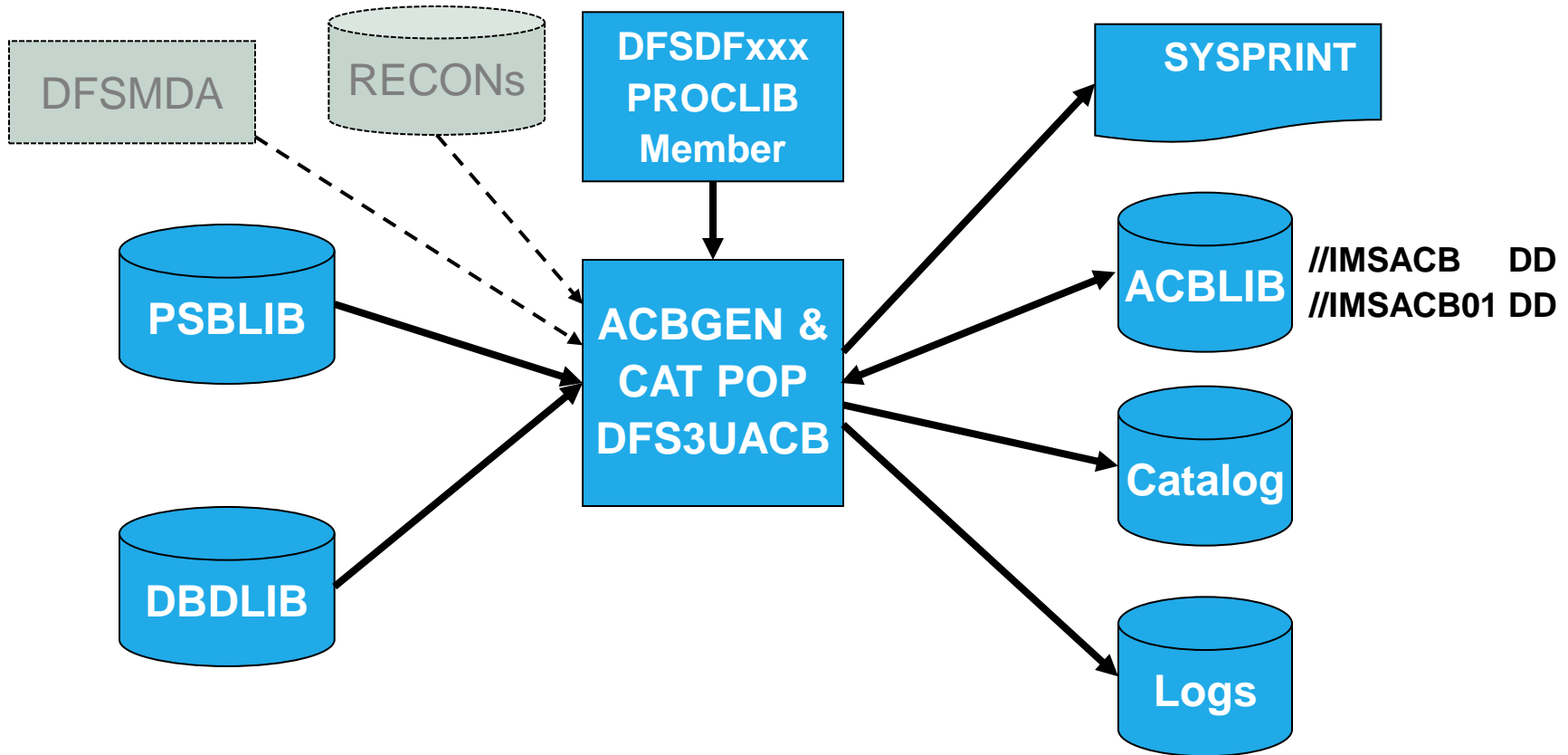
Adding IMS Catalog Metadata



- New ACB Generation - **DFS3UACB**
 - Replaces existing ACBGEN Utility, DFSUACB0 (if IMS catalog enabled)
 - Generate ACBLIB member and populate/create catalog metadata in a **single job step**
 - Phase 1 - ACBGEN
 - DBDLIB and PSBLIB members used as input
 - Validation is unchanged
 - ACB member is written to ACBLIB with new ACBGEN timestamp
 - Phase 2 – IMS catalog update
 - Generated ACB is decoded, converted to catalog segments , loaded into the catalog
 - DBD and PSB metadata created and inserted
 - Corresponding ACB member timestamp saved as timestamp in catalog DBD and PSB segments
 - ensures validity and consistency of ACBLIB and catalog
- New ACBGEN, DFS3UACB, and new Catalog Populate, DFS3PU00, utilities are the **only** updaters of the IMS catalog
 - IMS online and IMS batch regions will never update catalog data
 - IMS online and IMS batch regions will only retrieve data from the catalog

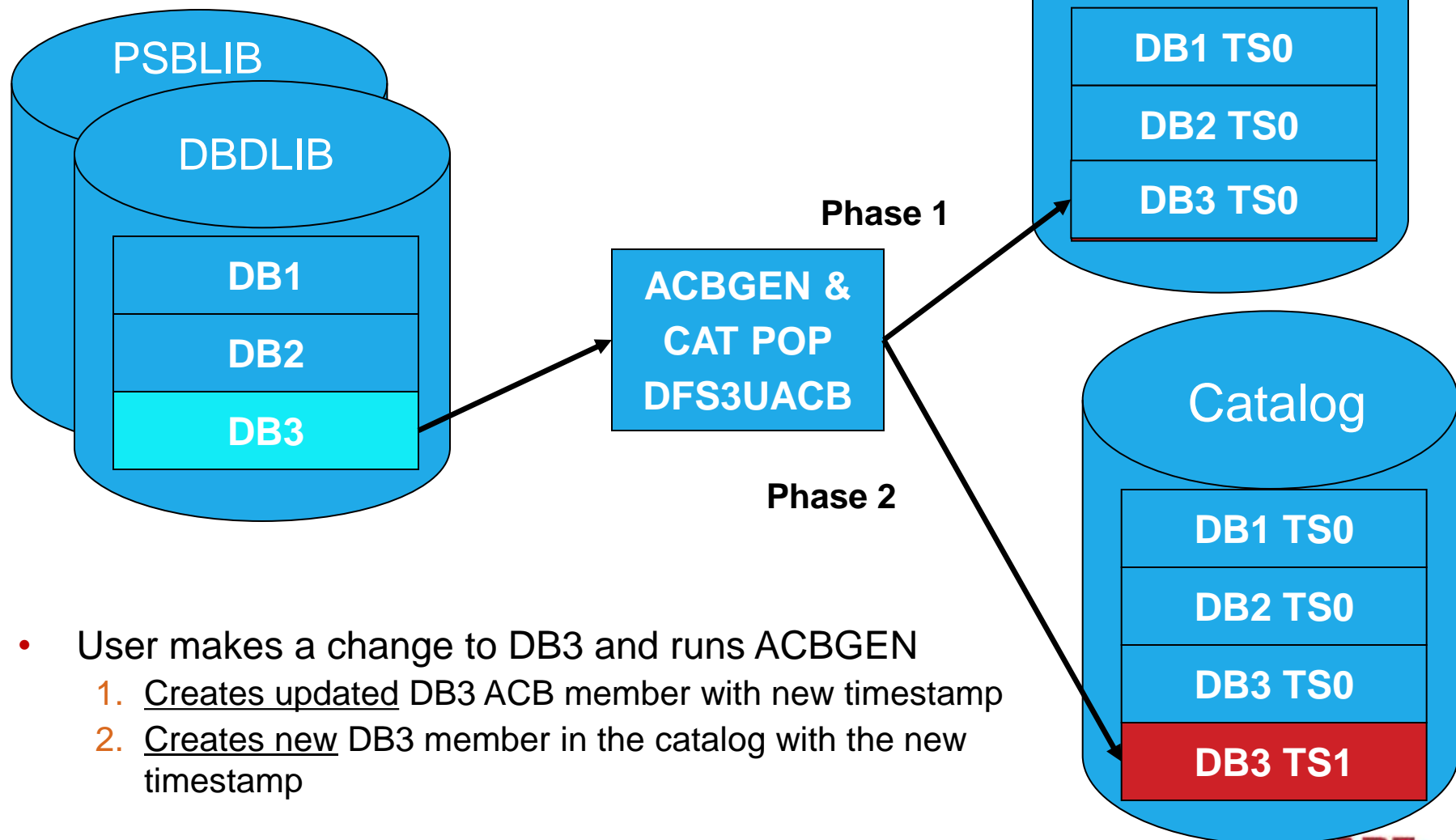


IMS Catalog Additions: New ACBGEN



- New integrated ACBGEN process includes updates to the IMS catalog
- DFSDFxxx PROCLIB member has the catalog information
- DFSMDA member used to dynamically allocate the catalog datasets

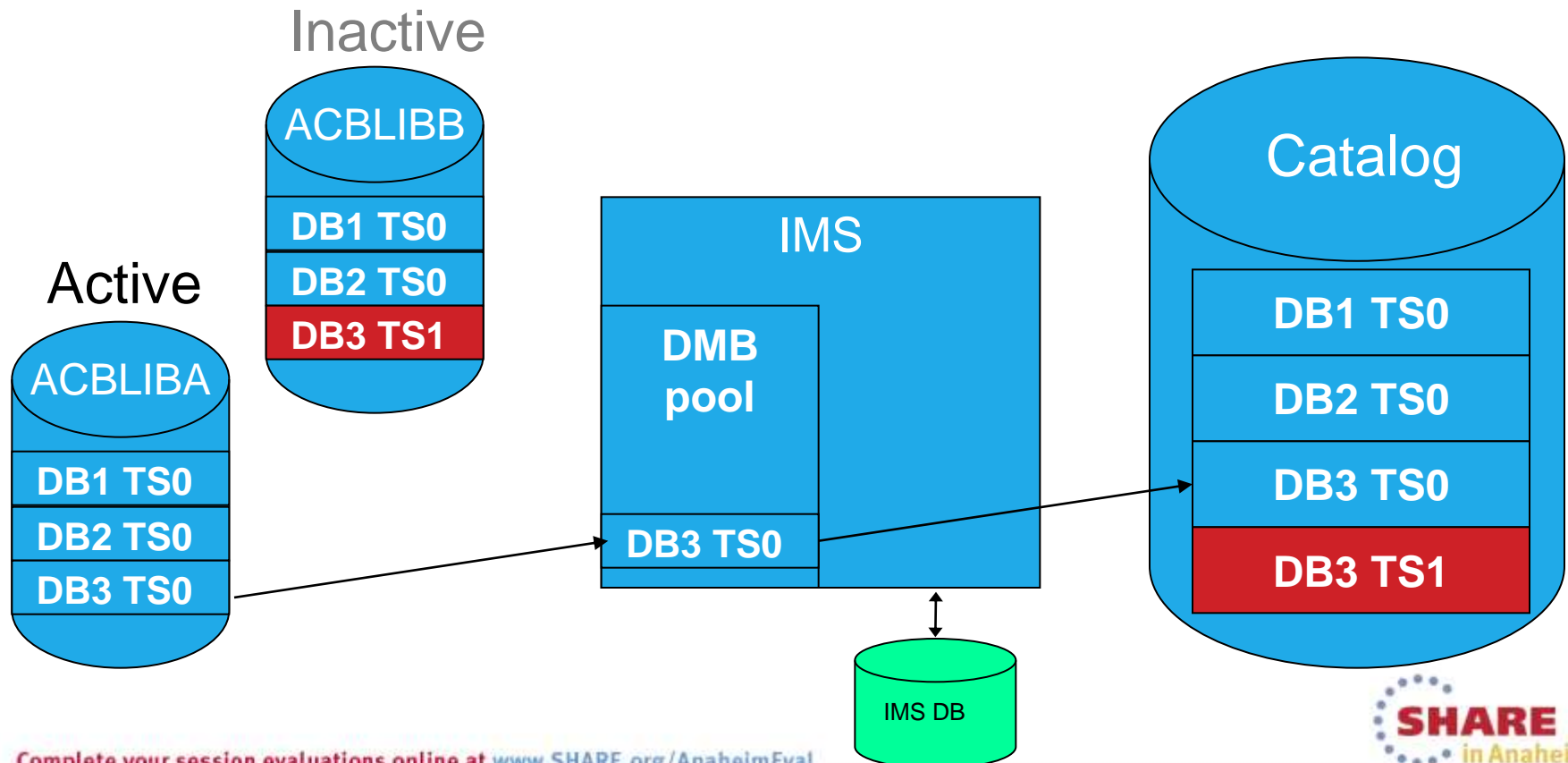
ACBLIB and Catalog After New ACBGEN



- User makes a change to DB3 and runs ACBGEN
 1. Creates updated DB3 ACB member with new timestamp
 2. Creates new DB3 member in the catalog with the new timestamp

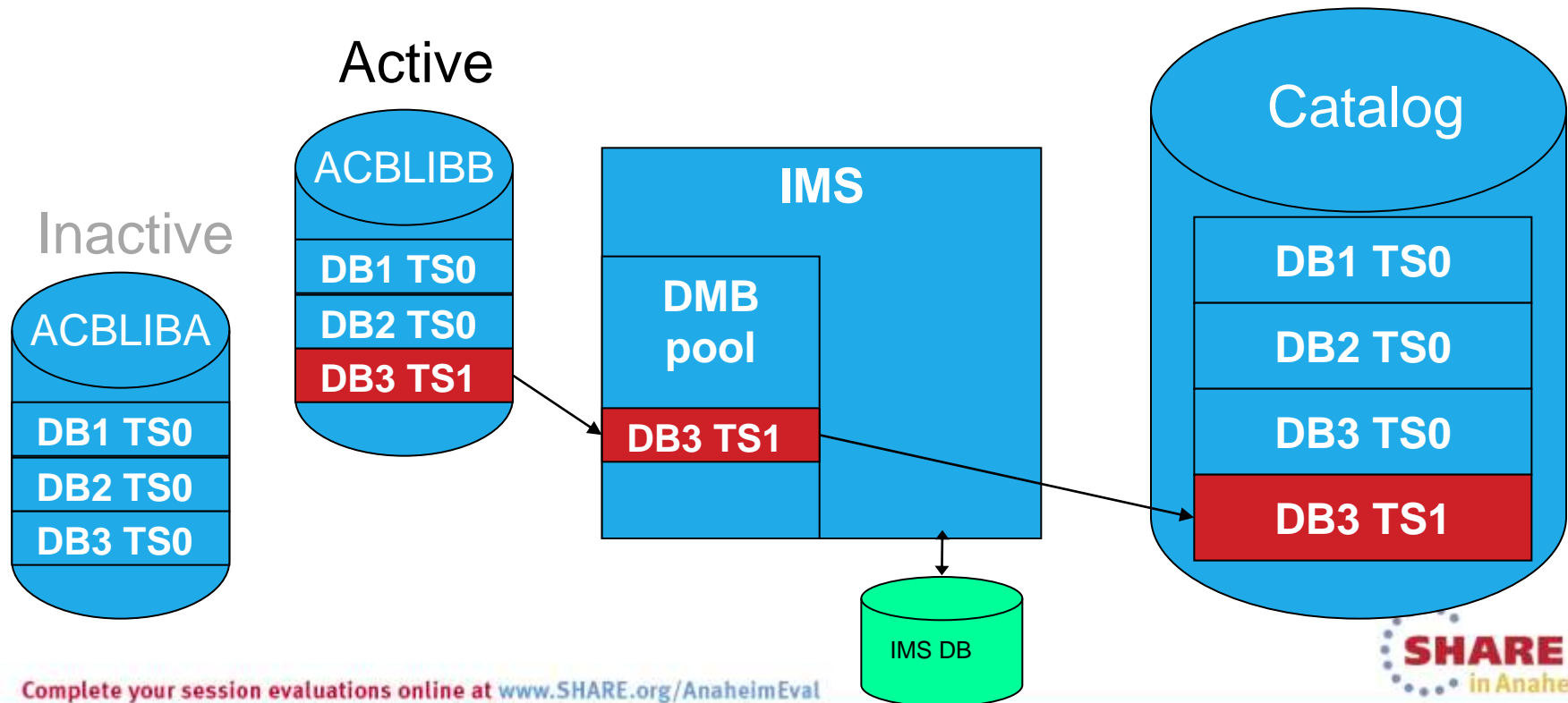
Catalog Member Timestamps

- Application request is made to read DB3
 1. IMS determines active DB3 member has timestamp TS0
 2. Internal DL/I call issued to retrieve member DB3 from IMS catalog
 3. IMS retrieves metadata from catalog member DB3 with timestamp TS0



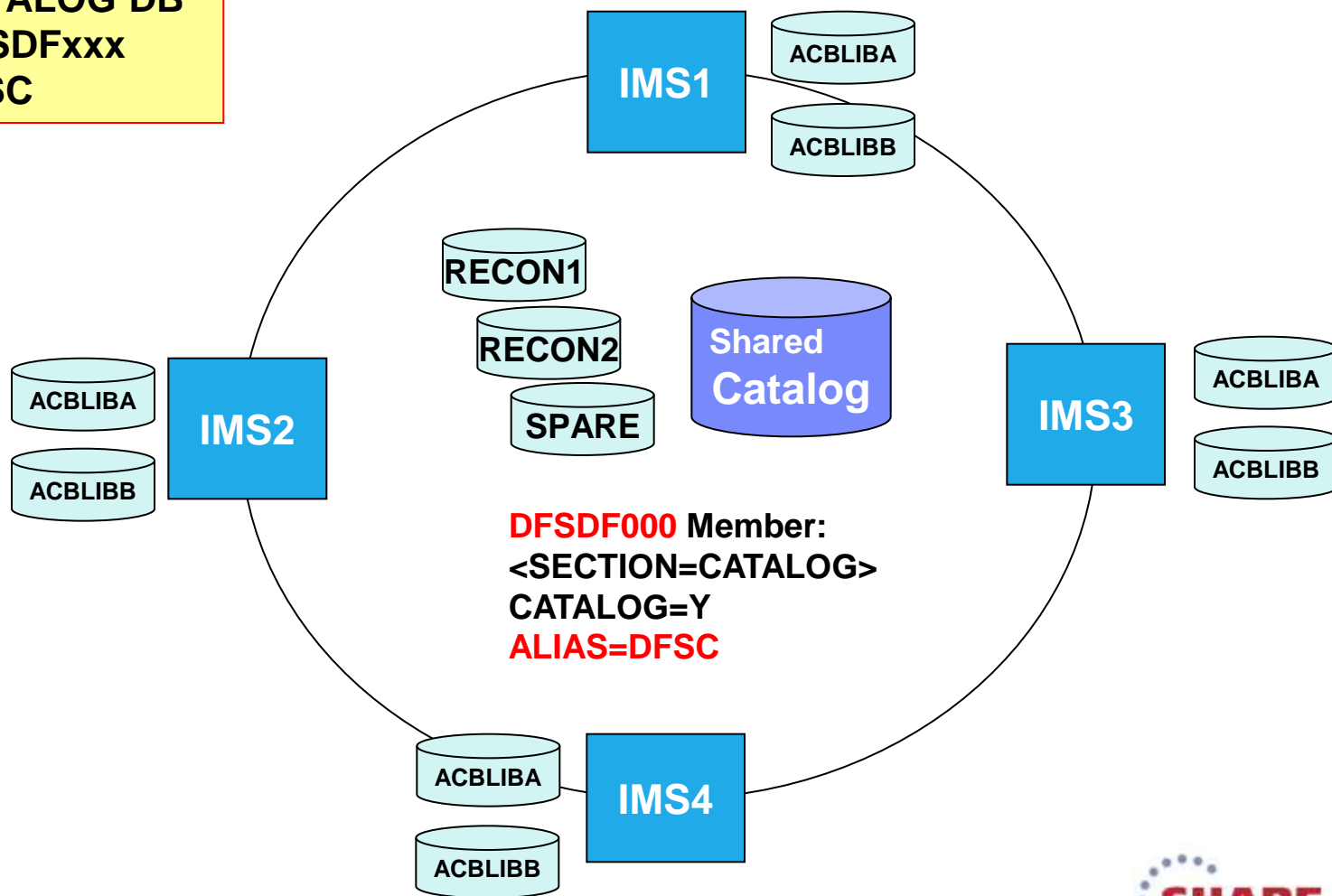
Catalog Member Timestamps

- Initiate OLC to switch from ACBLIBA to ACBLIBB
 - Activates DB3 ACB with timestamp TS1
- Application request is made to read DB3
 1. Internal DL/I call issued to retrieve member DB3 from IMS catalog
 2. IMS determines active DB3 member has timestamp TS1
 3. IMS retrieves metadata catalog member DB3 with timestamp TS1



Multiple IMSES, Cloned ACBLIBs, Shared Catalog

Shared CATALOG DB
Shared DFSDFXxx
ALIAS=DFSC

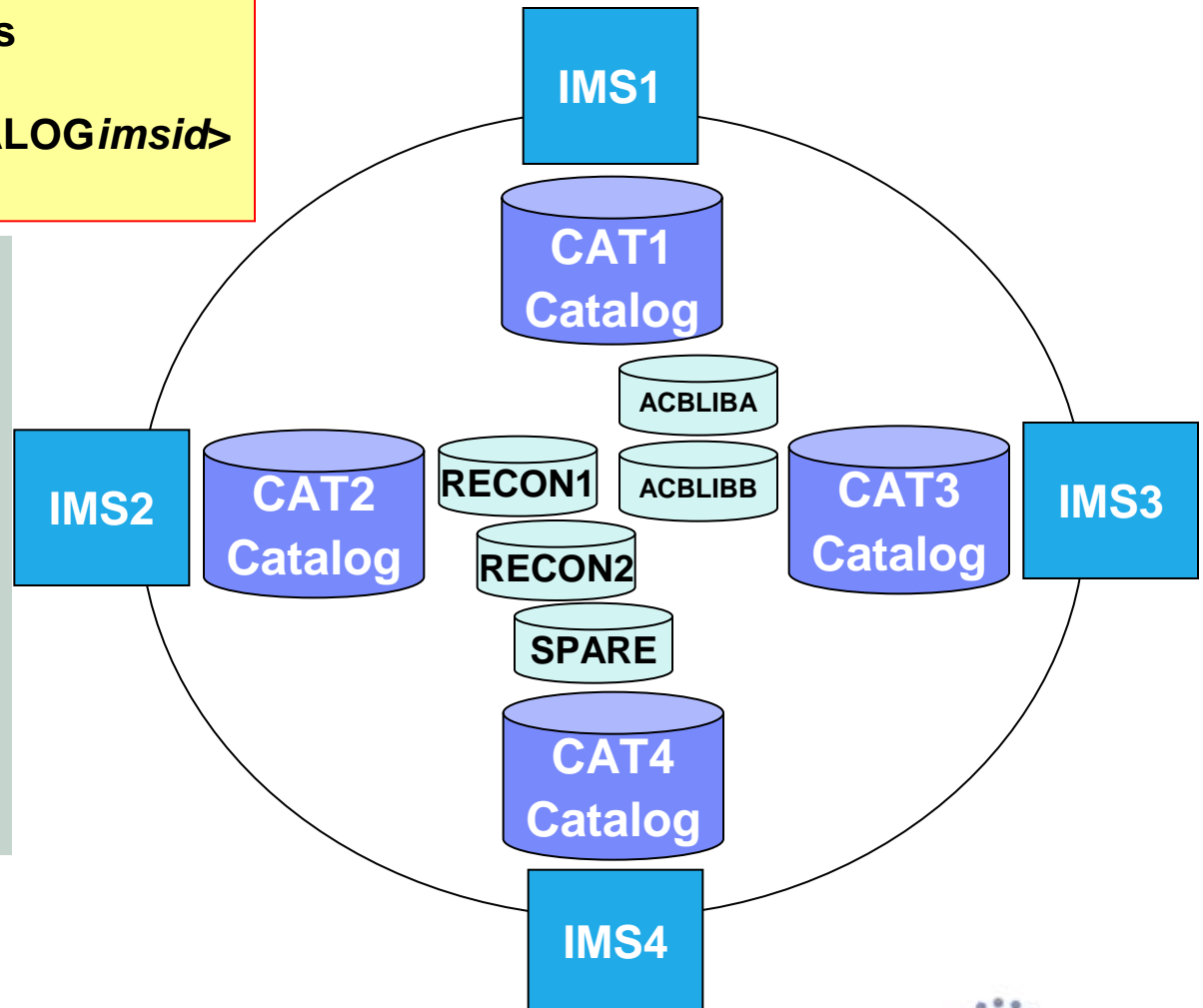


Multiple IMSes, Shared ACBLIBs, each IMS has it's own Catalog

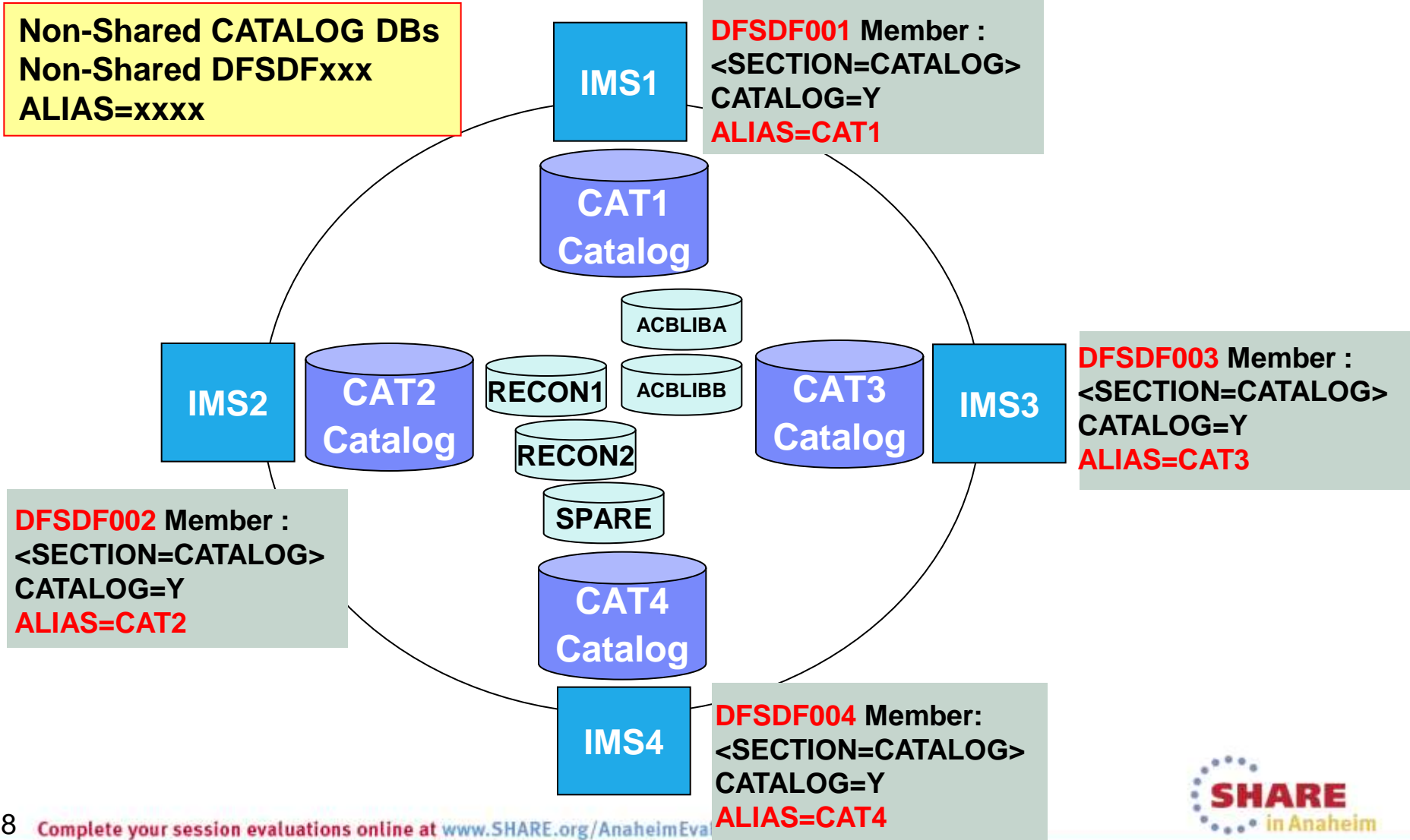
Non-Shared CATALOG DBs
Shared DFSDFxxx
Separate <SECTION=CATALOGimsid>
ALIAS=xxxx

DFSDF000 Member:

```
<SECTION=CATALOGIMS1>  
CATALOG=Y  
ALIAS=CAT1  
<SECTION=CATALOGIMS2>  
CATALOG=Y  
ALIAS=CAT2  
<SECTION=CATALOGIMS3>  
CATALOG=Y  
ALIAS=CAT3  
<SECTION=CATALOGIMS4>  
CATALOG=Y  
ALIAS=CAT4
```



Multiple IMSes, Shared ACBLIBs, each IMS has it's own Catalog



Application Programming with IMS Catalog



- IMS catalog PSBs resident for application programs
 - DFSCP000 - High-level assembler and COBOL applications
 - DFSCP002 - PL/I applications
 - DFSCP003 - PASCAL applications
- PCBs included to support different catalog processing models:
 - DFSCAT00 - Primary PCB to access all data in the catalog database
 - DFSCATSX – Used access the catalog database via the catalog 2nd index
 - DFSCATX0 – Used to access the catalog 2nd index as a database
- All catalog processing is performed with PROCOPT=G
- New “GUR” DL/I call to access Catalog DB records
 - Application programs can use the GUR call



NEW “GUR” DL/I Call



- Get Unique Record
 - Retrieves an XML formatted Catalog database “record”
 - Restricted to use with the IMS Catalog database
- Functions like a GU followed by a series of GNP calls
 - Less overhead than issuing GU & GNP to retrieve all the metadata for a catalog member
- Returns the entire database DBD or PSB record in one call
 - Using an AIB token, the call can be continued if the I/O area is too small for the entire catalog database record
- Data returned in XML format
 - The XML schemas are included in the IMS.ADFSSMPL data set:
 - DFS3XDBD.xsd (for DBD records)
 - DFS3XPSB.xsd (for PSB records)
- Support added for DFSDDLTO and IMS REXX



IMS Catalog Benefits



- Trusted, online source for IMS metadata
- Direct access to IMS metadata in the catalog
- Leveraged by the Universal Database Drivers
- No need for “local” Java metadata classes



Summary



- **IMS Catalog**

- Contains information about IMS database resources, program resources, and relevant application metadata
- Populate Utility and new ACBGEN
 - Used to populate and maintain the catalog metadata
- The IMS Catalog: The Next Step in Simplification and Enterprise Integration
<http://www-01.ibm.com/software/sw-events/teleconference/D108305C25358T22.html>

- **IMS Enterprise Suite Explorer for Development**

- Generates DBD/PSB source used to populate the catalog
- Enables IMS metadata discovery from the IMS Catalog
- IMS Enterprise Suite Explorer: The new face of IMS
<http://www-01.ibm.com/software/sw-events/teleconference/T929253D59548D24.html>





Two IMS White Papers!!



IBM Software
White Paper

What every enterprise architect needs to know about the evolution of IMS

Leverage IMS data and transactions as strategic elements of the enterprise architecture



IBM developerWorks

English Sign in (or register)

Technical topics Evaluation software Community Events Search developerWorks

developerWorks > Technical topics > Information Management > Technical library >

IBM Cognos Proven Practices: Best Practices for Using IMS Data in IBM Cognos BI Reports

Product(s): IMS, IBM Cognos BI Server; Area of Interest: Business Intelligence

[Richard Tran](#), Software Engineer, IBM
[Kevin Hite](#) (khite@us.ibm.com), IMS Solution Test, IBM
[Nigel Campbell](#), Senior Developer, IBM
[David Harrison](#), Information Developer, IBM

Summary: IBM Cognos BI v10.2 offers a data connection type for direct connections to IMS databases. This article describes best practices for configuring the IBM Cognos BI server, preparing the target IMS system for reporting applications and creating reports.

[View more content in this series](#)

Date: 03 Oct 2012
Level: Intermediate
PDF: A4 and Letter (261 KB | 13 pages) [Get Adobe Reader!](#)

Activity: 2000 views
Comments: 0 ([View](#) | [Add comment](#) | [Sign in](#))

★★★★★ Average rating (1 votes)
[+ Rate this article](#)

Download it today at ibm.co/PQ5Hdo

Download it today at ibm.com/ims





facebook.com/IMSFans



twitter.com/IBM_IMS



linkd.in/IBM_IMS



imsmadesimple.tumblr.com



IMSLISTSERV.BMC.COM

imslistserv.bmc.com



ibm.co/IMS_dWBlog



ibm.com/vrm/newsletter/11069



www.slideshare.net/ibmims



ibm.co/IMS_RUGs



bit.ly/IMS_YouTube



Connect with IBM System z on social media!

Subscribe to the new [IBM Mainframe Weekly](#) digital newsletter to get the latest updates on the IBM Mainframe!

LinkedIn

[System z Advocates](#) **
[IBM Mainframe- Unofficial Group](#)
[IBM System z Events](#)
[Mainframe Experts Network](#)
[SHARE](#)

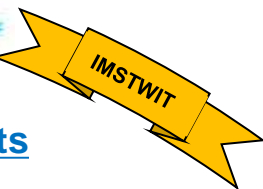


facebook

[IBM System z](#) **
[IBM Master the Mainframe Contest](#)
[IBM Destination z](#)
[SHARE Inc.](#)

twitter

[IBM System z](#) **
[IBM System z Events](#)
[Destination z](#)
[SHARE](#)



System z SMEs and Executives:

Deon Newman - [@deonnewm](#)
Steven Dickens - [@StevenDickens3](#)
Michael Desens - [@MikeDesens](#)
Patrick Toole - [@Pat Toole II](#)
Kelly Ryan - [@KellykmRyan](#)
Richard Gamblin - [@RichGx](#)

Blogs

[IBM Mainframe Insights](#) **
[Millennial Mainframer](#)
[#MainframeDebate blog](#)
[SHARE blog](#)
[IBM Destination z](#)



You Tube

[IBM System z](#) **
[Destination z](#)

tumblr.

[IBM Mainframe50](#)

Include the hashtag [#mainframe](#) in your social media activity and [#mainframe50](#) in 50th anniversary activity



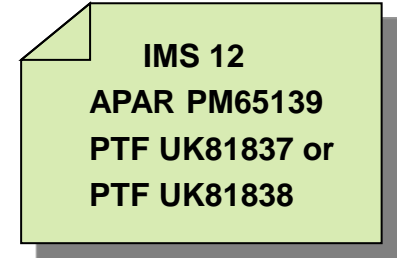
THANK YOU

BONUS SLIDES

SSA Command Code “O”: Qualify by Position



- New SSA command code “O”
 - Non-key field definitions not required in the DBD
 - Allows fields defined in database metadata to be used
 - Search by position and length vs. field name
 - 4 byte position, hex value
 - 4 byte length, hex value
 - Valid for HDAM, HIDAM, PHDAM, PHIDAM and DEBD databases
 - Valid for GU, GHU, GN, GNP, GHNP, ISRT calls
 - Plays well with other command codes
 - “GE” status code returned if field not found
 - Support for DFSDDLTO and IMS REXX
 - Performance will be the same as non-key field search



SSA with command code, position and length



- SSA with "O" command code, position and length

DBD

Field	Pos	Len
LASTNAME	1	10

Database Segments

0	1	2	3
<u>1234567890</u>	<u>1234567890</u>	<u>1235678901</u>	<u>234567</u>
LAST1	FIRST1	555-1111	1999999
LAST2	FIRST2	555-2222	2999999
LAST3	FIRST3	555-3333	3999999

COBOL Copybook

Field	Pos	Len
LASTNAME	1	10
FIRSTNME	11	10
EXTENSIN	21	10
ZIPCODE	31	7

'bb' Status Code: segments returned successfully

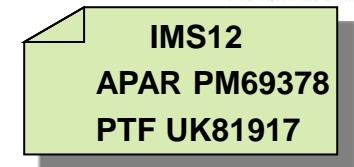
```
GU TELEBOOK*--(LASTNAME=LAST2      )
GU TELEBOOK*O-(000000010000000A=LAST3      )
GU TELEBOOK*O-(0000000B0000000A=FIRST1      )
```

}
}
 Position Length

(Position is relative to 1 vs. zero)



SSA Command Code “O”: Qualify by Position



- Support for IMS Universal Drivers
 - Allows users to issue queries qualified on non-searchable fields and have a field’s position and length generated automatically in the qualified SSA
 - SQL
 - Universal Drivers will detect a non-searchable field in the where clause based on database metadata and will internally convert the SSAList qualification
 - DLI
 - Universal Drivers will detect a non-searchable field in the SSAList based on database metadata and will internally convert the SSAList qualification



IMS Universal Driver Additional Enhancements



- Variable length segment support
 - VL segments contain a two byte length (LL) field that identify size of the segment instance
 - Universal Drivers are now sensitive to the LL field of a VL segment and will manage the IO area of the segment instance on all CRUD calls

FIELD=PERSONAL_INFO (VLOB min length=82 max length=112)			
INNER FIELD=LENGTH (2 bytes)	INNER FIELD=NAME (30 bytes)	INNER FIELD=ADDRESS (50 bytes)	INNER FIELD=EMAIL (optional field 30 bytes)
112	RICHARD	555 Bailey Ave	tran@abc123.com
82	KEVIN	555 Bailey Ave	<does not exist physically on disk>



Mapping Support

- A Map is metadata that describes how a field, or set of fields, are mapped for a particular segment instance
- Various mapping cases are captured in the metadata
 - defines the set of fields to be used for each case
- Maps can be defined in the IMS catalog
- Example:
 - Insurance segment mapped multiple ways depending on value of a 'Policy Type' field

Policy Type	Property Type	Rooms	Value	Address	Make	Model	Year	Value	Color
M	-	-	-	-	Ford	Escort	1989	2K	Red
H	Single Family	5	500K	555 Disk Drive Way, 95141	-	-	-	-	-