

Everything you ever wanted to know about the IMS Catalog but were afraid to ask

Session no: B05 & C01

Deepak Kohli

IMS Product Management Team



IMS Technical Symposium 2015

-
- Deepak Kohli
 - Six Pack
 - Pac-Man
 - James ←=====

© Copyright IBM Corporation 2015. All rights reserved.

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

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. 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. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. 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 ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

IBM, the IBM logo, ibm.com, and IMS 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 ibm.com/legal/copytrade.shtml

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

- Demo
- IMS Catalog Overview
- IMS Catalog Configuration
- IMS Catalog Enablement
- Living with the IMS Catalog ←=====

- Teleconference on IMS Catalog March 31st
 - <http://www-01.ibm.com/software/os/systemz/telecon/mar31/>
- IMS 12: The IMS Catalog, REDP-4812-00 Redpapers, published 20 September 2012
- IMS Version 13 Technical Overview, SG24-8224-00 Redbooks, published 15 September 2014,
- YouTube Video on Implementing the IMS Catalog:
 - <https://www.youtube.com/watch?v=lcrOhmGzUWY>
- More sessions here at the IMS Technical Symposium:
 - Exploiting the IMS Catalog using IMS Explorer for Development (Session D01)
 - Scaling your applications with the IMS Catalog (Session B06 & C15)



Demo Time !!!





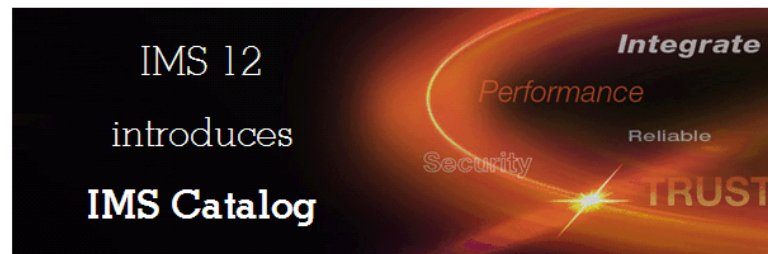
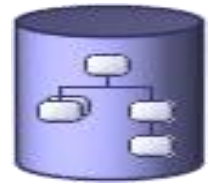
IMS Catalog Overview



What is the IMS Catalog?



- **Optional** system database from IMS 12 onwards
 - An IMS High Availability Large Database (HALDB)
- A trusted, online source for IMS Database and Application metadata information – fully managed by IMS
- Accessible via both standard JDBC/SQL, (and COBOL/SQL in IMS 13) and traditional DL/1 access
 - Applications (and tooling) can access the online IMS Catalog
- Leveraged by Universal JDBC Drivers for metadata exchange and discovery
 - Enabling better scalability for large scale Open Database Access deployment
- Significant component of the IMS simplification and integration strategy
- Foundation for features such as database versioning in IMS 13 and future DB enhancements



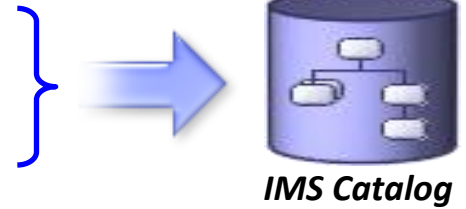
- IBM Products which can use the Catalog:
 - IMS Explorer for Development
 - Cognos 10.2 is certified to use the Catalog
 - InfoSphere Data Architect
 - IBM Data Studio
 - Rational Asset Analyzer
 - IBM Worklight
 - InfoSphere DataStage
- Enables scalable and flexible IMS Open Database solutions
 - Applications no longer need to maintain **local** Java metadata in order to interact with IMS databases
 - The metadata is used by the IMS Universal JDBC driver to allocate program specification blocks (PSBs), issue DL/I calls, perform data transformation, and translate SQL queries to DL/I calls
 - Applications can reference the online IMS Catalog
- Powerful tool for metadata discovery
 - Which PSBs have PCBs with labels?
 - Which databases have exits?
 - Which PSBs have update access to a specific database?
 - If I change a DBD, which PSBs are affected?
- IMS 13 features which **require** the Catalog:
 - Database Versioning
 - IMS Native SQL support for COBOL
 - .NET access to IMS data

B ■ Business metadata

- Business rules, definitions, terminology, glossaries, algorithms and lineage using business language
- **Audience:** Business users

T ■ Technical metadata

- Defines source and target systems
 - Table & Column / Segment & Field structures and attributes
 - Derivations and dependencies
- **Audience:** Specific tool users, AD, BI, ETL, profiling, modeling



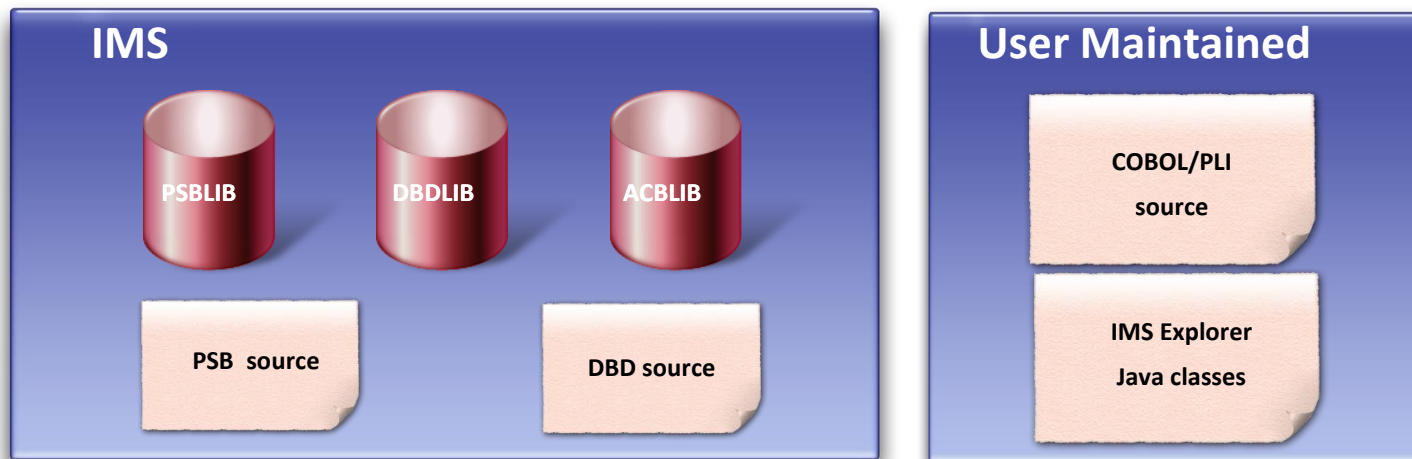
O ■ Operational metadata

- Information about application runtime
 - Frequency, record counts, component by component analysis and other statistics
- **Audience:** Operations, management and business users



Literally, “*data about data*” that *describes* your company’s information from *multiple perspectives*

- Databases partially defined in the IMS DBD
 - Only key/searchable fields needed by applications
 - Remaining segment data is not defined
- Remaining database 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

- Database and program resources defined to an IMS system and relevant application information stored as metadata in an IMS catalog
 - Databases, fields, segments, data types, and more ...
- The IMS catalog can be updated when you create, alter or delete IMS resource or application information
- Updates to the IMS catalog are done only via integrated IMS processes
 - Catalog Populate Utility (initial catalog load and member update)
- Catalog metadata can be used to:
 - Enhance understanding of the data
 - Improve consistency of the data
 - Improve impact analysis of the data
 - Improve development productivity
 - Improve data governance
- Key component of the IMS growth strategy:
 - Simplification
 - Integration


- Contains metadata related to an IMS system's databases and programs
 - DBDs and PSBs and Application info
- IMS PHIDAM/OSAM HALDB database
 - Defined with 4 DSGs (Data Set Groups)
- Has one Secondary Index
 - Can be used to determine which IMS programs (PSBs) reference a specific user database without processing the entire IMS catalog
- Unique features
 - DBRC use is optional for the IMS Catalog HALDB database
 - ONLY HALDB that isn't required to be defined in the DBRC RECONS
 - IMS can manage allocation/creation of catalog database data sets
 - Uses parameters in the "CATALOG" section of DFSDFxxx PROCLIB member

- IMS provides DBD and PSB source code for the Catalog database
- IMS provides object code for the Catalog DBDs and PSBs
- PHIDAM DBD reserved name is **DFSCD000**
- PSINDEX DBD reserved name is **DFSCX000**
 - Used to connect DBDs to PSBs that reference them
- PSBs provided to load, read and update the Catalog database
 - DFSCPL00 is used for initial load process
 - Used by the Catalog Populate Utility
 - DFSCP000 (COBOL/HLASM), DFSCP002 (PL/I), and DFSCP003 (PASCAL), are used for read access
 - DFSCP001 is used for **update** access
 - Used by ACBGEN and Catalog Populate Utility, and Purge Utility
- IMS automatically adds a PSB for the IMS catalog to each user PSB at run time (when the catalog is enabled)
- ACBGEN required for all catalog PSBs before enabling, populating and using the Catalog

A cartoon character of a man in a suit, pointing towards the text box.

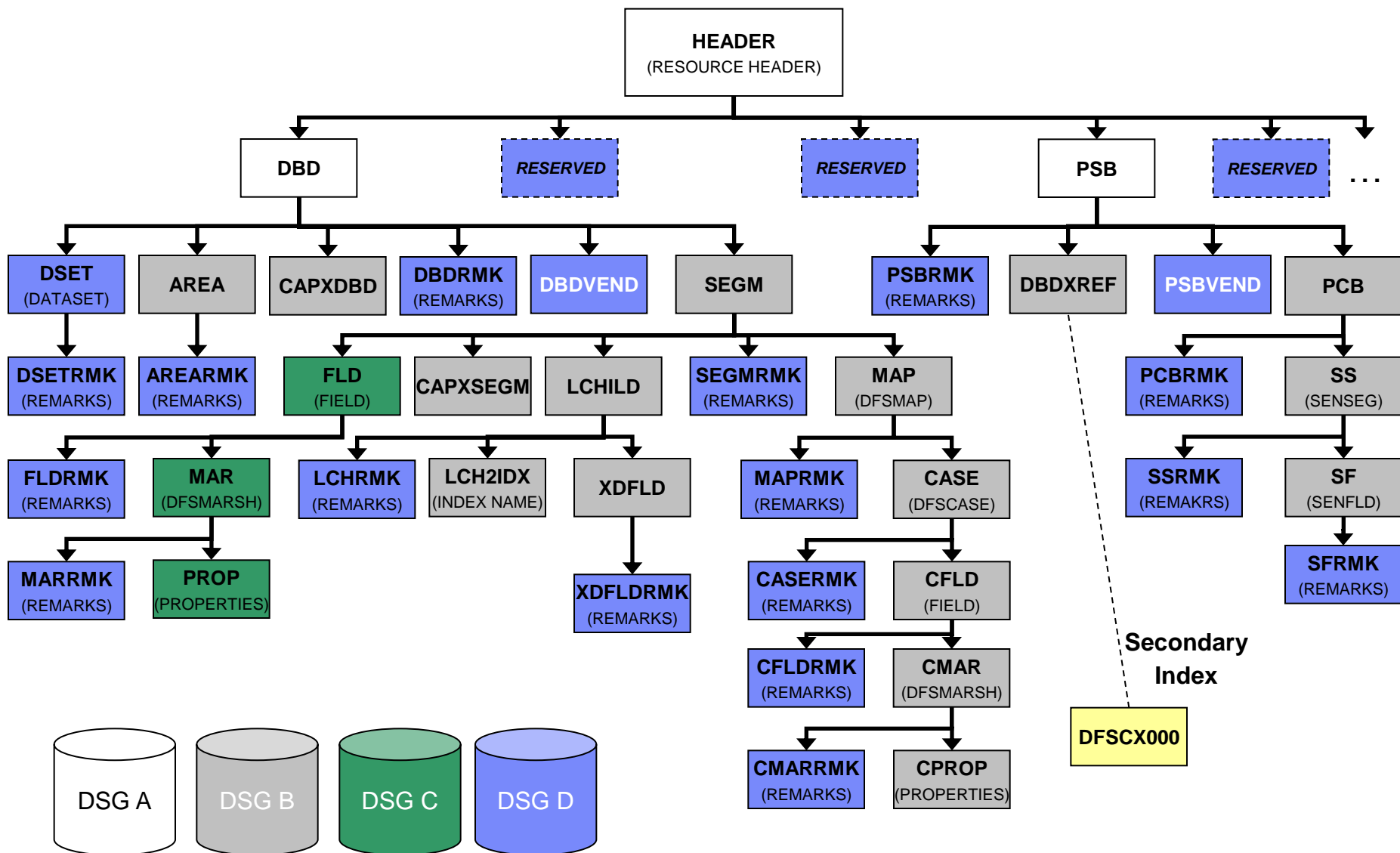
The catalog PSB itself occupies 12kb in the DLIPSB pool and 500 bytes CSAPSB pool for each user PSB that is using the catalog PSBs. You might need to increase the size of your storage pools, up to the maximum size of the catalog PSB in each pool multiplied by the number of user PSBs that concurrently access the catalog.

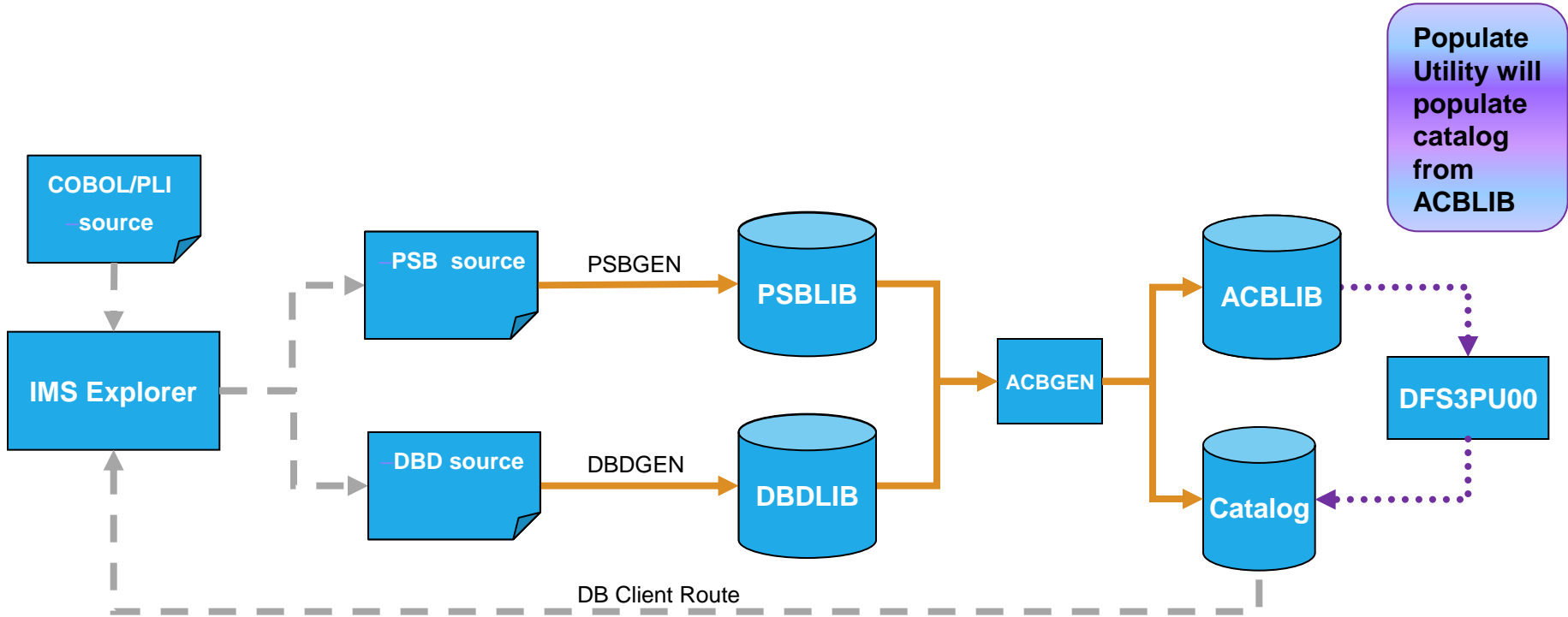
- Root segment of the Catalog database is a generic resource header
 - Indicates the type of resource → DBD or PSB
 - A dependent of the Root and its Children are a complete DBD or PSB
 - Multiple iterations/instances of a specific resource are supported
 - Most resources are differentiated by their ACBGEN timestamp
 - Logical DBDs and GSAM are differentiated by their DBDGEN timestamp
- Catalog database segments typically correspond to macro statements in the DBD and PSB source
- One segment at the first Child level under both the DBD and PSB segments is available for vendor/customer use
 - DBDVEND
 - PSBVEND

A cartoon character of a man in a white lab coat, standing with his hands on his hips, pointing towards the text box.

An “instance” of the IMS catalog metadata is **generated each time an ACBGEN is done**. The DFSDFxxx member of the IMS PROCLIB data set is used to configure how many instances are stored and when IMS removes old instances.

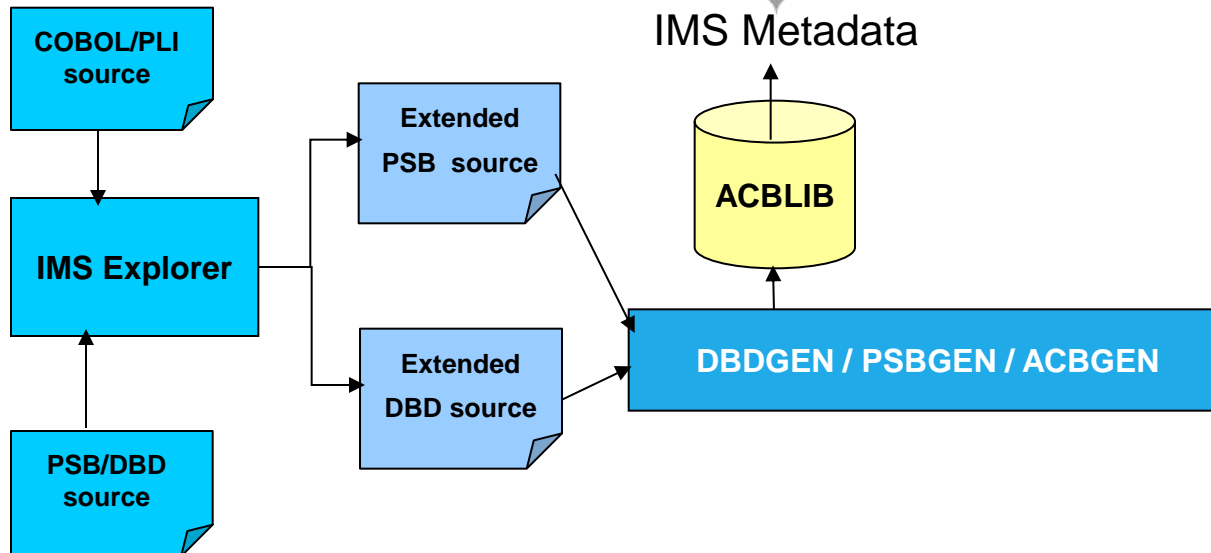
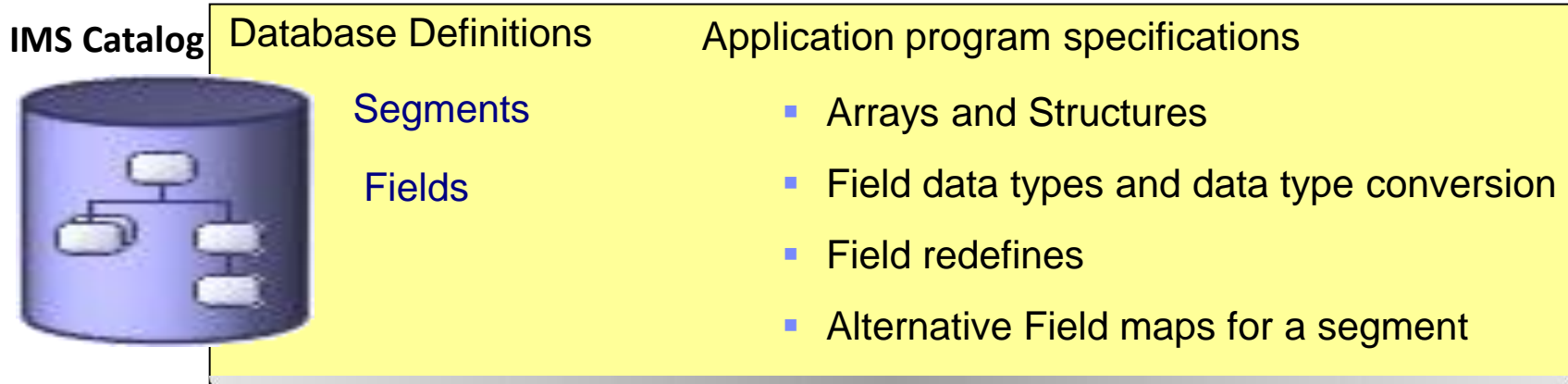
Physical Catalog Structure





- 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
 - 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

Contains IMS program and database resources, and relevant application metadata





■ PCB

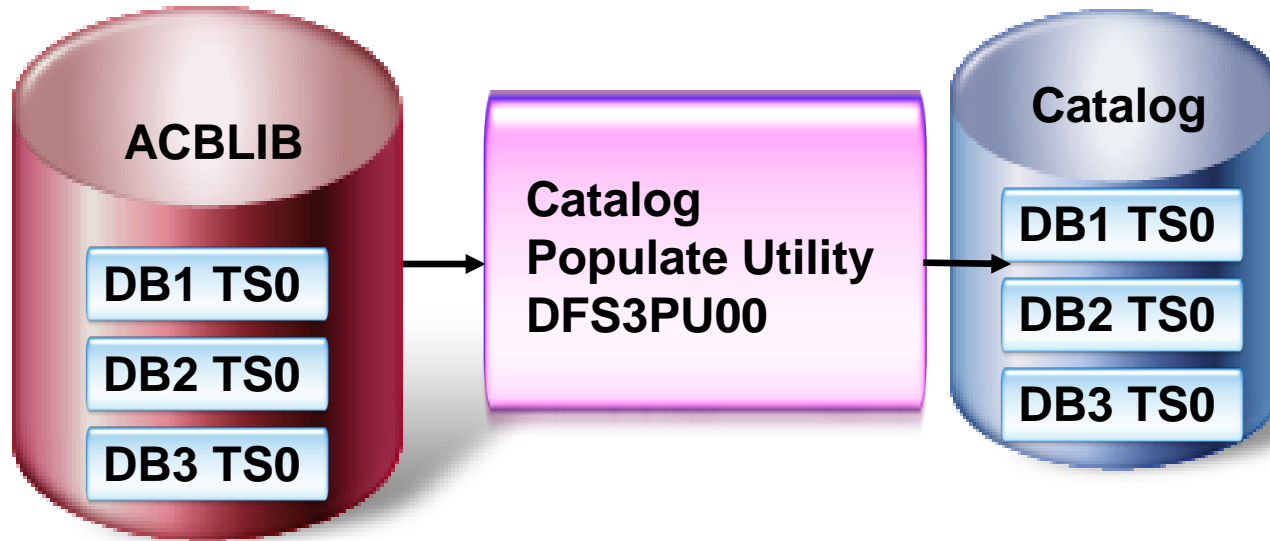
– EXTERNALNAME=

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

– REMARKS=

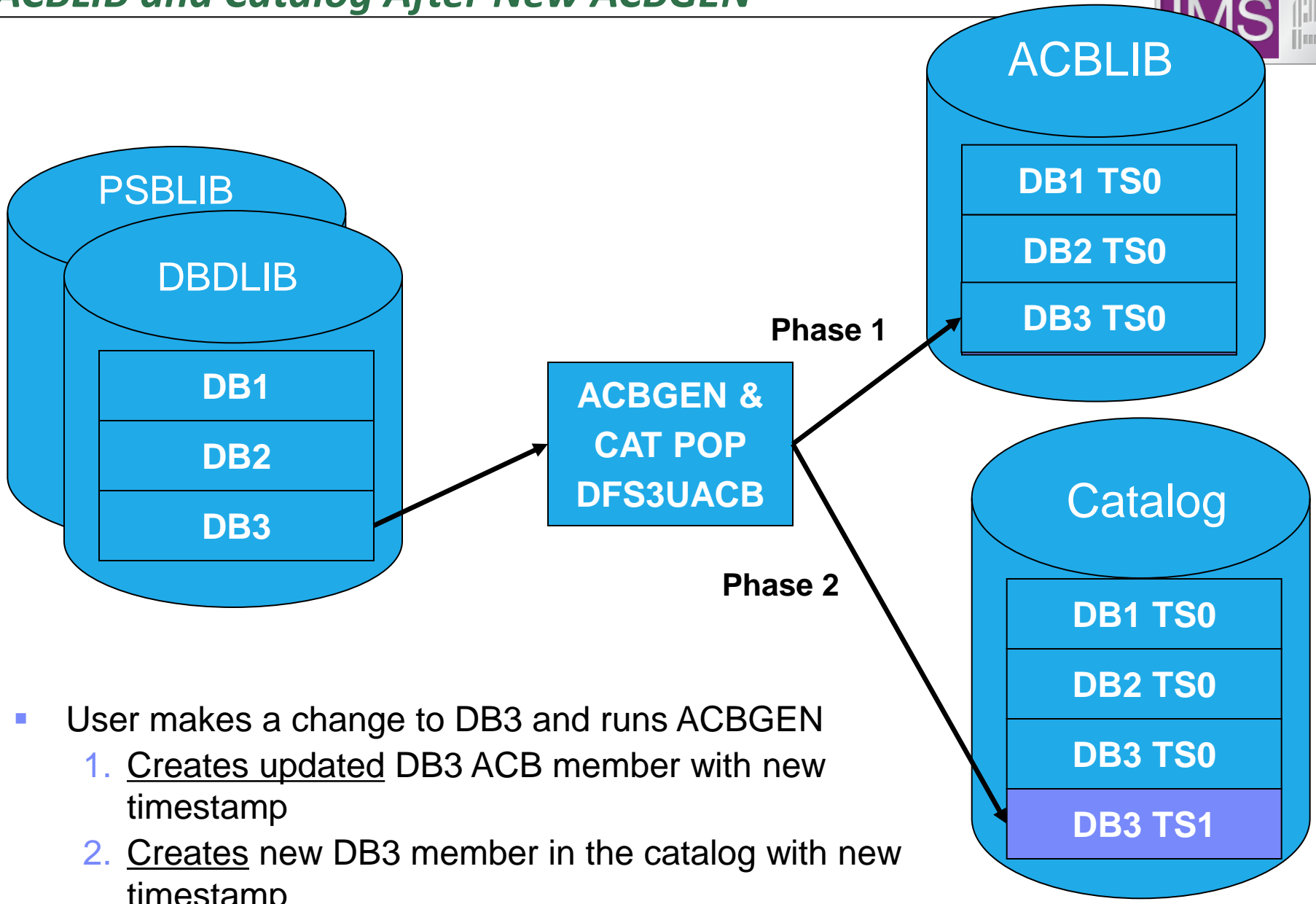
- Optional user comments. A 1- to 256-character field. Added to PSBGEN statements

- **DBD**
 - ENCODING=
 - Specifies the default encoding of all character data in the database defined by this DBD.
 - This value can be overridden in individual segments or fields.
- **SEGM**
 - EXTERNALNAME=
 - An optional alias for the NAME= parameter used by Java application programs to refer to the segment.
- **FIELD**
 - CASENAME=
 - The name of the map case when alternative mappings are defined for the fields in a segment
 - DATATYPE=
 - Specifies the external data type of the field.
 - EXTERNALNAME=
 - An optional alias for the NAME= parameter used by Java application programs to refer to the 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. A 1- to 256-character field. Added to DBDGEN statements



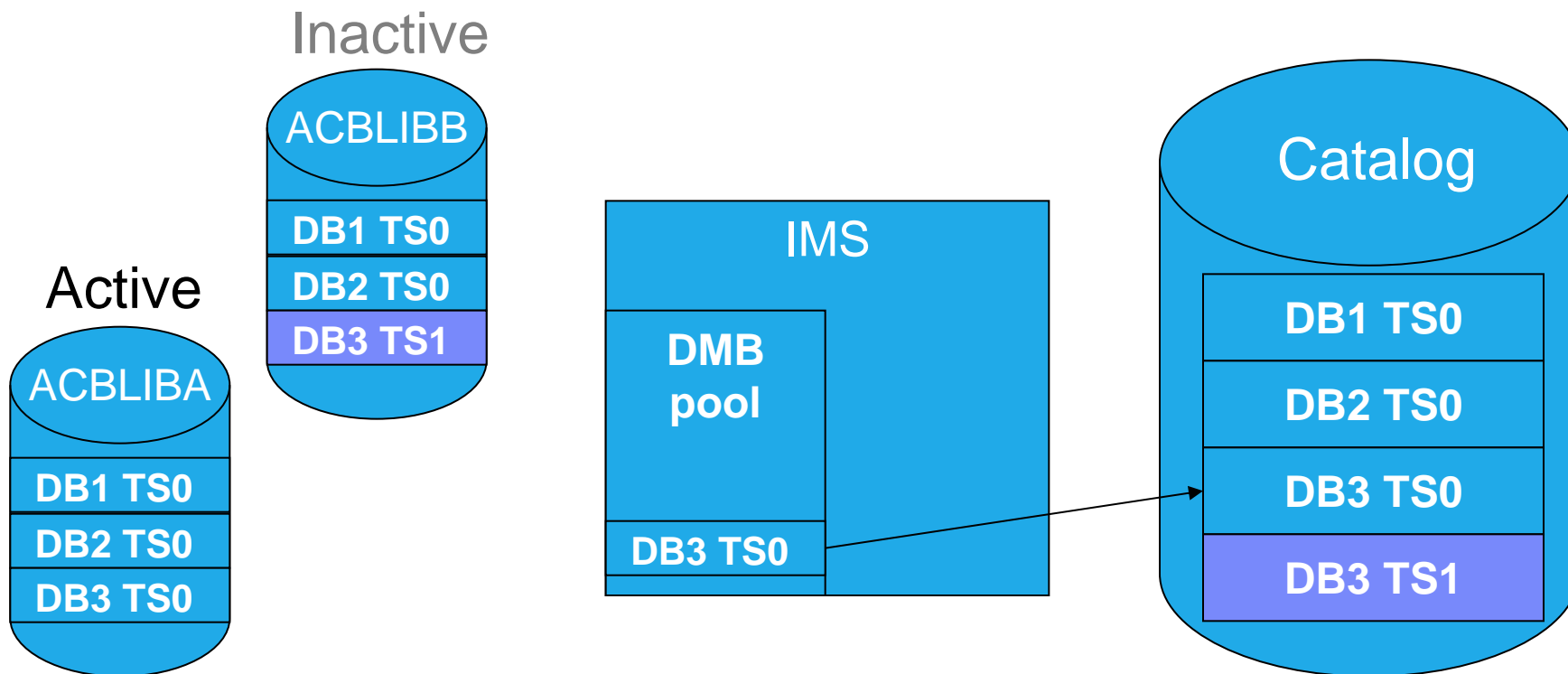
- There are 3 members in an ACBLIB
- Run the Catalog Populate Utility, DFS3PU00
- The catalog members will have a timestamp TS0
- This is the ACB member timestamp

ACBLIB and Catalog After New ACBGEN

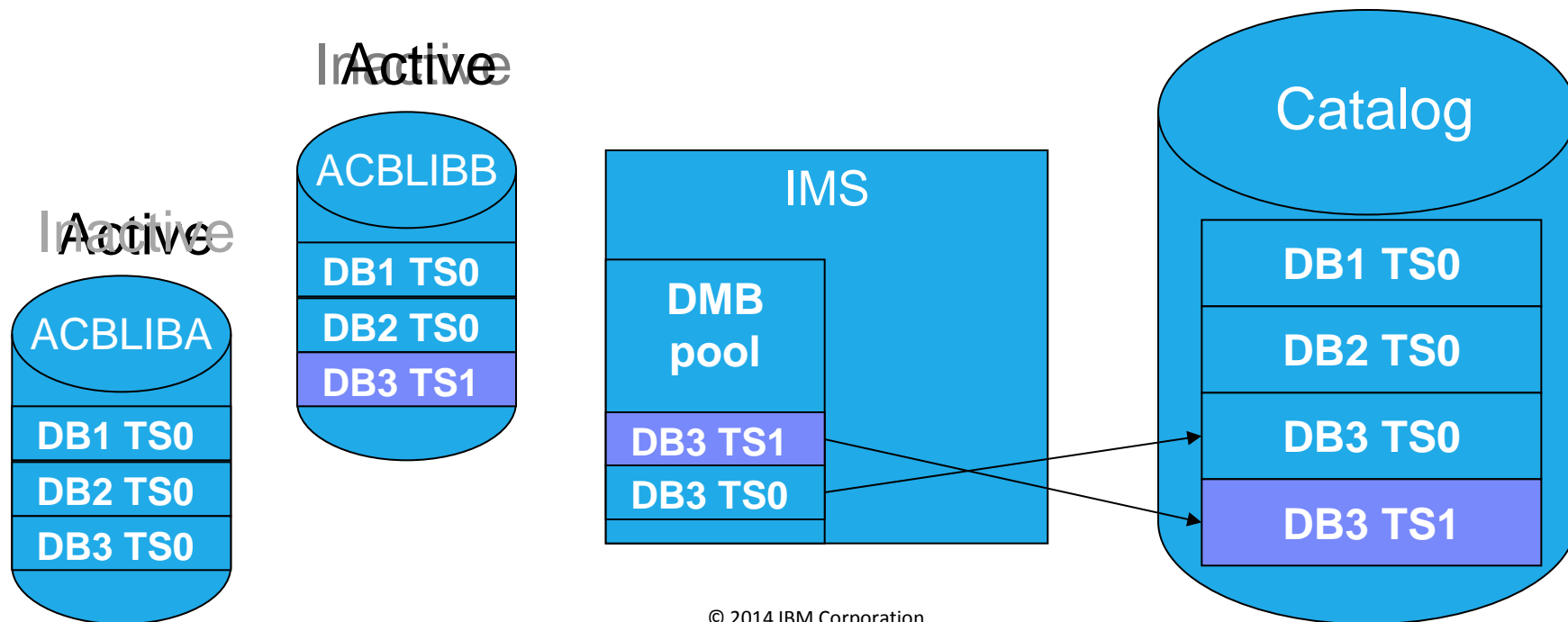


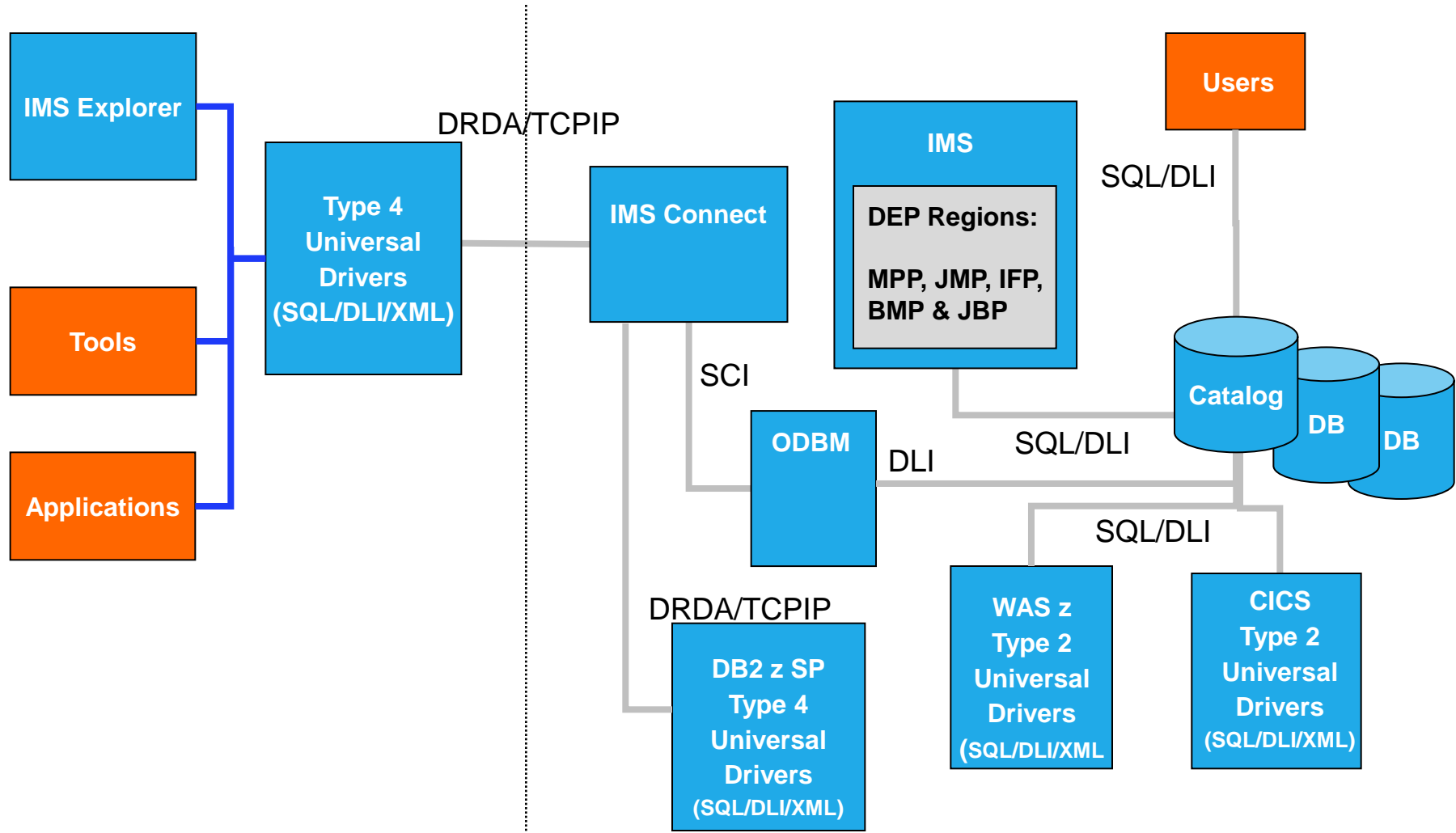
- User makes a change to DB3 and runs ACBGEN
 1. Creates updated DB3 ACB member with new timestamp
 2. Creates new DB3 member in the catalog with new timestamp

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



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







IMS Catalog Configuration

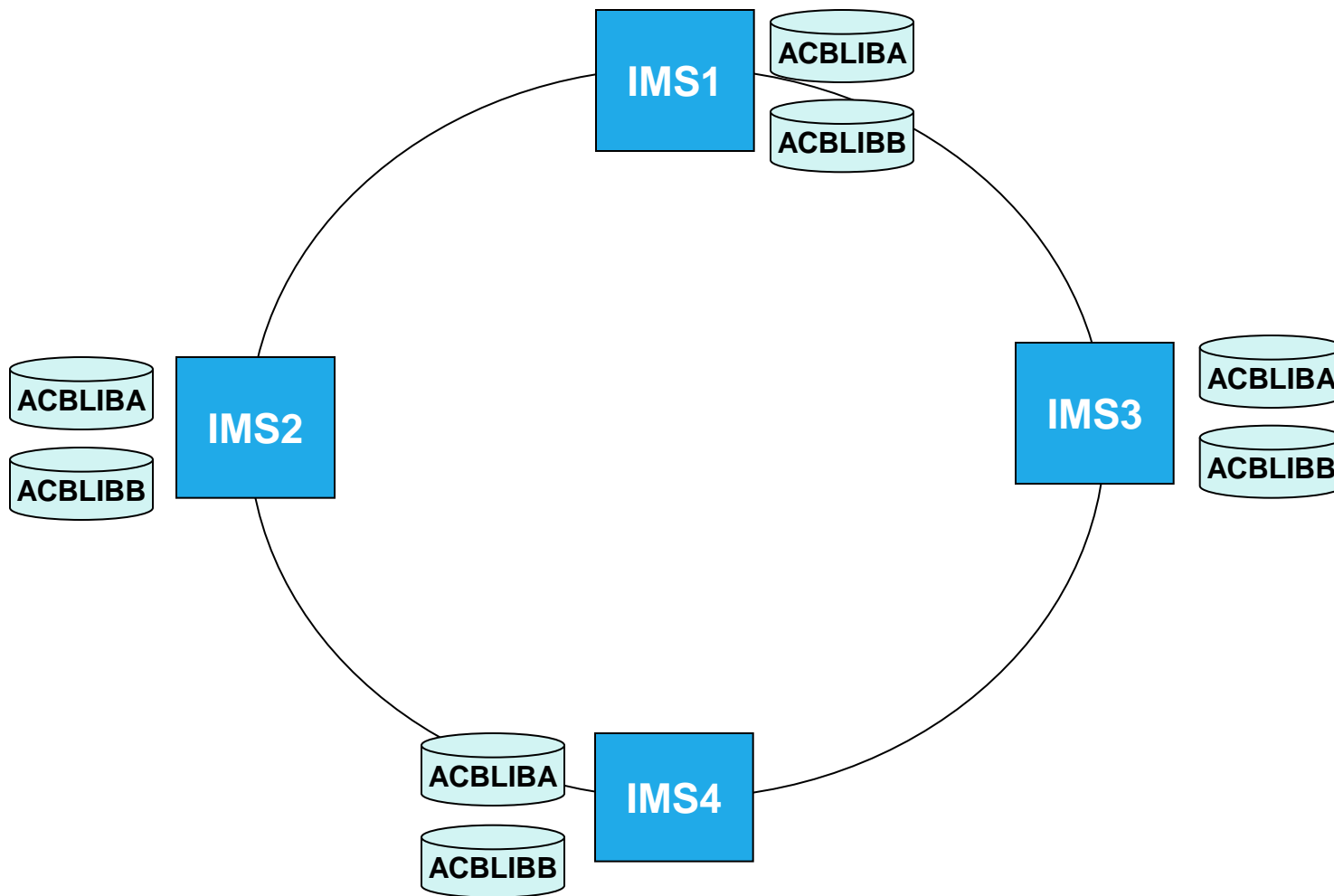




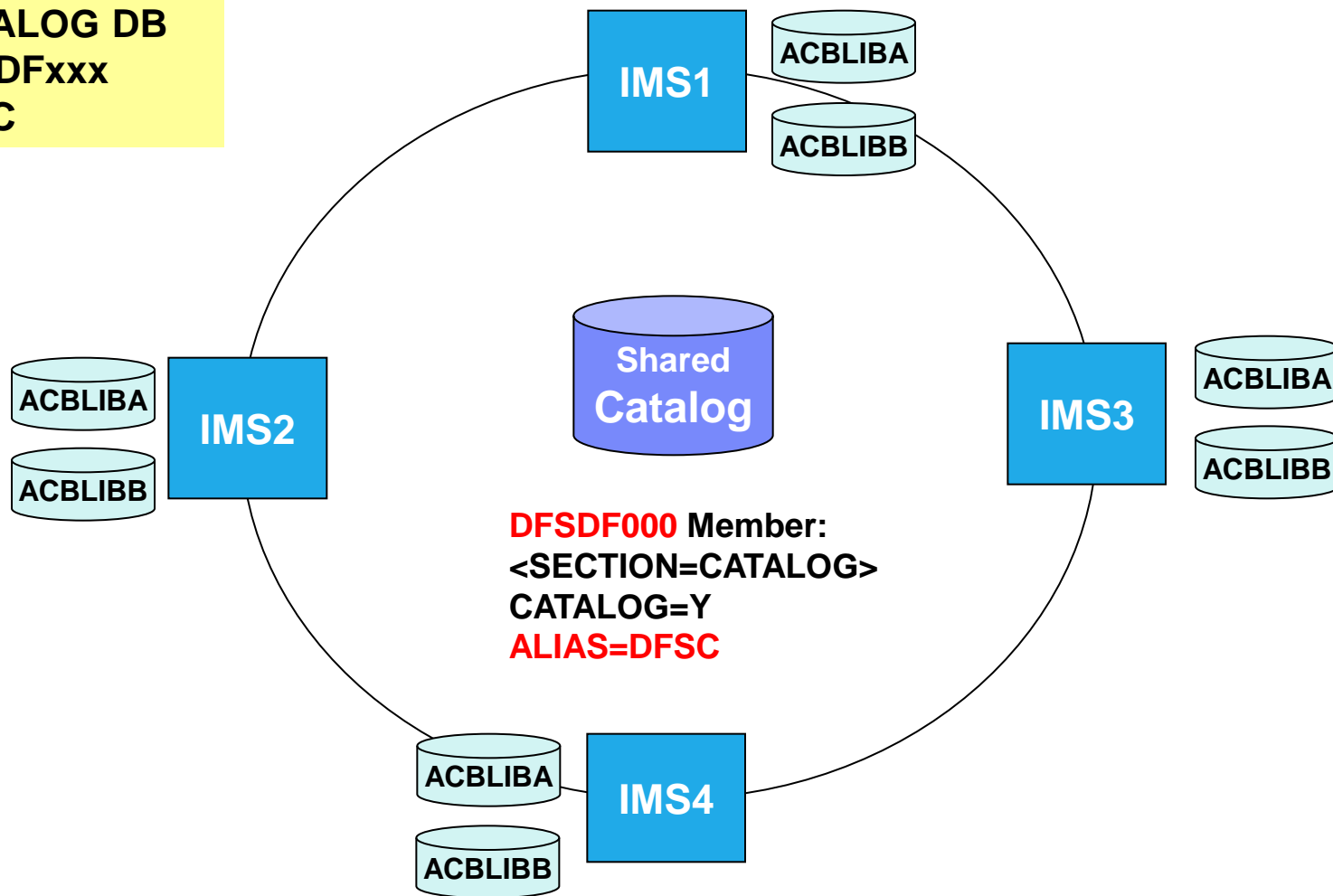
- The IMS catalog supports many environments:
 - one-catalog-per-system,
 - one-catalog-per-sysplex, or
 - any combination of shared and independent catalogs.
- Each IMS system can be linked to only one instance of the IMS catalog, regardless of system configuration.

Without an IMS Catalog

Multiple IMSes, each IMS has it's own cloned ACBLIBs

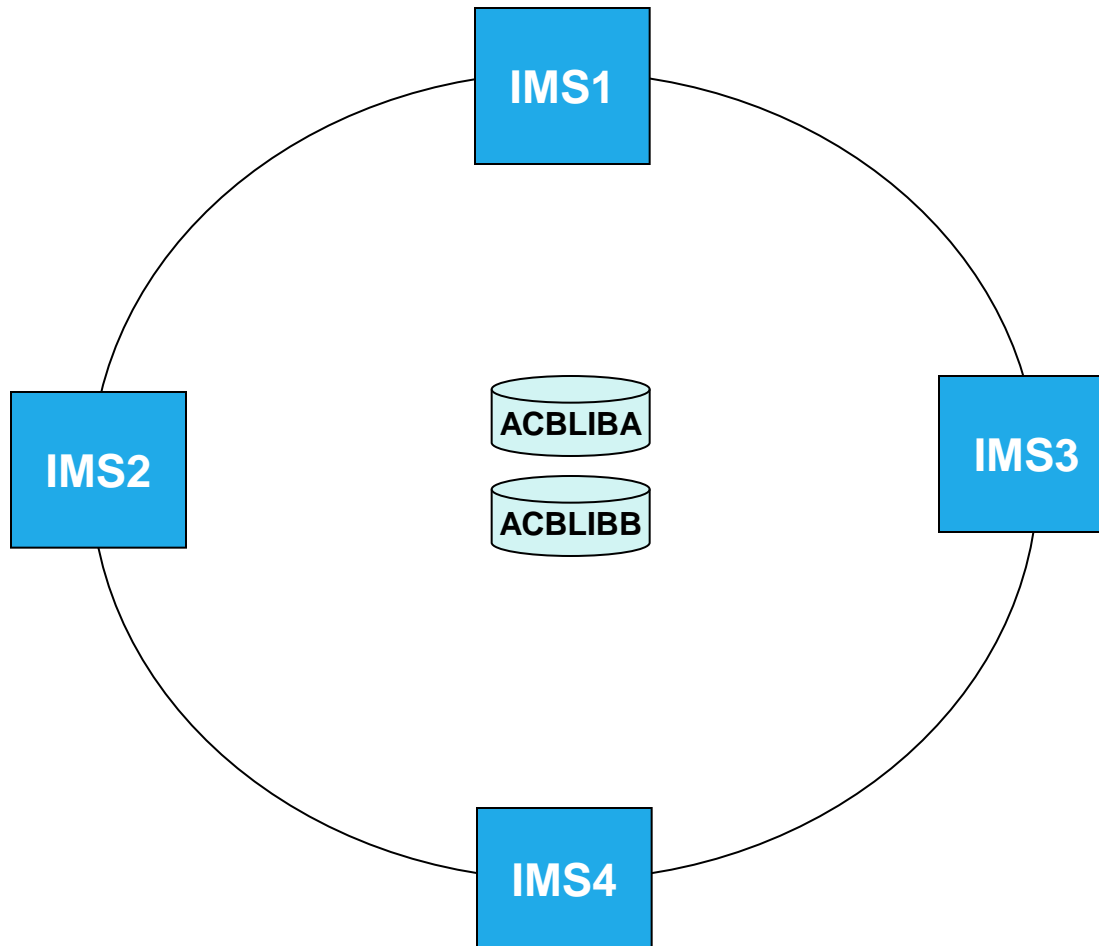


Shared CATALOG DB
Shared DFSDFxxx
ALIAS=DFSC

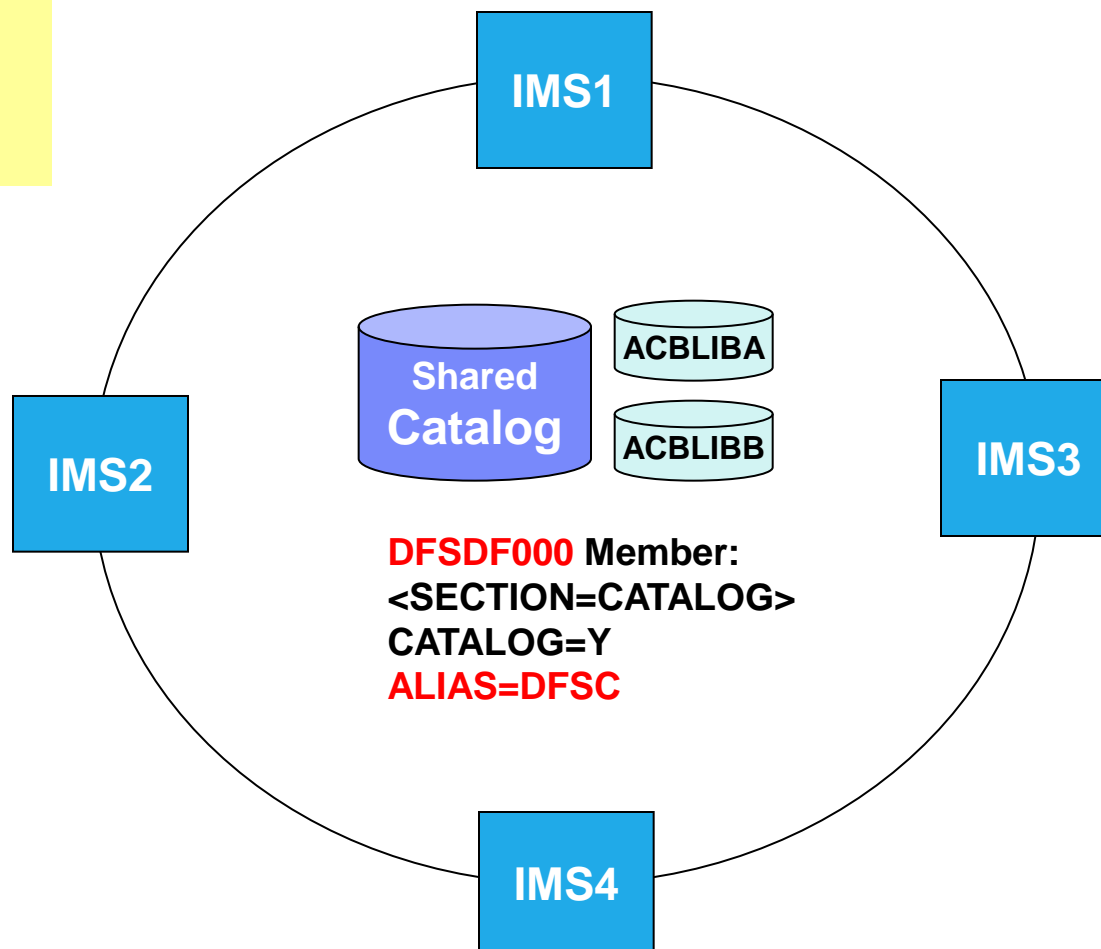


Without an IMS Catalog

Multiple IMSes, shared ACBLIBs



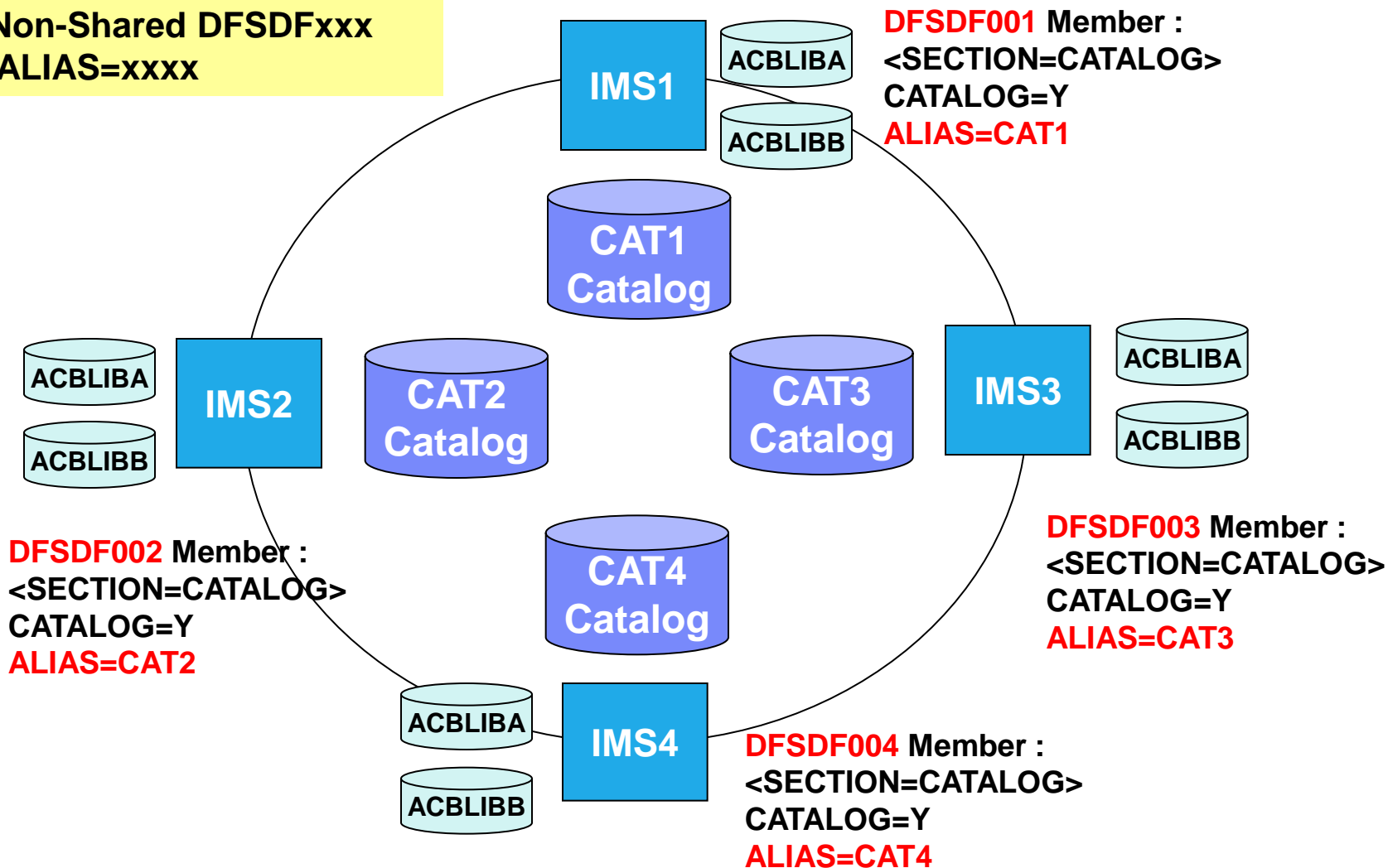
Shared CATALOG DB
Shared DFSDFxxx
ALIAS=DFSC



Multiple IMSes, Cloned ACBLIBs, each IMS has its own Catalog



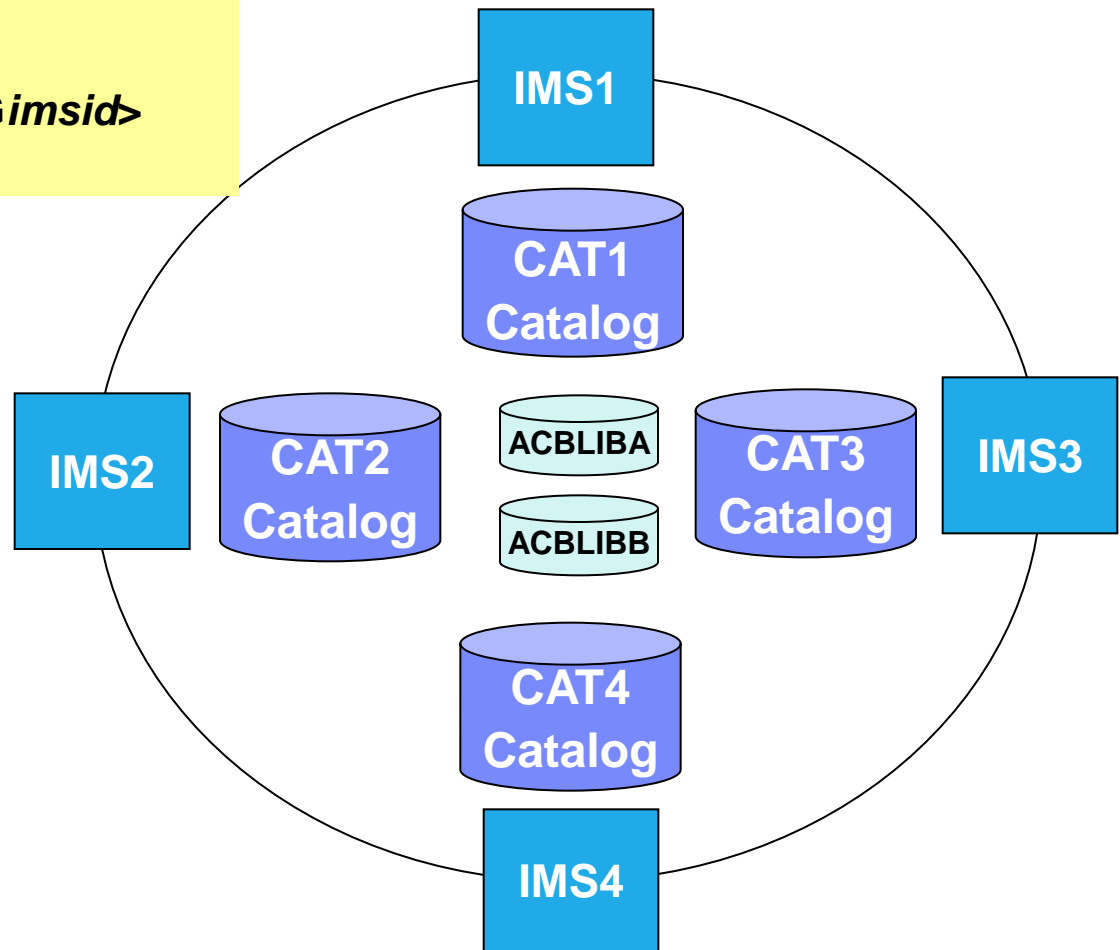
Non-Shared CATALOG DB
Non-Shared DFSDFxxx
ALIAS=xxxx



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
```

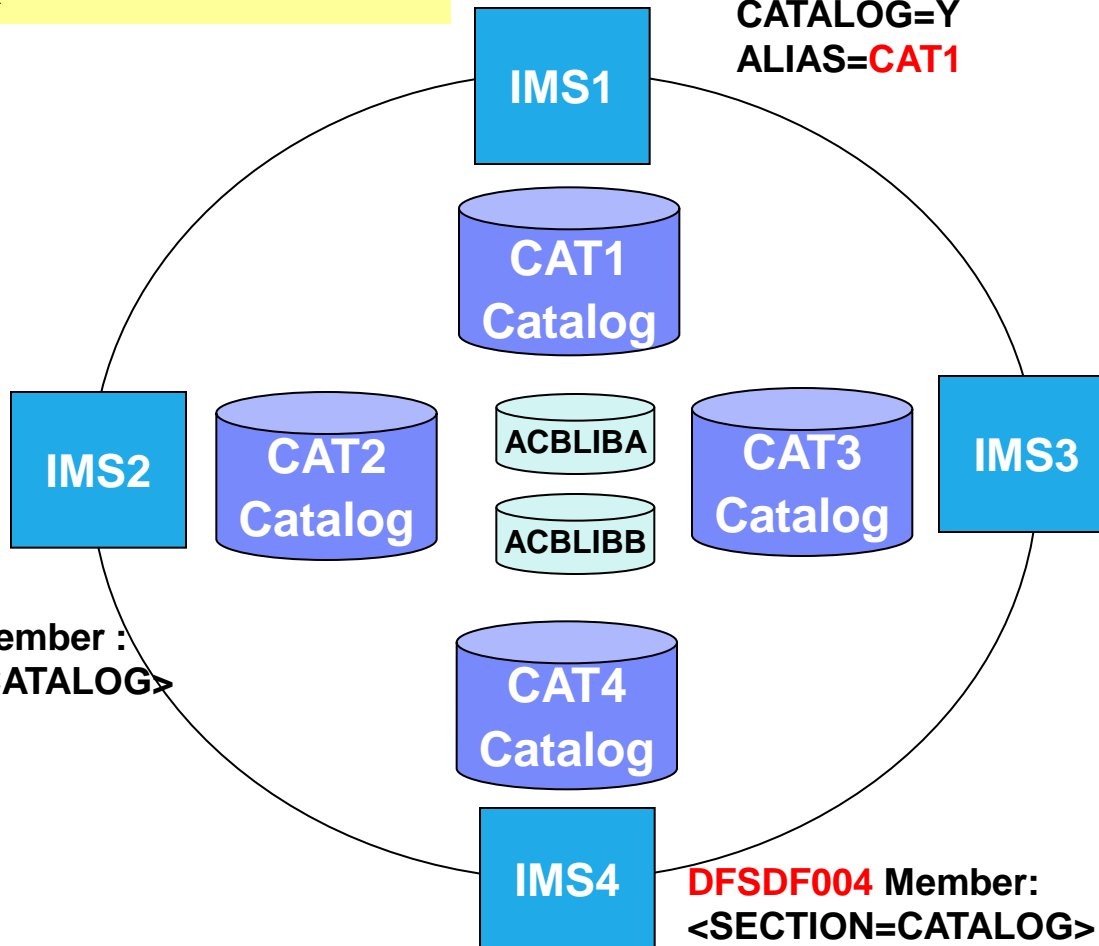


Multiple IMSes, shared ACBLIBs, each IMS has it's own Catalog



Non-Shared CATALOG DBs
Non-Shared DFSDFxxx
ALIAS=xxxx

DFSDF001 Member :
<SECTION=CATALOG>
CATALOG=Y
ALIAS=**CAT1**



DFSDF003 Member :
<SECTION=CATALOG>
CATALOG=Y
ALIAS=**CAT3**

DFSDF002 Member :
<SECTION=CATALOG>
CATALOG=Y
ALIAS=**CAT2**

DFSDF004 Member:
<SECTION=CATALOG>
CATALOG=Y
ALIAS=**CAT4**



IMS Catalog Enablement



- 1 ■ Copy the IMS supplied DBDs and PSBs for the IMS catalog from the IMS.SDFSRESL data set to your IMS.DBDLIB and IMS.PSBLIB data sets.
- 2 ■ Run the ACB Maintenance utility to generate the ACBs for the IMS catalog.
- 3 ■ Activate the ACB library that contains the IMS catalog ACB.
- 4 ■ Define the HALDB master and partitions of the IMS catalog.
- 5 ■ Code the CATALOG section of the DFSDFxXX member in the IMS.PROCLIB data set.
- 6 ■ If you need to manually allocate the database data sets for the IMS catalog, allocate the data sets now. Otherwise, IMS creates them automatically.
- 7 ■ Load the IMS catalog by running the DFS3PU00 utility.
- 8 ■ Image copy the Catalog
- 9 ■ Restart IMS



- 1 Add IMS supplied catalog DBDs and PSBs into your DBDLIB and PSBLIB
 - Copy DBD and PSB object code from SDFSRESL to your DBDLIB and PSBLIB

```
//CPYCMEM EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//SDFSRESL DD DSN=IMSCFG.IMB1.SDFSRESL,DISP=SHR
//DBDLIB DD DSN=IMSCFG.IMSB.DBDLIB,DISP=OLD
//PSBLIB DD DSN=IMSCFG.IMSB.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)
```




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



2 ■ Run an ACB Gen for the Catalog PSBs

```
//PROCS JCLLIB ORDER=(IMSCFG.IMSB.PROCLIB)
//ACBGEN EXEC PROC=ACBGEN,SOUT='*',COMP='POSTCOMP'
//G.SYSIN DD *
    BUILD PSB=(DFSCPL00)
    BUILD PSB=(DFSCP000)
    BUILD PSB=(DFSCP001)
    BUILD PSB=(DFSCP002)
    BUILD PSB=(DFSCP003)
//*
```



ACBGEN can be run into Staging Library

3 ■ Activate the ACB library that contains the IMS Catalog ACBs

- Run the Online Change Copy Utility to “Backup” Active ACBLIB in Inactive ACBLIB
- Initiate an ACB Member Online Change

```
INIT OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(DFSCD000,DFSCX000)
INIT OLC PHASE(COMMIT)
```

- Run the Online Change Copy Utility to Copy Active ACBLIB in Inactive ACBLIB

4 ■ 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 is 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 set** using the Catalog Partition Definition Data Set utility, **DFS3UCDO**
 - If an ALIAS is used in the CATALOG sections of the DFSDfxxx member, each alias Catalog database must be defined

- DBRC DSPURX00 utility and commands to define the Catalog DB to the RECONs

```
//D.SYSIN DD *
  INIT.DB      DBD(DFSCD000) TYPHALDB SHARELVL(3)
  INIT.PART    DBD(DFSCD000) PART(DFSCD01) -
              DSNPREFIX(IMSCFG.IMSC.DFSCD000) -
              BLOCKSIZE(4096,4096,4096,4096) -
              KEYSTRNG(X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF') -
              ICJCL(ICJCL) OICJCL(OICJCL) RECOVJCL(RECOVJCL) -
              NOREUSE RECOVPD(0) GENMAX(3) RECVJCL(RECVJCL)
  INIT.DB      DBD(DFSCX000) TYPHALDB SHARELVL(3)
  INIT.PART    DBD(DFSCX000) PART(DFSCX01) -
              DSNPREFIX(IMSCFG.IMSC.DFSCX000) -
              KEYSTRNG(X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF') -
              ICJCL(ICJCL) OICJCL(OICJCL) RECOVJCL(RECOVJCL) -
              NOREUSE RECOVPD(0) GENMAX(3) RECVJCL(RECVJCL)
```

- 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

- The following jobs and task are added to the IVP in support of the IMS Catalog:
 - IV_E319J – This job allocates and loads the IMS Catalog database
 - IV_O254T – This task provides an introduction to the IMS Catalog
 - IV_O260J – This job retrieves data from the IMS Catalog database using DFSDDLTO
 - Java Samples (Java suboption must be selected)
 - IV_O261J – SQL, Type 2
 - IV_O262J – DLI, Type 2
 - IV_O275J – SQL, Type 4
 - IV_O276J – DLI, Type 4




The Catalog is not registered to DBRC with the IVP samples



- 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) /*
```



DFSHDBSC is the DDNAME for the Catalog Partition definition data set. IVP provides sample jobs and tasks that demonstrate how to set-up a simple IMS Catalog.

**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

- 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

**Unregistered
Catalog database**

- 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
/*
```

**Unregistered
Catalog database**

- 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

5 ■ 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, aliased, 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**

The Catalog Definition user exit routine (DFS3CDX0) is provided as an alternative option for batch processing environments that do not use the DFSDFxxx member of the PROCLIB data set.



■ 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 - *the secondary index will be created with a CI SIZE of 18432 - so buffers of 20480 are required in the DFSVSMxx proclib member*
 - 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 data sets

■ DFSDFxxx PROCLIB Member

– CATALOG section parameters (continued)

- RETENTION=(INSTANCES=nnnnn,DAYS=nnnnn)
 - Specifies retention schedule for metadata in the IMS catalog
 - By default IMS keeps only two instances of the DBD or PSB in catalog
 - Metadata instances older than the specified retention period are not automatically deleted, but available for removal when a new instance of metadata is added
 - Default value of “0” disables this feature
 - You can specify the INSTANCES parameter, the DAYS parameter, or both.
- GURCACHE= 1
 - Specifies the amount of storage, in gigabytes, to allocate in 64-bit memory to cache XML documents generated as responses to GUR calls. The valid values are 1 through 999.

You need to run the Catalog Purge Utility to delete DBDs and PSBs from the Catalog – it doesn't happen automatically.



- 6 ■ If you want to manually allocate the database data sets for the IMS catalog, allocate the data sets now. Otherwise, IMS creates them automatically.

7 ■ Populate the IMS Catalog

- Load the IMS Catalog using IMS Catalog Populate utility, DFS3PU00
 - Each ACB member is decoded, converted to catalog format, loaded into the catalog
- Reads ACBLIB, DBDLIB and PSBLIB datasets as input
 - Data sets can be concatenated but only first occurrence of an ACB member is used
 - DBDLIB needed for Logical databases and GSAM databases
 - PSBLIB needed to determine which GSAM databases go into the catalog
- Catalog database DBD and PSB segments will have a version and contain a timestamp that matches the ACB member timestamp
 - Used to associate an ACB member with a catalog member
 - Timestamp exceptions
 - DBDGEN timestamp for Logical and GSAM DBs
 - PSBGEN timestamp for GSAM only PSBs

■ Populate the IMS Catalog ...

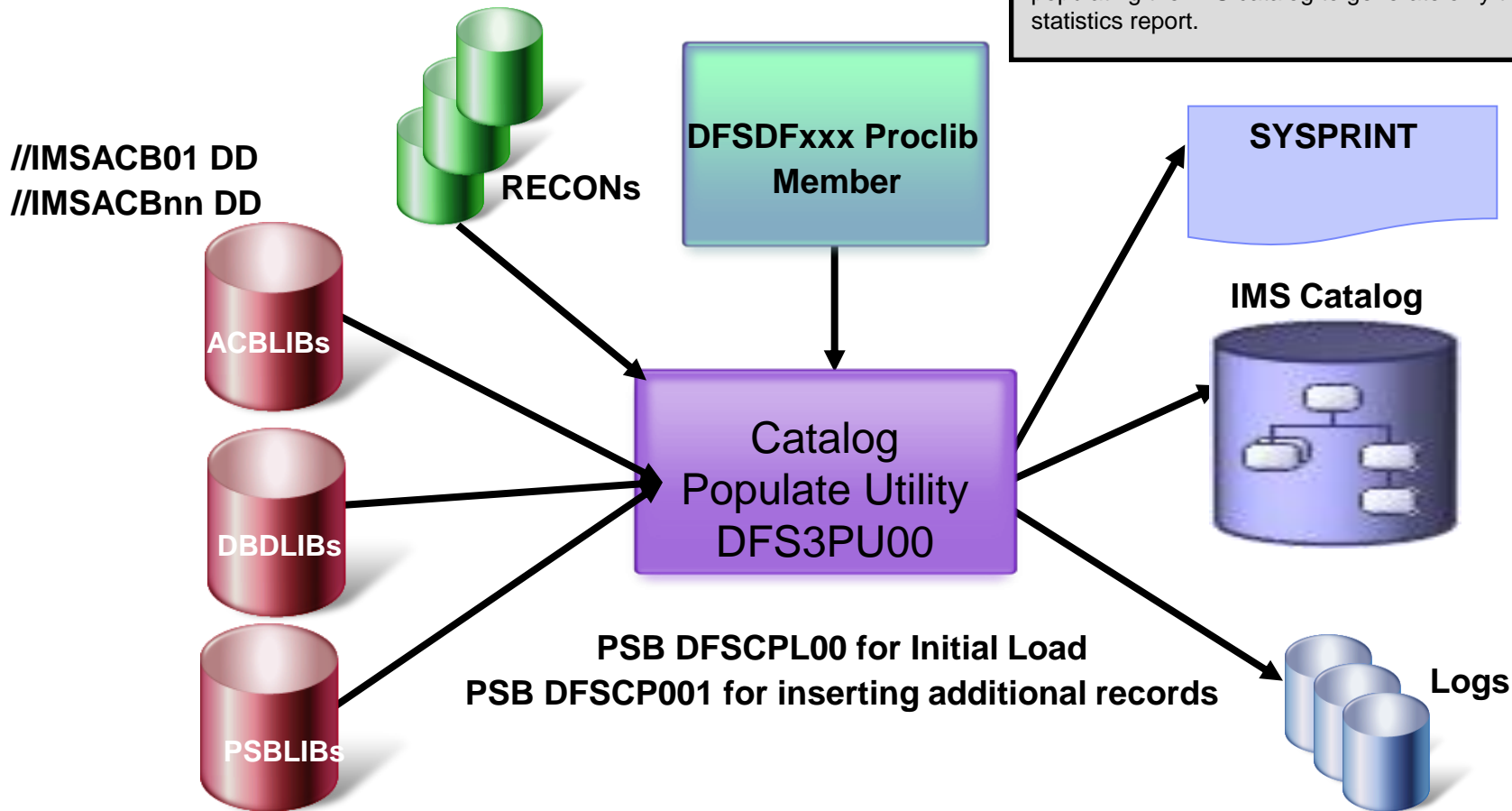
- If any of the database data sets do not exist, the DFS3PU00 utility creates them automatically: i.e. the DFSCD000 database data sets which are:
 - The primary index data set
 - The indirect list data set (ILDS)
 - Four data set group data sets for the segments of the IMS catalog
 - DFSCX000 secondary index data set
- DFS3PU00 calculates the size of the database data sets based on both the size of the ACB library and the expansion percentages that you can specify in the DFSDFxxx PROCLIB member
 - The DFSDFxxx member is also where you specify the volume serial number for the VSAM key-sequenced data sets or the SMS storage class, data class, and management class for all data sets

- IMS Catalog Populate utility
 - Can run as a typical IMS Batch or BMP job
 - Requires IMS logs for backout / recovery
 - Requires IRLM if catalog is shared and catalog active in an IMS subsystem
 - Business as usual for data sharing
 - Requires DBRC if catalog is defined in the RECON
 - If using Catalog Partition Definition Data Set
 - User's responsible to ensure online catalog access has ceased
 - Business as usual for non-registered full function DB

Catalog Populate Utility – DFS3PU00



If you need to know how much DASD storage the IMS catalog data sets will use before they are created, you can run the DFS3PU00 utility without populating the IMS catalog to generate only the statistics report.



Take an Image Copy of the Catalog database after the initial populate!
(Required if using DBRC)

8

▪ If the Catalog is registered to DBRC

- Catalog database datasets are now marked “Image Copy Needed” in the RECON
- Image Copy the databases

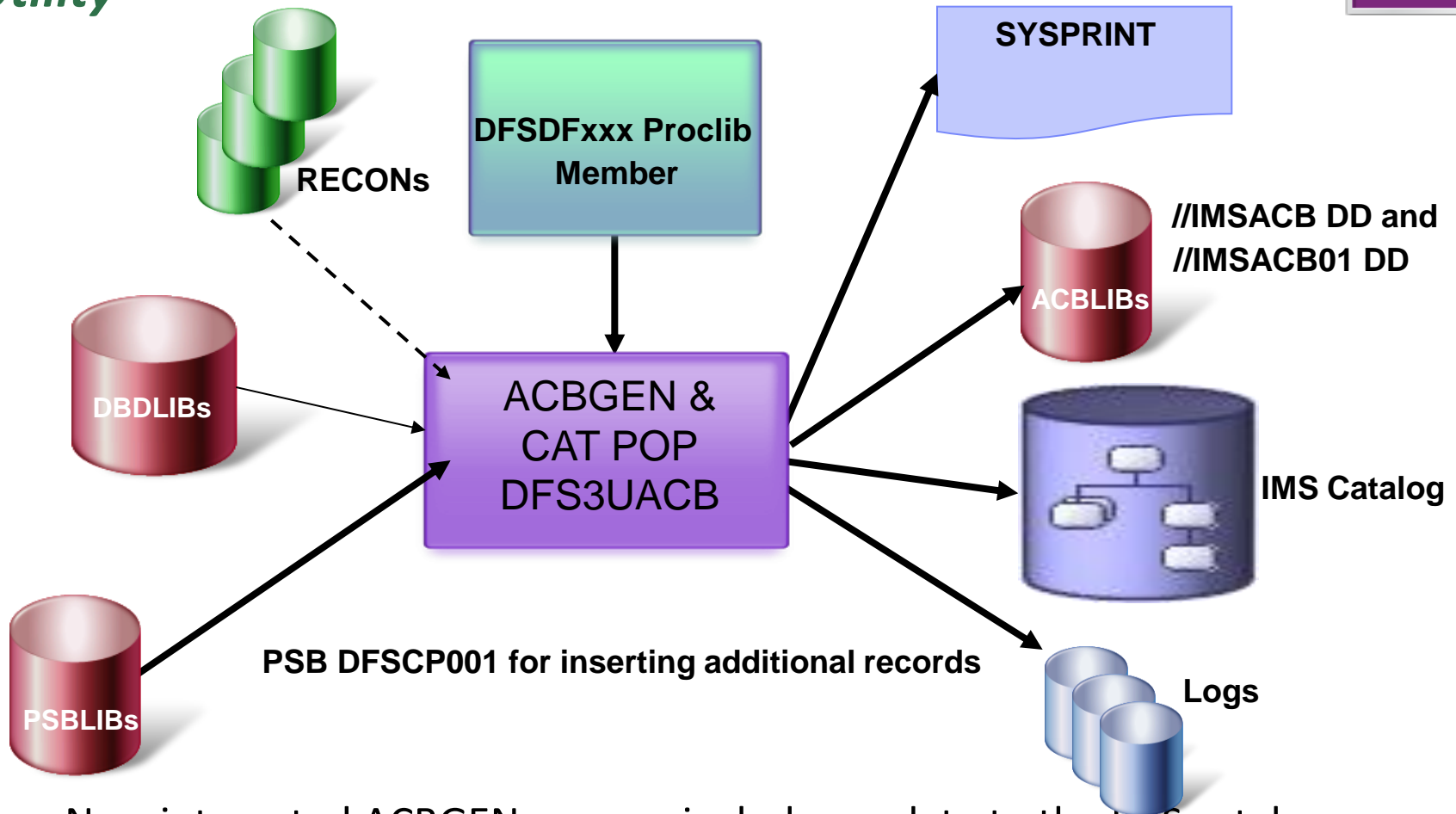
```
DSP0132I  IMAGE COPY NEEDED FOR
DSP0132I  DBDNAME=DFSCD01    DDNAME=DFSCD01A
DSP0132I  IMAGE COPY NEEDED FOR
DSP0132I  DBDNAME=DFSCD01    DDNAME=DFSCD01B
DSP0132I  IMAGE COPY NEEDED FOR
DSP0132I  DBDNAME=DFSCD01    DDNAME=DFSCD01C
DSP0132I  IMAGE COPY NEEDED FOR
DSP0132I  DBDNAME=DFSCD01    DDNAME=DFSCD01D
```

- 9 ■ Restart IMS causing DFSDFxxx to be read and Catalog to be enabled!

```
Response for: QRY DB NAME (DFSC*) SHOW (ALL) More:
DBName      PartName  MbrName      CC TYPE      Acc Status  LAcc LRsdnt  LclStat
DFSC*                IMB2          0              NONE
DFSCD000                IMB1          0 PHIDAM      READ Y
DFSCD000 DFSCD01   IMB1          0 PART        READ        NOTOPEN
DFSCD000 DFSCD01   IMB1          0 PART        READ Y      NOTOPEN
DFSCD000                IMB2          0 PHIDAM      READ Y
DFSCD000 DFSCD01   IMB2          0 PART        READ Y      NOTOPEN
DFSCD000 DFSCD01   IMB2          0 PART        READ        NOTOPEN
DFSCX000                IMB1          0 PSINDEX     READ Y
DFSCX000 DFSCX01   IMB1          0 PART        READ        NOTOPEN
DFSCX000 DFSCX01   IMB1          0 PART        READ Y      NOTOPEN
DFSCX000                IMB2          0 PSINDEX     READ Y
DFSCX000 DFSCX01   IMB2          0 PART        READ Y      NOTOPEN
DFSCX000 DFSCX01   IMB2          0 PART        READ        NOTOPEN
```

- **New ACB Generation and Catalog Populate utility, DFS3UACB**
 - Replaces existing ACBGEN Utility, DFSUACB0, if IMS catalog enabled
 - Generate ACBLIB member and create catalog metadata in a single job
 - 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 format, 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 Via New ACBGEN & Populate Utility



- New integrated ACBGEN process includes update to the IMS catalog
- DFSDFXxx PROCLIB member has the catalog information
- DFSMDA member used to dynamically allocate the catalog datasets (if not using DBRC for the Catalog)

- Catalog database management is required
 - Review/adjust database buffer pool definitions
 - Perform routine management and maintenance on the Catalog database
 - Image Copy, Pointer Checker, Reorg, etc...
 - Catalog database will need to be REORG'd
 - If Catalog database is defined to the RECONS →
 - HALDB OLR non-disruptive reorganization is supported
 - If Catalog database is not defined to the RECONS →
 - HALDB OLR can't be supported and a reorg utility must be employed
- Support with existing backup and recovery procedures
 - Image copy, recovery, backout utilities etc...
 - If Catalog database is not defined to the RECONS → recovery is limited
 - Same as non-registered, full function database recovery procedure

- Run the Catalog Purge Utility DFS3PU10 to remove Catalog entries

```
//BATCH EXEC PGM=DFSRR00,  
// PARM=(DLI,DFS3PU10,DFSCP001,,,,,,,,,IMB1,,Y,Y,IRLB,,,,,,,,,  
//      'DFSDF=0B1')  
//STEPLIB DD DSN=IMSCFG.IMB1.SDFSRESL,DISP=SHR  
//      DD DSN=IMSCFG.IMSB.USERLIB,DISP=SHR  
//DFSRESLB DD DSN=IMSCFG.IMB1.SDFSRESL,DISP=SHR  
//PROCLIB DD DSN=IMSCFG.IMSB.PROCLIB,DISP=SHR  
//IMS      DD DSN=IMSCFG.IMSB.PSBLIB,DISP=SHR  
//      DD DSN=IMSCFG.IMSB.DBDLIB,DISP=SHR  
//SYSUT1 DD DSN=IMSCFG.IMSB.CATPURG,DISP=OLD  
//IEFRDER DD DSN=&&IMSLOG1,DISP=(NEW,DELETE),  
// DCB=(RECFM=VB,LRECL=4092,BLKSIZE=4096,BUFNO=5),  
// SPACE=(CYL,(200,75)),UNIT=SYSDA  
//IEFRDER2 DD DSN=&&IMSLOG2,DISP=(NEW,DELETE),  
// SPACE=(CYL,(200,75)),UNIT=SYSDA  
//DFSVSAMP DD DSN=IMSCFG.IMSB.PROCLIB(DFSVSMB),DISP=SHR  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
MODE BOTH  
//*
```



Living with the Catalog



- Sizing the Catalog
 - How many Partitions based on the number of PSBs & DBDs
 - What Block size/CISIZE to use for performance & locking issues
- Backup & Recovery of the Catalog
- Changes to the ACBGEN processes which involve change control and fall back plans
- Default started access for the catalog is READ. So if you plan to do ACBGEN using the CATPOP as a BMP you need to start the catalog in UP mode.
- Dont forget the Catalog Purge Utility

- IMS Catalog is key element of IMS strategy
 - For integration and simplification

 - More and more products/processes use it

 - IMS 13 features which pre-req it

 - It's simple to implement – it's an IMS database after all!