

The IMS Catalog Revolution

Deepak Kohli

deepakk@us.ibm.com

IMS Product Management Team

Virtual IMS User Group
August 9, 2016



Disclaimer



© Copyright IBM Corporation 2016. 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.

Agenda



- IMS Catalog Overview
- IMS Catalog Configuration
- IMS V14 Catalog related Enhancements
 - Dynamic Database Definition
 - IMS Managed ACBs
- Living with (or concerns with) the IMS Catalog ←=====

But First ... some more resources



- **IMS 12: The IMS Catalog, REDP-4812-00, published 20 September 2012**

- **IMS Version 13 Technical Overview, SG24-8224-00, published 15 September 2014**

- **YouTube Video on Implementing the IMS Catalog (pre-V14):**
 - <https://www.youtube.com/watch?v=lcrOhmGzUWY>

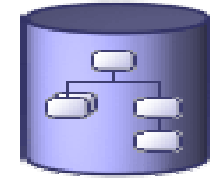
IMS Catalog Overview



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 available in IMS 13 & IMS V14 & future DB enhancements





- 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?



- 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 MobileFirst (formerly IBM Worklight Foundation)
 - InfoSphere DataStage
 - InfoSphere Metadata Asset Manager (IMAM)



- IMS 13 features which require the Catalog:
 - Database Versioning
 - IMS Native SQL support for COBOL
 - .NET access to IMS data

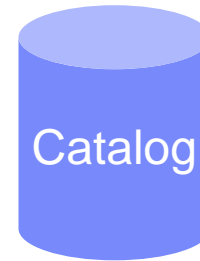
- IMS 14 features which require the Catalog:
 - IMS V14 Dynamic Database Definition
 - IMS V14 Managed ACBs

Types of technical metadata and storage medium



▪ DB

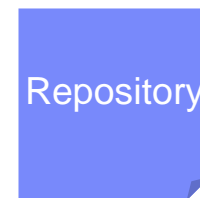
- PSB/DBD resources
 - Database structure definitions
 - Physical database definitions
 - Segment definitions
 - Field definitions
- Application
 - Data types
 - Application defined fields
 - Encodings
 - Redefines
 - User defined types
 - Structures



IMS database

▪ TM

- MODBLKS resources
 - Program definitions
 - Transaction definitions
- FORMAT resources
- Application
 - Input/output message definitions

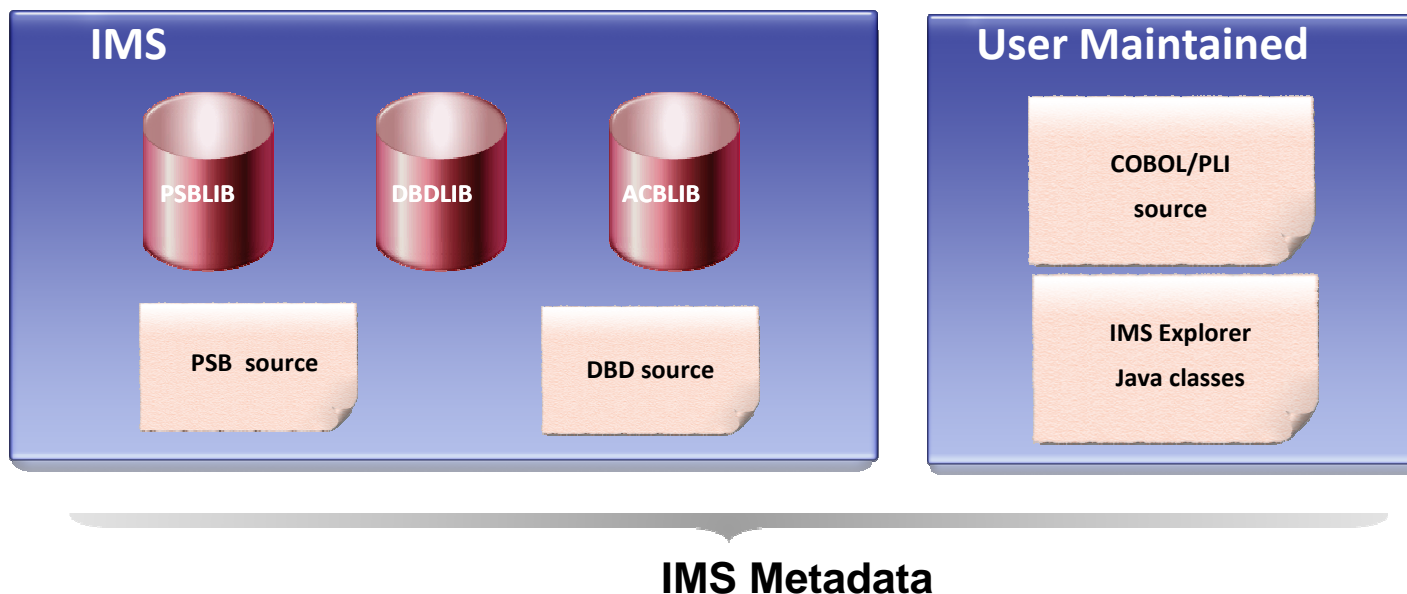


VSAM

Before the IMS Catalog



- 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



With the IMS Catalog



- 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

The IMS Catalog Database



- 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

The IMS Catalog Database & PSBs



- 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

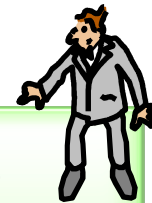
The catalog PSB itself occupies 12kb in the DLIPSB pool and 500 bytes CSAPSB pool for each user PSB that is using the catalog PSBs. You might need to increase the size of your storage pools, up to the maximum size of the catalog PSB in each pool multiplied by the number of user PSBs that concurrently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



The IMS Catalog Database

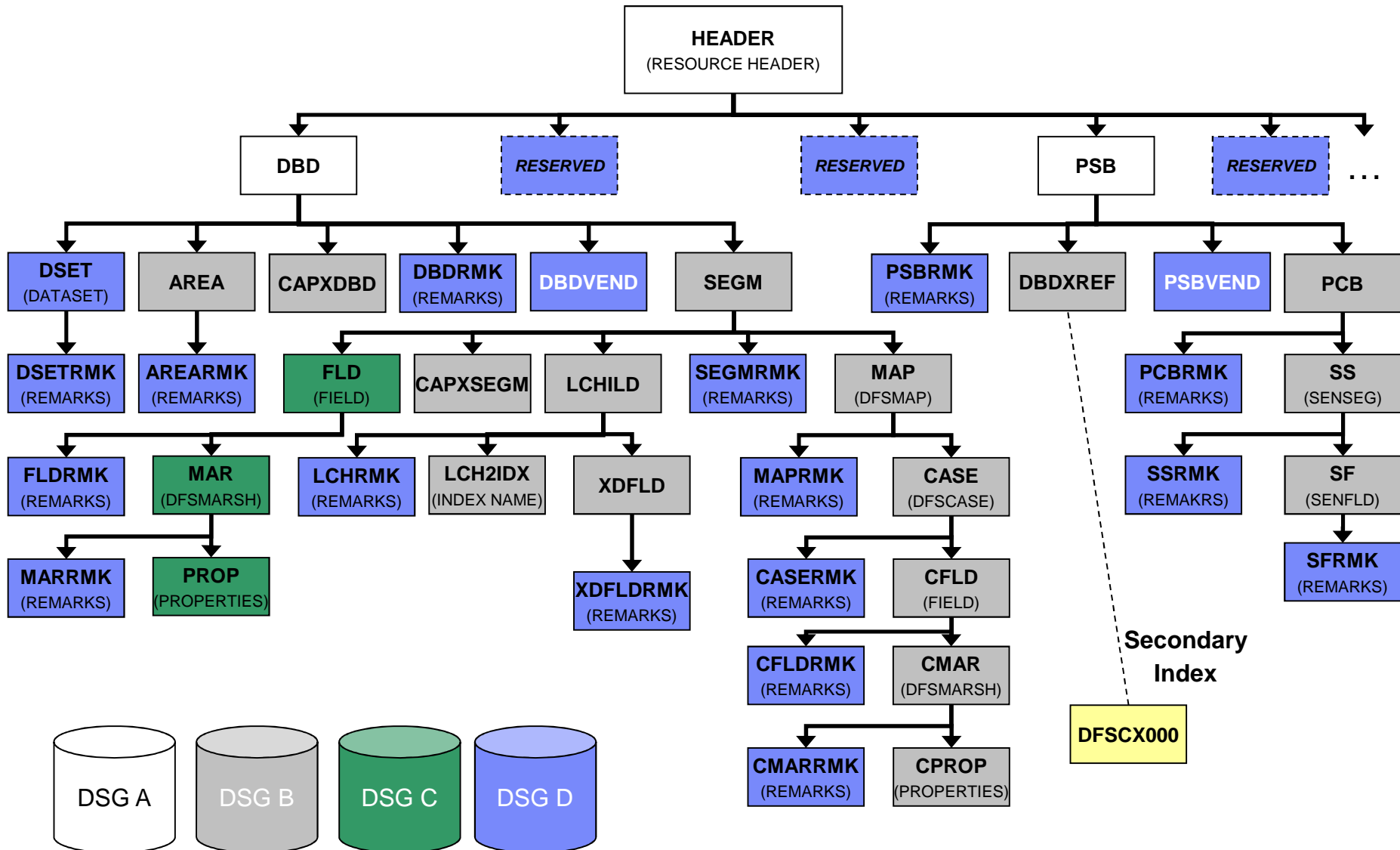


- Root segment of the Catalog database is a generic resource header
 - Indicates the type of resource → DBD or PSB
 - A dependent of the Root and its Children are a complete DBD or PSB
 - Multiple iterations/instances of a specific resource are supported
 - Most resources are differentiated by their 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

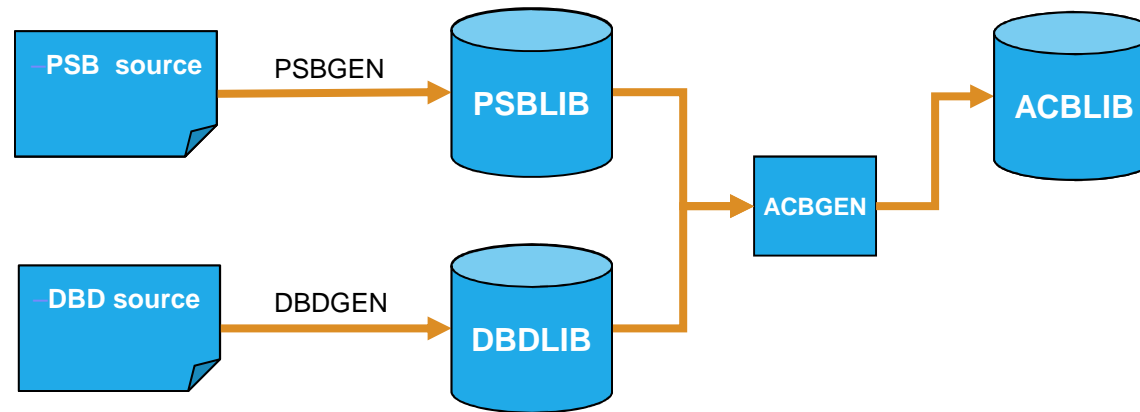


An “instance” of the IMS catalog metadata is **generated each time an ACBGEN is done**. The DFSDFxxx member of the IMS PROCLIB data set is used to configure how many instances are stored and when IMS removes old instances.

Physical Catalog Structure



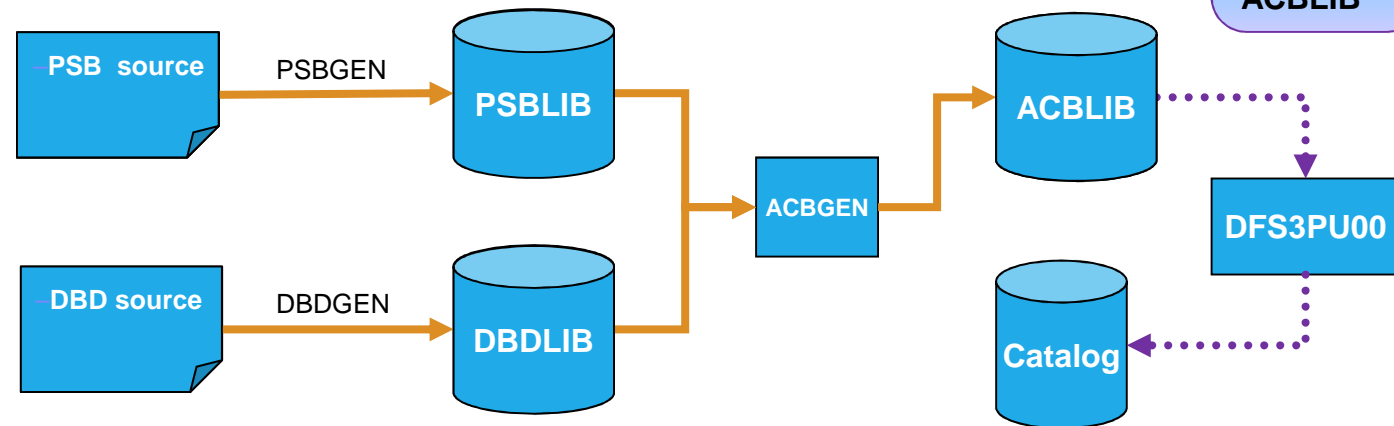
The ACBGEN Process Today



Populating the IMS Catalog (from ACBLIB)

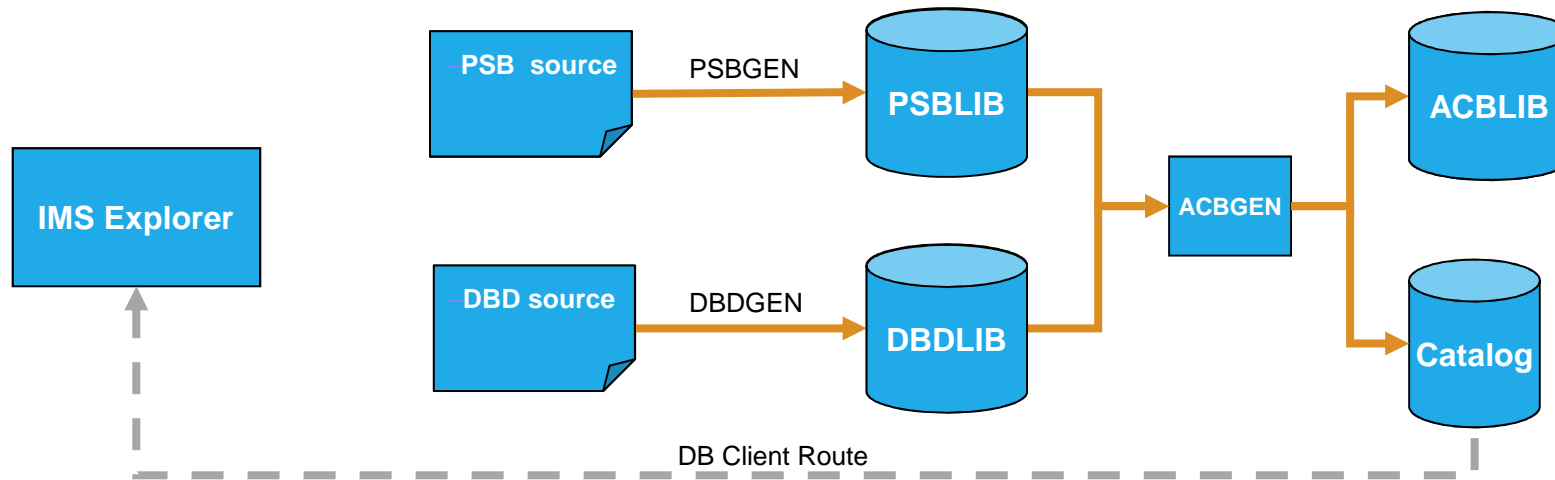


Populate Utility will populate catalog from ACBLIB



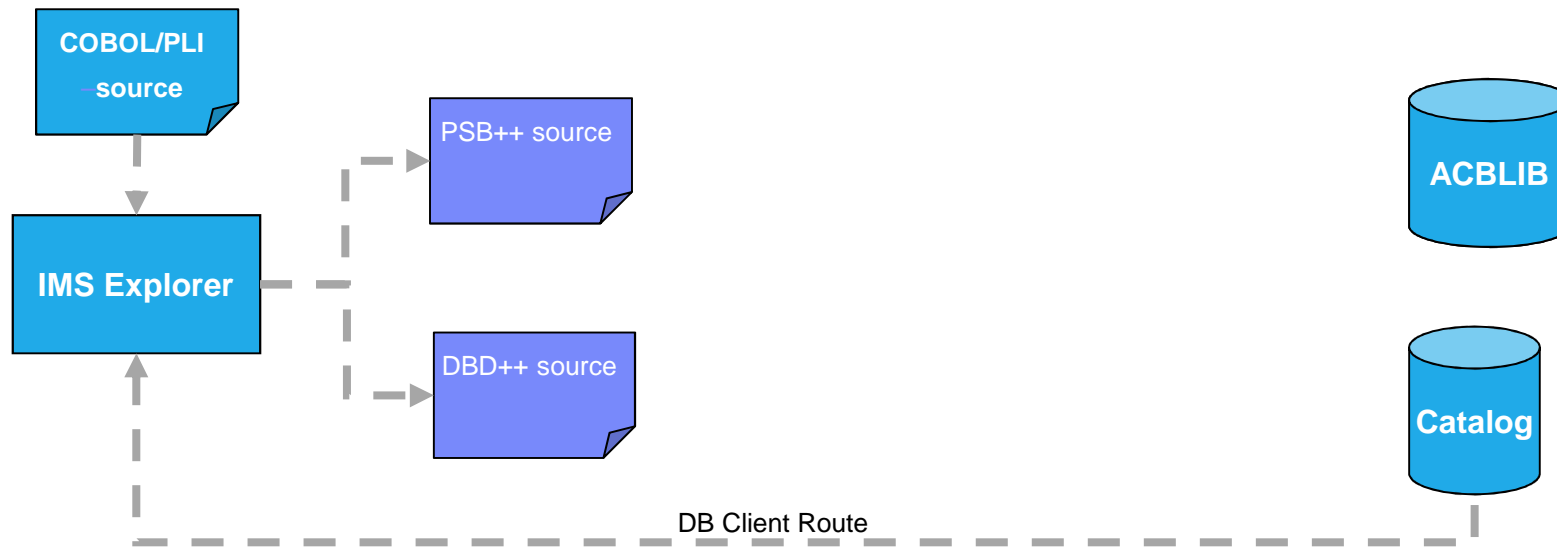
- Initially populate the IMS catalog with the Catalog Populate Utility (only done once)

IMS Catalog Lifecycle 1/3

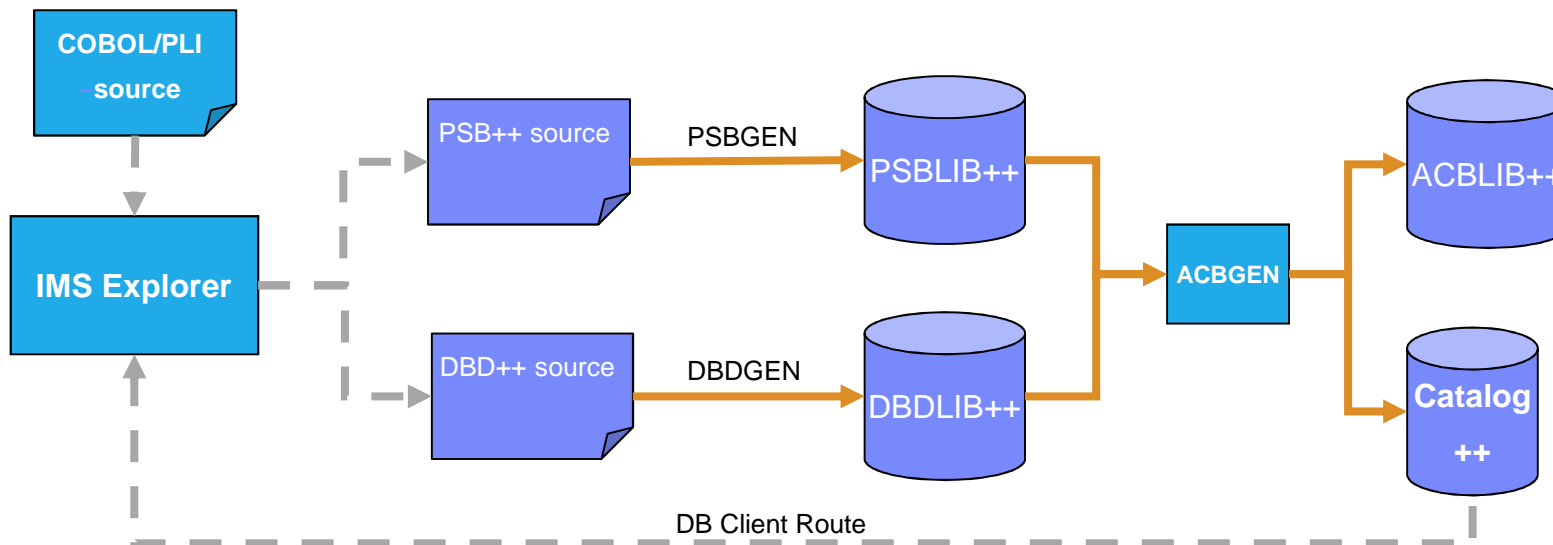


- PSB & DBD information in the catalog can be viewed by the IMS Explorer

IMS Catalog Lifecycle 2/3

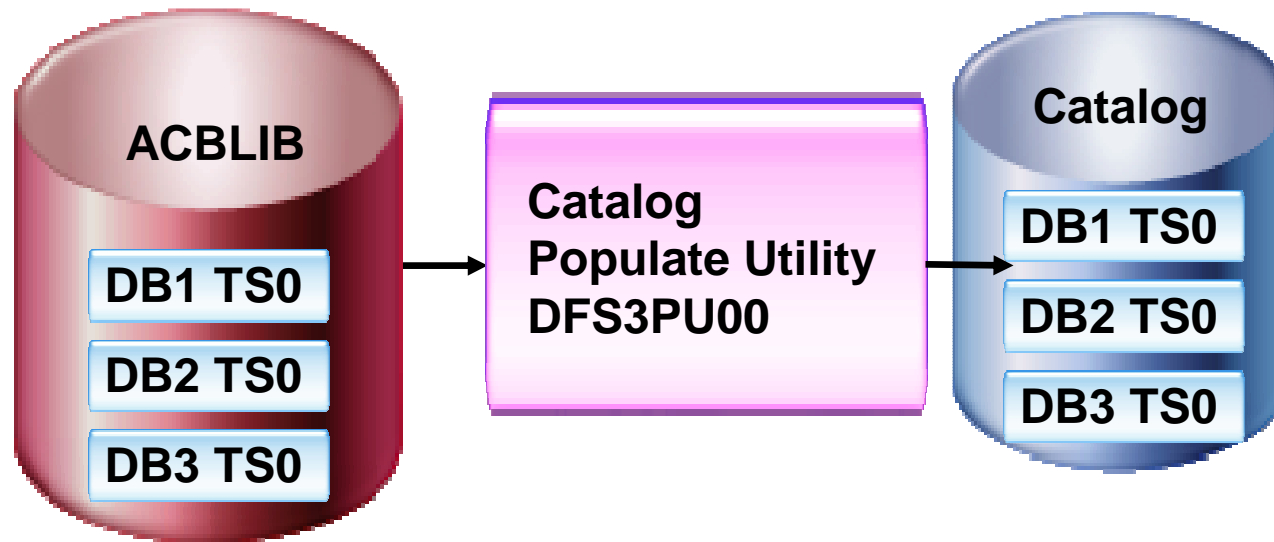


- ❑ Incorporate application metadata & generate new DBD & PSB source via the IMS Explorer



- 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

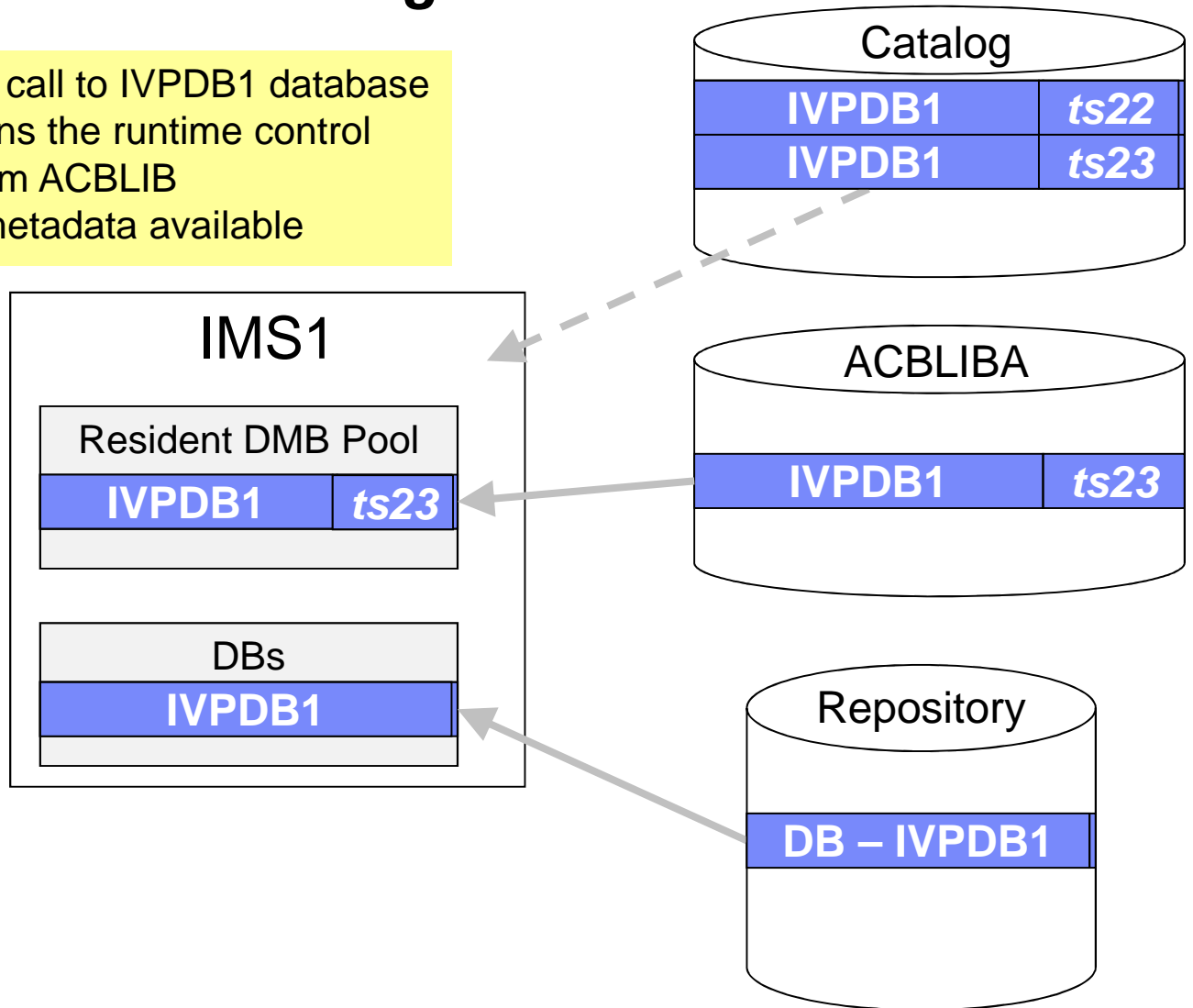
IMS Catalog Members After Populate Utility



- 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

Running with IMS Catalog and ACBLIB

- Appl SQL call to IVPDB1 database
- IMS obtains the runtime control blocks from ACBLIB
- Catalog metadata available



IMS Catalog Configuration



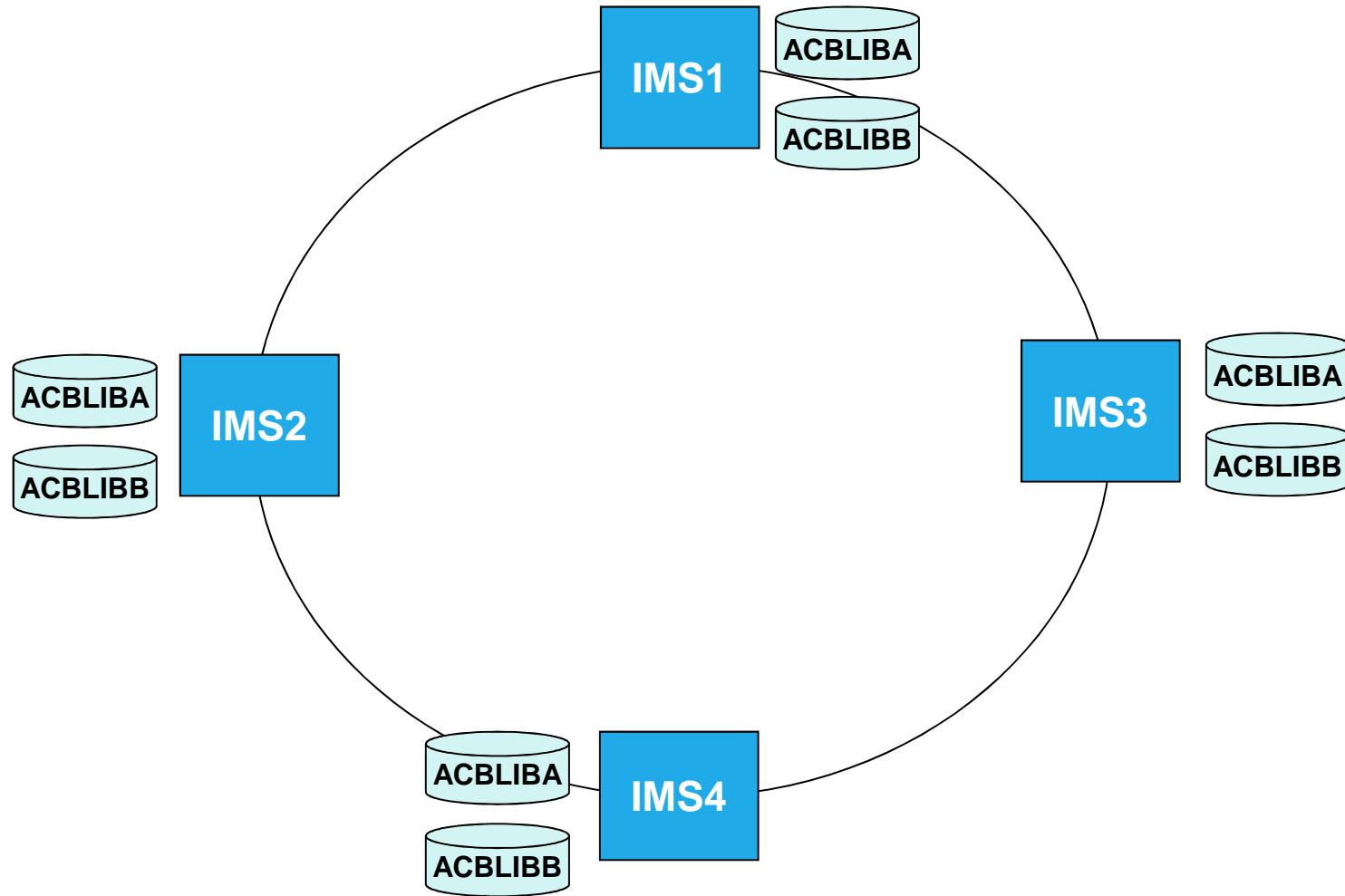
Catalog Configurations



- 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

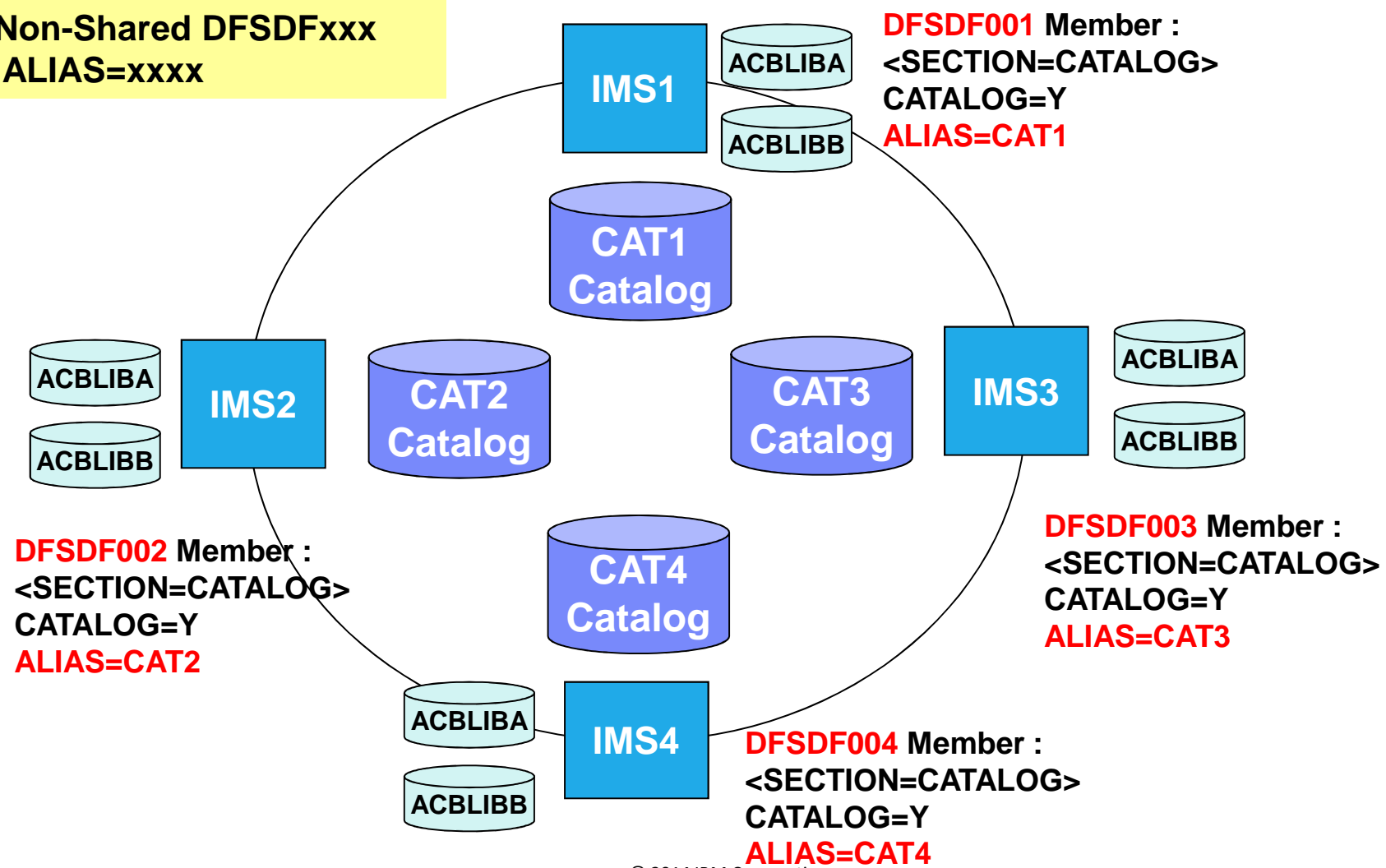
Multiple IMSES, each IMS has its own cloned ACBLIBs



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



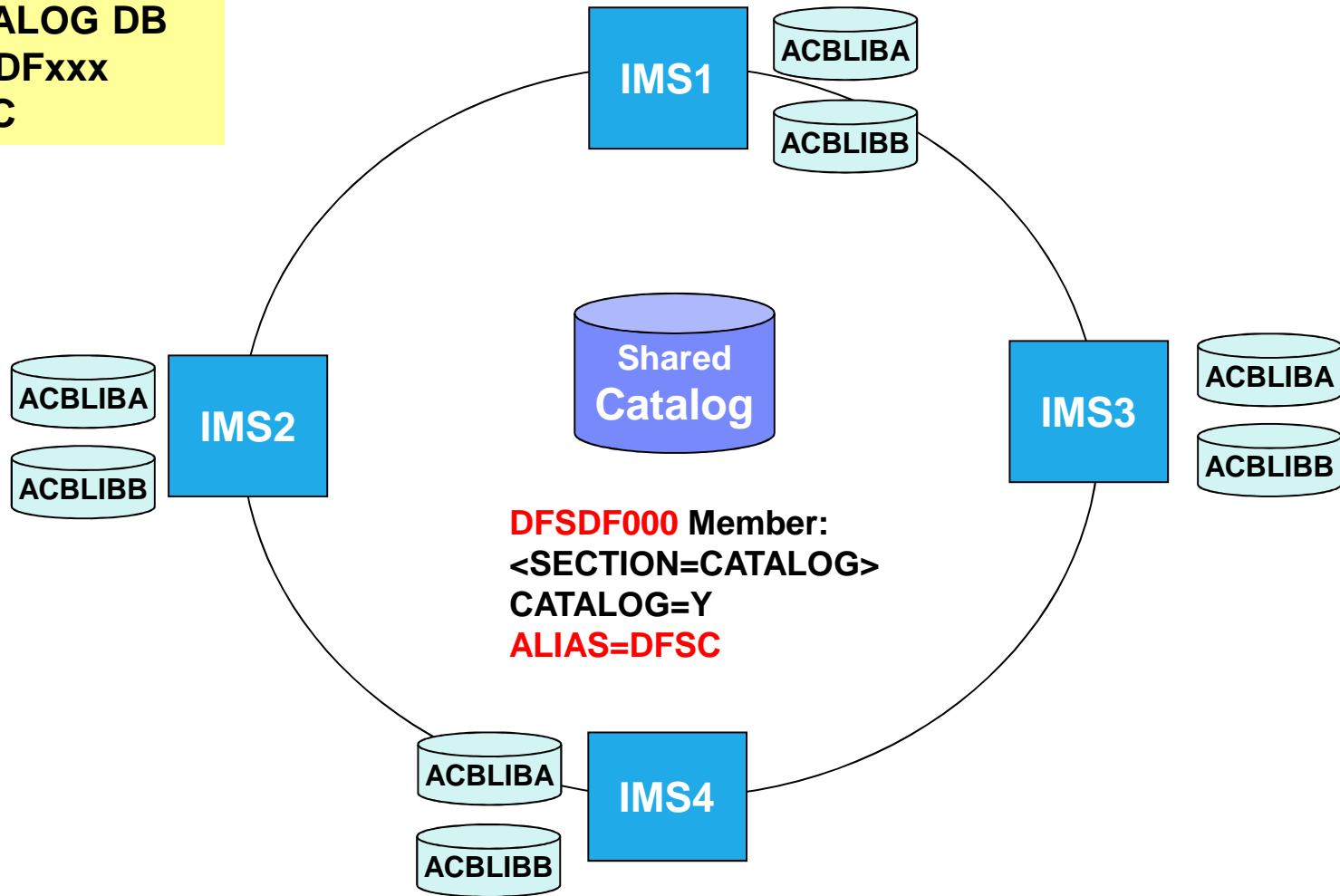
Non-Shared CATALOG DB
Non-Shared DFSDFxxx
ALIAS=xxxx



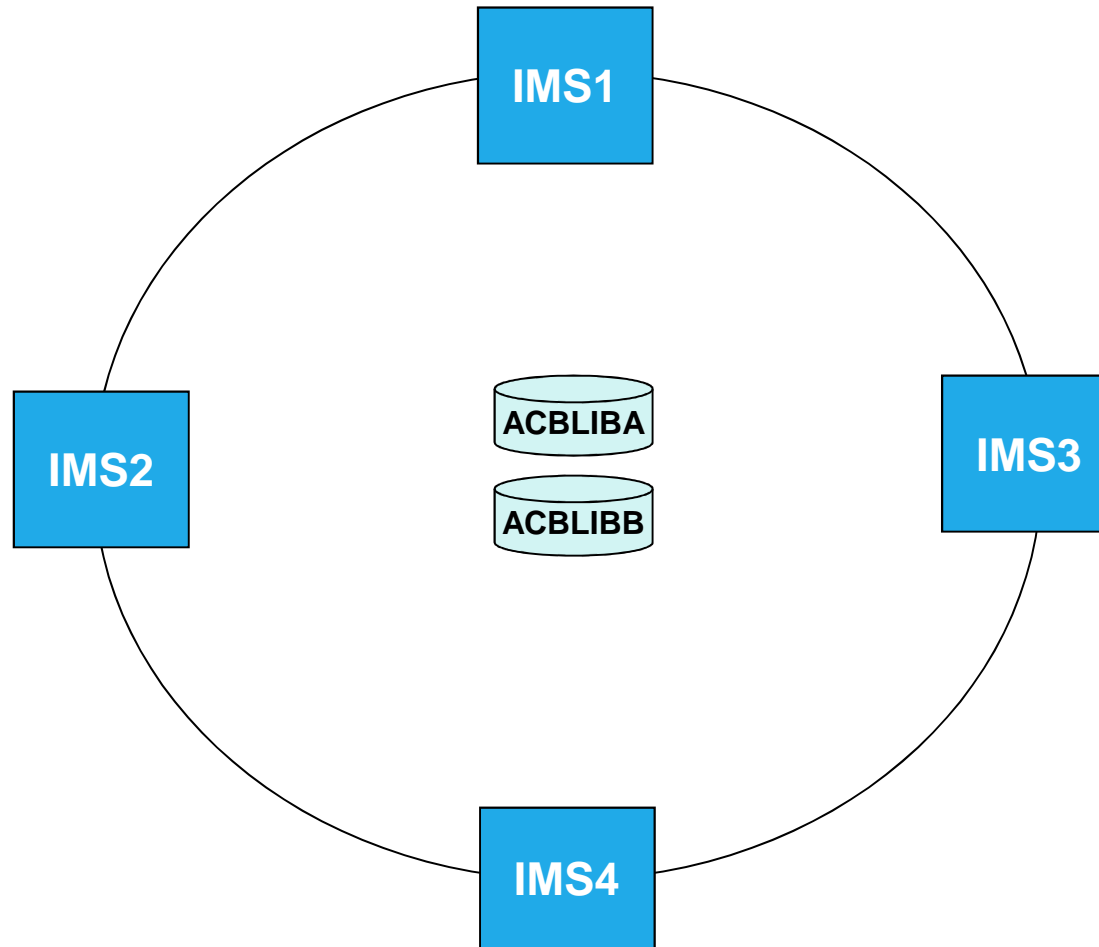
Multiple IMSes, Cloned ACBLIBs, Shared Catalog



Shared CATALOG DB
Shared DFSDFxxx
ALIAS=DFSC



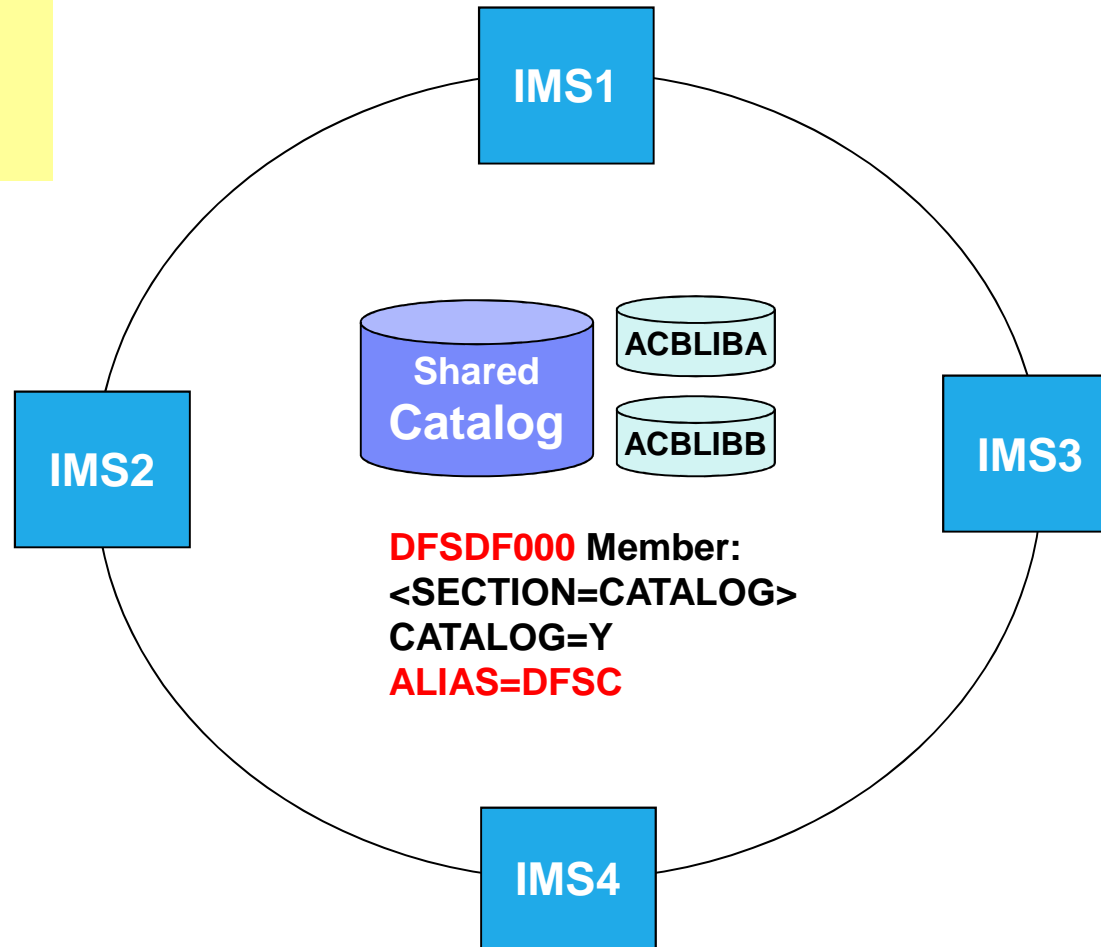
Without an IMS Catalog
Multiple IMSES, shared ACBLIBs



Multiple IMSes, Shared ACBLIBs, Shared Catalog



Shared CATALOG DB
Shared DFSDFxxx
ALIAS=DFSC





IMS V14 Catalog related enhancements

- IMS 14 enables **dynamic definition and modification** of IMS database and program resources directly into the IMS catalog
 - Leverages the use of industry standard Data Definition Language (DDL) commands
 - Provides an alternative to the existing DBDGEN, PSBGEN, and ACBGEN
 - Provides an audit trail to capture information about catalog updates
- IMS 14 enables IMS to manage loading of runtime ACBs, application control blocks, from the IMS catalog (**IMS Managed ACBs**)
 - DBD, PSB, and ACB libraries are no longer required for IMS runtime
 - Libraries possibly needed for tools, utilities and other processes
 - User responsible for keeping DBD, PSB, and ACB resources in sync with the IMS catalog
 - Online Change process no longer needed to in effect ACBs

Database Definition Language for IMS

- Use DDL to create, modify, and delete IMS databases and application program resources and activate definitions in a target IMS system immediately
 - no gen processes required
 - no Online Change process required
- DDL commands can be submitted from a Java client using the IMS Universal Drivers or through tools that support the Universal Drivers
- DDL commands supported
 - CREATE DATABASE
 - CREATE TABLE
 - CREATE TABLESPACE
 - CREATE PROGRAMVIEW
 - ALTER DATABASE
 - ALTER TABLE
 - ALTER TABLESPACE
 - DROP DATABASE
 - DROP TABLE
 - DROP TABLESPACE
 - DROP PROGRAMVIEW
 - COMMENT ON
- The IMS catalog and “directory” are required to use DDL definitions

Resource Name Comparison

- IMS will support the standard DDL syntax for CREATE, ALTER and DROP of Databases and Tables
 - Consume the standard DDL generated without IMS affinity
- The equivalent IMS to DDL statements are shown in the Table here.

IMS	GEN statement	DDL
Database	DBD	DATABASE
Segment	SEGM	TABLE
Field	FIELD	COLUMN
Dataset	DATASET	TABLESPACE
Area	AREA	TABLESPACE
Program	PSBGEN	PROGRAMVIEW
PCB	PCB	SCHEMA
Senseg	SENSEG	SENSEGVIEW

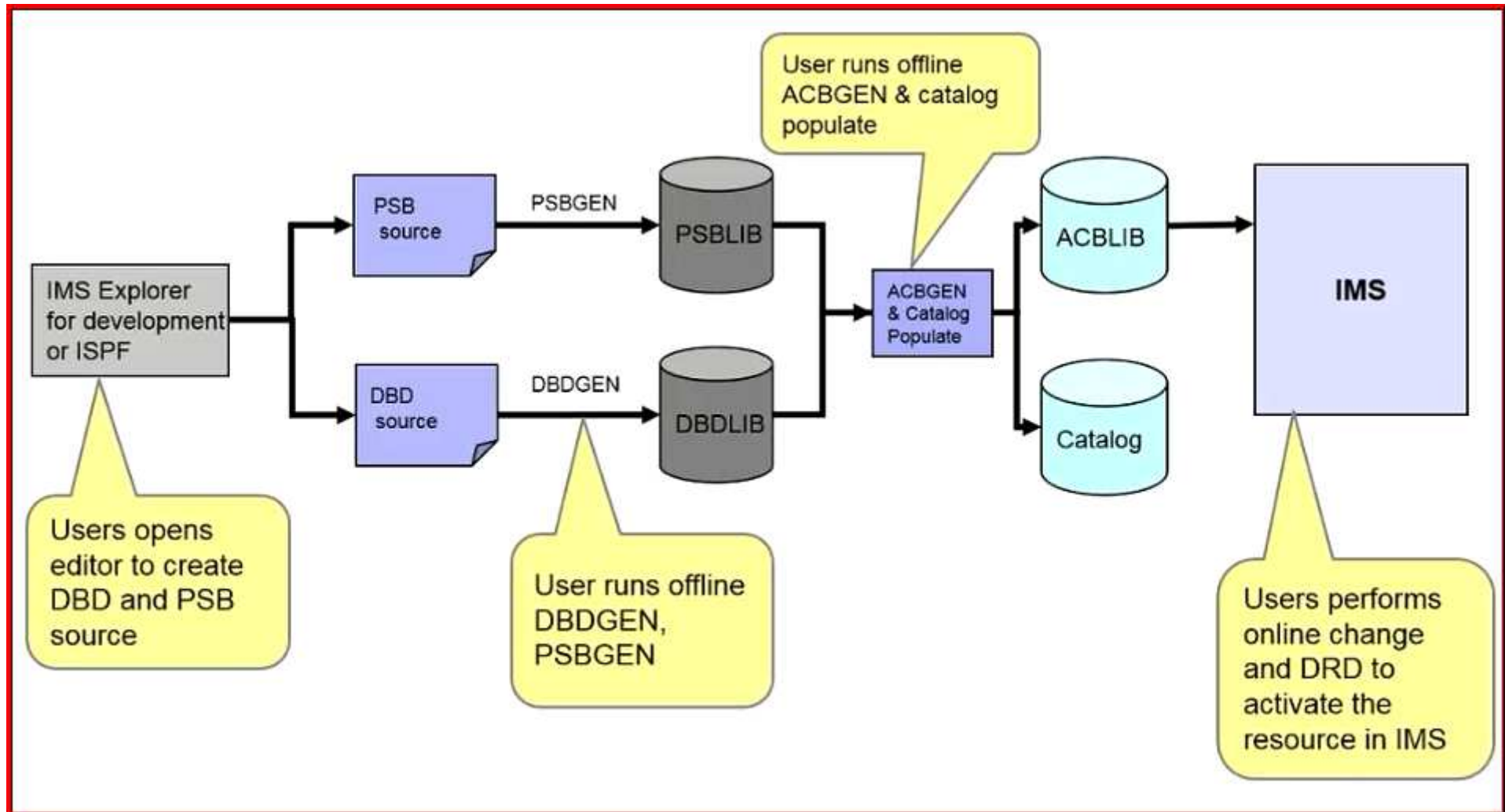
IMS DDL syntax – CREATE DATABASE & TABLESPACE Example

```
DBD  NAME=COGDBD, C
      ENCODING=Cp1047, C
      ACCESS=(HDAM,OSAM), C
      RMNAME=(DFSHDC40,3,3,25), C
      PASSWD=NO
DATASET DD1=COGDATA, C
        DEVICE=3390, C
        SIZE=(8192), C
        REMARKS='Dataset Group 1'
SEGM  NAME=ROOT, C
      PARENT=0, C
      BYTES=(20), C
      RULES=(LLL,HERE)
FIELD NAME=(ROOTKEY,SEQ,U), C
      BYTES=12, C
      START=1, C
      TYPE=C, C
      DATATYPE=CHAR
FIELD NAME=TABTYPE,
      BYTES=8,
      START=13,
      TYPE=C,
      DATATYPE=CHAR
```

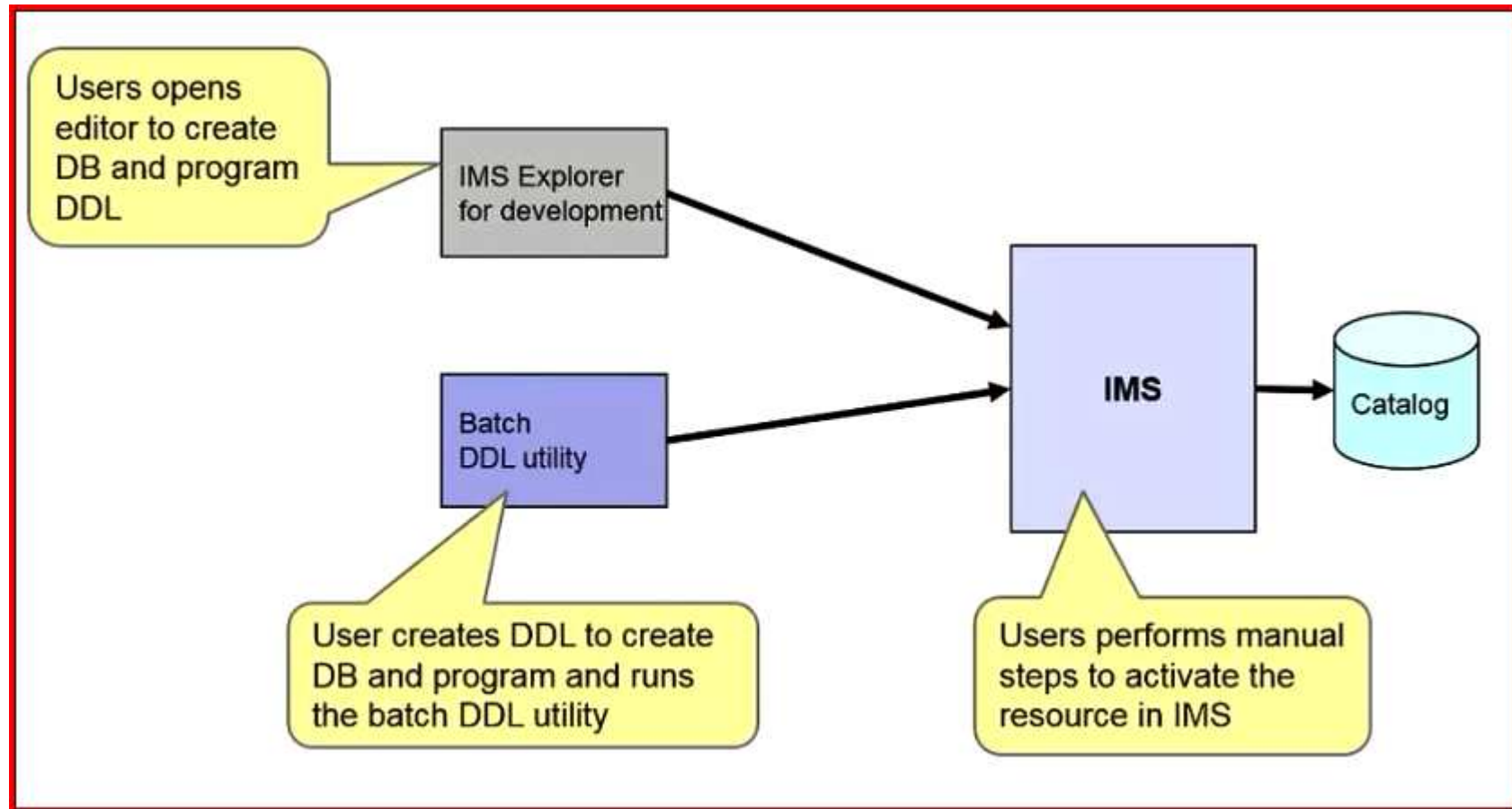
```
CREATE DATABASE cogdbd
  ACCESS HDAM OSAM
  RMNAME(DFSHDC40 3 3 25);

CREATE TABLESPACE cogdata
  IN cogdbd
  SIZE PRIMARY 8192;
COMMENT ON TABLESPACE cogdata IN cogdbd IS 'Dataset Group 1';
```

Before IMS 14 – Creating a PHDAM Database



IMS 14 – Creating a PHDAM Database

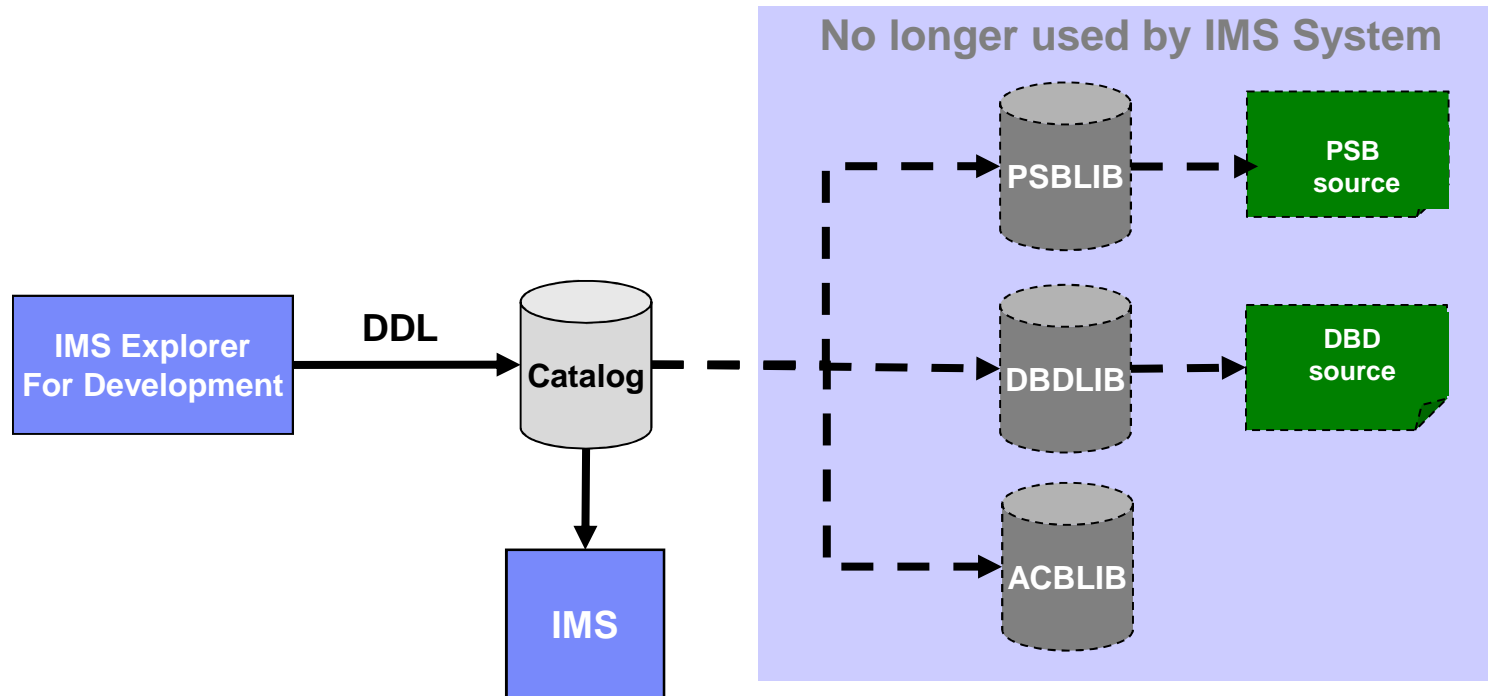


Batch SQL Utility

- The Batch SQL Utility is made as a way to invoke DDL statements via a JCL on the z platform

- **Requirements:**
 - The utility is bundled in the IMS JDBC driver (imsudb.jar)
 - APAR PI30848
 - Uses IMS JDBC Type-4 connections to invoke SQL statements
 - IMS Connect, ODBM, SCI
 - IBM Java for z/OS (JZOS) Batch Launcher

Advantage of IMS ACB Management



- Create DDL statements to build or modify IMS database or program resources
- IMS processes SQL DDL statements
- Activation of some DDL commands can happen automatically or commands are held in pending activation state until an IMPORT command is entered
 - **IMPORT DEFN SOURCE(CATALOG)**
- IMS application control blocks will be loaded from the catalog at runtime

IMS Management of ACBs

- In IMS14, customers may *optionally* enable IMS management of ACB control blocks for database and program views

- Meaning:
 - IMS can build, activate & load ACBs into memory dynamically from the Catalog when database and program view definitions are created via DDL statements

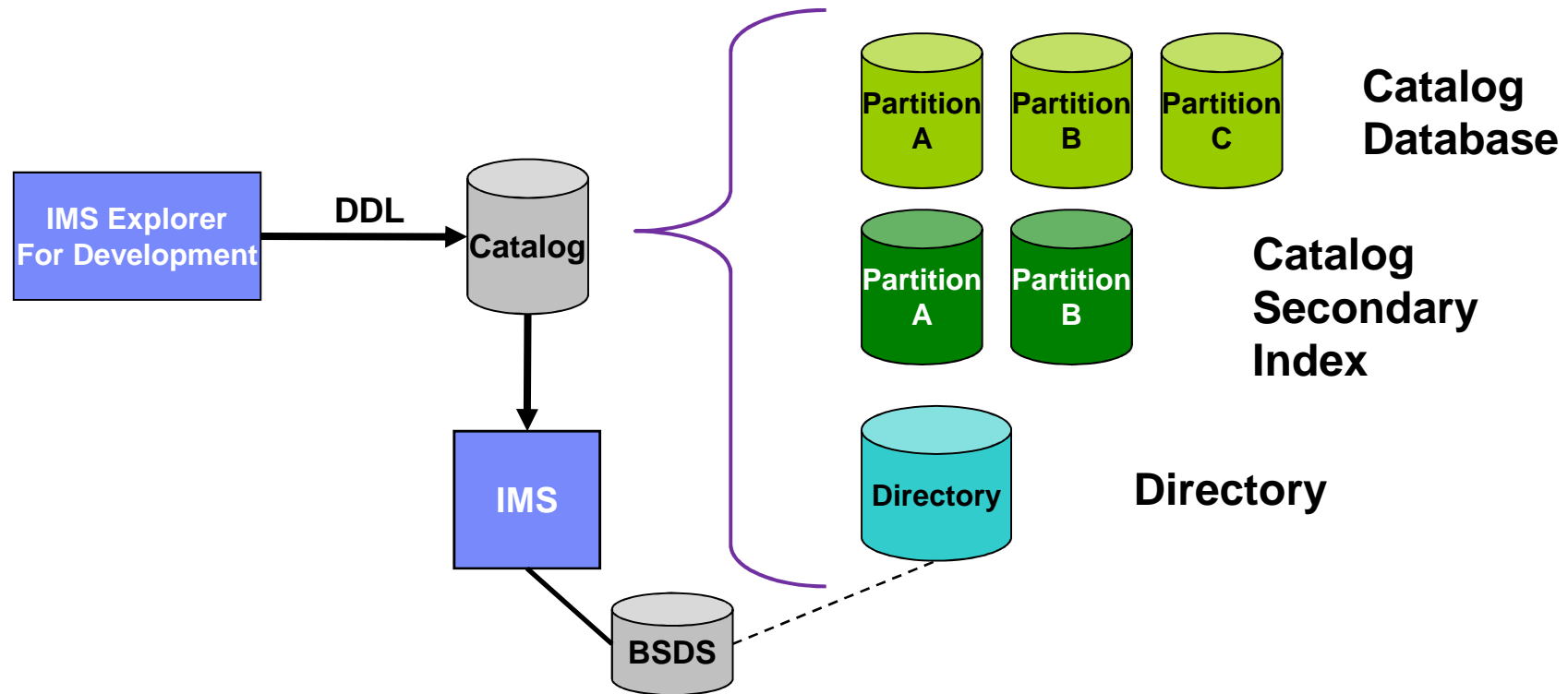
- ACBs are stored in the IMS *directory* which is an extension of the Catalog.

- IMS management of ACBs implemented by specifying:
 - **ACBMGMT=CATALOG** in the <CATALOG> section of the DFSDFxxx proclib member

IMS Management of ACBs

- The directory has functionality similar to an ACBLIB
 - Directory records have a format much like the ACBs in an ACBLIB
 - The directory is kept in sync with the catalog
- IMS maintains a boot strap data set (BSDS) containing information about the directory data sets
- IMS will self-manage the IMS directory data sets
 - Directory data sets are automatically allocated
 - another data set is allocated when a directory data set becomes full
- When IMS ACB management is enabled
 - IMS will reference the directory data sets to get the runtime application control blocks
 - IMS will use the directory to indicate which members are active in the IMS catalog
- 1:1 relationship between ACBs in the directory and the catalog
 - If a catalog is shared within an IMSplex, then the directory data sets are shared
 - If a catalog is non-shared then the directory data sets are non-shared

IMS 14 Catalog



- An “IMS Catalog” has several components
 - A Catalog HALDB Database (many partitions, 4 Data Set Groups)
 - A Secondary Index
 - Directory Data Set(s)

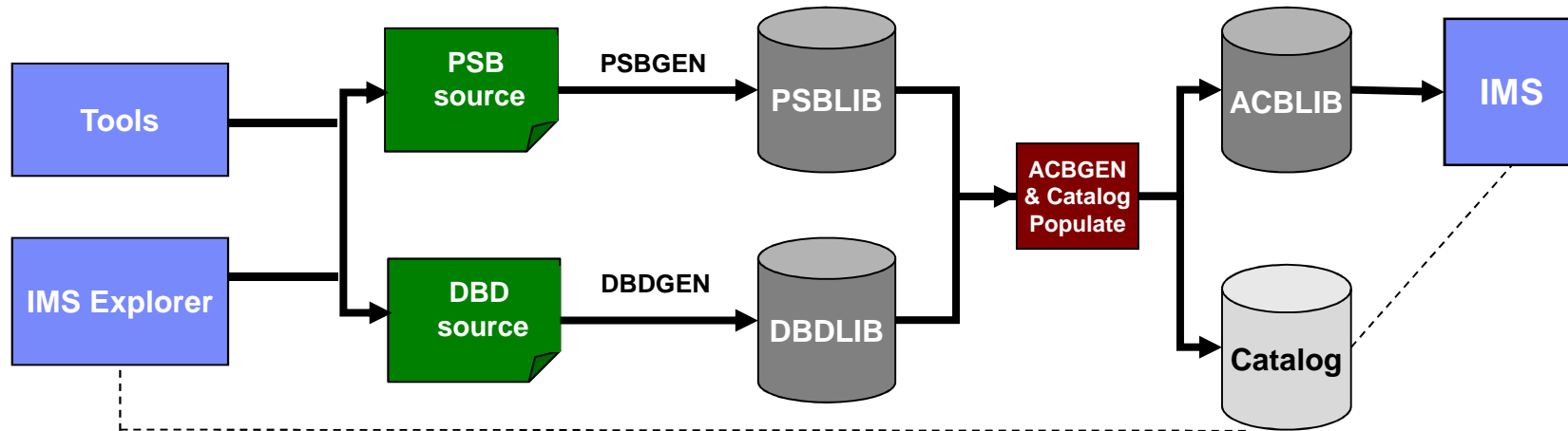
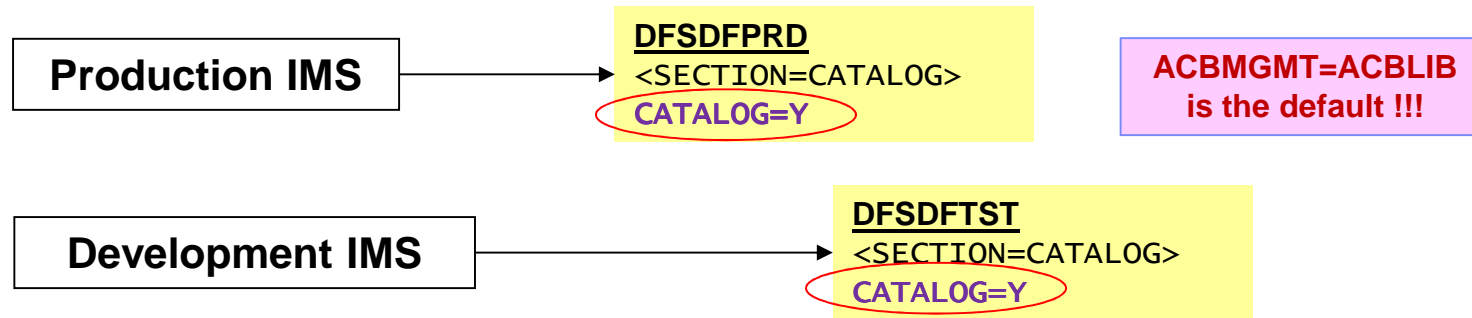
Details: IMS Catalog with Managed ACBs

- Definition of the ACB related data sets is automatic
 - IMS creates directory data sets as PDSEs
 - Data set names will be an extension of the HALDB catalog data set name
 - <HALDB Catalog data set prefix>.DI<suffix>
 - Example
 - IMSTEST.DFSCD000.DI01**001**
 - IMSTEST.DFSCD000.DI01**002**
 - Suffix will be automatically created by IMS
 - Allows IMS to automatically support multiple catalog directory data sets
 - IMS will use DFSDIRxx DD for directory data set allocation
 - If multiple directory data sets are allocated by IMS
 - 'xx' starts with 01 and will be incremented by 1

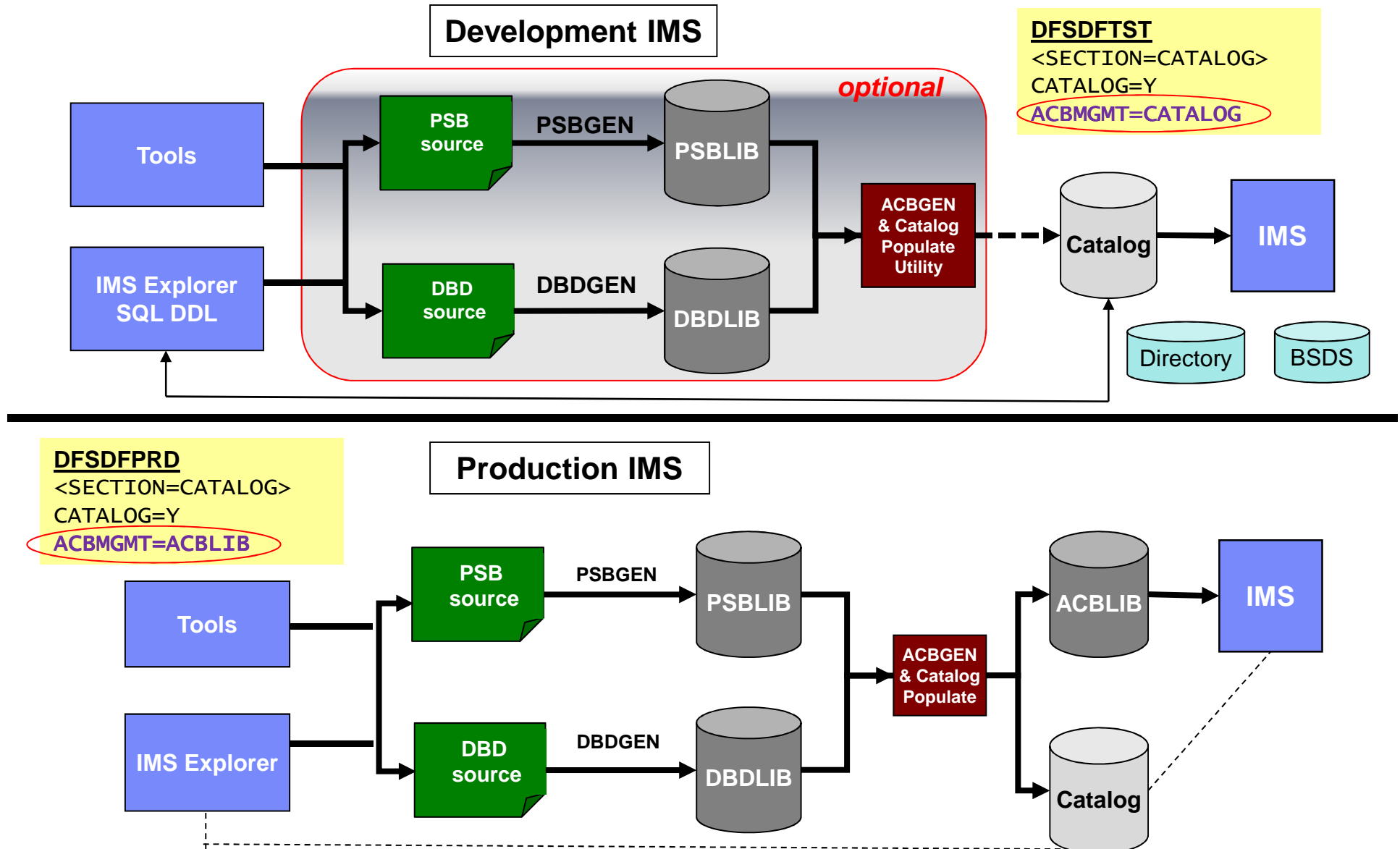
Details: IMS Catalog with Managed ACBs

- Maintaining IMS release independence of ACB members
 - IMS catalog is release independent
 - IMS 14 catalog is compatible with IMS 12 or 13
 - IMS 12 or 13 catalog is compatible with IMS 14 without the directory
 - ACB members are still release dependent
 - Long term direction is release independent ACBs
 - In the future, ACBs will be release independent ...
 - IMS catalog directory data sets will have a specific identifier within the suffix for each IMS release
 - Examples
 - IMSTEST.DFSCD000.DI01001 → Directory for V14
 - IMSTEST.DFSCD000.DI01002 → Directory for V14
 - IMSTEST.DFSCD000.DI02001 → Directory for V14+1
 - IMSTEST.DFSCD000.DI02002 → Directory for V14+1

Pre-IMS 14: IMS with Catalog and ACBLIB



IMS 14: Transition to IMS Catalog Managed ACBs



Batch Implications

- Batch jobs can be enabled for IMS management of ACBs
 - Specify use of the IMS catalog and ACB management
 - DFSDFxxx PROCLIB member added to JCL parm or
 - User exit DFS3CDX0 can be used as an alternative to modifying the JCL
 - IMS will load application control blocks from the catalog
 - If present, IMS ignores DBDLIB and PSBLIB or ACBLIB
 - Overrides the DLI or DBB processing option specified in the JCL
- When IMS is managing ACBs but the batch job needs to use DBDLIB/PSBLIB or ACBLIB
 - Do not specify a DFSDFxxx PROCLIB member or use the DFS3CDX0 exit
 - User can build DBDLIB and PSBLIB from the catalog using the IMS Catalog Database Library utility, DFS3LU00
 - ACBLIB can be created from DBD and PSB library members
 - User must keep the DBD and PSB library members in sync with the catalog
 - If used, ACBLIB must also be kept in sync with catalog
- Batch regions running in a system without the catalog and ACB management enabled
 - IMS continues to use DBDLIB and PSBLIB or ACBLIB

Summary

- Two Major Enhancements in IMS V14:
 - IMS DDL support
 - IMS Managed ACBs
- DDL requires IMS managed ACBs. DDL doesn't require DBDGEN, PSBGEN, ACBGEN.
- IMS managed ACBs does not require DDL. You can keep your old DBD/PSB source and use the populate utility to keep the catalog in sync with your source updates.

The IMS Catalog is a Database



- 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

Living with (or concerns with) the IMS Catalog 1/2



- Catalog is populated from ACBLIB (initially)
 - Batch PSBs or DBDs accessed only in batch may not be in the ACBLIB
- Sizing the Catalog
 - How many Partitions will your Catalog have?
 - Depeneds on the number of PSBs & DBDs
 - IMS Catalog is a Database. So treat it like a database!!
- Backup & Recovery of the Catalog
 - How often do you image copy the Catalog?
 - IMS Catalog is a database. So treat it like a database!!
- Will need to run the Catalog Purge utility. How often?
- The ACBGEN process
 - How do you handle fall back plans?

Summary



- 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

 - IMS 14 enhances the use of Catalog
 - Dynamic database management
 - IMS Managed ACBs

 - **It's simple to implement but ...**
 - **Remember it's an IMS database so treat it like a database!!**

Thank You!

