Db2 for z/OS and DFSMS for the DBA

Hendrik Mynhardt
mynhardt@us.ibm.com

IBM

Agenda

- Introduction to DFSMS family
- DFSMS Constructs for DBAs
- DFSMS - how it affects Db2 pagesets
- IBM FlashCopy Overview
- Q/A

IBMs DFSMS Family

DFSMS and related components

Data **F**acility **S**torage **M**anagement **S**ubsystem (DFSMS) has the following components:

- DFSMSdfp - a base element of z/OS
- DFSMSdss - optional feature of z/OS
- DFSMShsm - optional feature of z/OS
- DFSMSrmm - optional feature of z/OS
- DFSMStvs - optional feature of z/OS

DFSMS – Components

- DFSMS**dfp** – A data and space management tool to provides storage, data, program, and device management. It is comprised of programs like IDCAMS, SMS, ISMF etc.
- ** DFSMS**dss** – Provides data movement, copy, backup, and space management functions – known as program ADRDSSU
- ** DFSMS**hsm** – A management and productivity tool for managing low-activity and inactive data. Provides backup, recovery, migration, and space management functions. It does invoke DFSMSdss for certain of its functions.
- DFSMS**rmm** - To manage your removable media and provides management functions for tapes
- DFSMS**tvs** - Enables batch jobs and CICS online transactions to update shared VSAM data sets concurrently.

** Used by Db2 utilities for fast replication functions

DFSMSdss and DFSMShsm basics

- DFSMSdss - Data Set Services (dss) component is a disk storage management utility. It can be invoked using ISMF or by running a batch job
- It allows you to:
  - COMPRESS datasets
  - CONVERTV – converts volumes to and from SMS managed
  - COPY – move datasets, volumes etc. from one to another
  - DUMP – Dump data to tape or disk etc.
- Db2 utilities uses – E.g. Db2 FCIC uses it to drive FlashCopy
- Below an example to delete logs not referenced for 15 days

```
//STEP1    EXEC  PGM=ADRDSSU,REGION=4M
//DASD0002 DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 DUMP DATASET(INCLUDE(DB9AU.ARCHLOG%.**) -
 BY(REFDT LT *,-15)) -
 OUTDDNAME(DASD0002) DELETE PURGE
```

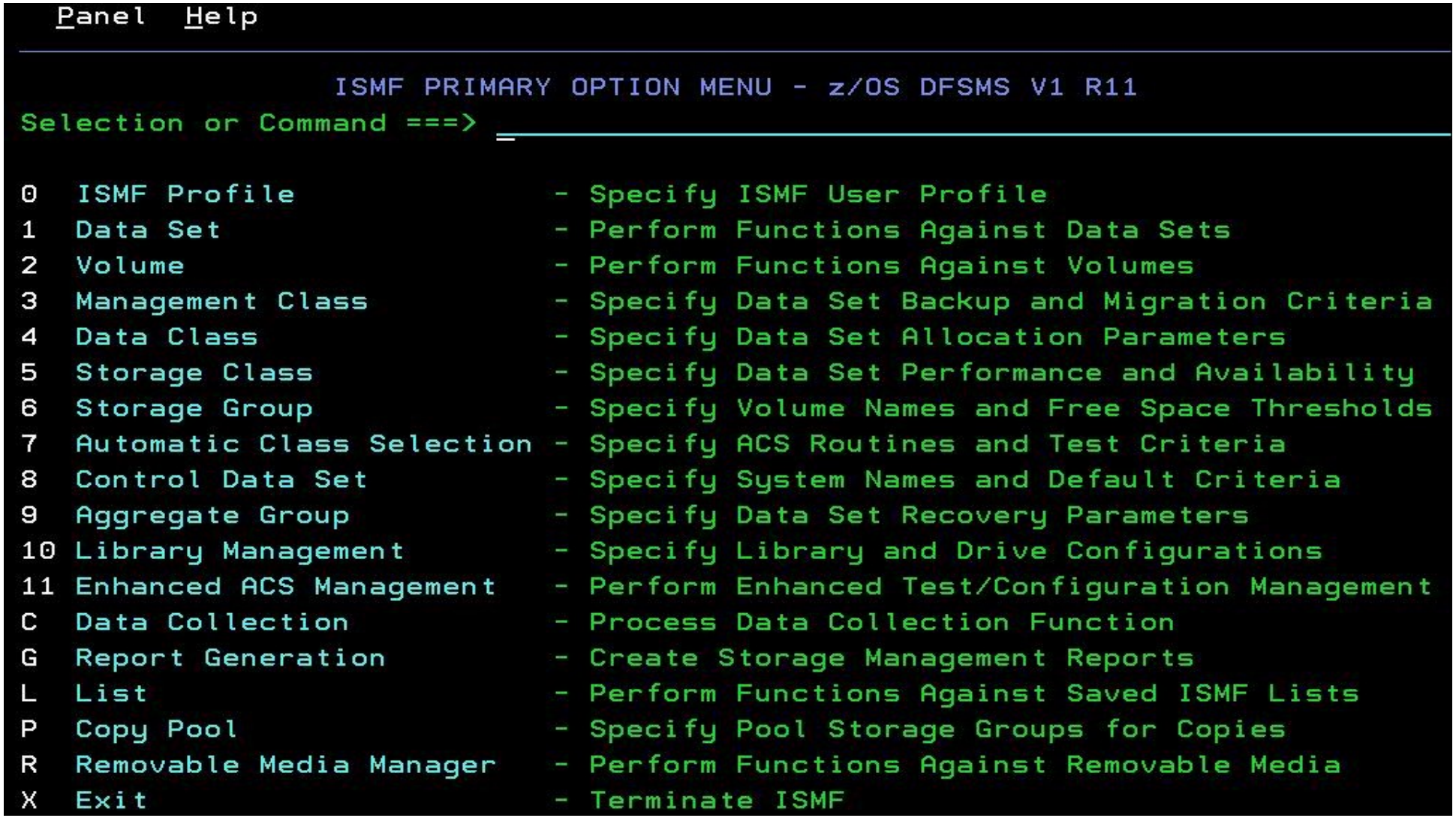
DFSMSdss and -hsm for Db2 Utilities

DFSMSdss

The following Db2 utilities can invoke DFSMSdss copy function:

- CHECK DATA with SHRLEVEL CHANGE
- CHECK INDEX with SHRLEVEL CHANGE
- CHECK LOB with SHRLEVEL CHANGE
- COPY with FLASHCOPY YES or FLASHCOPY CONSISTENT
- LOAD with FLASHCOPY YES or FLASHCOPY CONSISTENT
- REBUILD INDEX with FLASHCOPY YES or FLASHCOPY CONSISTENT
- RECOVER with FLASHCOPY YES or FLASHCOPY CONSISTENT
- REORG INDEX with FLASHCOPY YES or FLASHCOPY CONSISTENT
- REORG TABLESPACE with FLASHCOPY YES or FLASHCOPY CONSISTENT

DFSMS**dfp** – ISMF Menu

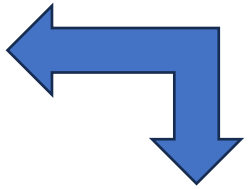


- Only seen with Storage Administrator view

DFSMSdss – building a simple *copy* job

DFSMSdss – building a *copy* job

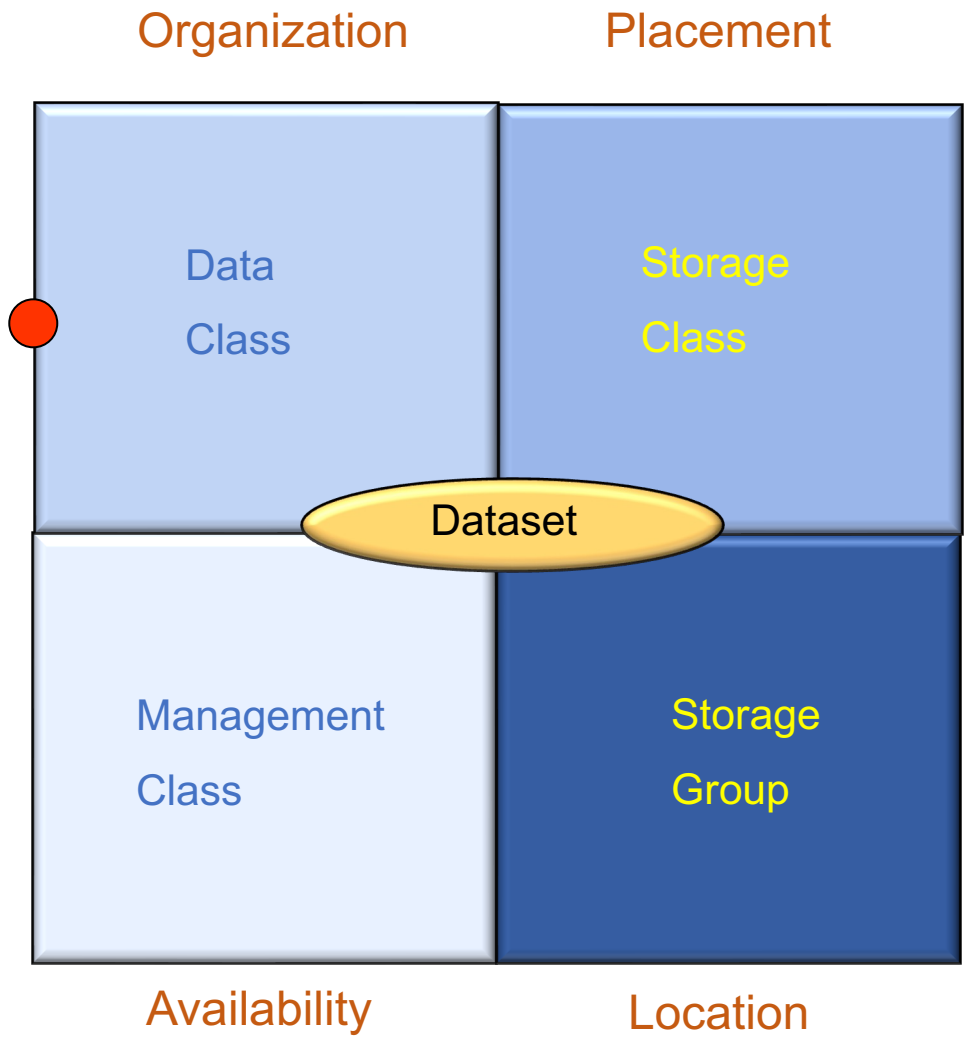
```
                          COPY ENTRY PANEL                      Page 2 of 8
Command ===> ____________________________________________________________

Optionally Specify one or more for
  Data Set: ADLAB.MASTER.INSTALL
    Source Data Set Password . . . _                  (Password or blanks)

  Select Disposition of Source Data Set after Successful Copy:
    3  1. Keep
       2. Uncatalog
       3. Uncatalog and Scratch

  Enter "/" to select option
    _   Concurrent Copy, specify  __   (AP, CP, VP, AR, CR, VR, N, S o
    _    If Concurrent Copy, Notifyconcurrent
    _   DFSMSdss Administrator Mode              _  FCnocopy
    _   FCtoPPRCprimary                          _  FCnocopytocopy

  Debug(FRmsg) . . . . . _   (M=Minimal, S=Summarized, D=Detailed o
  Fastreplicate  . . . . _   (R=Required, P=Preferred, N=None or bl
```



```
//STEP1  EXEC  PGM=ADRDSSU,REGION=2048K
//SYSPRINT  DD  SYSOUT=*
//SYSIN    DD    *
 COPY DATASET( -
       INCLUDE( -
             ADLAB.MASTER.INSTALL -
             )) -
     CANCELERROR -
     CATALOG -
     DELETE PURGE -
     INCAT(CATALOG.PRODUCTS.UCAT) -
     TGTALLOC(SOURCE) -
     FASTREP(PREF) -
     WAIT(2,2)
```

SMS constructs for the Db2 DBA

SMS Construct Relationship



Data Class – what is it?

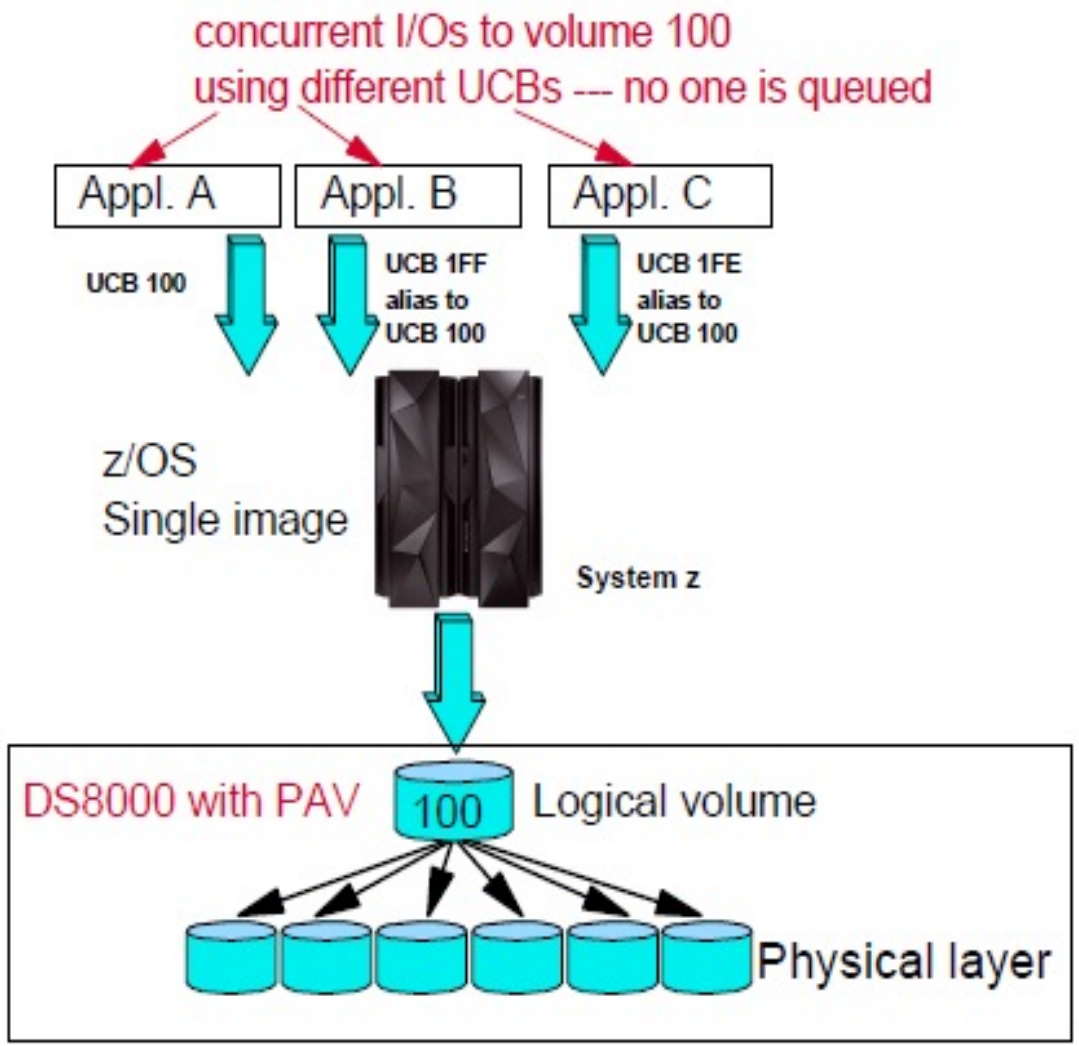
- Data classes are used to define model allocation characteristics for data sets.
- Some of the most common reasons to use a Data Class in a Db2 environment include:
  - Enabling EF and/or EA;
  - Bypassing the 255 extent rule for data sets;
  - Bypassing the 5 extent rule for allocations of data sets;
  - Reducing space requirements when no volume meets the space requirement;
  - Allocating data sets with common DCB and/or space characteristics; Specifying additional volumes for Db2 or utility data sets;
  - Specifying different data sets types, such as PDSE, large format, etc.

Extended Address Volumes - EAV

- An Extended Address Volume (EAV) is a volume with more than 65,520 cylinders.
- Only 3390 Model A devices can be EAV.
  - EAV is supported by all z/OS versions
  - The size is limited to 223 GB (262,668 cylinders)
- Why was this needed at the time?
  - Running out of z/OS addressable disk storage. The four-digit device number limit (actually 65,280 devices) is fast approaching and each volume is limited to about 54 GB
  - Define larger volumes by increasing the number of cylinders beyond 65520

What is PAV - Parallel Access Volumes

- **PAV**s allow simultaneous access to logical volumes by multiple users or jobs from **one** system.
- Reads are simultaneous
- Writes to different domains are simultaneous
- Writes to same domain are serialized
- Eliminates or sharply reduces IOSQ
- High I/O activity, particularly to large volumes (3390 mod 9, 27, and 54) greatly benefits from the use of PAV.



What is DFSMSdfp – e.g. List a Data Class

What is DFSMSdfp – E.g. List a Data Class

```
                          DATA CLASS DISPLAY                    Page 1 of 5
Command ===> _____________________________________________________________

CDS Name  . . . : ACTIVE
Data Class Name : DB2EDC

Description : DATA CLASS FOR DB2 CATALOG AND DIRECTORY DATA SETS

Recfm  . . . . . :
Lrecl  . . . . . :
Override Space . . . . . : NO
Space Avgrec . . . . . :
```

Can you override the attributes

```
                          DATA CLASS DISPLAY                    Page 2 of 5
Command ===> _____________________________________________________________

CDS Name  . . . . . : ACTIVE
Data Class Name . . : DB2EDC

Data Set Name Type  . . . . . : EXTENDED
  If Extended . . . . . : PREFERRED
  Extended Addressability . . : YES
  Record Access Bias  . . . . : USER
Space Constraint Relief . . . : NO
```

EA provides datasets > 4GB

DFSMS Data Class

```
                                DATA CLASS DISPLAY
Command ===>

CDS Name  . . . : ACTIVE
Data Class Name : DB2EDC

Description : DATA CLASS FOR DB2 CATALOG AND DIRECTORY DATA SETS

Recfm  . . . . . . . . . :
Lrecl  . . . . . . . . . :
Override Space . . . . . : NO
Space Avgrec . . . . . . :
Retpd Or Expdt . . . . . :
Volume Count . . . . . . : 1
Add'l Volume Amount  . . :
Data Set Name Type  . . . . . : EXTENDED
  If Extended . . . . . . . . : PREFERRED
  Extended Addressability . . : YES
  Record Access Bias  . . . . : USER
Space Constraint Relief . . . : NO
```

Data Class when creating a STOGROUP

Can a dataset be allocated in extended format

```
Data Set Name Type  . . . . . : EXTENDED
  If Extended . . . . . . . . : PREFERRED
  Extended Addressability . . : YES
  Record Access Bias  . . . . : USER
Space Constraint Relief . . . : NO
```

DFSMS Data Class and Db2

Data Class and Db2:

- Old history - The Db2 Catalog and Directory were user-defined
  - Used to define the cluster with only a single volume
- With Data Class you can specify:
  - Use **VOLUME COUNTs**
  - Use **EF** (Extended Format):
    - Use this for >4GB datasets (LOBs, XML, large datasets)
    - For > 255 Extents
- With Data Class you also bypass the DFP 255 extent rule
  - 123 extents / volume, up to 59 volumes (123 * 59 = 7,257 extents)

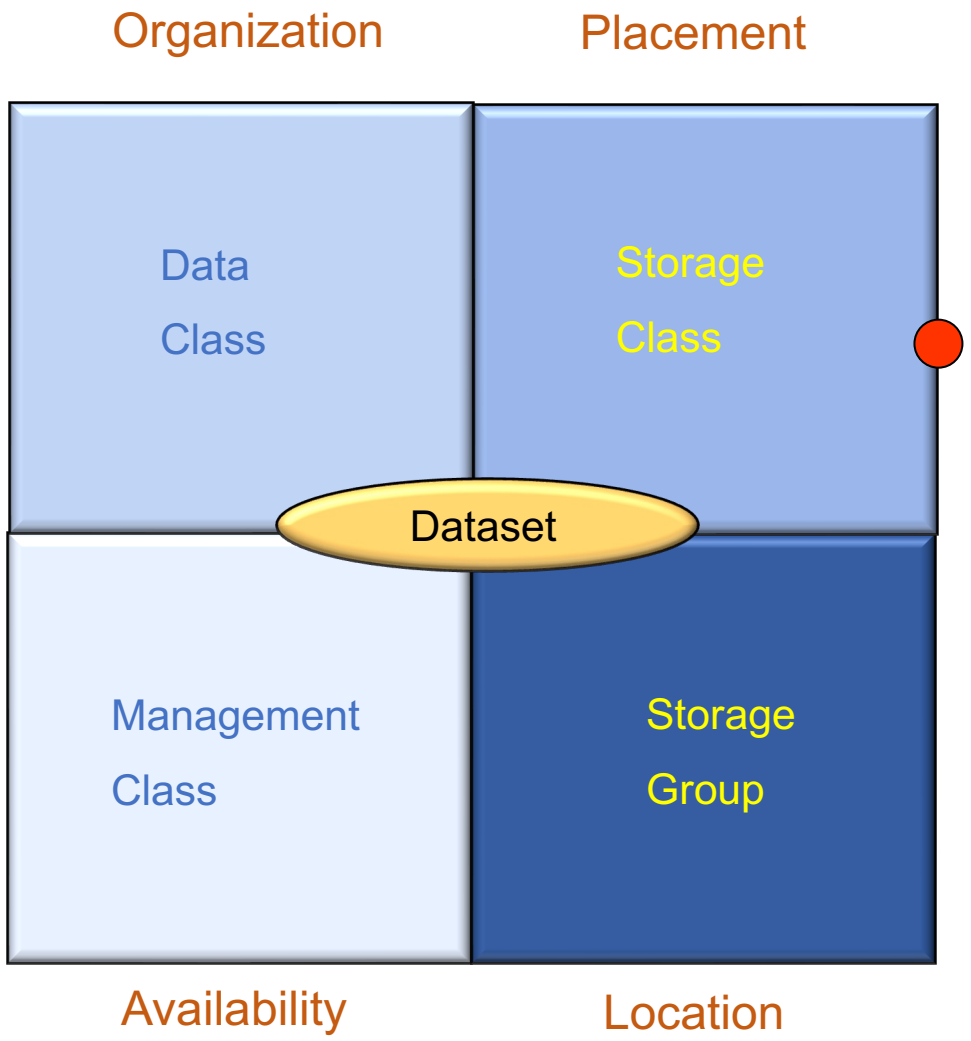
DFSMS Data Class and Db2

Db2 Catalog and Directory

- Enable EF and EA
- Provides for a dynamic volume count.
  - The amount is installation specific, but generally 2 or volumes should be sufficient. This would be removed when Db2 Catalog and Directory data sets are eventually converted to Db2 managed.
- Set Extent Constraint Removal to YES
  - Removed the extents - i.e. >255

SMS constructs for the Db2 DBA

SMS Construct Relationship



SMS Storage Class

A storage class is a collection of performance goals and availability requirements that you define. The storage class is used to select a device to meet those goals and requirements

- Enable the Guaranteed Space attribute
  - Allows the volumes on which an SMS data set is to reside to be chosen explicitly when the data set is created.
  - If GUARANTEED SPACE is used and if any of the volumes specified are not part of one of the storage groups to which the storage class maps, the data set creation fails

SMS Storage Class

- Prior to SMS-managed storage, you specified and/or separated important datasets to specific volumes manually for performance

- The Storage Class construct allows you to:
  - Specify the ***performance characteristics***
  - ***Striping*** in combination with Data Class EF

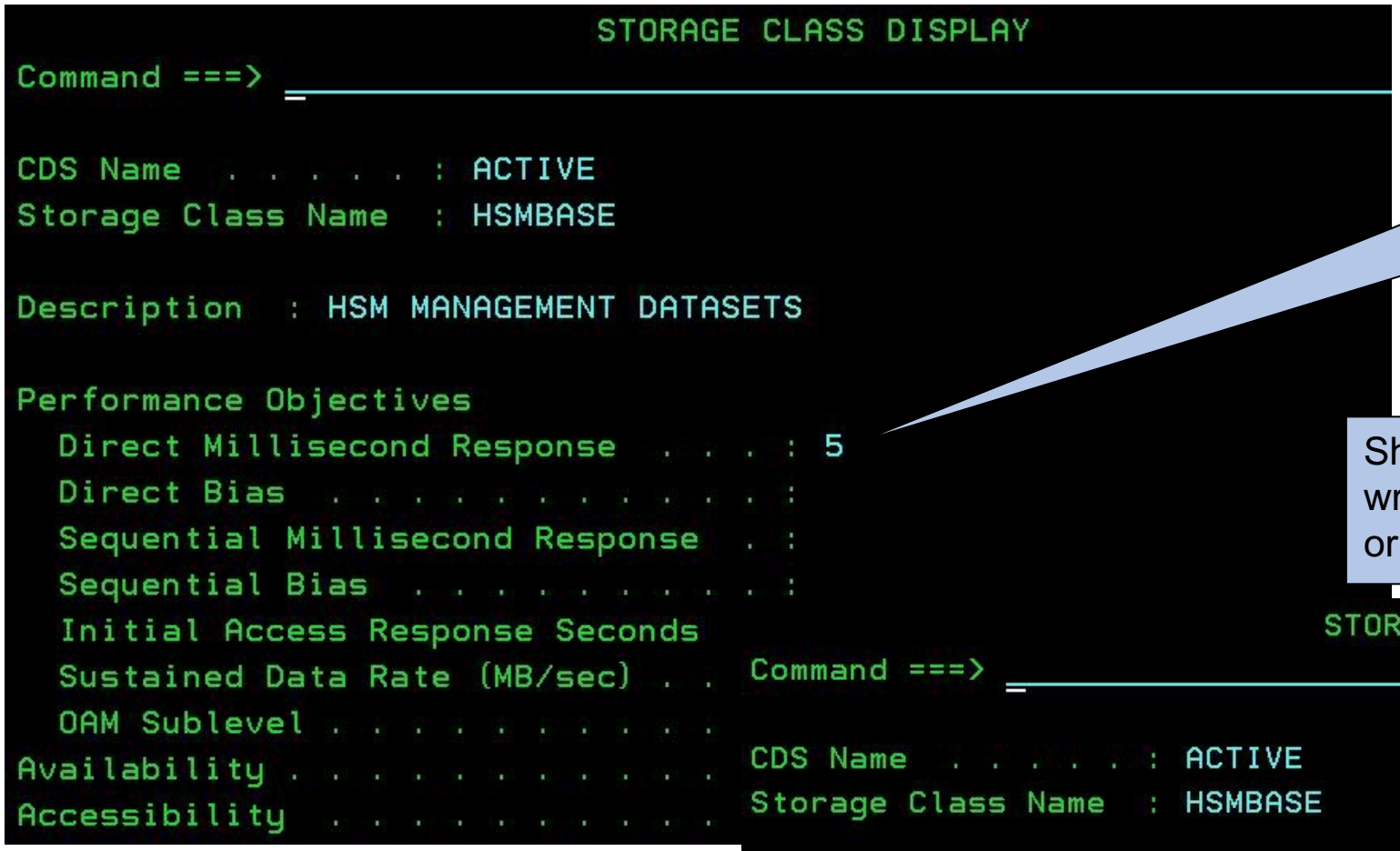
  - Use of ***guaranteed space***
    - Use a Separation Profile for LOGCOPY1 and 2 (disk controller level)
    - Use Separation Profile to specify datasets on different volumes

SMS Storage Class – Info in ISMF

Above is a list of Storage Classes from ISMF

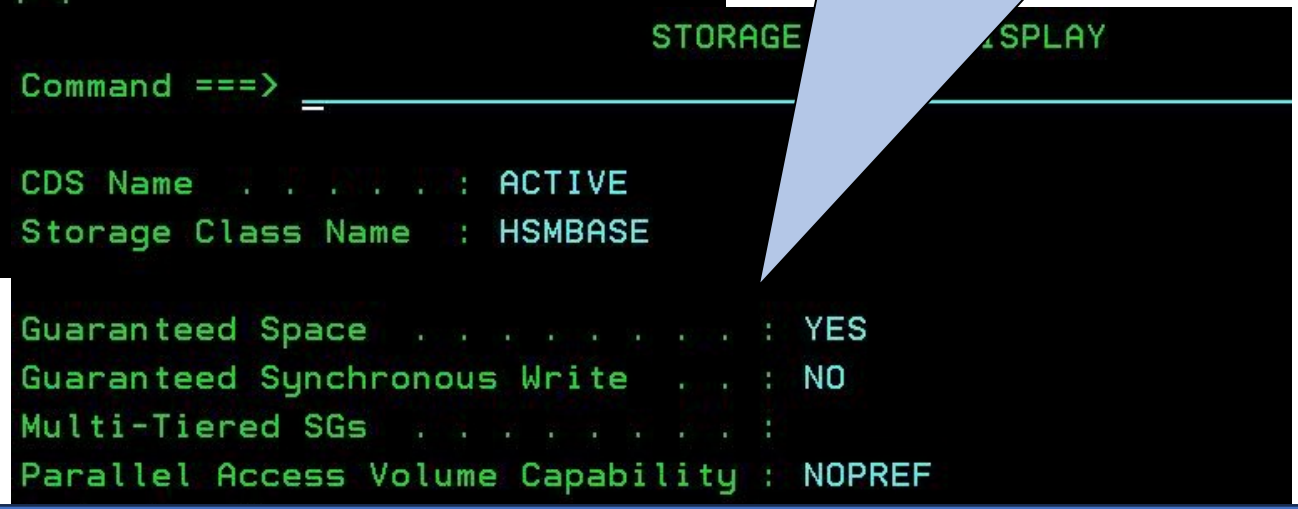
The Guaranteed Space attribute denotes whether a user can pre-allocate space for multi-volume datasets.

SMS Storage Class – **performance** clauses



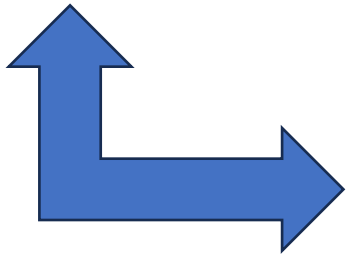
Shows the direct access response time required for data sets in this storage class to write a block of data

Should the system check (wait) prior to writing out to the disk – Sync (after - NO) or Async (before – YES)



Db2 and DFSMS relationship

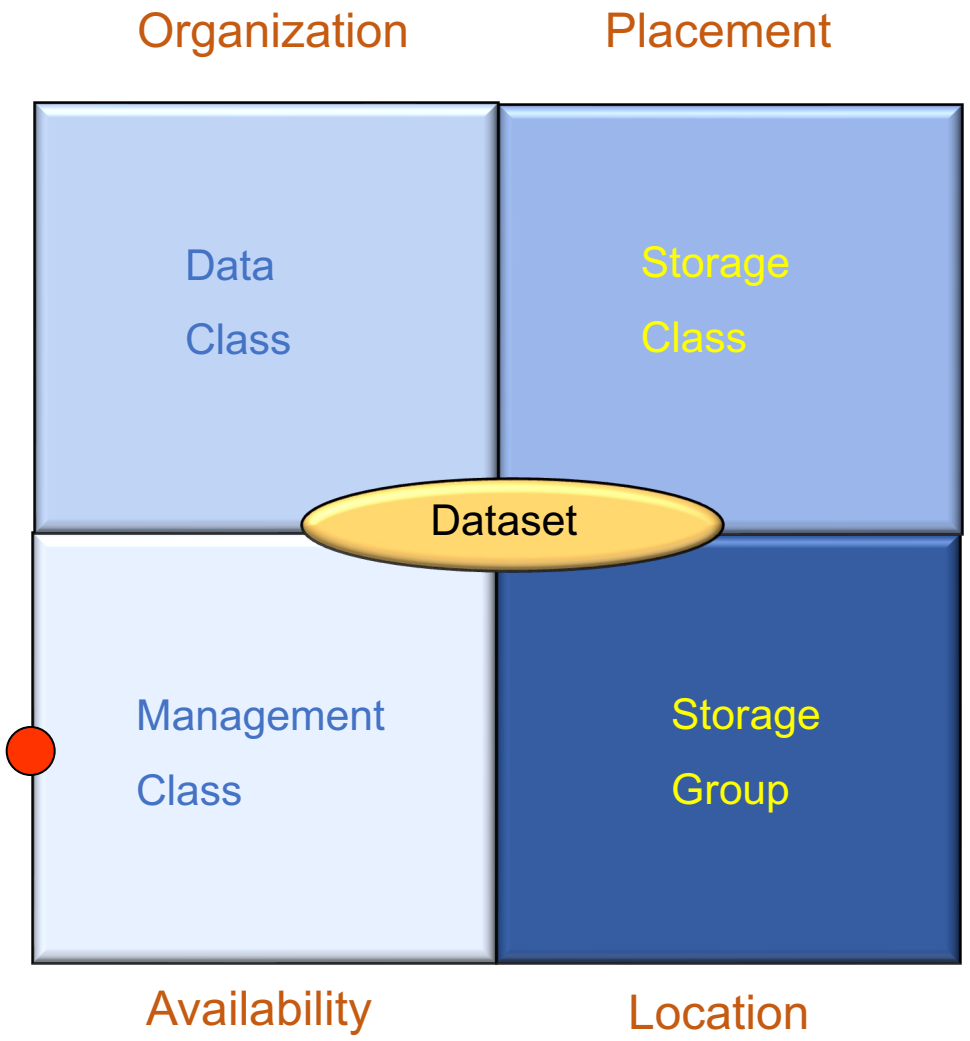
```
CREATE STOGROUP STOSMS1
  VOLUMES("*")
  VCAT Db2
    DATACLAS Db2EDC
    MGMTCLAS DMMGMT
    STORCLAS Db2DASD;
```



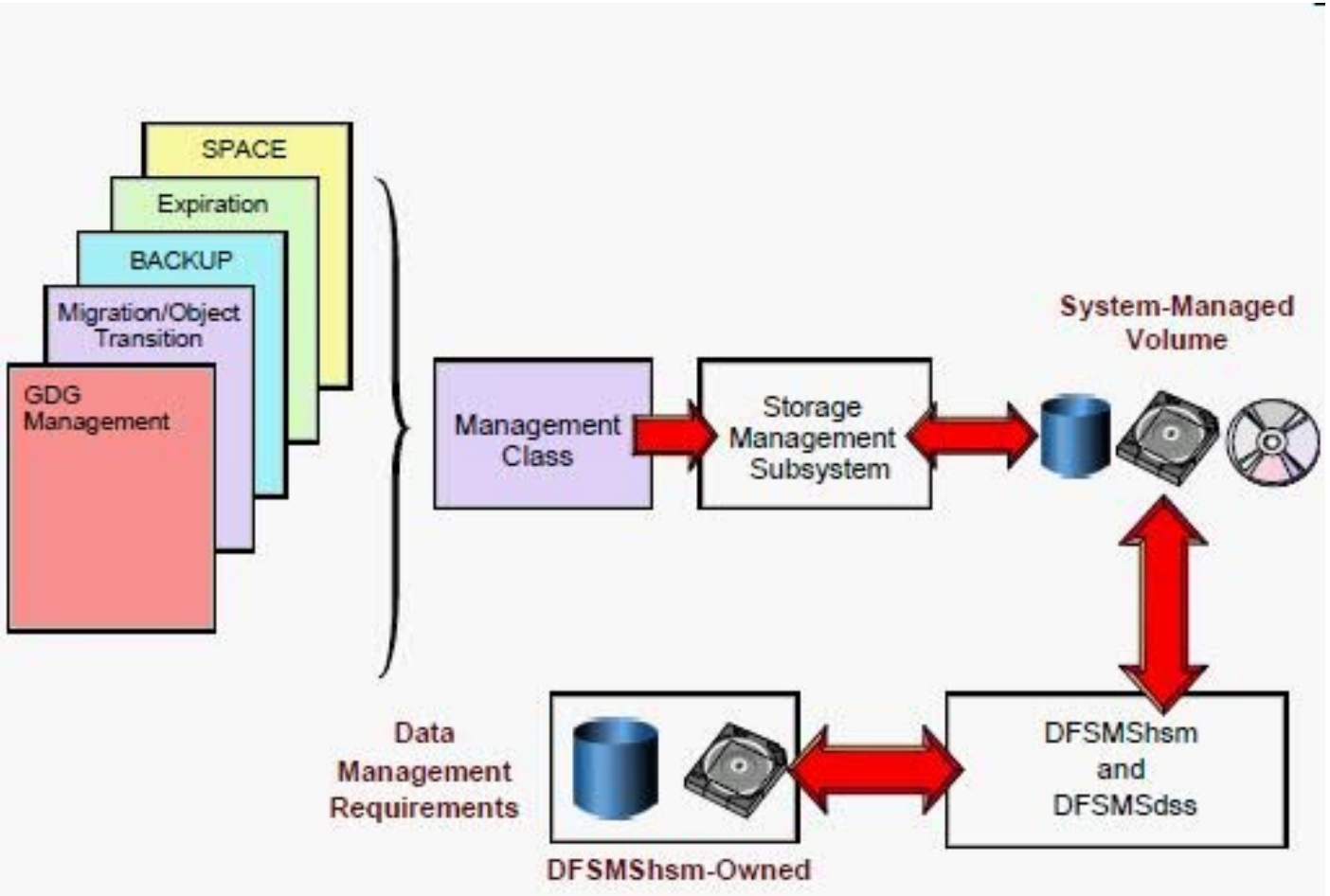
```
DB2 Admin ------- DB1S Interpretation of an Object in SYSSTOGROUP -
Option ===> _

Details for storage group  : STOSMS1

Storage group owner        : DNET963
Created by                 : DNET963
Created timestamp          : [redacted]-25-10.25.05.401281
Altered timestamp          : [redacted]-25-10.25.05.401281
Creator type               : Auth ID
VSAM catalog name          : DB2
DASD space allocated (KB)  : -1    (Float: -1.00000000000000E+00)
Last update of space field :        (yyddd)
STOSPACE last executed     : 0001-01-01-00.00.00.000000
SMS Data class             : DB2EDC
SMS Management class       : DMMGMT
SMS Storage class          : DB2DASD
Created in DB2 Version     : M - DB2 V9
```

SMS constructs for the Db2 DBA

SMS Construct Relationship



SMS Management Class



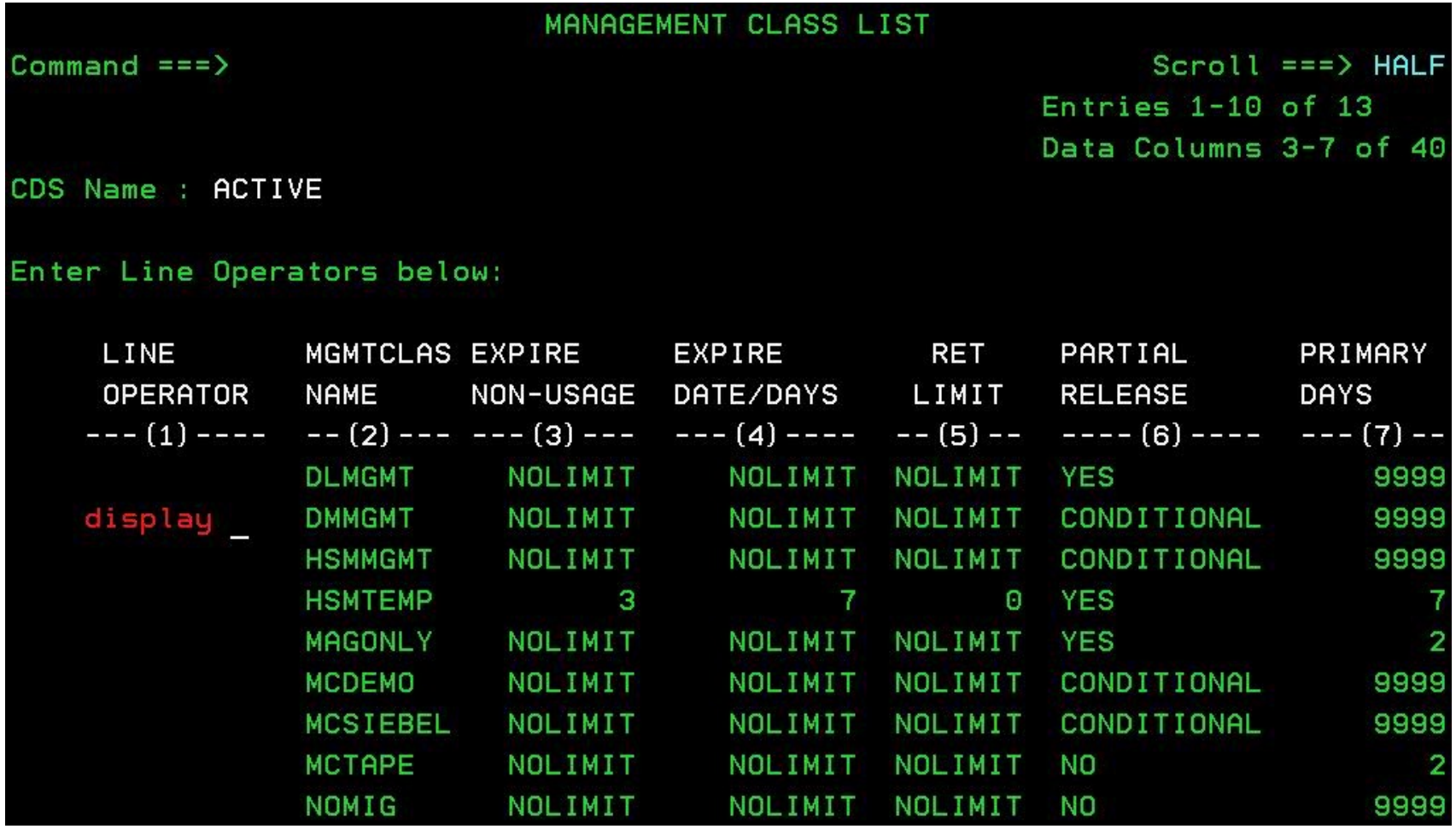
Management classes let you define management requirements for **individual data sets**, rather than defining the requirements for entire volumes

SMS Management Class

- Prior to SMS Managed Class, *HSM* managed the data sets at volume level, applying a standard management criteria for all data sets on a given volume:

  - The SMS Management Class can be used for:
    - Retention
    - Backup and Migration
    - Expiration
    - Management of generation data set groups (GDGs) and their data sets (GDSs)
    - Space release
    - ABARS management
  - Also details the performance requirements for a data set (Response times)

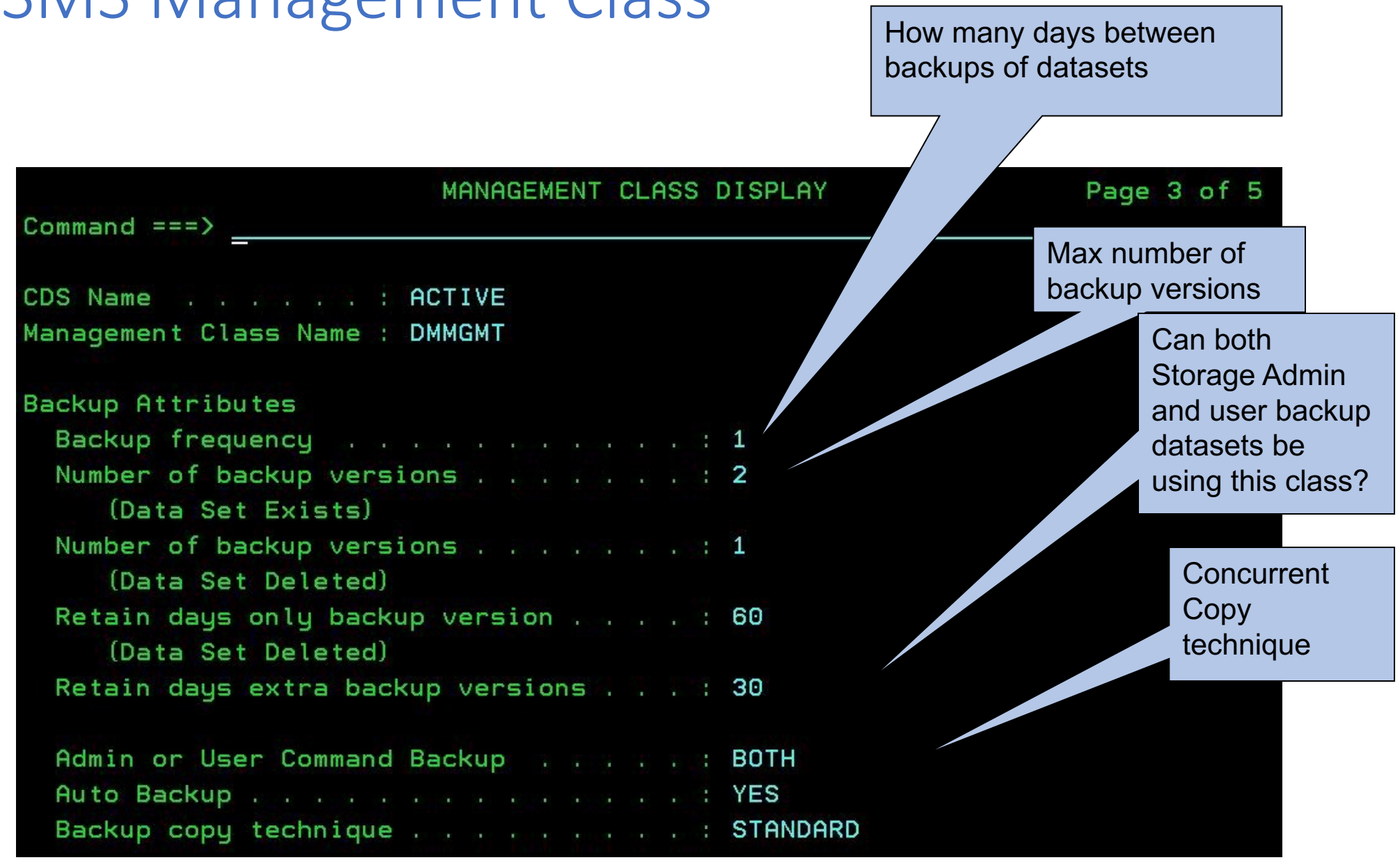
SMS Management Class



```
                          MANAGEMENT CLASS LIST
Command ===>                                                 Scroll ===> HALF
                                                          Entries 1-10 of 13
                                                          Data Columns 3-7 of 40
CDS Name : ACTIVE

Enter Line Operators below:

     LINE        MGMTCLAS EXPIRE     EXPIRE          RET    PARTIAL       PRIMARY
     OPERATOR    NAME     NON-USAGE  DATE/DAYS       LIMIT  RELEASE       DAYS
    ---(1)----   --(2)--- ---(3)---  ---(4)----      --(5)-- ----(6)----   ---(7)--
                 DLMGMT     NOLIMIT     NOLIMIT     NOLIMIT  YES              9999
    display _    DMMGMT     NOLIMIT     NOLIMIT     NOLIMIT  CONDITIONAL      9999
                 HSMMGMT    NOLIMIT     NOLIMIT     NOLIMIT  CONDITIONAL      9999
                 HSMTEMP          3           7           0  YES                 7
                 MAGONLY    NOLIMIT     NOLIMIT     NOLIMIT  YES                 2
                 MCDEMO     NOLIMIT     NOLIMIT     NOLIMIT  CONDITIONAL      9999
                 MCSIEBEL   NOLIMIT     NOLIMIT     NOLIMIT  CONDITIONAL      9999
                 MCTAPE     NOLIMIT     NOLIMIT     NOLIMIT  NO                  2
                 NOMIG      NOLIMIT     NOLIMIT     NOLIMIT  NO               9999
```

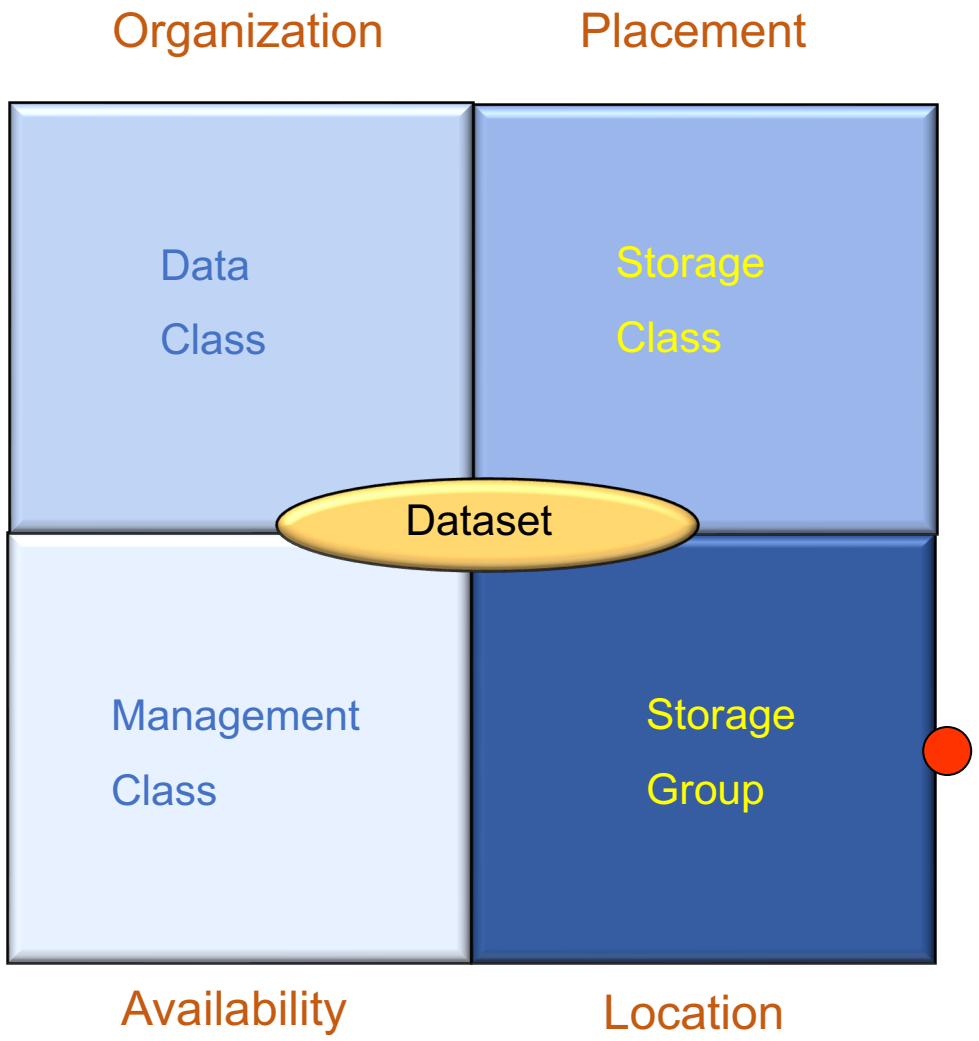
Will unused space be released – only for VSAM EF datasets

SMS Management Class



SMS constructs for the Db2 DBA

SMS Construct Relationship



Differences between a SMS Storage Group and Db2 STOGROUP

Db2 STOGROUP	SMS Storage Group
Different STOGROUPS share the same DASD volumes	A disk volume can only belong to ONE SMS Storage Group
The VOLSERs are specific	Consider coding VOLUMES("*"). This will enable SMS management. Consider avoiding Guaranteed Space and specific VOLSERs where possible. This defeats the purpose of SMS.
SYSIBM.SYSVOLUMES contains one row for each volume in the VOLID column	SYSIBM.SYSVOLUMES contains an "*" for each volume in the VOLID column. This is true when your specified VOLUMES("*").
Limited to manage up to 133 volumes	No limit
The volume selection is based on free space	The volume selection is based upon SMS algorithms

Db2 and SMS – Storage Groups

DFSMS Storage Groups and Db2 recommendation:

- Db2 Catalog and Directory
- Active log and boot strap data sets
- User table spaces and indexes

  Work files (DSNDB07 and equivalent) can reside in their own Storage Group. An alternative if a sufficient amount of addresses are assigned using PAV, is to combine the work files in the same Storage Group as the ones used by the Db2 Catalog and Directory or the user table spaces and indexes
- Archive log data sets if not created on tape
- Image copy data sets if not created on tape

Db2 and SMS – Storage Groups

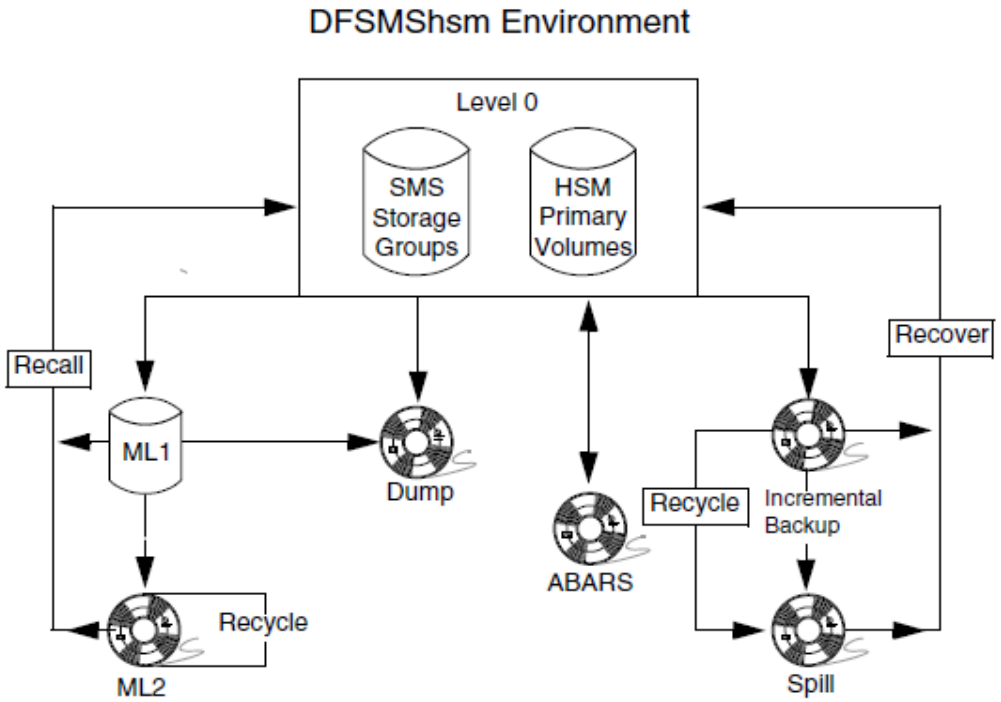
- SMS Storage Group has a Guaranteed Space attribute
  - Think of it as "guaranteed volsers"
  - When allocating onto multiple volumes, the PRIQTY is allocated and not the SECQTY
    - E.g. 100 prime, 10 secondary
      - When dataset extents onto second volume it will use 100 and not 10
      - APAR PK83108 (old) resolved some of the problem when the object has Guaranteed Space enabled with specific entries in the Db2 Stogroup. In this case if a volume in the Db2 Stogroup does not have enough space, it will redrive the request onto the next volume
  - ZPARM SVOLARC – allows archive log to reside on one volume – without this, Db2 may request large allocation up to 15 volumes at the same time

Db2 and SMS – SMS Storage Groups

- ***Alternative*** to Guaranteed Space is to use **Separation Profile**
- Separate specific data sets onto different disk controllers during allocation.
  - For example, LOGCOPY1 and LOGCOPY2 data sets must be allocated on different disk controllers.
- Separate specific data sets on different volumes to avoid hot spots. The volume separation, as opposed to the controller separation
- Striping (also part of **Storage Class**)
  - Must have DSVCI = YES (page size = CI size to avoid partial writes)

What is DFSMShsm

Hierarchical Storage Manager (DFSMShsm) is a disk storage management and productivity product for managing low activity and inactive data



- ML0 (Migration level 0) is the on-line data that is accessed by applications and users.
- ML1 is a dedicated pool of disks, which are non-SMS managed.
- If a dataset continues to be unused, it will eventually be migrated off to ML2 (migration level 2), which is usually high capacity cartridge

What is DFSMShsm

Functions of interest:

- **Incremental backups**
  - HSM takes a copy of a dataset if it changed since the last backup – similar to Db2
- **Full Volume Dumps**
  - By invoking -DSS, -HSM dumps the volume based upon a *Dump Class* (daily, weekly etc)
- **Aggregate backup**
  - *ABARS* – a process of backing up user defined groups of datasets
  - Disk, tape or DFSMShsm volumes
- **Recovery**
  - Can be individual datasets or physically a full volume
  - Uses DFSMSdss

Db2 and SMS – SMS Management Class

- FlashCopy COPY POOL BACKUP – usage of pool of volumes can be used for:
  - DFSMShsm *migration*
  - DFSMShsm *full volume dump*
  - DFSMShsm *incremental backup*

SMS and Db2 – PDS and PDSE

What is a PDSE?

- Like a PDS:
  - Each member name is 8 bytes long
  - Has a directory
  - Can be created via JCL etc
- Also different to a PDS:
  - Can have 123 extents (PDS only 16)
  - Has no limit on # of directory blocks
  - Does not require compression to consolidate space for reuse
- Libraries ADSNLOAD, ADSNLOD2, SDSNLOAD and SDSNLOD2 are delivered as PDSE

SMS and Db2 objects

Active logs and BSDS recommendation

- Separate Logs and BSDS onto their own volumes for performance and recovery
- Data Class:
  - Enable EF if striped – not recommended for Db2 logs
  - Set Compaction=YES if compressing the archive log data sets.
- Storage Class:
  - Assign Guaranteed Space to both or a Separation Profile
- Management Class:
  - Assign datasets to a class with no actions – MCNOACT
  - If migrating the archive log data sets, assign a Management Class to migrate to ML1, then eventually to ML2 if desired, or directly to ML2.
- Storage Groups:
  - Define a separate Storage Group for above
  - Assign a Copy Pool backup if planning to use Flashcopy

SMS and Db2 objects

Db2 Catalog and Directory

- EF/EA enabled
- Data Class:
  - Enable EF and EA
  - Set Extent Constraint Removal to YES (Consolidates adjacent extents for Db2 LDS - VSAM data sets when extending on the same volume)
  - Automatic and requires no action on your part
- Storage Group
  - Define one specifically for the Db2 Catalog and Directory
  - Assign Copy Pool backup SG for FlashCopy

SMS and Db2 objects

Image Copies on disk

- Data Class
  - LARGE or EF must be used If the data set is greater than 4,369 cylinders
  - If the image copy data sets are striped, EF must be used.
  - Volume Count and Dynamic Volume Count can be used if image copy data sets are requested as multi-volume.
  - Set Space Constraint Relief to YES
- Storage Group
  - Should be for the subsystem or Data Sharing group only and not share with others.

SMS and Db2 objects

Db2 application pagesets

- Recommend this to be SMS managed
- Data Class:
  - Enable EF and EA
  - Set Extend Constraint Removal to YES
- Management Class – Consider MCNOACT
  - Management Class that takes no action, no migration, nor backups
- Storage Group:
  - Define separate Storage Group
  - Assign Copy Pool for FlashCopy
  - Use Separation Profile for volumes – avoid known hot spots

Copy Pool Storage Group

- SMS construct, consists of SMS storage groups.
  - Versions attribute allow specification of the number of copy versions to be maintained on DASD (max is 85).
  - Each version is a complete set of the source DASD, so greater than 2 are unlikely – 1 is okay
  - Advanced capabilities of FlashCopy are specified on the ISMF Copy Pool backup storage group definitions.
- Each Db2 system or data sharing group has two HSM copy pool backup storage groups with prescribed Db2 naming convention (30 bytes in length)
    - DATABASE COPYPOOL (DSN$location_name$DB)
    - LOG COPYPOOL (DSN$location_name$LG)

FlashCopy Basics

- Volume and dataset-level FC
  - All types of data sets are supported (sequential, partitioned, VSAM data sets)
  - No volume size restriction
  - No location restriction – can copy to same volume
- At any point-in-time, a volume, or a data set, can be only a source or a target

IBM Fast Replication / FlashCopy basics

FlashCopy options

- Volume and Dataset level FC
- Copy and NoCopy FC
- Incremental FC
- Space Efficient FC
- Consistency Group FC
- Remote Pair FC
- Fast Reverse Restore

FlashCopy basics …

- It can be invoked as follows:
  - DFSMSdss – ADRDSSU batch program
  - TSO
  - ICKDSF – A disk utility that initialize volumes, creates the VTOC etc.

- There are 3 ‘phases’:
  - Establish the relationship
  - Copy the data
  - Terminate the relationship

FlashCopy – How to determine if you can use it

- Produces instant copy of a volume or dataset
- Source and Target volumes require real disk space AND must be on the same DASD ssid
- Source and target volumes must be same track geometry

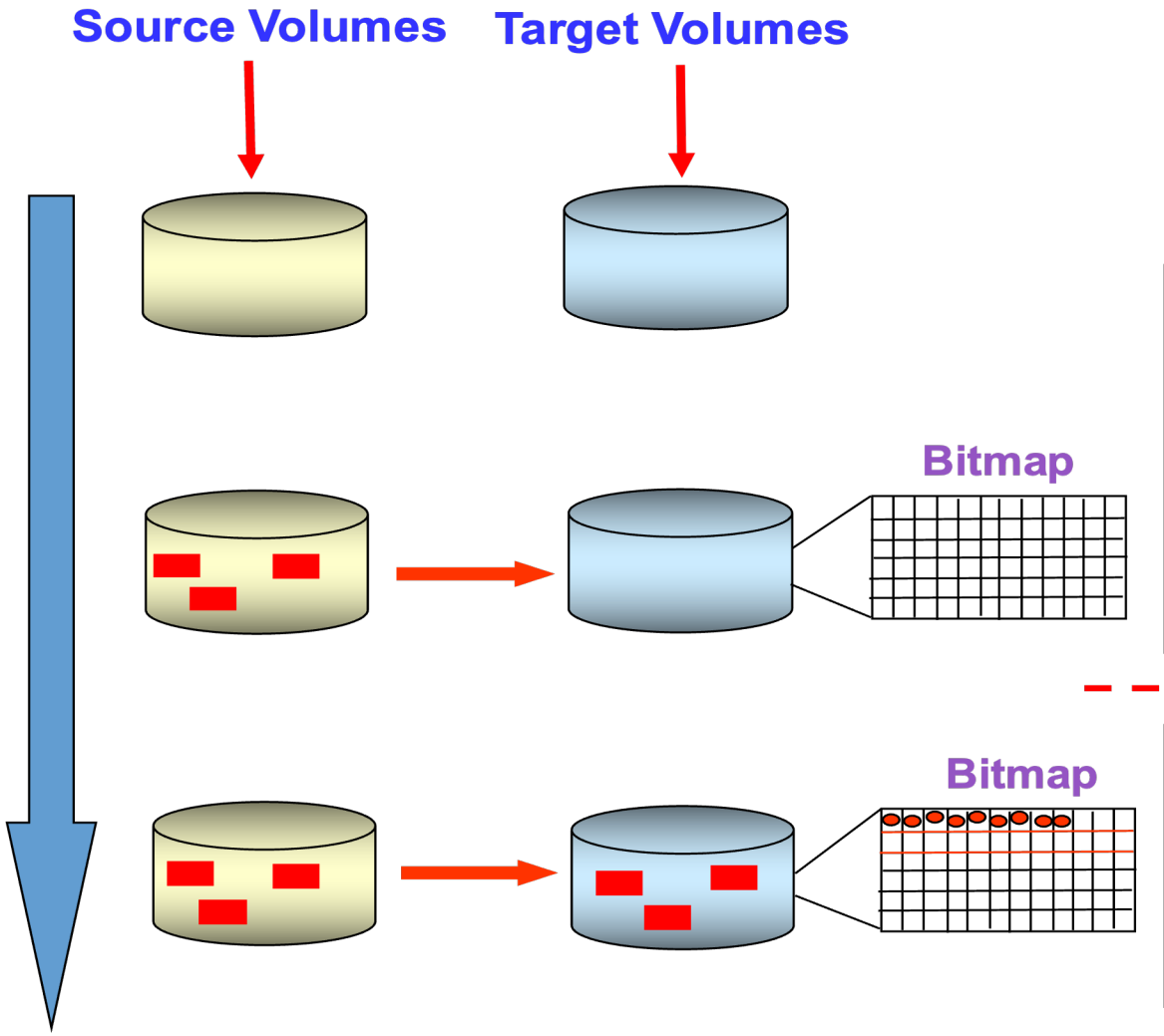
```
//STEP1    EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD    *
    COPY DATASET(   -
            INCLUDE(  -
               DBA.DSNDBC.FLASHDB1.**
                                DB2A.DSNDBC.MYOWNDB.** )) -
                    FASTREPLICATION(PREFERRED)
```

```
ADR711I (001)-NEWDS(01), DATA SET BEPD.DSNDBC.BEDPRDDB.BAADJUDS.I0001.A001 HAS BEEN ALLOCATED
 WITH NEWNAME BEPD.FAST.BEDPRDDB.BAADJUDS.I0001.A001 USING STORCLAS STANDARD, DATACLAS
ADR806I (001)-T0MI (03), DATA SET BEPD.DSNDBC.BEDPRDDB.BAADJUDS.I0001.A001 COPIED USING A FAST
 REPLICATION FUNCTION
```

FlashCopy – basic steps

1. DFSMSdss checks if source and target are eligible for FlashCopy. If not, DFSMSdss will do a normal copy.
2. Once the FlashCopy "Logical Complete" occurs, the DFSMSdss or Db2 utilty job is completed.
   - It does not wait until the copy is physically complete – that is performed within the (which is performed by the ESS/DS8K hardware)
3. Once the copy is physically complete, the relationship between source and target is ended.
4. NOCOPY reserves space (for copied tracks), but does not start the background copy task. Target is used as a cache for updated tracks only. Secondary Relationship stays till terminated or until all source tracks have been copied because they were all updated. For DR tape copies, you would explicitly withdraw it.

FlashCopy Basics



Flashcopy command sent to HSM
- **Relationship established**
- **Bitmap created**
- **Source vol avail for updates**

LOGICAL COPY complete

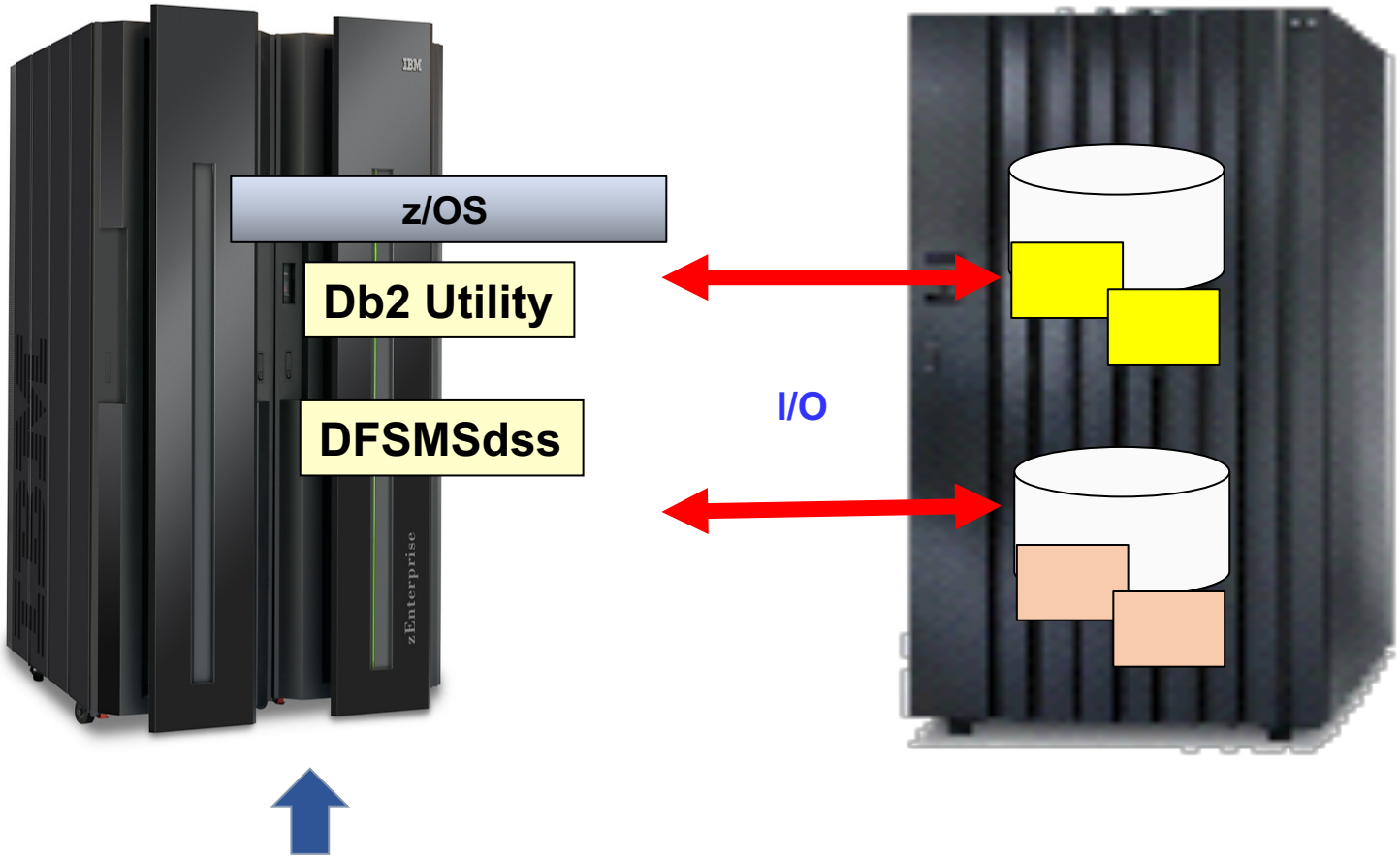
- **Tracks are copied from source to target volumes**
- **Bitmap updated**

PHYSICAL COPY

FlashCopy Basics …

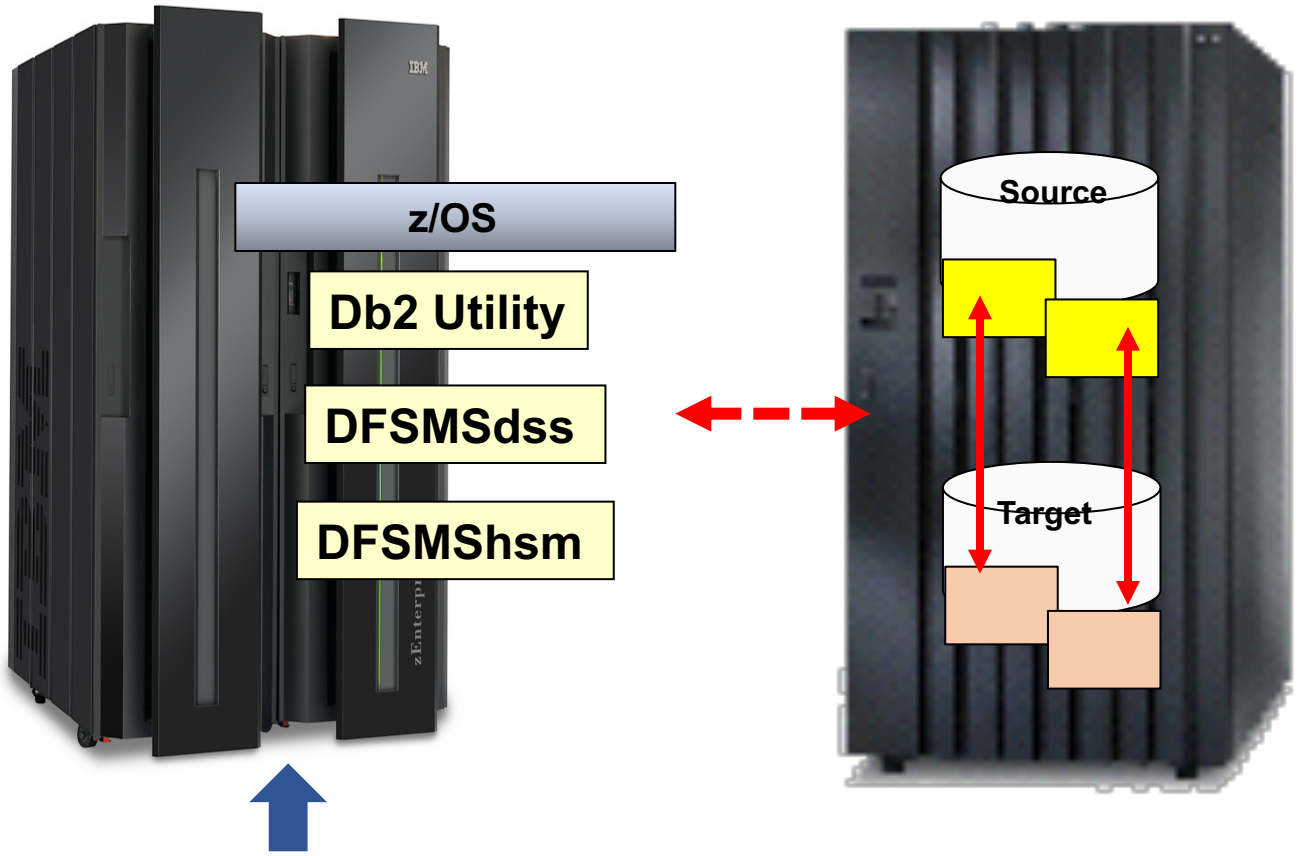
- FlashCopy – Volume and dataset level
  - Volume level used by BACKUP / RESTORE SYSTEM utilities
  - Dataset level used by Db2 FCIC and other utilities
- Incremental – only for volume-based copies
  - Supported by BACKUP / RESTORE SYSTEM
- FlashCopy SE – Space efficient
  - Volume level only
  - Uses ‘virtual’ Space Efficient volumes as target volumes and a repository
  - Only uses the space needed for updates to the source volume

Fast Replication – Host based - non-FC



All I/O and CPU costs are host based and charged

Fast Replication – Storage based – fast replication



An instant copy of a volume/data set at a specific point in time

Data movement (CPU and I/O) offloaded to storage processor

Frees up resources on host processor

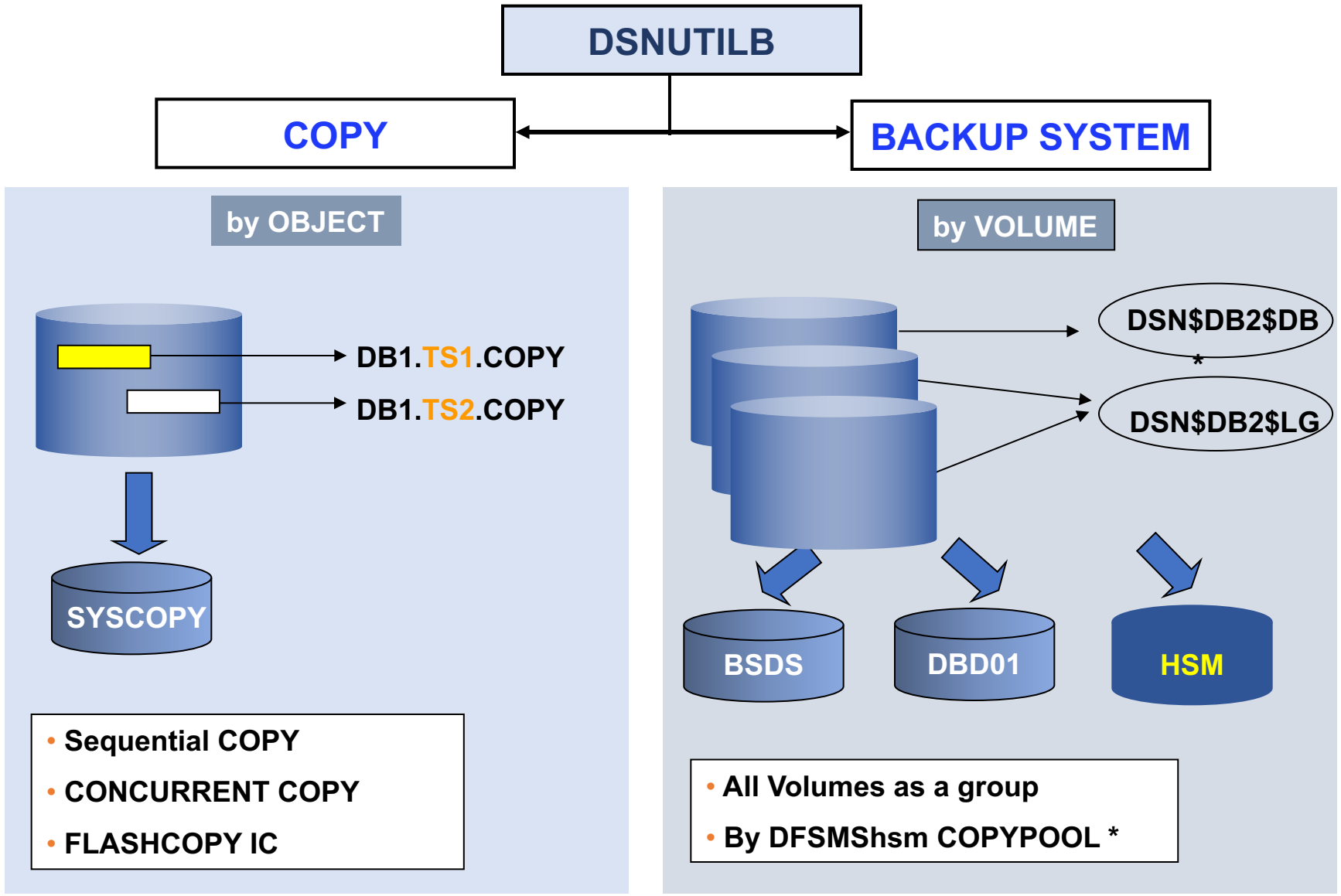
Volume and dataset level FlashCopy on IBM, EMC and HDS DASD

No host or I/O costs for the data movement

Fast Replication uses by Db2

- BACKUP SYSTEM / RESTORE SYSTEM
- Dataset FC support for CHECK INDEX SHRLEVEL CHANGE
- Incremental FC support for BACKUP SYSTEM
- Dataset FC for RECOVER with system-level backup (SLB) as input
- Dataset FC for CHECK DATA SHRLEVEL CHANGE and CHECK LOB SHRLEVEL CHANGE
- Dataset FC for COPY
- Dataset FC for inline copy in REORG TABLESPACE, REORG INDEX, REBUILD INDEX and LOAD
- FC image copies with consistency and no application outage (SHRLEVEL CHANGE)
- FCIC accepted as input to RECOVER, COPYTOCOPY, DSN1COPY, DSN1COMP and DSN1PRNT

Db2 backup utilities – two flavors

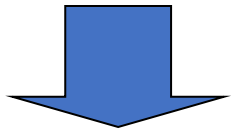


FlashCopy Image Copies - FCIC

- Activated through utility statement FLASHCOPY YES or DSNZPARMs
  - FLASHCOPY_COPY
  - FLASHCOPY_LOAD
  - FLASHCOPY_REORG_TS & FLASHCOPY_REORG_IX
  - FLASHCOPY_REBUILD_IX

```
//UTIL EXEC
DSNUPROC,SYSTEM=DB0B,UID=,A',UTPROC=''
//DSNUPROC.SYSCOPY DD
DSN=DB0BI.DB1.TS1.FC01,
//     DISP=(MOD,CATLG), UNIT=SYSDA,
//     SPACE=(16384,(20,20),,,ROUND)
//DSNUPROC.SYSIN    DD  *
  COPY  TABLESPACE DB1.TS1
```

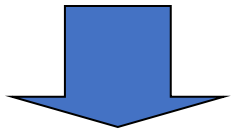
FLASHCOPY_COPY=NO



Sequential file:
DB0BI.DB1.TS1.FC01

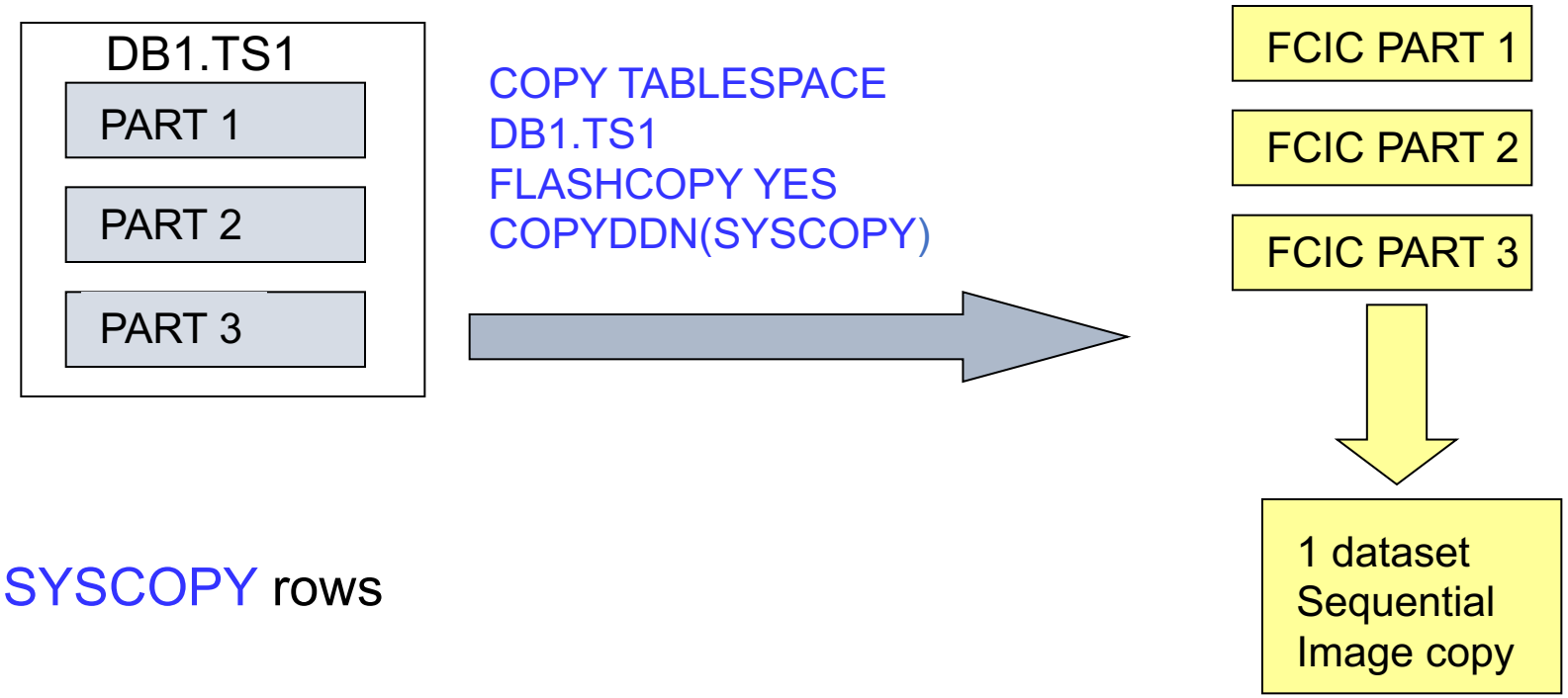
```
//UTIL EXEC
DSNUPROC,SYSTEM=DB0B,UID=,A',UTPROC=''
//DSNUPROC.SYSCOPY DD
DSN=DB0BI.DB1.TS1.FC01,
//     DISP=(MOD,CATLG), UNIT=SYSDA,
//     SPACE=(16384,(20,20),,,ROUND)
//DSNUPROC.SYSIN    DD  *
  COPY  TABLESPACE DB1.TS1
```

FLASHCOPY_COPY=YES



VSAM cluster:
DB0BI.DB1.TS1.LOCAL.COPYFC
(name generated using FCCOPYDDN)

FCIC - Partitioned page sets



SYSCOPY rows

DSNUM	ICTYPE	ICDATE	START_RBA	PIT_RBA	STYPE	ICBACKUP
0	F	100827	00002DC978EF	00002DC97923	Q	
0	F	100827	00002DC978EF	00002DC97923	T	FC
3	F	100827	00002DC978EF	00002DC97923	T	FC
2	F	100827	00002DC978EF	00002DC97923	T	FC
1	F	100827	00002DC978EF	00002DC97923	T	FC

FCIC SYSCOPY Records

- Records for FlashCopy copies have an ICTYPE=**F** and ICBACKUP=**FC**
- START_RBA is the point at which the pages for the object was externalized to disk
- PIT_RBA is the point of consistency, i.e. more or less the point when the FLASHCOPY relationship was established
- SYSCOPY records for each piece/part.
  - STYPE = T - FlashCopy copy is **consistent**
  - STYPE = N - FlashCopy copy is **not consistent**
  - STYPE = Q - Sequential copy is **consistent**
  - STYPE = U - Sequential copy is **not consistent**
  - TTYPE - one character indicating type of utility which made the copy
- The SYSCOPY record for a RECOVER with the BACKOUT keyword specified has a **ICTYPE= P** and a **STYPE=B**.

Db2 COPY utility – sequential copies – 1 of 2

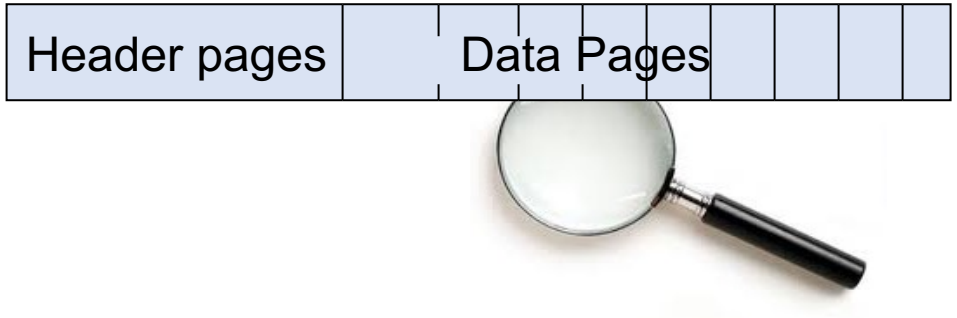
```
COPY TABLESPACE DSN8D81A.DDS1011B
```

```
14:14:46.55 DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DNET963.DNET963D
14:14:46.56 DSNUGTIS - PROCESSING SYSIN AS EBCDIC
14:14:46.57 DSNUGUTC -  COPY TABLESPACE DSN8D81A.DDS1011B DSNUM ALL FULL YES SH
14:14:46.72 DSNUBBID - COPY PROCESSED FOR TABLESPACE DSN8D81A.DDS1011B
    NUMBER OF PAGES=3
    AVERAGE PERCENT FREE SPACE PER PAGE = 32.66
    PERCENT OF CHANGED PAGES =  0.00
    ELAPSED TIME=00:00:00
16 14:14:46.79 DSNUBAFI - Db2 IMAGE COPY SUCCESSFUL FOR TABLESPACE DSN8D81A.DDS
```

You can use the output as input for:

- **DSN1COMP**
- **DSN1COPY**
- **DSN1PRNT**

Output format



Db2 COPY utility – sequential copies – 2 of 2

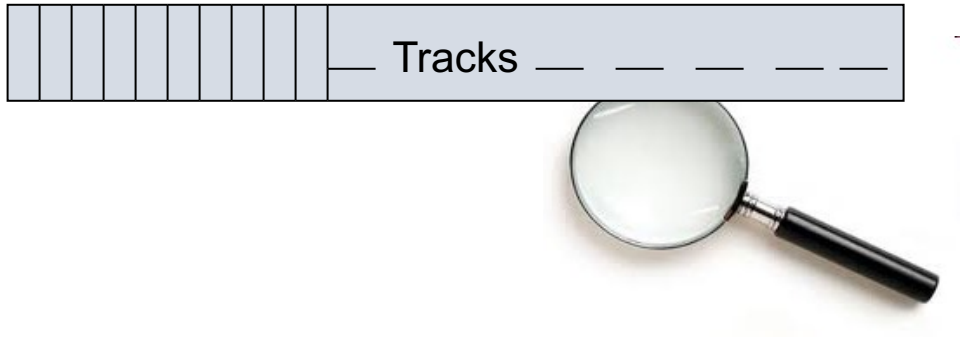
COPY TABLESPACE DSN8D81A.DDS1011B **CONCURRENT**

```
PAGE 0001     5695-DF175  DFSMSDSS V1R12.0 DATA SET SERVICES     2012.016 14:19  PARALLEL
ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'PARALLEL '
 DUM OPT(3) DATAS(INCL(DSNSCAT.DSNDBC.DSN8D81A.DDS1011B.J0001.A%%%)) -
 CAN CONC SHA TOL(ENQF) WAIT(0,0) -
 OUTDD(SYSCOPY)
ADR101I (R/I)-RI01 (01), TASKID 002 HAS BEEN ASSIGNED TO COMMAND 'DUM '
ADR109I (R/I)-RI01 (01), 2012.016 14:19:16 INITIAL SCAN OF USER CONTROL STATEME
ADR014I (SCH)-DSSU (02), 2012.016 14:19:16 ALL PREVIOUSLY SCHEDULED TASKS COMPL
```

You CANNOT use the output as input for:

- ~~DSN1COMP~~
- ~~DSN1COPY~~
- ~~DSN1PRNT~~

Output format



Db2 COPY utility – Db2 FCIC

COPY TABLESPACE DSN8D81A.DDS1011B **FLASHCOPY YES**

```
 FLASHCOPY YES
 DSNU421I    016 15:20:51.67 DSNUGFUM - START OF DFSMS MESSAGES
PAGE 0001     5695-DF175  DFSMSDSS V1R12.0 DATA SET SERVICES
ADR030I (SCH)-PRIME( 0), DCB VALUES HAVE BEEN MODIFIED FOR SYSPRINT
 COPY DATASET(INCLUDE( -
 DSNBCAT.DSNDBC.DSN8D51A.DSN8S51E.I0001.A001 , -
...
RENAMEU( -
 (DSNBCAT.DSNDBC.DSN8D51A.DSN8S51E.I0001.A001 , -
ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ADR109I (R/I)-RI01 (01), 2012.016 15:20:51 INITIAL SCAN OF USER CONTROL STATE
ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE
ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ADR006I (001)-STEND(01), 2012.016 15:20:51 EXECUTION BEGINS
ADR711I (001)-NEWDS(01), DATA SET DSNBCAT.DSNDBC.DSN8D51A.DSN8S51E.I0001.A001
                  HLQ.DSN8D51A.DSN8S51E.N00001.DNDJZ8MW USING STORCLAS
                  USRMGMT
ADR755W (001)-PROTD(01), SOURCE DATA SET DSNBCAT.DSNDBC.DSN8D51A.DSN8S51E.I00
                  TARGET DATA SET HLQ.DSN8D51A.DSN8S51E.N00001.DNDJZ8M
```

Use COPYTOCOPY to make sequential copy when needed to DASD or TAPE

Output format VSAM Cluster

FlashCopy – in-line FCIC

```
//DSNUPROC.SYSIN    DD  *
  COPY TABLESPACE HASHDB.HASHTS FLASHCOPY CONSISTENT SHRLEVEL CHANGE
```

```
1PAGE 0001     5695-DF175  DFSMSDSS V1R11.0 DATA SET SERVICES     2010.239 18:58
-ADR030I (SCH)-PRIME(01), DCB VALUES HAVE BEEN MODIFIED FOR SYSPRINT
  COPY DATASET(INCLUDE( -
  DB0BD.DSNDBC.HASHDB.HASHTS.J0001.A001 )) -
  RENAMEU( -
  (DB0BD.DSNDBC.HASHDB.HASHTS.J0001.A001 , -
  DB0BI.HASHDB.HASHTS.N00001.CY25YIE5 )) -
  REPUNC ALLDATA(*) ALLEXCP CANCELERROR SHARE -
  WRITECHECK TOLERATE(ENQF)
 ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'COPY '
 ADR109I (R/I)-RI01 (01), 2010.239 18:58:45 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED
 ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE
 ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
0ADR006I (001)-STEND(01), 2010.239 18:58:45 EXECUTION BEGINS
0ADR711I (001)-NEWDS(01), DATA SET DB0BD.DSNDBC.HASHDB.HASHTS.J0001.A001 HAS BEEN ALLOCATED WITH NEWNAME
                  DB0BI.HASHDB.HASHTS.N00001.CY25YIE5 USING STORCLAS DB0BDATA, DATACLAS DB0B,
                  AND MGMTCLAS   MCDb22
0ADR806I (001)-T0MI (03), DATA SET DB0BD.DSNDBC.HASHDB.HASHTS.J0001.A001 COPIED USING A FAST REPLICATION FUNCTION
0ADR801I (001)-DDDS (01), DATA SET FILTERING IS COMPLETE. 1 OF 1 DATA SETS WERE SELECTED: 0 FAILED 0ADR454I (001)-DDDS
(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
```

FlashCopy – Recover using FCIC

RECOVER TABLESPACE DSN00020.FLASHTES TOCOPY

```
DB0AU.DSN00020.FLASHTES.N00001.CV11XBMY
 DSNU421I    173 17:40:46.58 DSNUGFUM - START OF DFSMS MESSAGES
1PAGE 0001     5695-DF175  DFSMSDSS V1R11.0 DATA SET SERVICES     2010.173 17:40
-ADR030I (SCH)-PRIME(01), DCB VALUES HAVE BEEN MODIFIED FOR SYSPRINT
  COPY DATASET(INCLUDE( -
  DB0AU.DSN00020.FLASHTES.N00001.CV11XBMY )) -
  RENAMEU( -
  (DB0AU.DSN00020.FLASHTES.N00001.CV11XBMY , -
  DB0AU.DSNDBC.DSN00020.FLASHTES.I0001.A001 )) -
  ALLDATA(*) ALLEXCP CANCELERROR SHARE -
  REPUNC TOLERATE(ENQF) DEBUG(FRMSG(DTL))
 ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'COPY '
 ADR109I (R/I)-RI01 (01), 2010.173 17:40:46 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED
 ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE
 ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ADR006I (001)-STEND(01), 2010.173 17:40:46 EXECUTION BEGINS
ADR442I (001)-PREVS(01), DATA SET DB0AU.DSN00020.FLASHTES.N00001.CV11XBMY PREALLOCATED WITH NEW NAME
                    DB0AU.DSNDBC.DSN00020.FLASHTES.I0001.A001, FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
                    DB0AU.DSN00020.FLASHTES.N00001.CV11XBMY
```

Thank you for attending

email: mynhardt@us.ibm.com