BACKUP SYSTEM / RESTORE SYSTEM

Guy_delaporte@fr.ibm.com

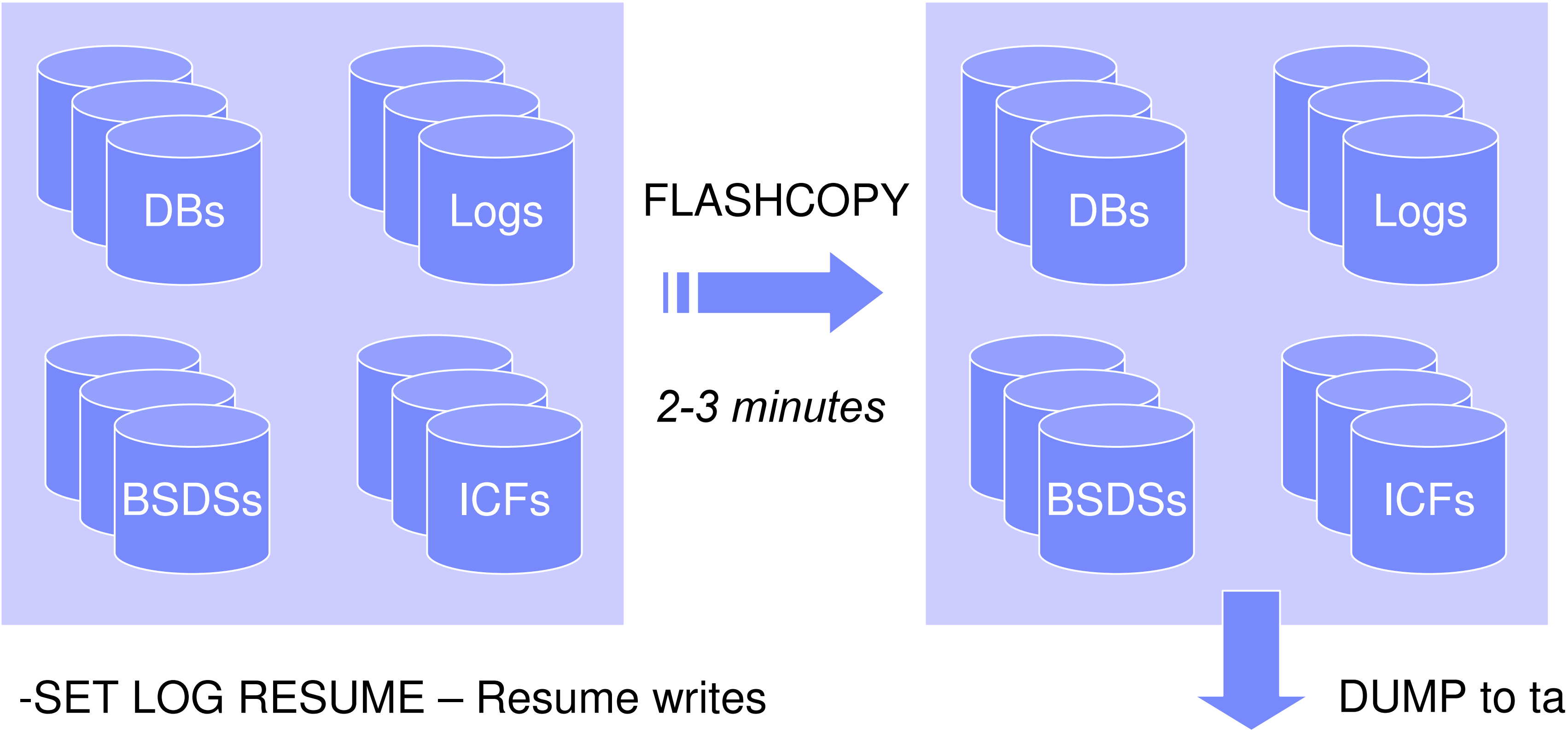
DB2 Information Management Software

Customer Requirements

- Need a fast non-disruptive way to backup/recovery all DB2 data
  - It is too slow and difficult to manage backups at table/index level
  - Need a way to ensure data and logs in backup are consistent for restart recovery
  - Consistent backup with no impact to applications
- Backups can be used to support
  - Point-In-Time Recovery on application errors
  - Disaster Recovery or Cloning Systems

Prior to DB2 V8 – Volume-level backup

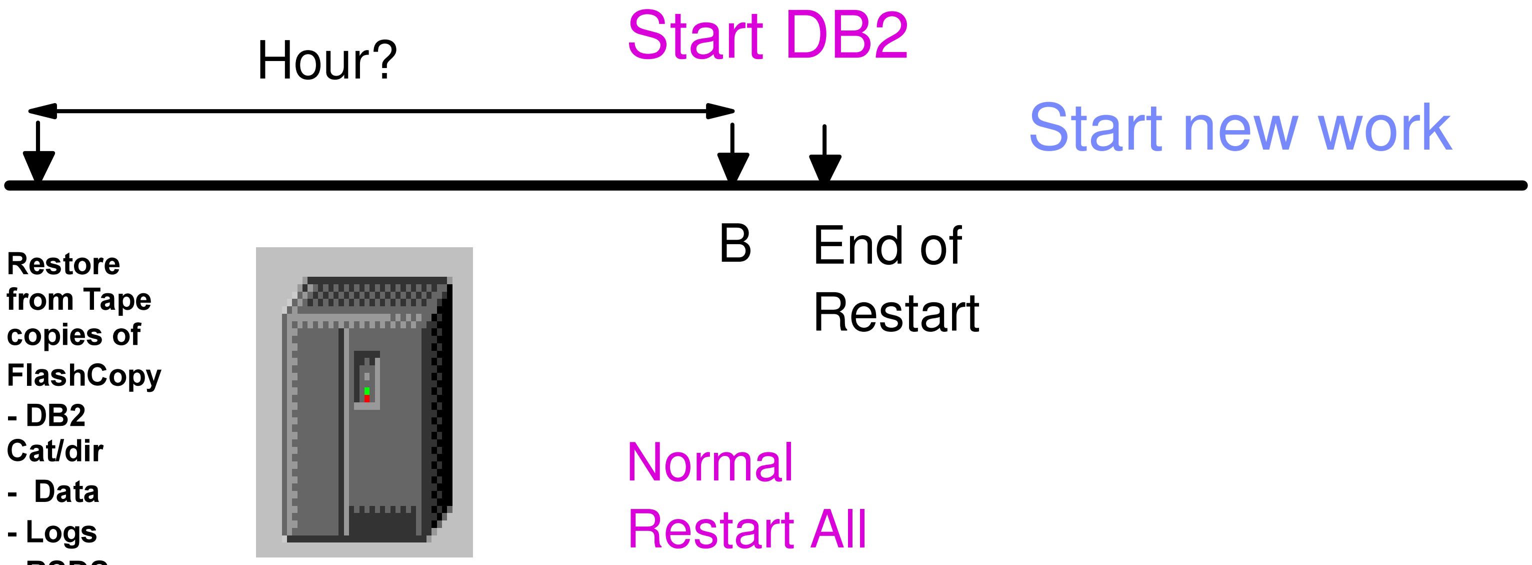
-SET LOG SUSPEND – Suspend writes



-SET LOG RESUME – Resume writes

DUMP to tape

Prior to DB2 V8 – Recovery to the time of the backup



Hour?

Start DB2

Start new work

B

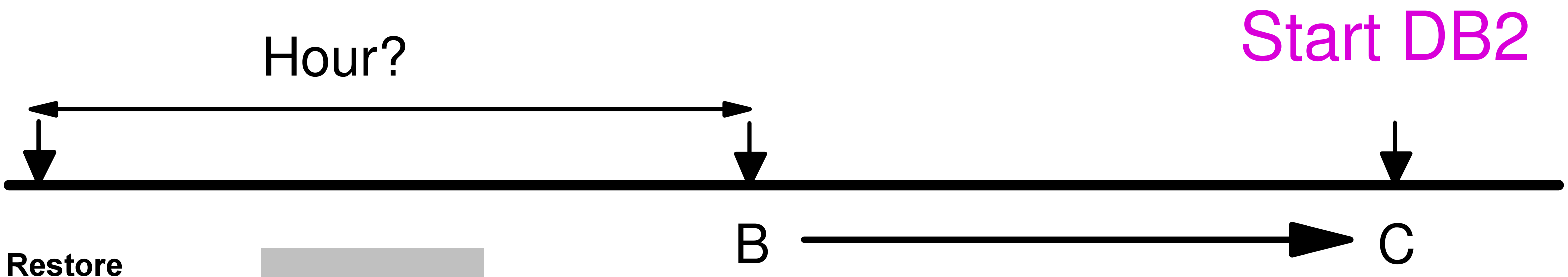
End of
Restart

Restore from Tape copies of FlashCopy
- DB2 Cat/dir
- Data
- Logs
- BSDS

Normal
Restart All

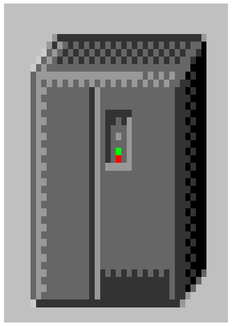
Analysis: Fast Restart – loss of data since FlashCopy

Prior to DB2 V8 – Recovery to any point-in-time



6 hours of log

Restore from Tape copies of FlashCopy
- DB2 Cat/dir
- Data
- Logs
- BSDS



Conditional
Endrba=C
Defer All
+
RECOVER
LOGONLY

Analysis: Longer to come up – reduced data loss
Complex DB2 D/R procedures required

With DB2 V8 – System PITR

- Provides an easier and less disruptive way for fast volume-level backup and recovery
  - Use FlashCopy API to backup DB2 data and logs
  - No longer need to suspend logs
  - Backup and recovery are managed by DB2 and DFSMShsm
- Two new utilities:
  - BACKUP SYSTEM
  - RESTORE SYSTEM
- Backup copies can also be used for:
  - Disaster recovery
  - System cloning

DFSMS constructs and definitions

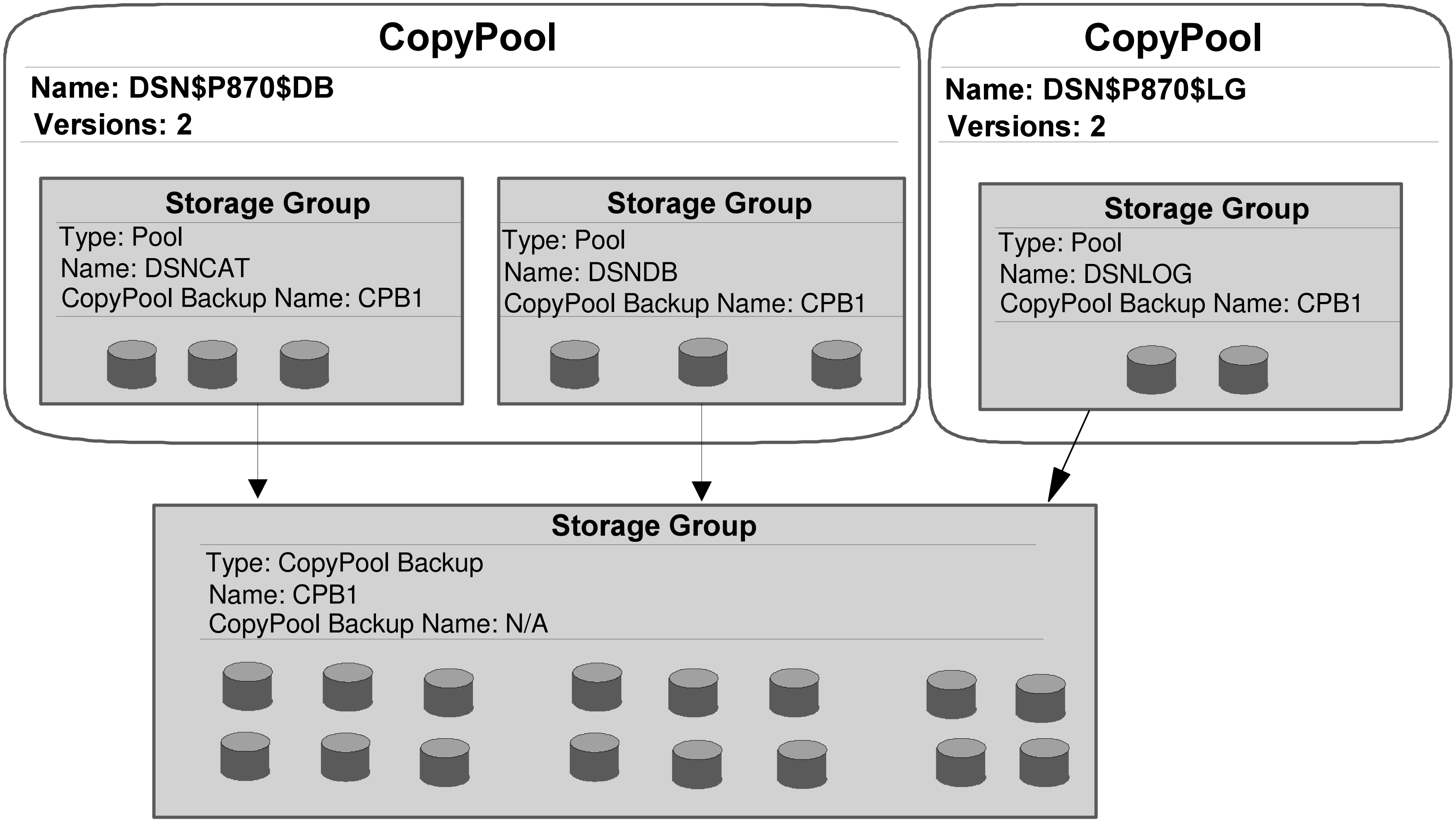
- COPYPOOL
  - New SMS construct
  - Set of SMS storage groups – maximum 256 storage groups
  - Has a VERSIONS attribute to specify the number of copy versions to be maintained on DASD – maximum is 85
  - Each DB2 system can have two SMS COPYPOOLs

    DATABASE COPYPOOL (DSN$location_name$DB)

    LOG COPYPOOL (DSN$location_name$LG)

- COPYPOOL BACKUP storage group
  - New storage group type
  - Used to hold volume copies of DASD defined in the COPYPOOL

Example of SMS definitions



CopyPool

Name: DSN$P870$DB
Versions: 2

Storage Group
Type: Pool
Name: DSNCAT
CopyPool Backup Name: CPB1

Storage Group
Type: Pool
Name: DSNDB
CopyPool Backup Name: CPB1

CopyPool

Name: DSN$P870$LG
Versions: 2

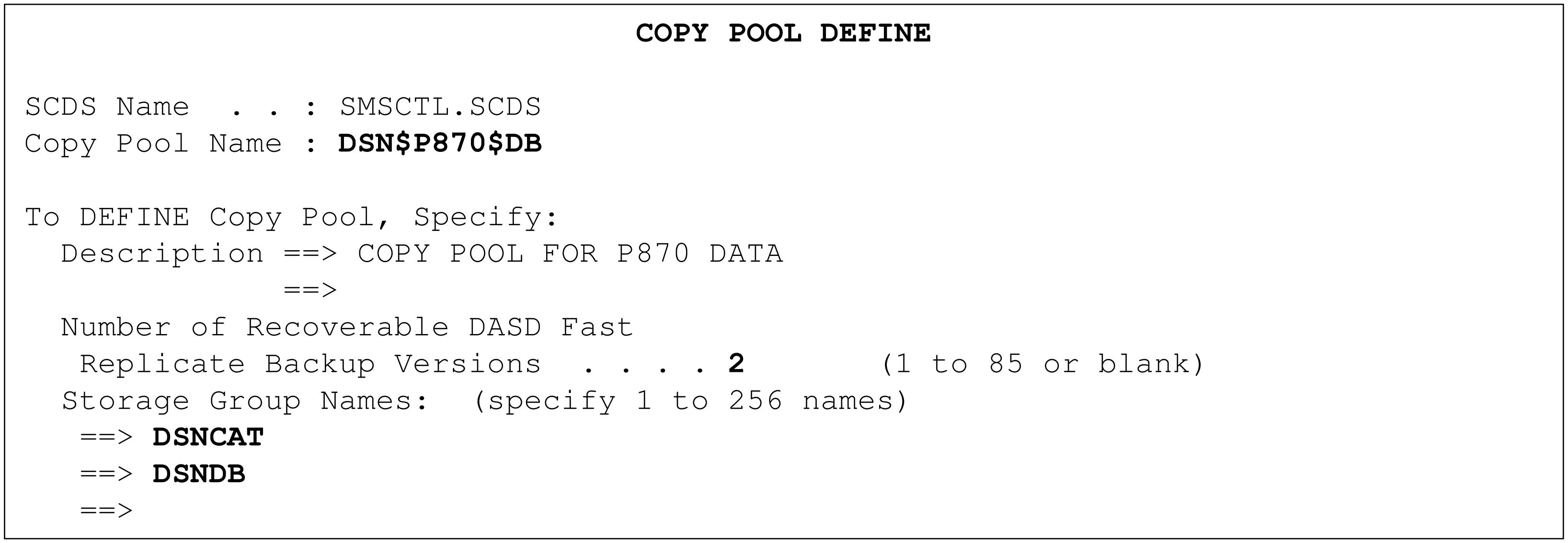
Storage Group
Type: Pool
Name: DSNLOG
CopyPool Backup Name: CPB1

Storage Group
Type: CopyPool Backup
Name: CPB1
CopyPool Backup Name: N/A

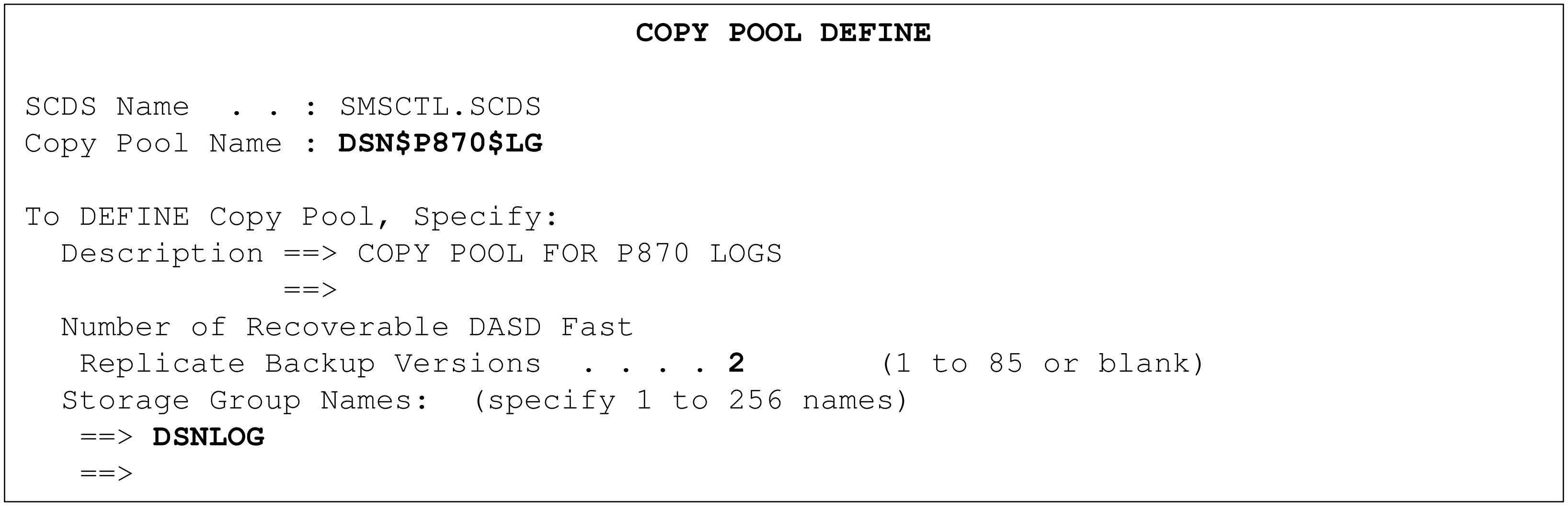
Requirements in a DB2 environment

- All volumes containing DB2 data sets, including the DB2 catalog and directory, the logs, and the BSDS, must be SMS managed.
- DB2 logs and BSDS should be separated from the rest of the DB2 data sets:
  - Own pool of volumes
  - Own ICF catalog
  - Defined in a separate SMS storage group
- Special care should be taken to ensure that the ICF catalogs stay synchronized with the data.
  - Must have separate ICF catalog for data and logs
  - Do not share ICF catalog with non-DB2 data

```
                      ISMF PRIMARY OPTION MENU - z/OS DFSMS V1 R5
Enter Selection or Command ===> P

Select one of the following options and press Enter:
0  ISMF Profile              - Specify ISMF User Profile
1  Data Set                  - Perform Functions Against Data Sets
2  Volume                    - Perform Functions Against Volumes
3  Management Class          - Specify Data Set Backup and Migration Criteria
4  Data Class                - Specify Data Set Allocation Parameters
5  Storage Class             - Specify Data Set Performance and Availability
6  Storage Group             - Specify Volume Names and Free Space Thresholds
7  Automatic Class Selection - Specify ACS Routines and Test Criteria
8  Control Data Set          - Specify System Names and Default Criteria
9  Aggregate Group           - Specify Data Set Recovery Parameters
10 Library Management        - Specify Library and Drive Configurations
11 Enhanced ACS Management   - Perform Enhanced Test/Configuration Management
C  Data Collection           - Process Data Collection Function
L  List                      - Perform Functions Against Saved ISMF Lists
P  Copy Pool                 - Specify Pool Storage Groups for Copies
R  Removable Media Manager   - Perform Functions Against Removable Media
X  Exit                      - Terminate ISMF
Use HELP Command for Help; Use END Command or X to Exit.
```



```
                              COPY POOL DEFINE

SCDS Name  . . : SMSCTL.SCDS
Copy Pool Name : DSN$P870$DB

To DEFINE Copy Pool, Specify:
  Description ==> COPY POOL FOR P870 DATA
              ==>
  Number of Recoverable DASD Fast
   Replicate Backup Versions  . . . . 2        (1 to 85 or blank)
  Storage Group Names:  (specify 1 to 256 names)
   ==> DSNCAT
   ==> DSNDB
   ==>
```



```
                              COPY POOL DEFINE

SCDS Name  . . : SMSCTL.SCDS
Copy Pool Name : DSN$P870$LG

To DEFINE Copy Pool, Specify:
  Description ==> COPY POOL FOR P870 LOGS
              ==>
  Number of Recoverable DASD Fast
   Replicate Backup Versions  . . . . 2        (1 to 85 or blank)
  Storage Group Names:  (specify 1 to 256 names)
   ==> DSNLOG
   ==>
```

DFSMShsm commands

- FRBACKUP – Create a Fast Replication backup
  - ➢ `FRBACKUP COPYPOOL(DSN$locn$DB) PREPARE`
  - ➢ `FRBACKUP COPYPOOL(DSN$locn$DB) NOVTOCENQ TOKEN(token)`
- FRRECOV – Recover a pool of volumes (or a target volume)
  - ➢ `FRRECOV COPYPOOL(DSN$locn$DB) VERIFY(YES) TOKEN(token)`
- QUERY – Determine status of physical backup
  - ➢ `QUERY COPYPOOL(DSN$locn$DB)`
- LIST – Provide information about each backup version of copy pool
  - ➢ `LIST COPYPOOL(DSN$locn$DB)`

BACKUP SYSTEM utility

- Invokes DFSMShsm to take fast volume copies of the DB2 data and / or logs
- No DB2 quiesce point is required, nothing stops as in SET LOG SUSPEND
  - BACKUP SYSTEM FULL

    Allow recovery of the entire system in later stage

    Have to define the "database" and "log" COPYPOOLs

    Backup both database and then log (active logs and BSDS)

  - BACKUP SYSTEM DATA ONLY

    Only "database" COPYPOOL has to be defined for database backup

BACKUP SYSTEM process

```
//BACKUP EXEC DSNUPROC,PARM=’P870,BACKUP`
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
BACKUP SYSTEM DATA ONLY
```

- When BACKUP SYSTEM is issued, DB2
  - Suspends 32K page writes
  - Suspends data set creation, deletion, rename and extension operations
  - Prevents data sets from pseudo-close
  - *Records* the Recover Based Log Point (RBLP) in DBD01

    Starting point for RESTORE SYSTEM to apply log records

    Data sharing – Oldest system checkpoint across all members of group

BACKUP SYSTEM process…

- During the backup . . .
  - Invokes DFSMShsm to take FlashCopy of ’DB‘ COPYPOOL

    ```
    FRBACKUP COPYPOOL(DSN$locn$DB) NOVTOCENQ TOKEN(token)
    ```

  - Invokes DFSMShsm to take FlashCopy of the 'LG‘ COPYPOOL, if for Backup System Full

    ```
    FRBACKUP COPYPOOL(DSN$locn$LG) NOVTOCENQ TOKEN(token)
    ```

  - Each member updates BSDS with the system backup information

    Up to 50 entries

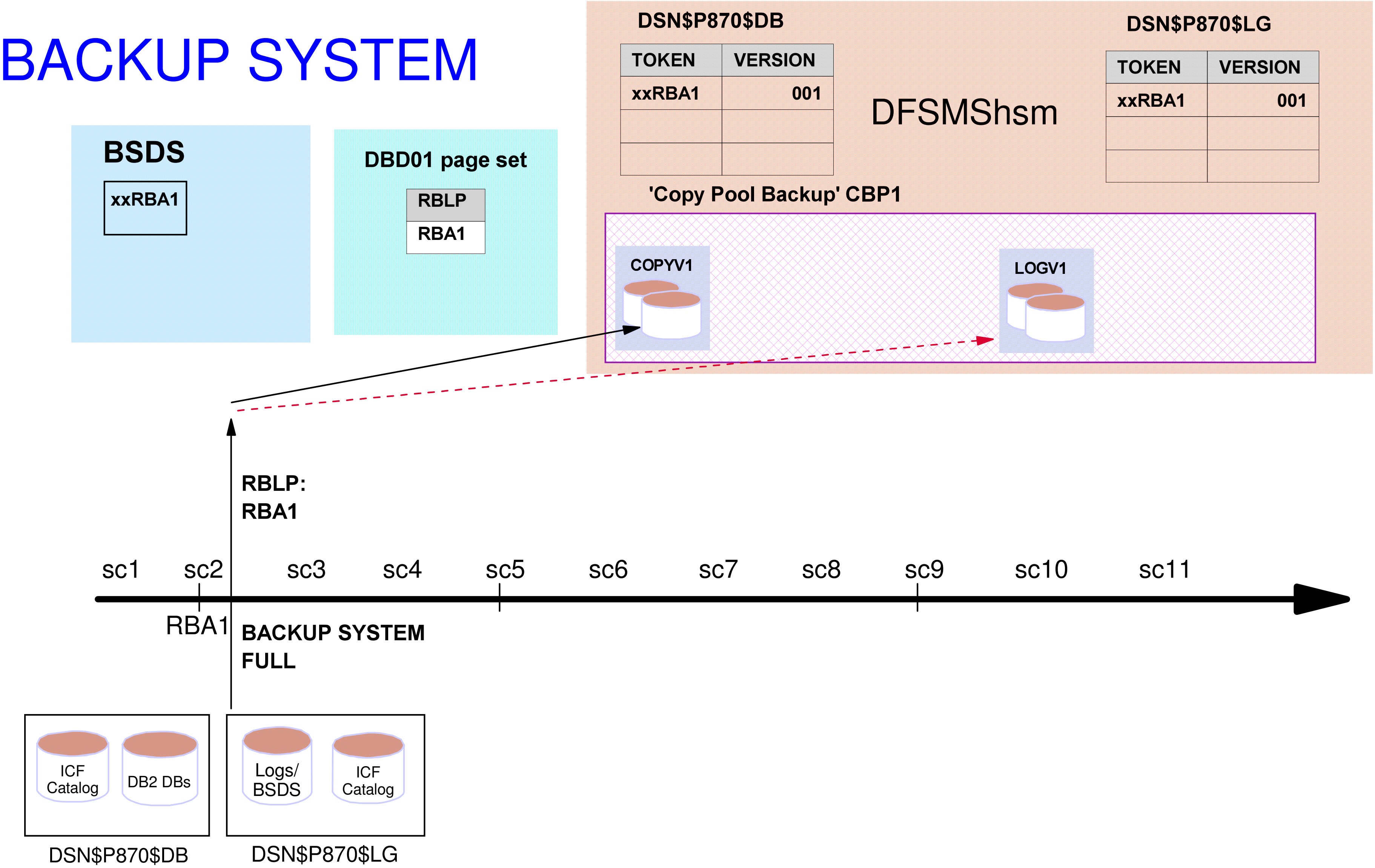
    In data sharing only the submitting member logs BSDS information

  - Resume the quiesced activities

```
DSNU000I DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = BACKUP
DSNU1044I DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I DSNUGUTC - BACKUP SYSTEM DATA ONLY
DSNU1600I DSNUVBBD - BACKUP SYSTEM UTILITY FOR DATA STARTING,
COPYPOOL = DSN$P870$DB
TOKEN = X'D7F8F7F0BB8E5142855EF907BB8E502C0AC4'.
DSNU1614I DSNUVBBD - BACKUP SYSTEM UTILITY FOR DATA COMPLETED
SUCCESSFULLY,
COPYPOOL = DSN$P870$DB
TOKEN = X'D7F8F7F0BB8E5142855EF907BB8E502C0AC4'
ELAPSED TIME = 00:00:16.
DSNU1602I DSNUVBBD - BACKUP SYSTEM UTILITY COMPLETED, ELAPSED TIME =
00:00:19.
```

```
                        BACKUP  SYSTEM UTILITY HISTORY
                              SUBSYSTEM ID P870
                            20:59:59 JULY 22, 2004

 START STCK                                       DATA COMPLETE  DATA/LOG   COMPLETE
    DATA                LOG              RBLP          LRSN          DATE      LTIME
---------------- ---------------- ------------ ------------ --------------------
BB8E5142855EF907 0000000000000000 BB8E502C0AC4 BB8E502C0AC4 2004/07/22 13:59:45
                TOKEN = D7F8F7F0BB8E5142855EF907BB8E502C0AC4
BA1CCAA9A3820B8A BA1CCAB1FF964845 00120CDAC090 00120CDAC090 2003/10/02 16:12:43
                TOKEN = D7F8F7F0BA1CCAA9A3820B8A00120CDAC090
```

BACKUP SYSTEM



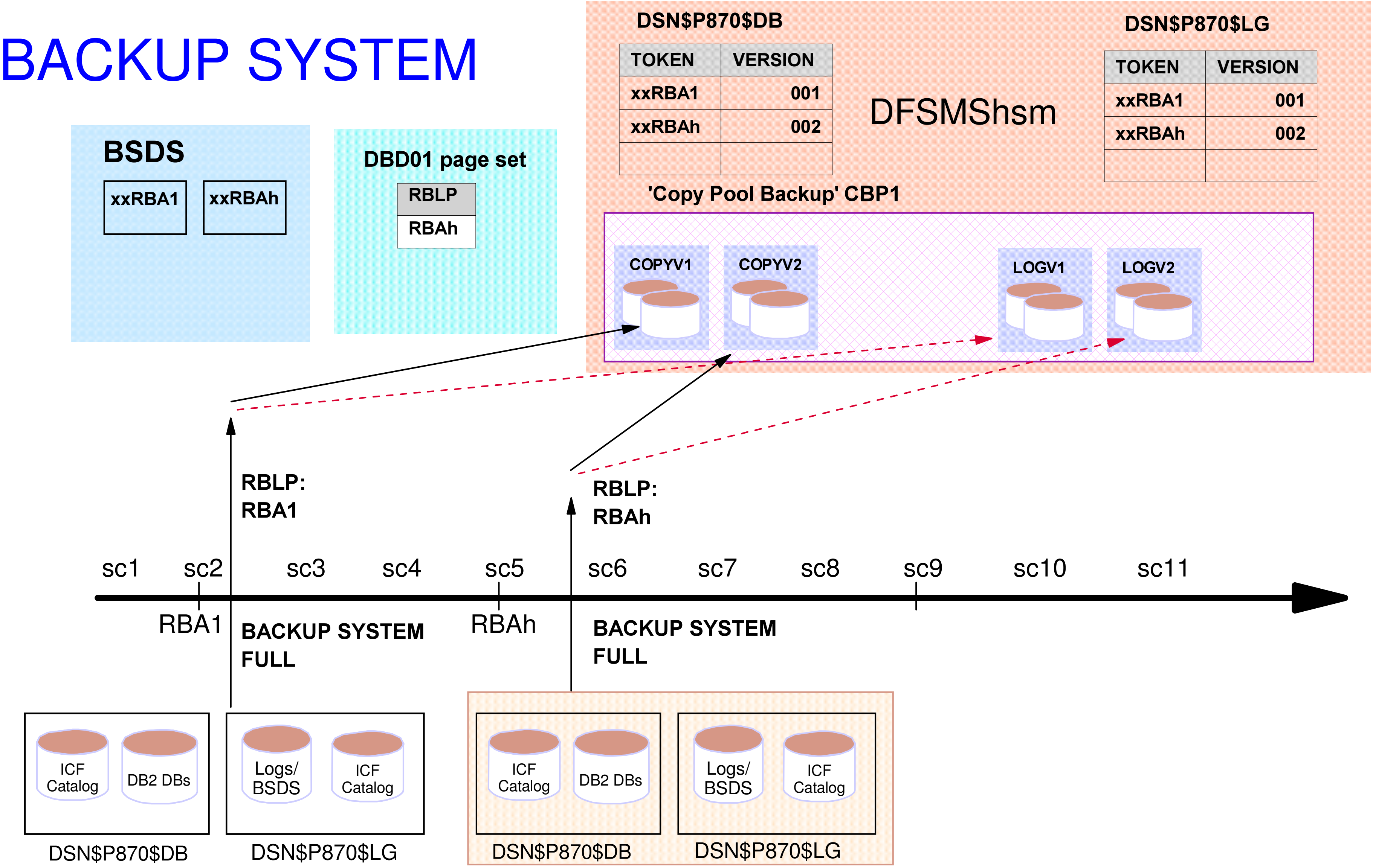
DSN$P870$DB

TOKEN	VERSION
xxRBA1	001

DSN$P870$LG

TOKEN	VERSION
xxRBA1	001

BACKUP SYSTEM



DSN$P870$DB

TOKEN	VERSION
xxRBA1	001
xxRBAh	002

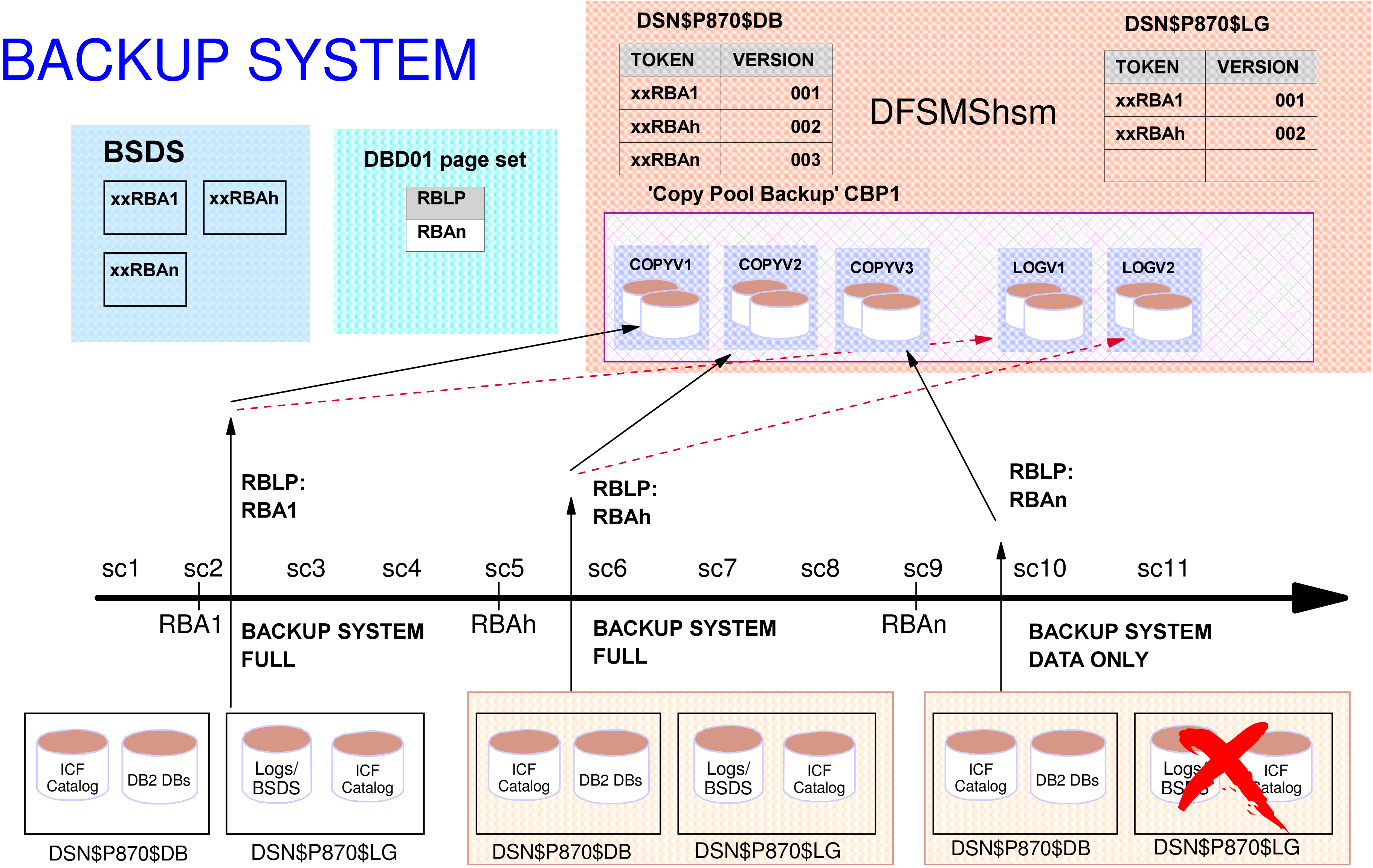
DFSMShsm

DSN$P870$LG

TOKEN	VERSION
xxRBA1	001
xxRBAh	002

'Copy Pool Backup' CBP1

BACKUP SYSTEM



DSN$P870$DB

TOKEN	VERSION
xxRBA1	001
xxRBAh	002
xxRBAn	003

DSN$P870$LG

TOKEN	VERSION
xxRBA1	001
xxRBAh	002

RESTORE SYSTEM utility

- RESTORE SYSTEM utility is only needed to recover system to an arbitrary PIT
- To recover system only to the PIT at which the backup copy was taken
  - Use copies from BACKUP SYSTEM FULL to restore the database and log copypool

    ```
    FRRECOV COPYPOOL(DSN$locn$DB) VERIFY(YES) TOKEN(token)
    FRRECOV COPYPOOL(DSN$locn$LG) VERIFY(YES) TOKEN(token)
    ```
  - Start DB2 and inflight URs are backed out

System level restore to an arbitrary PIT

- RESTORE SYSTEM utility is needed
  - Use copies from BACKUP SYSTEM FULL or DATA ONLY

    RESTORE SYSTEM does not restore LOG backup copies, therefore copies from DATA ONLY is enough
- Two phases:
  - RESTORE phase: recover the database volumes from the latest BACKUP version prior to the arbitrary PIT
  - LOG APPLY phase: apply log records to recover DB objects to the PIT

RESTORE SYSTEM

BSDS

xxRBA1
xxRBAh
xxRBAn

DBD01 page set

RBLP
RBAn

DSN$P870$DB

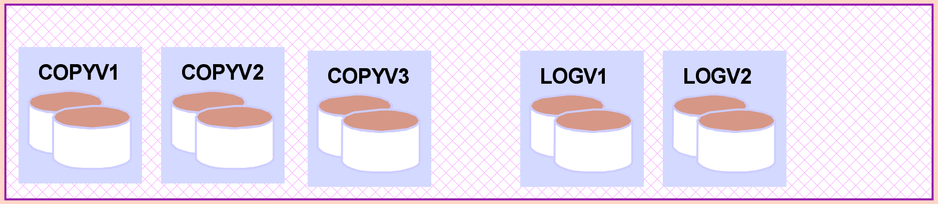
TOKEN	VERSION
xxRBA1	001
xxRBAh	002
xxRBAn	003

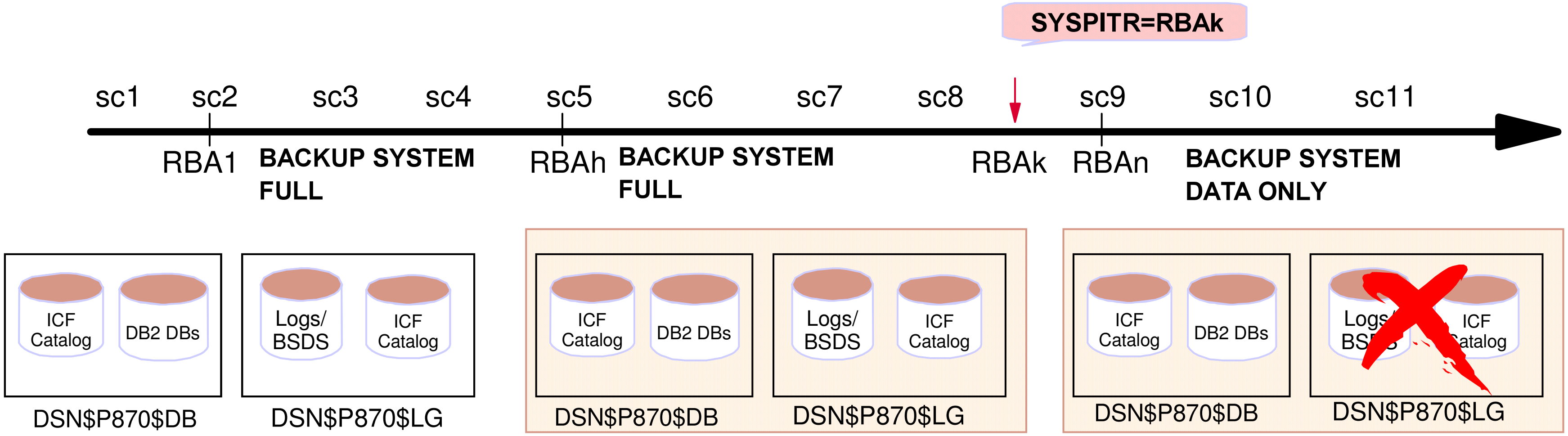
DFSMShsm

DSN$P870$LG

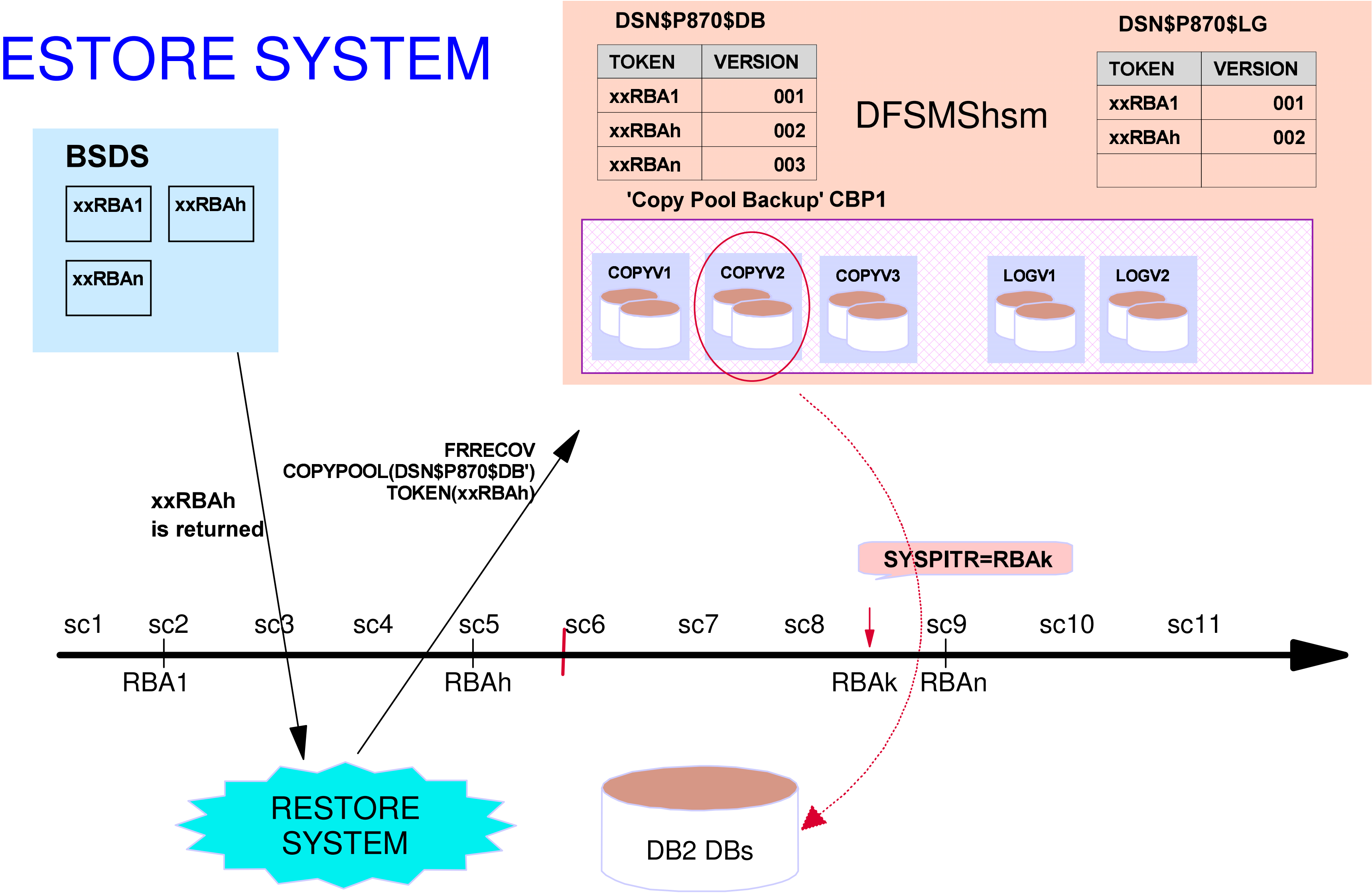
TOKEN	VERSION
xxRBA1	001
xxRBAh	002

'Copy Pool Backup' CBP1





RESTORE SYSTEM



BSDS

xxRBA1	xxRBAh
xxRBAn	

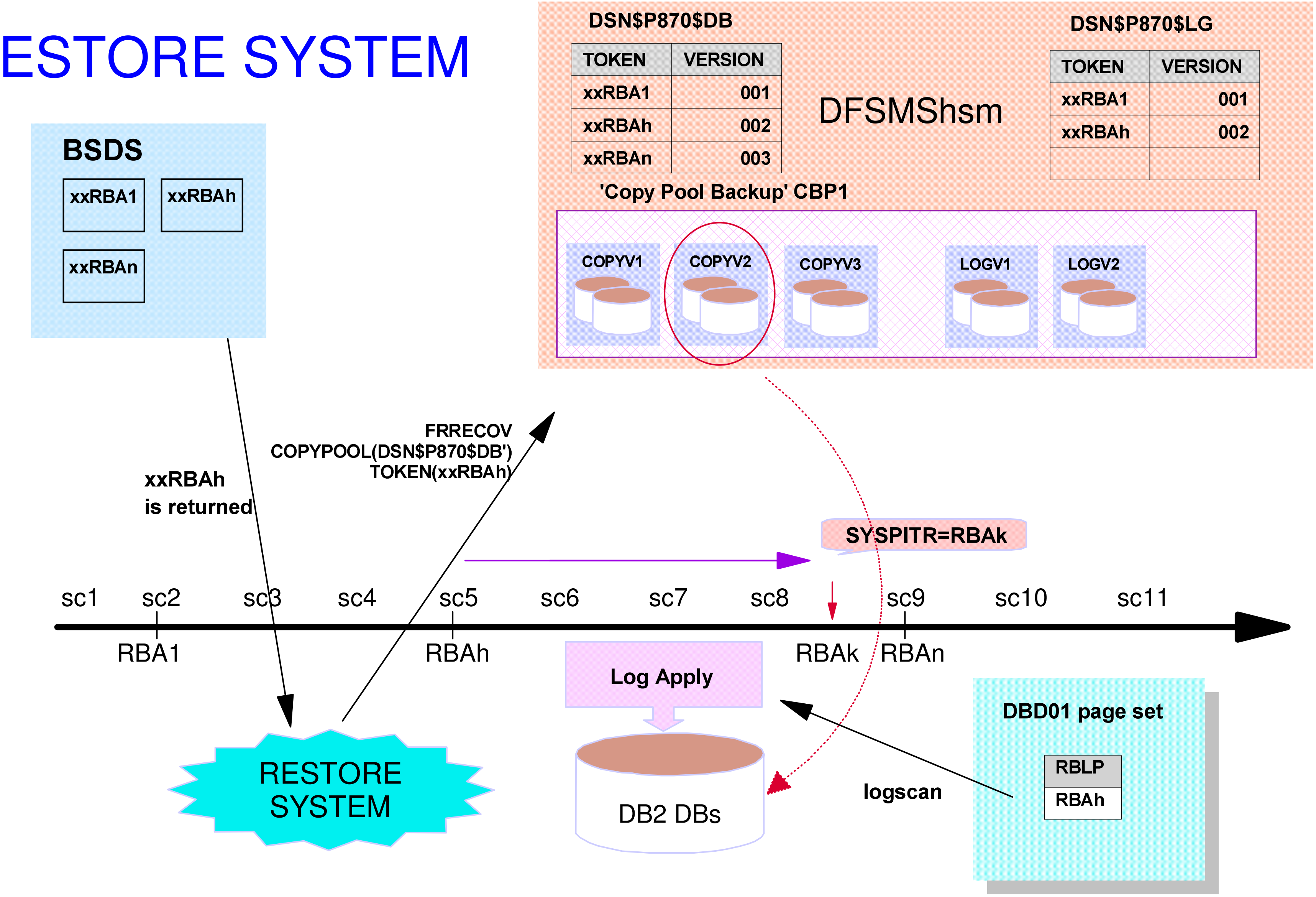
DSN$P870$DB

TOKEN	VERSION
xxRBA1	001
xxRBAh	002
xxRBAn	003

DSN$P870$LG

TOKEN	VERSION
xxRBA1	001
xxRBAh	002

RESTORE SYSTEM



DFSMShsm

DSN$P870$DB

TOKEN	VERSION
xxRBA1	001
xxRBAh	002
xxRBAn	003

DSN$P870$LG

TOKEN	VERSION
xxRBA1	001
xxRBAh	002

'Copy Pool Backup' CBP1

BSDS

DBD01 page set

RBLP
RBAh

FRRECOV
COPYPOOL(DSN$P870$DB')
TOKEN(xxRBAh)

xxRBAh
is returned

SYSPITR=RBAk

System level restore to an arbitrary PIT

- Establish the ’PITR’ conditional restart record
  - CRESTART CREATE SYSPITR=log-point (*truncation RBA value*)
  - MUST be in New Function Mode
- Start DB2 with a PITR CRCR
  - DB2 system enters into System Recover Pending mode
  - Implictly apply DEFER ALL, FORWARD = NO (except for in-doubt URs) and Access(Maint)
  - Write logs to rollback uncommitted changes
  - Reset database restrict status and utility job status

System level restore to an arbitrary PIT

- RESTORE SYSTEM
  - Restore the "database" COPYPOOL version that was taken by BACKUP SYSTEM prior to the specified PIT recovery point
  - Perform log apply function
- RESTORE SYSTEM with LOGONLY specified
  - Performs log apply function only
  - Note: this option can run in z/OS 1.3 without BACKUP SYSTEM utility

    Using Log Suspend/Resume and backup volumes manually
- Stop DB2 --> resets the system recover-pending status
- Recover all objects that are marked in RECP or RBDP state

System level restore to an arbitrary PIT – Data Sharing

- Establish the LRSN truncation point on all active members
  - CRESTART CREATE SYSPITR= end-lrsn
- Delete all CF structures
- Group restart each active member with the SYSPITR CRCR
  - All members MUST be restarted
- Restore system
  - Similar to the steps as in the non-data sharing environment

System level restore – Notes

- When DB2 is in System Recover Pending state
  - Only RESTORE SYSTEM utility is allowed
  - START DATABASE command is not allowed
  - TERM UTIL command is not allowed
  - DISPLAY UTIL command will display only the status of RESTORE SYSTEM utility
  - SQL operation is not allowed

    Claim request on any DB2 objects will be rejected with a

    -904 SQL code (reason code of 00C20269)

- Restore of the database volumes is done in parallel

System level restore – Log recovery

- DB2 reads the DBD01 header page to retrieve the recovery base log point (RBLP) → starting point for the log scan
- Handles table spaces and index spaces
  - CREATEs - will define data sets
  - DROPs
  - LOG NO events

    If LOG NO, the associated object is entered into RECP or RBDP state. Table spaces and indexes with COPY YES attribute will be marked RECP and the indexes with COPY NO will be marked RBDP
- Uses fast log apply (FLA) to recover objects in parallel

System level restore – Log recovery (cont.)

- Log apply phase takes periodic checkpoints
  - ➢ Forces modified data pages to DASD and trigger system checkpoint
  - ➢ Updates the DBD01 header page with the new RBLP value
- At the end of log apply phase
  - ➢ Issues informational message if any object is marked RECP, RBDP or LPL during the log apply phase -> RESTORE SYSTEM RC=4
  - ➢ Reset the PITR state of each member
- RESTORE SYSTEM utility is restartable
  - ➢ In data sharing, only that member which issued the original RESTORE SYSTEM can issue the restart request

Hardware and software prerequisites

- z/OS V1R5 and DFSMShsm
- DFSMShsm BCDS requires a block size of 6544
- DASD control units which support ESS Flashcopy APIs
- FlashCopy V2 is strongly recommended – allows 'source' and 'target' to be across LSS boundary, but must be within same ESS
- DB2 datasets must reside on SMS-managed volumes
- Must be in New Function Mode
- RESTORE SYSTEM LOGONLY can be executed under z/OS 1.3
  - Assumes you have used -Set Log Suspend
  - Manually dumped volumes (like today)
  - Manually restored volumes (like today)
  - Eliminates complex recovery procedures for Disaster Recovery

Future directions

- *The FlashCopy source and target volumes can reside on different ESS*
- *DFSMShsm will automatically manage FlashCopy target volumes to tapes*
- *Use volume level backups as the source for DB2 object level recovery*
- *Manage data set level FlashCopy*
- *Support object level point-in-time recovery*
  - *Rollback uncommitted changes*

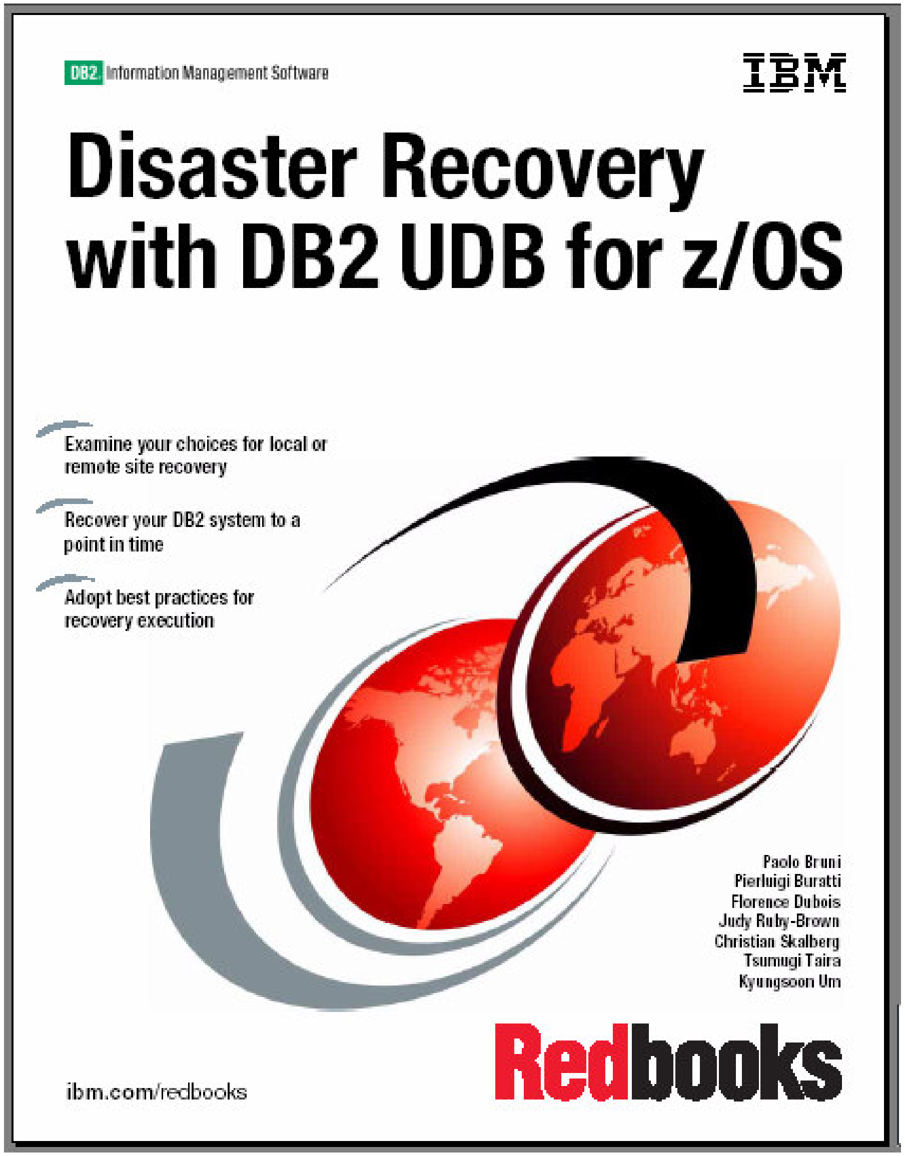
Just published...

Disaster Recovery with DB2 UDB for z/OS
SG24-6370

Chapter 6 - SMS copy pools and DB2 point in time recovery

Chapter 19 - Local recovery: System PITR

Chapter 20 - Restart using tape dump of copy pools



DB2 Information Management Software

IBM

Disaster Recovery with DB2 UDB for z/OS

- Examine your choices for local or remote site recovery
- Recover your DB2 system to a point in time
- Adopt best practices for recovery execution

Paolo Bruni
Pierluigi Buratti
Florence Dubois
Judy Ruby-Brown
Christian Skalberg
Tsumugi Taira
Kyungsoon Um

ibm.com/redbooks

Redbooks