Extended RBA and LSRN Support in DB2 11 for z/OS

Triangle DB2 User's Group Meeting
December 4, 2014

Charles Lewis
Technical Sales Specialist
IBM Mid-Atlantic Region
lewisc@us.ibm.com

Agenda

- Back in time --- the basic stuff
  - RBA & LRSN
  - DB2 logging by example -- INSERT
- The challenges with 6- byte format
- The solution – the new EXTENDED 10-byte format in DB2 11
- How to get there – conversion and controls

The Basic Stuff – What is an RBA?

- RBA = Relative Byte Address
  - The position from the very beginning of the DB2 log
- Since the first release of DB2 defined as a 6-byte value

C	9	C	1	2	3	1	1	3	9	3	F

- Can address up to 2^{48} bytes
  - Range from x'000000000000' up to x'FFFFFFFFFFFF'
  - 256 TB of log record addressing capability
- Each record in the DB2 log is identifiable by the RBA of the first byte of its header

The Basic Stuff – What is a LRSN?

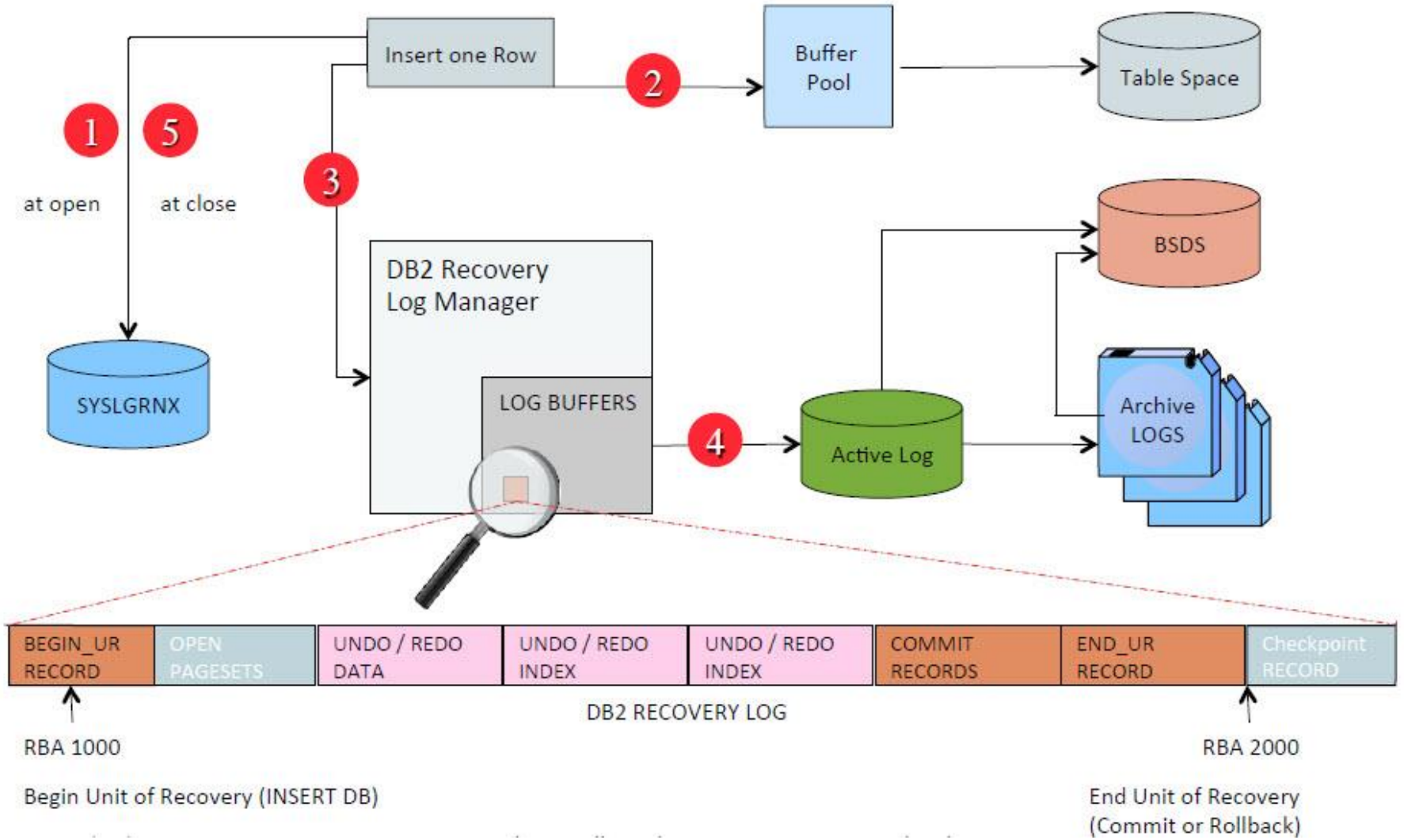
- LRSN = Log Record Sequence Number
  - Introduced with Data Sharing in DB2 V4
  - Used to serialize log records from all DB2 members of a data sharing group into the correct sequence during restart and recovery actions

- It is the high-order six bytes of the stored clock (STCK) value
  - The LRSN is incremented every 16 microseconds
  - However, it is only alike a TIMESTAMP
    - Can have an offset from actual STCK value
    - LRSN overflows on 2042/09/17 at the latest (if no LRSN delta exists)

    | C | 9 | C | 1 | 2 | 3 | 1 | 1 | 3 | 9 | 3 | F |

  - Kept in sync across all data sharing group members using the SYSPLEX timer

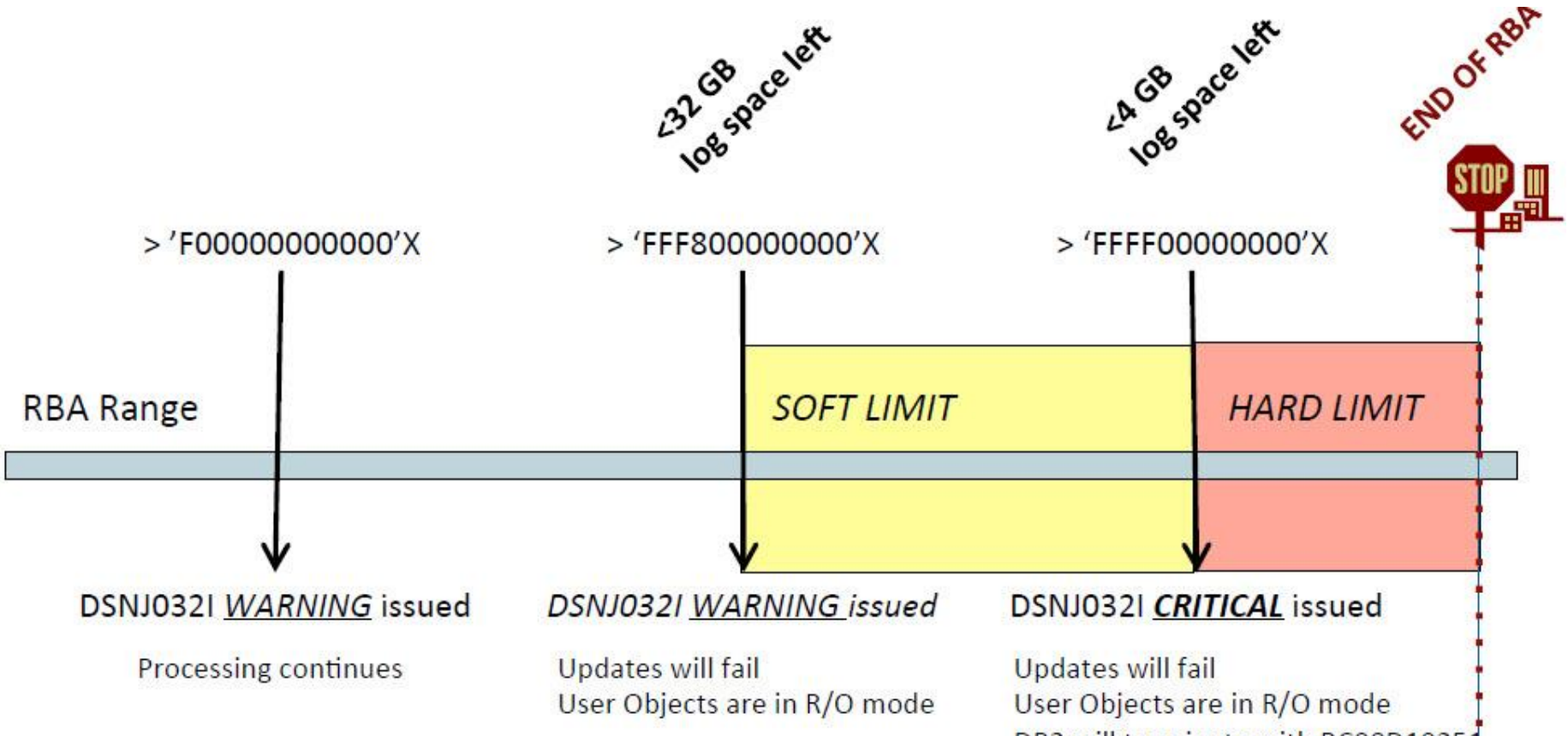
A Sample INSERT process



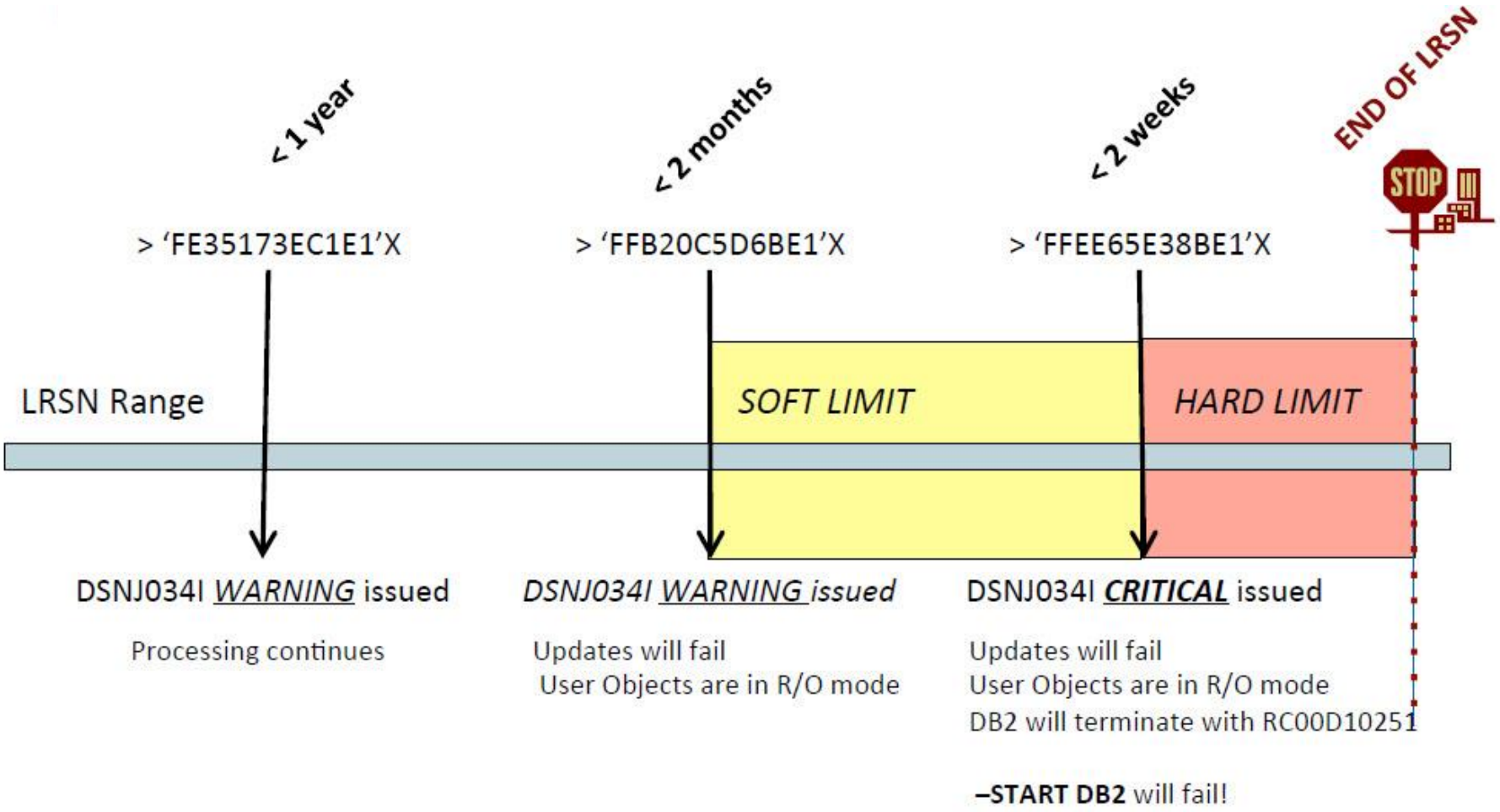
What is the problem with 6-byte RBA/RSN values?

- Due to a high DB2 logging volume and as hardware becomes faster the total range (256 TB) of an RBA can be exhausted
  - Required to execute the RBA reset procedure more often (requires a an outage in a non-data sharing DB2)
- LRSN value based on 16 micro seconds granularity
  - Can cause delays on faster CPUs with high update rate to same page
  - Possible CPU spin to get unique LRSN leading to more CPU overhead
- Leapfrog LRSN range when converting to Data Sharing
  - Highest log RBA of all members used + x (LRSN delta)
    - Ensures LSRN values are larger than any RBA before converting to data sharing
    - May cause DB2 to reach end of LRSN range before year 2042
    - And there is NO reset procedure for exhausted LRSN!!!

Reaching the RBA limit



Reaching the LRSN limit



When the hard limit is reached

- BSDS must be converted before start of DB2 after hard limit
  - Otherwise –START DB2 command will fail
- It is strongly recommended that you convert SYSUTILX, SYSTSCPY, and SYSLGRNZX tables to extended 10-byte format before the hard limit is reached.
  - Utilities might not be able to convert these objects to extended 10-byte format at the hard limit!

Preparing for the end of the RBA

- Determine the remaining log RBA range using DSNJU004

  [remaining log RBA range] = X'FFFFFFFFFFFF' - *[HIGHEST RBA WRITTEN]*

  [remaining log RBA range] = X'FFF80000000' - *[HIGHEST RBA WRITTEN]* (32GB log space left)

- Determine the average number of bytes logged per day using DSNJU004 utility

  (b) = *# of daily created archive logs*

  [average number of RBAs logged per day] = *b* * (*ENDRBA – STARTRBA*)

- Estimate remaining time

  [remaining time] = *[remaining log RBA range]* / *[average number of bytes logged per day]*

- Follow well documented RBA reset procedures!
  - https://ibm.biz/BdDZK6 - data sharing environment
  - https://ibm.biz/BdDZac - non-data sharing environment

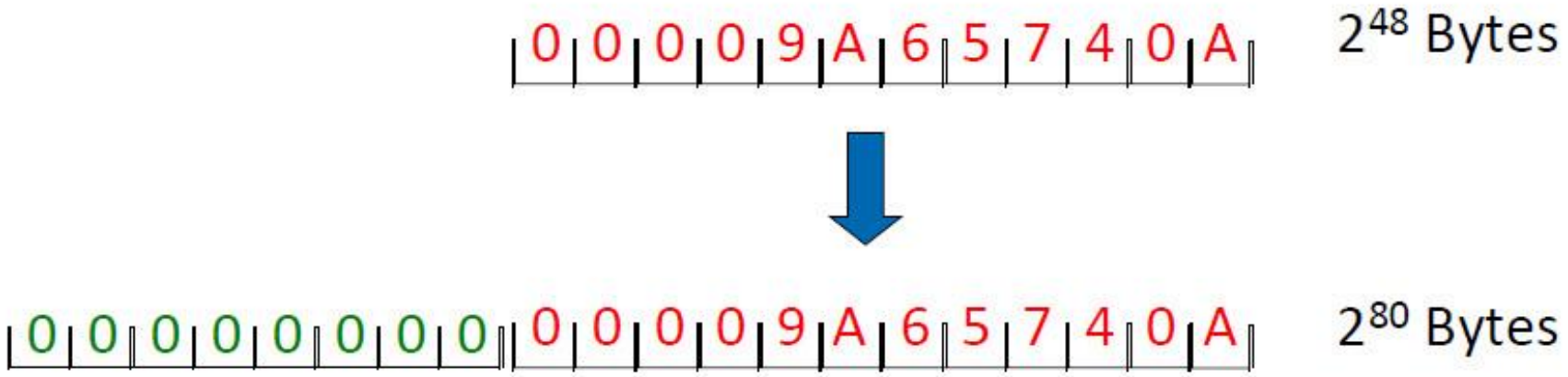
Another challenge with 6-byte format: LRSN spinning

- DB2 9 NFM introduced a function called LRSN spin avoidance
  - Duplicate LRSN values are allowed for consecutives log updates on different pages

- DB2 10 extended that relief to support duplicate LRSN values for updates on the same data page
  - But limited to INSERT log types
    - UPDATE and DELETE log type still require a unique LRSN value

The new 10-byte RBA format in DB2 11

- 6-byte to 10-byte RBA extension:
  - Add 4 bytes at the beginning for larger addressing range
  - Total addressability: 2^{80} Bytes = 1 Yottabyte



The new 10-byte LRSN format in DB2 11

- The 6-byte LRSN (BASIC) format

C	C	6	2	A	6	9	D	5	E	3	5

- Will be extended on the high-order and low-order bit side to the new 10-byte (EXTENDED) format
  - high-order 10 bytes of the stored clock (STCK) value

1 additional byte will extend the LRSN range to cover 36,534 years

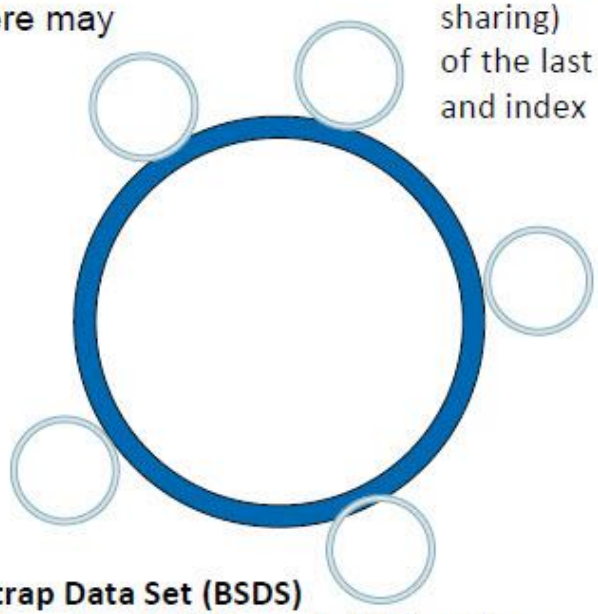
3 bytes at the end will increase the granularity to 1 picosecond

0	0	C	C	6	2	A	6	9	D	5	E	3	5	3	3	1	0	0	0

DB2 objects affected by the new format

System Common Area (SCA)
The SCA structure is used to track and communicate data pertinent to a data sharing group. This data always includes some LRSN and RBA values and there may be a large number of such values depending on the exception states, if any, of database objects.

Database Objects
The RBA (non-data sharing) and LRSN (data sharing)
of the last change in each page of every table and index



DB2 Catalog columns
The DB2 catalog and directory contain RBA and LRSN information
in several tables.

Bootstrap Data Set (BSDS)
Contains the LRSN and RBA values that bound each
active and archive log data set as well as a number of others that have various purposes.

Recovery Log
The log records themselves are assigned RBA values so that they can be located.
In a data sharing environment each log record has an associated LRSN value that is based on the time the log record was created (plus any LRSN delta).

The LRSN value is used to sequence log records from multiple members in a data sharing group.

How DB2 uses the new RBA/LRSN format

- DB2 11 is fundamentally a 10-byte format system
- All values are processed in 10-byte format internally
  - Avoids the complexity of a dual code path for dealing with 6-byte and 10-byte formats
  - Conversion is done for DB2 objects still in BASIC format
    - 6-byte values are mapped into a 10-byte bucket
- DB2 messages use external 10-byte format in all DB2 11 modes
  - To ensure a consistent message format in DB2 11
  - e.g. for QUIESCE or REPORT RECOVERY output
    - LRSN values contain either padded x'00' or precision value
- You need to be aware of this when going to DB2 11 CM!
  - Maintenance procedures and 3rd party tools must support DB2 11
    - E.g. IFCID 306 (always new log format)

How DB2 converts EXTENDED to BASIC format

- Happens when either the BSDS and log or the DB2 database object is not yet converted to EXTENDED format

0	0		C	9	C	1	2	3	1	1	3	9	F	A		1	4	F	6	A	4

- DB2 log manager will create the next valid LRSN value based on 6-byte, truncating the existing 10-byte format value

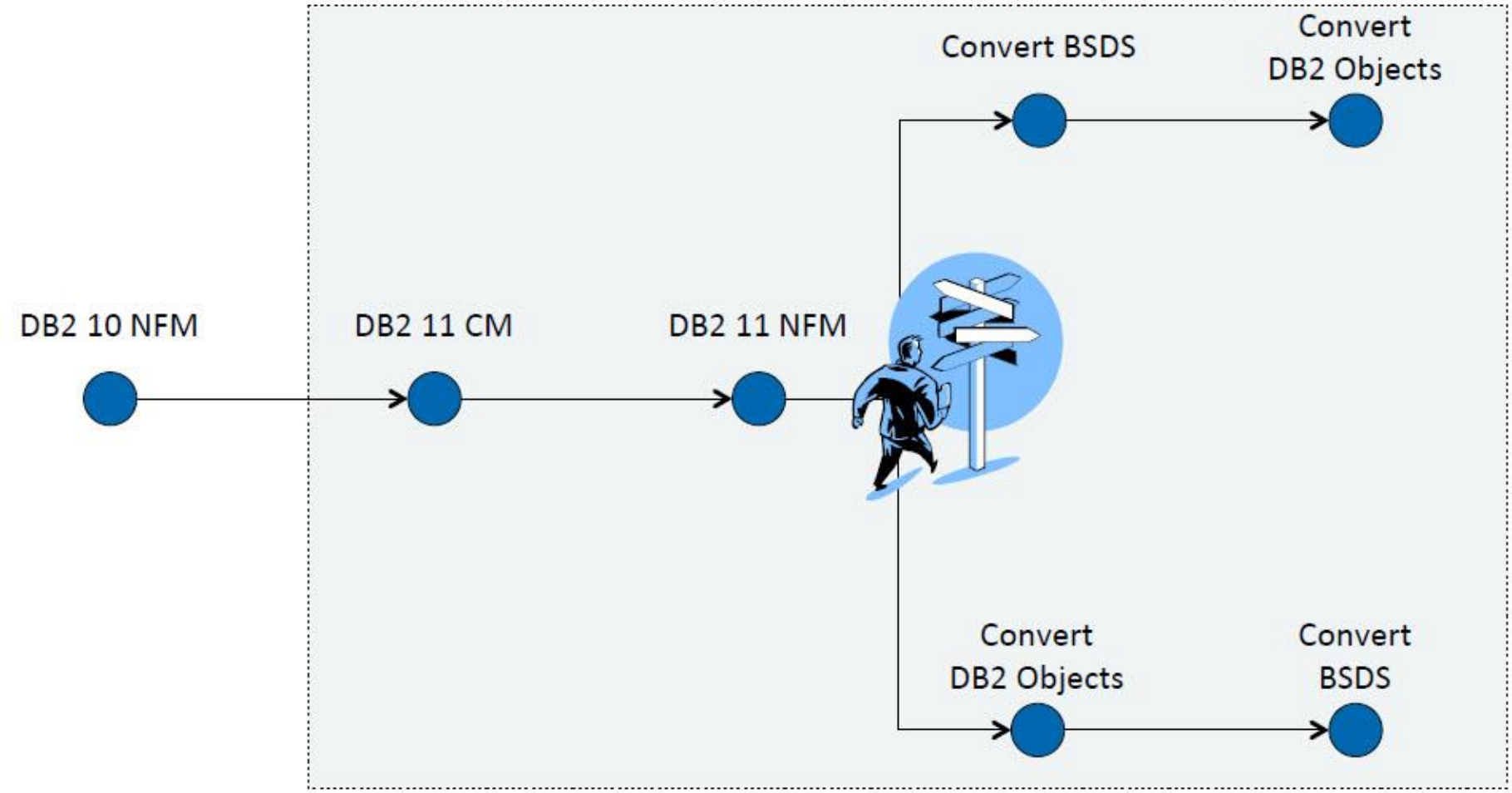
C	9	C	1	2	3	1	1	3	9	F	A

  - If the next valid LRSN value would be beyond valid range
    - UPDATE will be not possible
      - Rollback and abort processing are still allowed.
      - Return Code 00C2026D is issued
    - User Object is put in R/O mode until converted to EXTENDED format

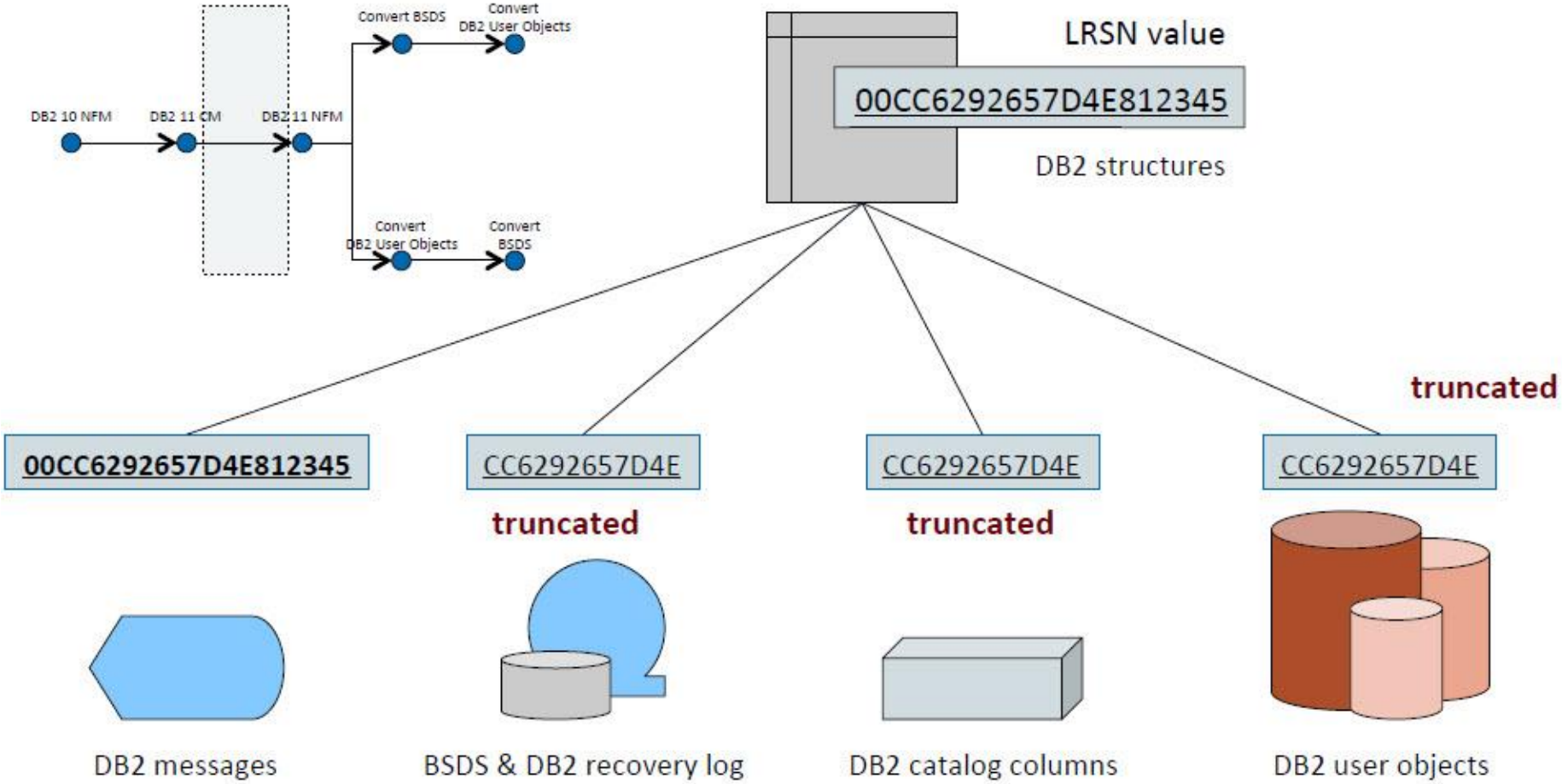
How to get there ?

CONVERSION PROCESS

DB2 11 Migration Overview (simplified)



LRSN values in DB2 11 CM mode



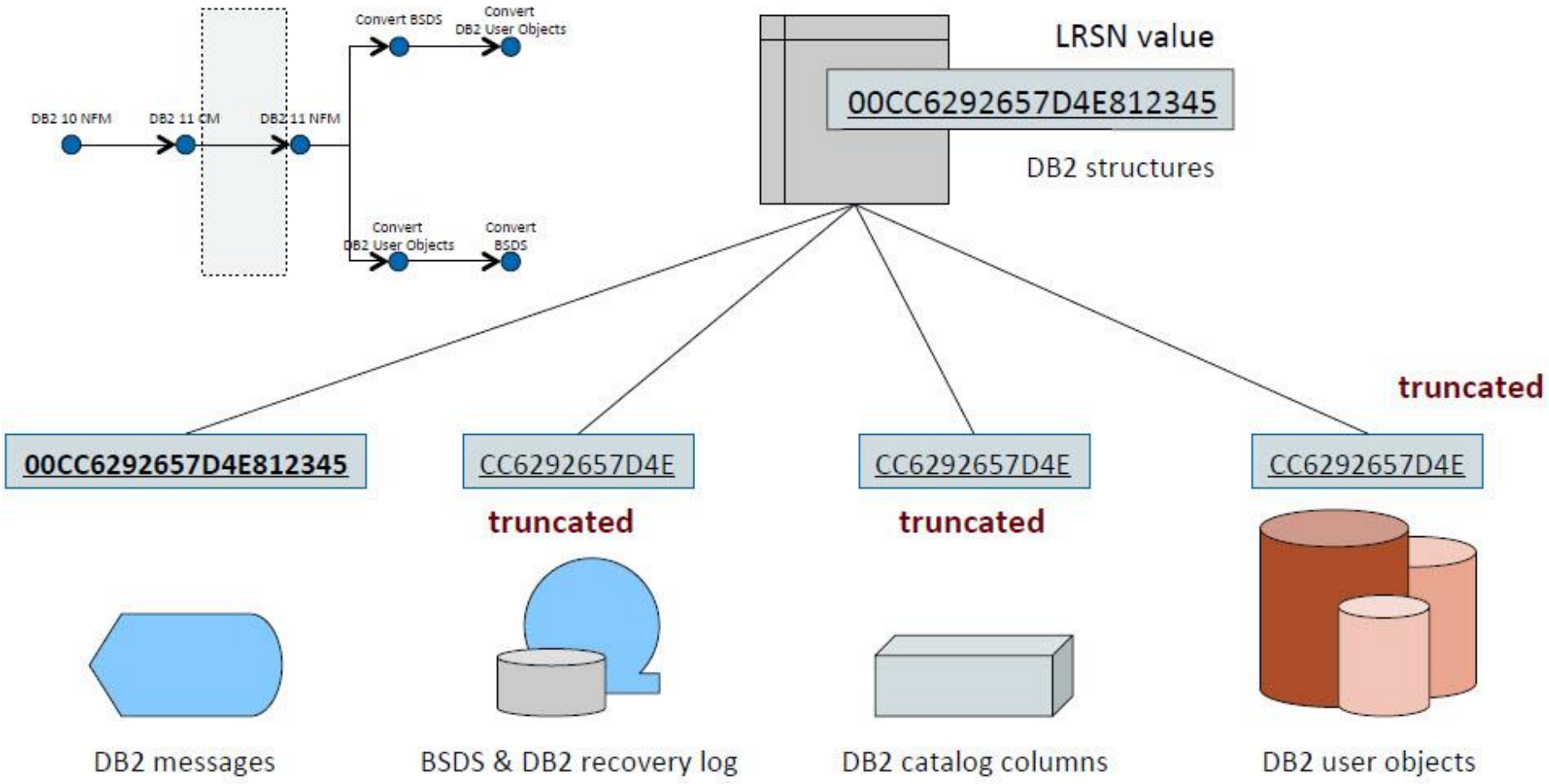
DB2 11 Coexistence with DB2 10

- Messages are generally in EXTENDED format as of CM
  - DB2 internally performs conversion between BASIC and EXTENDED
- APAR PM90175 & PM90247 for DB2 10 add sanity checks for provided RBA/LRSN support

```
RECOVER TABLESPACE DSNDB06.SYSTSTPT TOLOGPOINT X'00CB378D5A99'
DSNU510I   =D2T1 105 20:27:59.87 DSNUCASA - NO RECOVERY BASE AVAILABLE FOR RECOVERY OF
TABLESPACE DSNDB06.SYSTSTPT
DSNU017I       105 20:27:59.87 DSNUGBAC - UTILITY DATA BASE SERVICES MEMORY EXECUTION
ABENDED, REASON=X'00E40119' CAUSE=X'E4D9D4D3'
```

```
RECOVER TABLESPACE DSNDB06.SYSTSTPT  TOLOGPOINT X'00CB378D5A9900000000'
```

LRSN values in DB2 11 NFM – after job DSNTIJNF



New DB2 subsystem parameter

- **OBJECT_CREATE_FORMAT** subsystem parameter
  - Controls if DB2 uses BASIC or EXTENDED format when creating new table space or index objects
  - Acceptable values are
    - BASIC - new objects will be in 6-byte format
    - EXTENDED (default) - new objects will be in 10-byte format
- Before DB2 11 NFM, the setting of this ZPARM is ignored
  - BASIC format is always used
- WORKFILE objects
  - Will always be created in EXTENDED format!
    - No SYSPLEX QUERY PARALLELISM support between V10 and V11 exists if work files are involved

Converting the BSDS and LOG to EXTENDED format

- Run *DSNJCNVT* utility to convert the bootstrap data set (BSDS) and the recovery log
  - Can be run at any time after migration to DB2 11 NFM
  - In data sharing mode, BSDS of each member can be converted one at a time

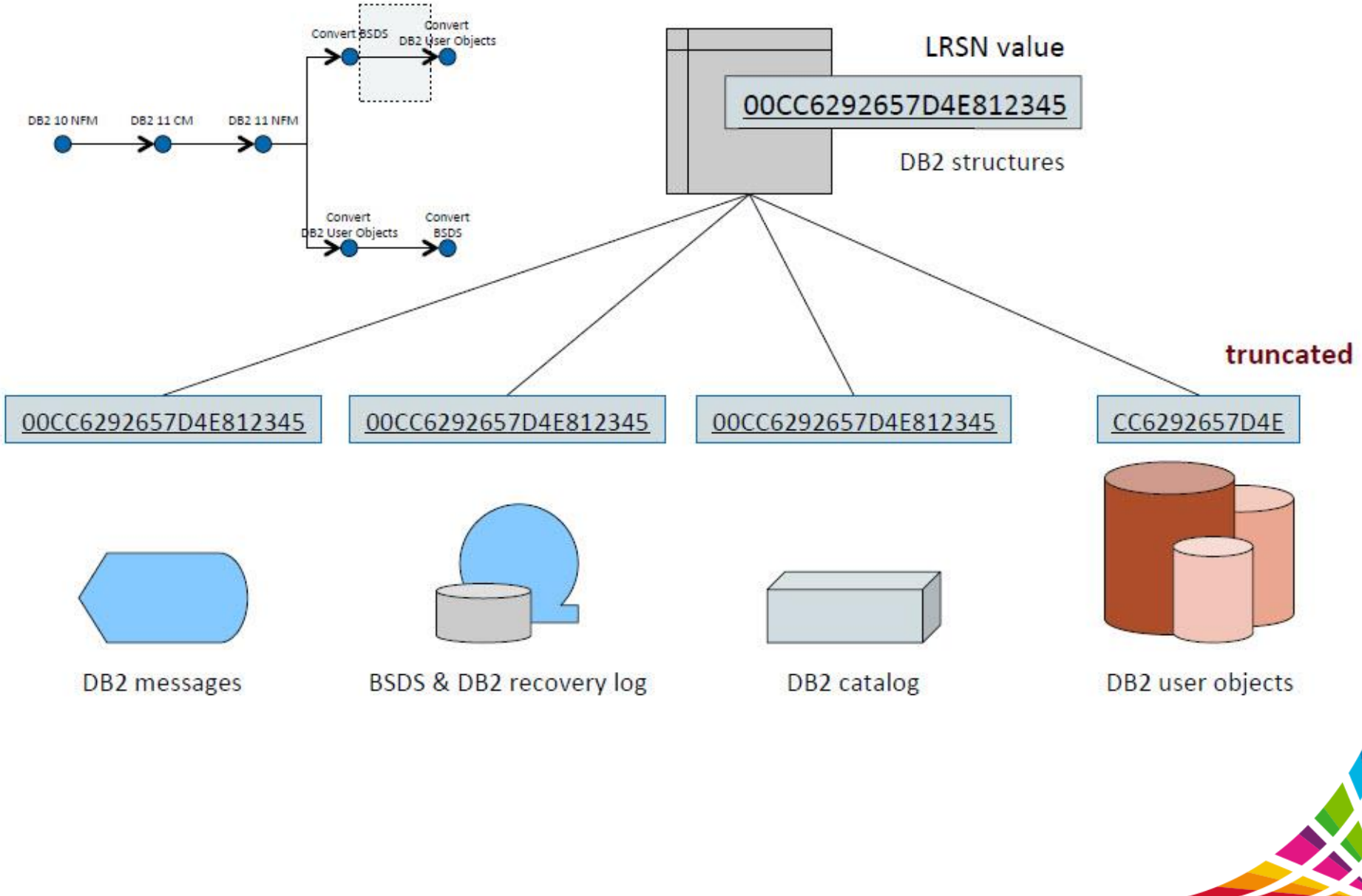
```
//CONVERT EXEC PGM=DSNJCNVT,REGION=64M
//SYSUT1 DD DSN=DB2A.OLD.BSDS01,DISP=SHR
//SYSUT2 DD DSN=DB2A.OLD.BSDS02,DISP=SHR
//SYSUT3 DD DSN=DB2A.BSDS01,DISP=OLD
//SYSUT4 DD DSN=DB2A.BSDS02,DISP=OLD
//SYSPRINT DD SYSOUT=*
```

```
CRCR convert started
DSNJ200I DSNJCNVT CONVERT UTILITY PROCESSING COMPLETED
SUCCESSFULLY FOR MEMBER 'xxxxxxxx'
```

Converting the BSDS and LOG to EXTENDED format …

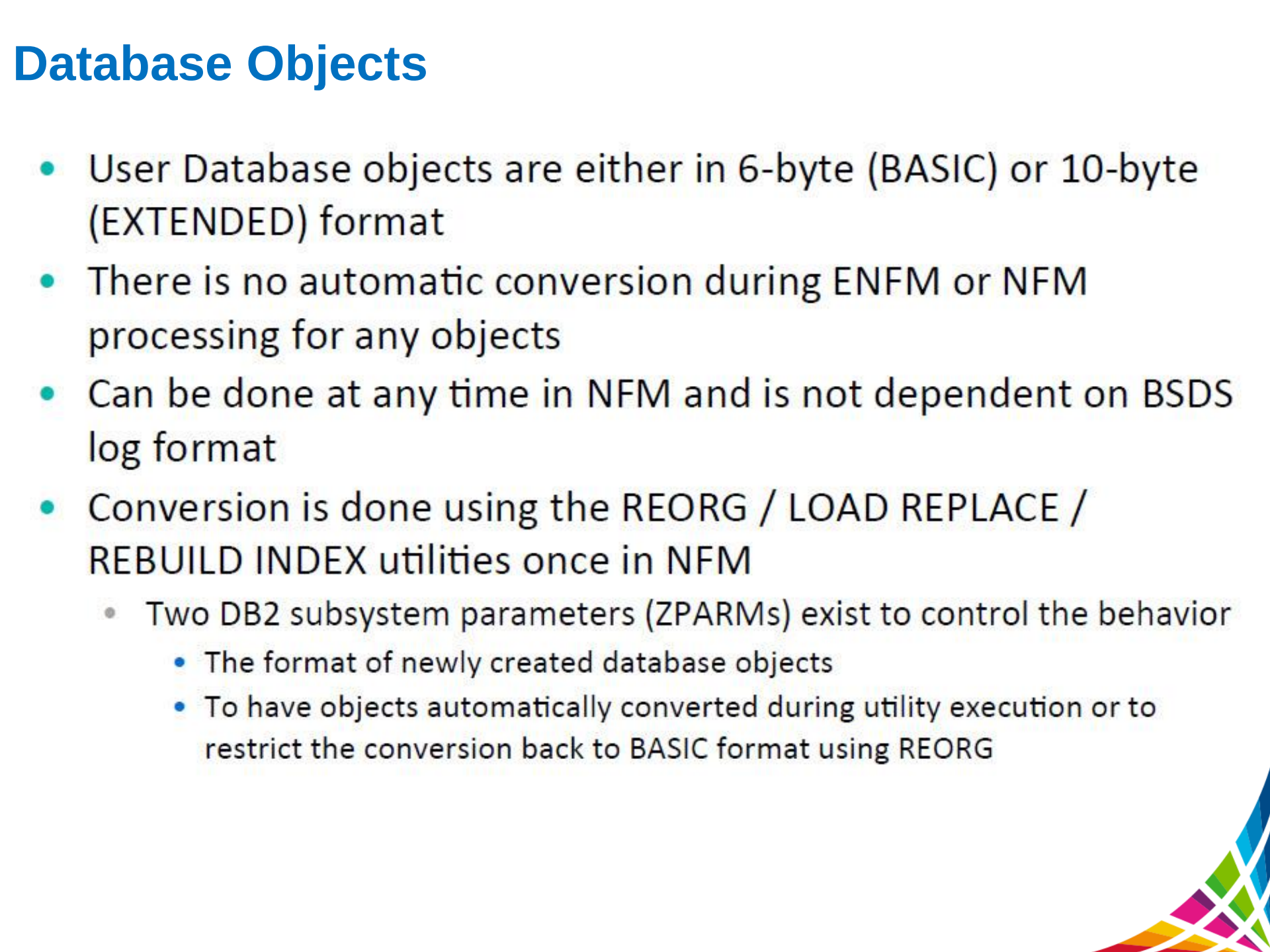
- Things to consider before running DSNJCNVT
  - Stop DB2 subsystem that owns the bootstrap data set
  - Any utility that reads from peer BSDS must be finished in data sharing
  - RACF user ID running DSNJCNVT must have read/write access on the new BSDSs , and read access on the old BSDSs
- Special considerations for Data Replication
  - Stop any data replication process to ensure successfully renamed and replaced BSDS
  - Best practice is to stop the data replication process first, then stop the DB2 subsystem
    - allows sharing systems to de-allocate the BSDSs when the state of the member changes to inactive.

DB2 11 NFM – only BSDS converted to EXTENDED



Database Objects

- User Database objects are either in 6-byte (BASIC) or 10-byte (EXTENDED) format
- There is no automatic conversion during ENFM or NFM processing for any objects
- Can be done at any time in NFM and is not dependent on BSDS log format
- Conversion is done using the REORG / LOAD REPLACE / REBUILD INDEX utilities once in NFM
  - Two DB2 subsystem parameters (ZPARMs) exist to control the behavior
    - The format of newly created database objects
    - To have objects automatically converted during utility execution or to restrict the conversion back to BASIC format using REORG



Converting DB2 user objects to EXTENDED format

- DB2 user-objects can be converted to EXTENDED by running
  - REORG TABLESPACE or REORG INDEX
  - LOAD REPLACE
  - REBUILD INDEX
- LISTDEF enhanced to filter object based on format

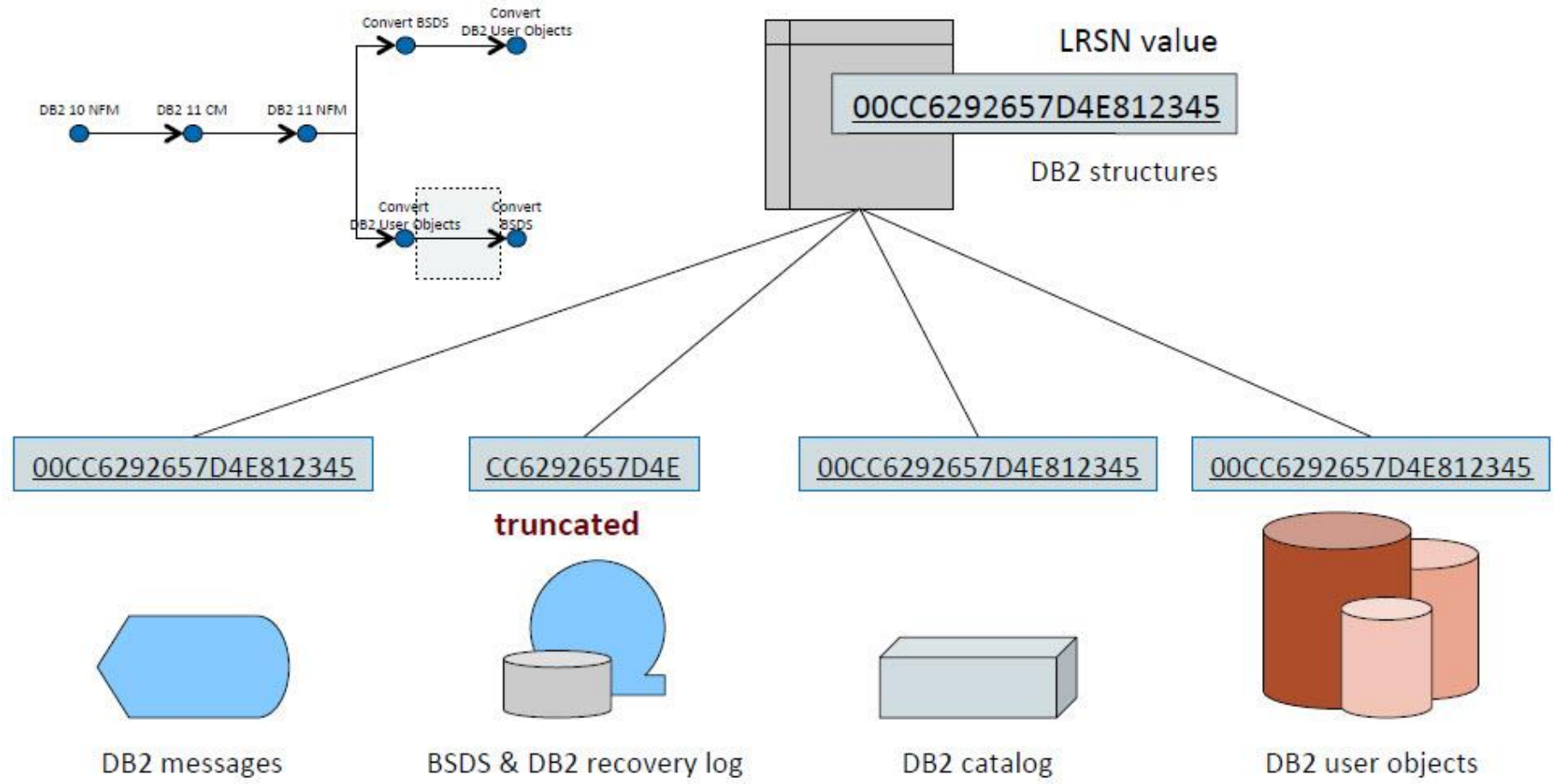
```
LISTDEF REORG_LIST
  INCLUDE TABLESPACE  DSNDB01.DBD01     BASIC YES
  INCLUDE TABLESPACE  DSNDB01.SPT01     BASIC YES
  INCLUDE TABLESPACE  DSNDB01.SCT02     BASIC YES

  REORG TABLESPACE LIST REORG_LIST SHRLEVEL CHANGE
        LOG NO COPYDDN(SYSCOPY)
        RBALRSN_CONVERSION EXTENDED
        RETRY 255 TIMEOUT TERM RETRY_DELAY 1 DRAIN_WAIT 1
        SORTDATA
```

New DB2 subsystem parameter …

- **UTILITY_OBJECT_CONVERSION** subsystem parameter
  - Controls if DB2 utilities that accept the RBALRSN_CONVERSION option will perform any format conversion for existing table spaces or indexes
  - The default behavior normally applies when the utility control statement does not specify the RBALRSN_CONVERSION option
  - Acceptable values are
    - BASIC - convert existing object to BASIC 6-byte format
    - EXTENDED - convert existing object to EXTENDED 10-byte format
      - Only allowed when **OBJECT_CREATE_FORMAT = EXTENDED**
    - NOBASIC - disables RBALRSN_CONVERSION = BASIC setting
    - NONE (default) - does not perform automatic conversion
- Before DB2 11 NFM, the setting of this ZPARM is ignored

DB2 11 NFM – only DB2 user objects converted to EXTENDED



DB2 catalog columns

- DB2 catalog and directory object definitions are changed from 6-byte to 10-byte format during ENFM processing as part of job ***DSNTIJEN***.
- The values stored in the DB2 catalog and directory objects can be 6-byte or 10-byte format
  - The format depends on which conversion step is executed in ENFM
    - Not a locked atomic operation
- In NFM, ENFM*, CM*, 10-bytes values are stored.
- DB2 catalog and directory table spaces and indexes are still in BASIC format after NFM conversion
  - Job ***DSNTIJCV*** can be used to convert catalog and directory to EXTENDED format once in DB2 11 NFM

Shared Communication Area (SCA) format

- The structure of the SCA is reformatted to handle the new 10-byte format
  - A rebuild is triggered during NFM processing as part of job **DSNTIJNF**
- This is not an optional step!
- A Group Restart can also trigger a rebuild of the SCA
  - When still in old format
  - And Data Sharing group is not in CM mode.

Bootstrap Data Set (BSDS)

- The BSDS can be converted at any time after NFM migration by running job ***DSNTIJCB*** or stand-alone utility ***DSNJCNVT***
- The owning DB2 member must be stopped before migration
  - In data sharing one member at a time can be migrated
- Mixed-mode of BSDSs is supported in a data sharing group
  - DB2 takes care of necessary conversions
  - Limitations of BASIC format will continue until all members are converted
    - E.g. LRSN spinning for subsequent transactions involving members with BASIC and EXTENDED log formats
- The BSDS conversion is a one way process
  - No fallback to 6-byte format possible
  - Truncation of log record information to 6-byte format not possible

Recovery Log

- In DB2 11 CM the recovery log content is identical to DB2 10
- DB2 internally operates with 10 byte format
  - Which will be truncated to 6-bytes format log records
- Once the BSDS is in EXTENDED format, DB2 Log Manager will generate 10-byte format log records
  - The BSDS must be converted to ensure enough room to store extended 10-byte format log records

Recovery considerations

- It is important that the disaster recovery process does not convert any objects to or from 10 byte extended RBA or LRSN format during the recovery and rebuild process.
  - If some objects are in extended 10-byte format
    - temporarily change the UTILITY_OBJECT_CONVERSION subsystem parameter to NONE before you begin a disaster recovery
    - do not specify the RBALRSN_CONVERSION keyword in the control statements.
  - After the disaster recovery is complete, change the UTILITY_OBJECT_CONVERSION subsystem parameter to its original value.
- DSN1COPY considerations
  - Will not update DB2 catalog tables to reflect actual format
    - Use of *REPAIR CATALOG TABLESPACE dbname.tsname* update values

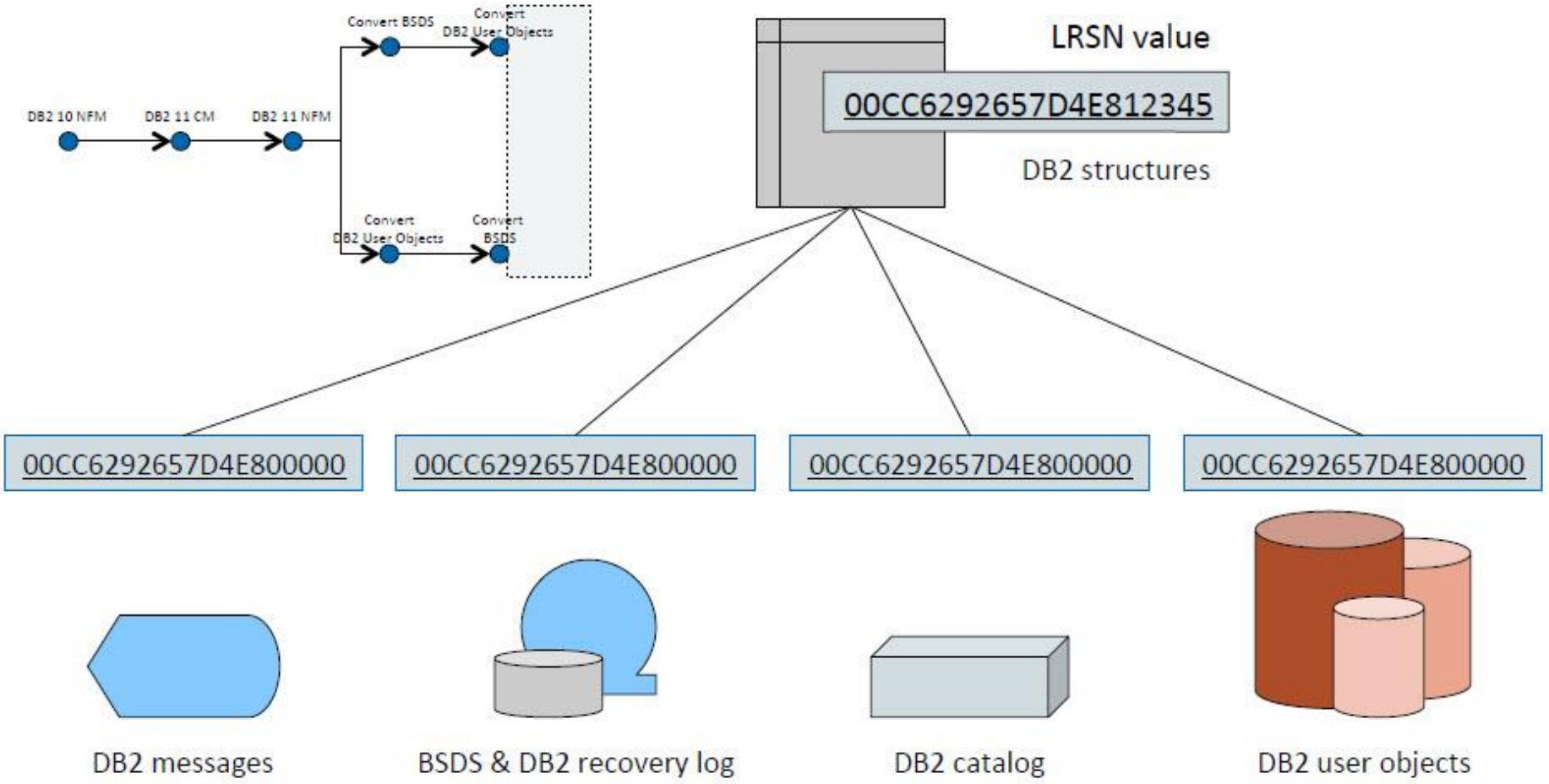
One more word on LRSN spinning

- With the extended 10-byte format in V11, the LRSN spin problem is basically eliminated
- But only after
  - BSDS converted to EXTENDED format
  - Associated user object is converted to EXTENDED format as well

Until you are in a true 10-byte DB2 11 System

- Once the BSDS is converted, DB2 system soft and hard limits are now based on the 10-byte format
  - Checked during startup and active log switches
- But once DB2 passes the end of the 6-byte log range
  - Any Update to a user object in BASIC format puts it into R/O mode
- Objects in BASIC format can be identified by
  - Checking RBA_FORMAT column in SYSTABLEPART and SYSINDEXPART
    - 'B' – BASIC format
    - 'E' – EXTENDED format
    - 'U' – DEFINE NO option used
    - 'blank' – for migrated objects
- Until all user objects are converted to EXTENDED
  - Monitor the current high RBA/LRSN value used by DB2
  - And check against the DB2 10 SOFT & HARD limits

True 10-byte format environment in DB2 11 NFM



Key Facts Summary

- DB2 11 is fundamentally a 10-byte format system as of CM
  - 3rd party tools and procedures must be compatible
- When you convert to EXTENDED format once in NFM is up to you
  - Mixed mode support available for DB2 Data Sharing
- Until in true EXTENDED format
  - Monitor your system against the V10 soft limit RBA/LRSN values


Thank You