Embrace The Future Today By Eliminating Deprecated Objects

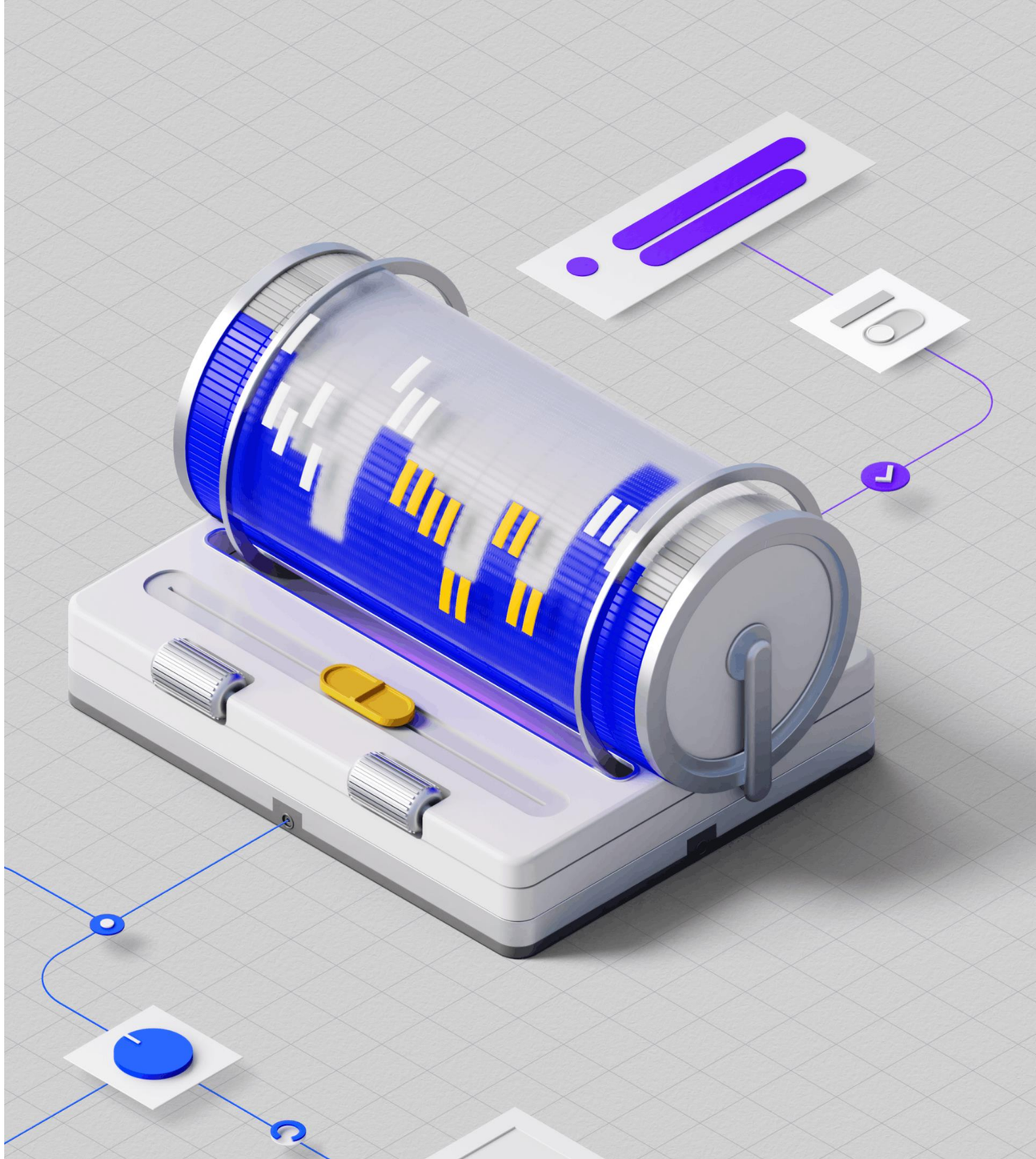
Central Canada Db2 Users Group

Mark Rader
Db2 for z/OS Specialist
IBM Z Worldwide System Center
mrader@us.ibm.com

Richard Ruppel
Brand Technical Specialist
Data and AI for IBM Z
IBM Technology - Americas
rmruppel@us.ibm.com

11 – 13 May 2026

IBM



SHARE
EDUCATE • NETWORK • INFLUENCE

Embrace The Future Today By Eliminating Deprecated Objects

Feb 25th, 2026

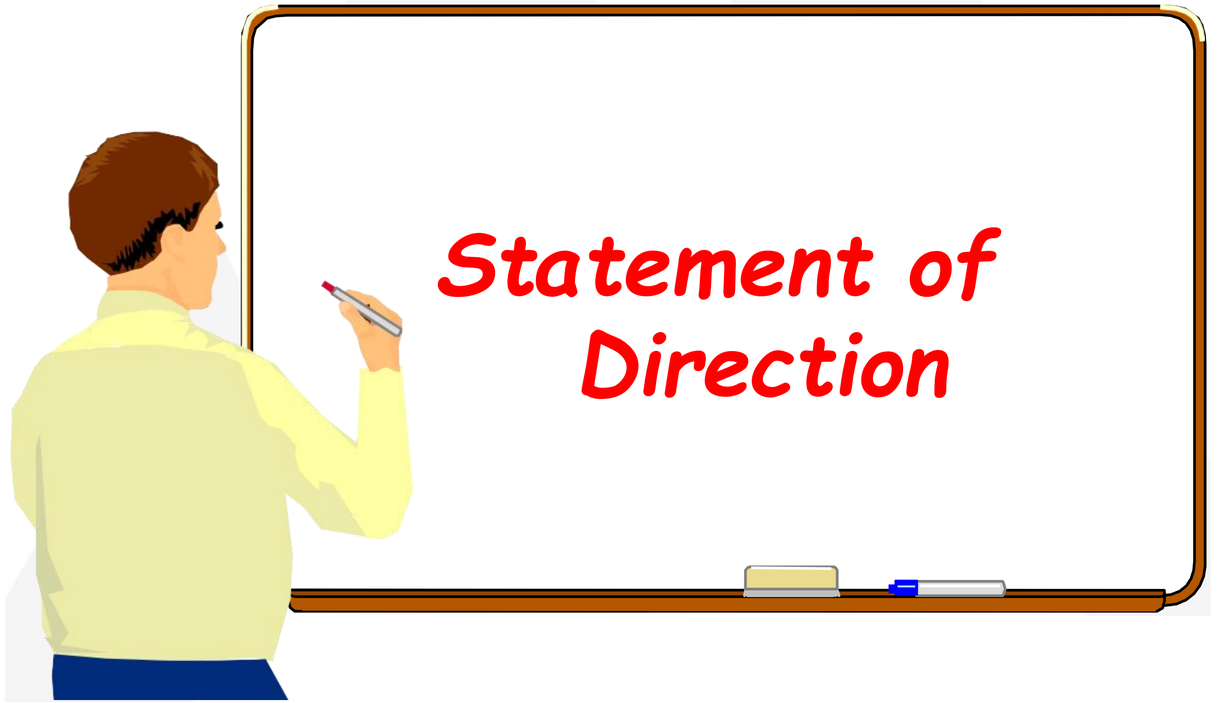
Craig Friske
Director, Software Engineering
Rocket Software
cfriske@rocketsoftware.com

John Lyle
Senior Db2 Developer
IBM
jlyle@us.ibm.com

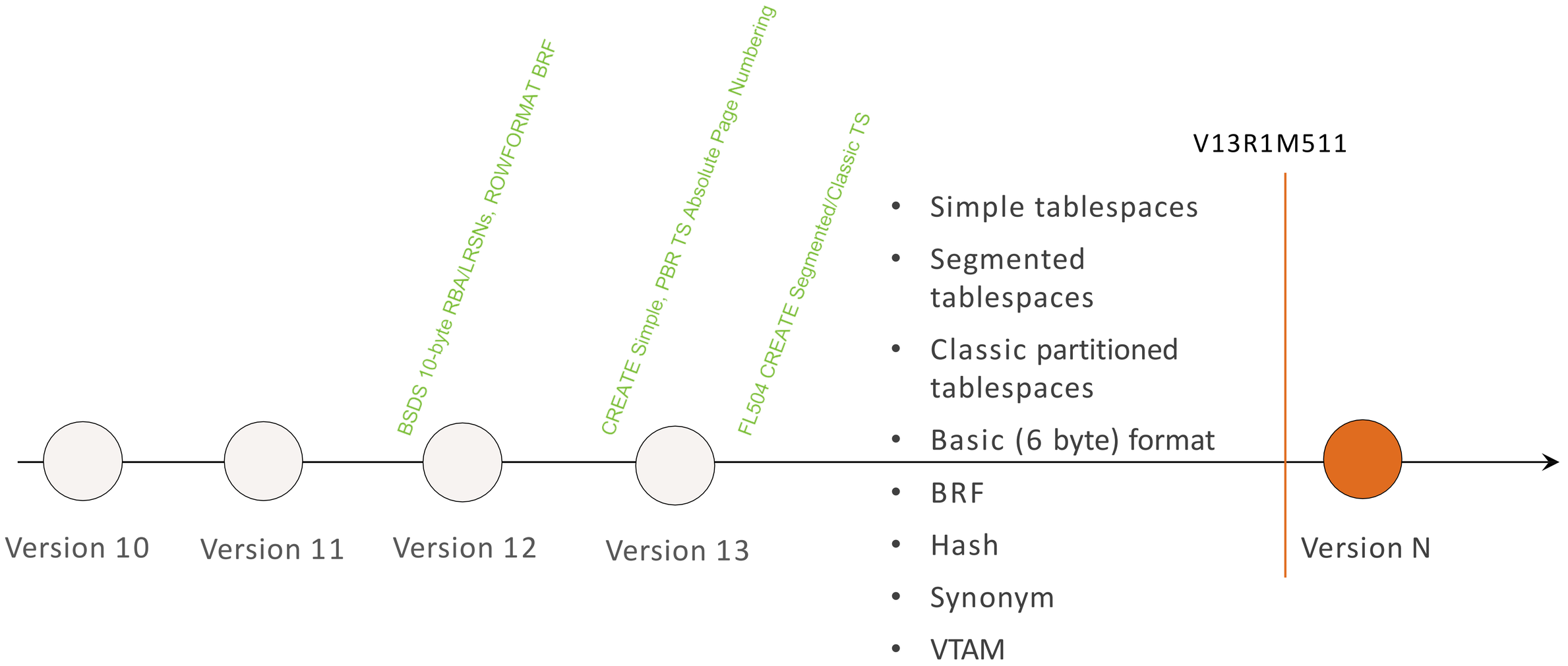
Copyright© by SHARE Association Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license. http://creativecommons.org/licenses/by-nc-nd/3.0/

Agenda

- IBM Db2 statement of direction
- Eliminating deprecation for Db2 objects
- Handling deprecated objects within the Db2 catalog and directory
- Assessing readiness for FL511 and beyond
- Summary


Statement of
Direction

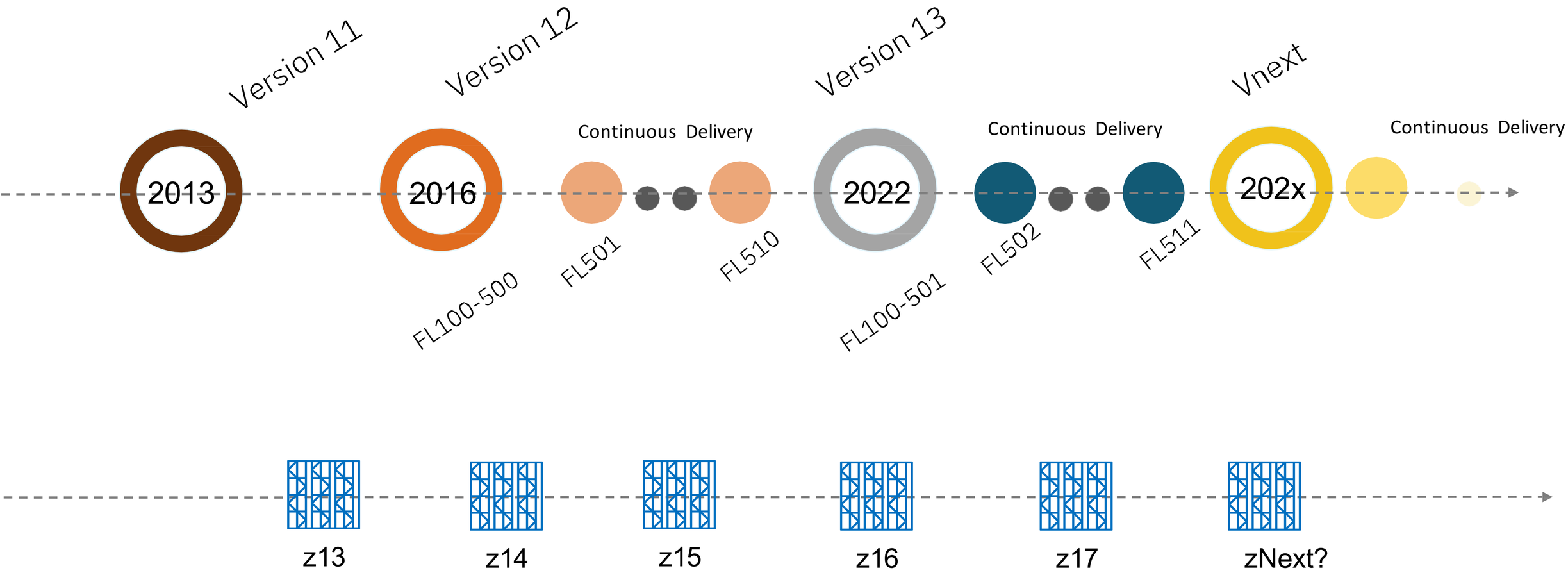
Db2 for z/OS Past, Present and Future

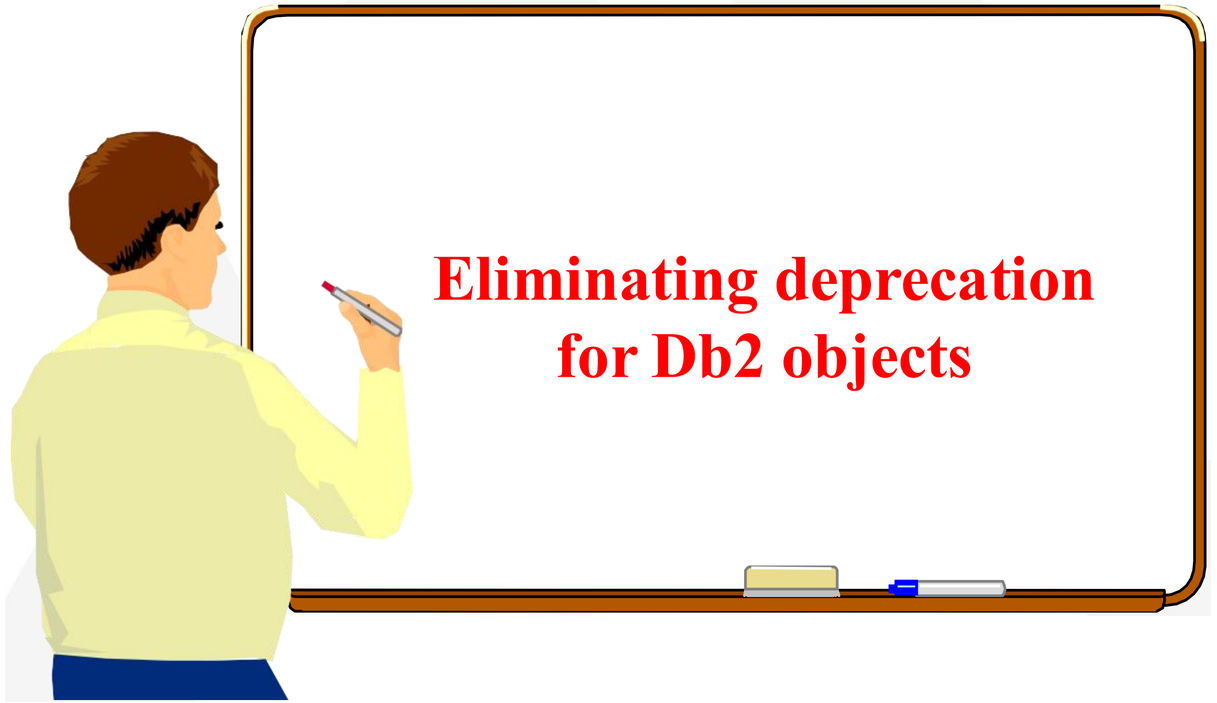


Major Theme of Db2 Vnext

- New foundation for future innovation
- Simplification and automation
- Scalability and performance

Db2 for z/OS Release Plan




Eliminating deprecation
for Db2 objects

Deprecation Overview

- Table spaces with Basic Row Format (BRF)
- Table spaces with 6-byte RBA/LRSNs
- Simple table spaces
- Segment table spaces
- Classic partitioned table spaces
- Hash organized tables
- Synonyms

Converting BRF to Reorder Row Format (RRF)

- Why convert? Performance and logging reduction
  - For example - SELECT with WHERE and column matching

- Identify all user table spaces
  - SYSTABLEPART Column FORMAT (‘’ – BRF, ‘R’ – RRF)
  - Not for LOB or XML table spaces
  - EDITPROCs and FIELDPROCS must be dropped

- REORG or LOAD without syntax option ROWFORMAT BRF
  - Specify ROWFORMAT RRF or take the default
  - No zparm Db2 12 and beyond

Find BRF Table Spaces, Edit/Validation Routines

```
SELECT DISTINCT A.DBNAME, A.TSNAME, A.FORMAT, A.TYPE, B.TYPE AS TBTYPE (table, view, clone)
    FROM SYSIBM.SYSTABLEPART A, SYSIBM.SYSTABLES B
  WHERE A.DBNAME=B.DBNAME
    AND A.TSNAME=B.TSNAME
    AND (A.FORMAT = ' ')                    Format is BRF (i.e. not 'R')
    AND (A.TYPE <> 'O')                     Not a LOB table space
    AND (A.TYPE <> 'P')                     Not an XML table space
```

```
SELECT DBNAME, TSNAME, NAME, EDPROC, VALPROC
    FROM SYSIBM.SYSTABLES
  WHERE EDPROC <> ''                        Edit Proc
        OR VALPROC <> ''                    Validation Proc
```

RRF Performance Boost

CREATE TABLE TB1 (C1 INTEGER, C2 VARCHAR(10), C3 CHAR(10), C4 VARCHAR(20)

Basic Row Format

C1	C2	C3	C4
8000000A	0007 WILSON	ANDREW	0009 SAN JOSE

2 bytes length

Reordered Row Format

C1	C3	O2	O4	C2	C4
8000000A	ANDREW	14	1B	WILSON	SAN JOSE

Offset to C2 Offset to C4

SELECT C4 FROM t1 WHERE C3 = ‘ANDREW’ AND C4 = ‘SAN JOSE’

Converting 6-byte to 10-byte RBA/LRSNs

- Why convert?
  - Keep individual objects updateable!
  - BSDS conversion for the system in Db2 12

- Identify all user table spaces
  - SYSTABLEPART Column RBA_FORMAT for table spaces
    - ‘E’ (10 bytes), ‘B’ (6 bytes), ‘U’ (undefined), or ‘ ‘ (6 bytes before V10)

- REORG, LOAD REPLACE, REBUILD INDEX without syntax option RBALRSN_CONVERSION
  - Specify RBALRSN_CONVERSION EXTENDED, remove for the default
  - No zparm Db2 12 and beyond

Find All 6-byte RBA/LRSN Objects

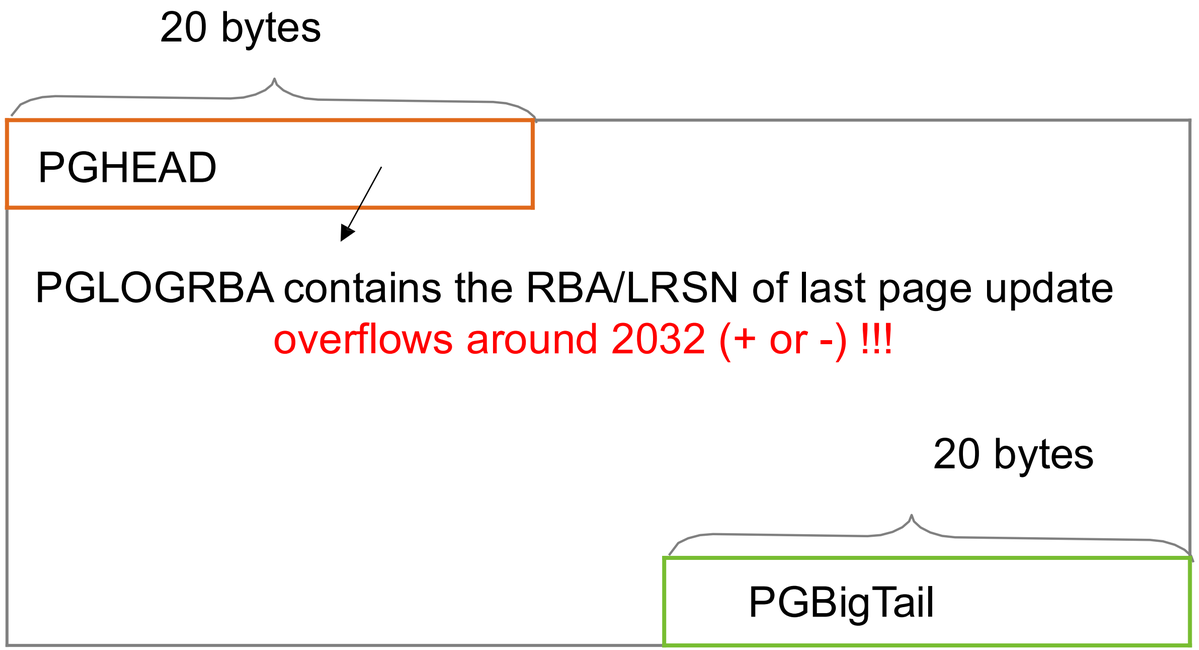
Identify all user table spaces

```
SELECT DISTINCT A.DBNAME, A.TSNAME, A.RBA_TYPE, A.TYPE, B.TYPE AS TBTYPE
    FROM SYSIBM.SYSTABLEPART A, SYSIBM.SYSTABLES B
  WHERE A.DBNAME=B.DBNAME
    AND A.TSNAME=B.TSNAME
    AND ((A.RBA_FORMAT = 'B')          Format is 6-byte RBA/LRSN
      OR (A.RBA_FORMAT = ' '))         Format is not 10-byte RBA/LRSN value of 'E'
```

Identify all user index spaces

```
SELECT DISTINCT IXNAME, RBA_FORMAT
    FROM SYSIBM.SYSINDEXPART
    WHERE RBA_FORMAT = 'B'             Format is 6-byte RBA/LRSN
     OR RBA_FORMAT = ' '               Format is not 10-byte RBA/LRSN value of 'E'
```

Page RBA/LRSN Detail



- New page format to accommodate the 10-byte RBA/LRSN in pages.
- No room in PGHEAD to extend the current PGLOGRBA
- **Extended page format has a 'big tail' aka PGBigTail, which has PGBigRBA to hold a 10-byte RBA/LRSN**

At the 6-byte limit, updates will fail with 00C2026D, and objects will be read only until they are converted to 10-byte support

Why Convert to UTS?

- **Performance and Scalability:**
  - **Efficient Mass Deletes:** Segmented space management within partitions allows for fast mass deletes, as Db2 can quickly free up space without scanning the entire table.
  - **Reduced Locking Contention:** UTS supports "currently committed" locking, which improves concurrency by preventing applications from waiting on uncommitted row deletions.
  - **Large Table Handling:** UTS PBR RPN allows for massive table sizes and efficient, partition-level operations.

- **Modern Feature Support:** Many new Db2 for z/OS features, such as LOB in-lining (storing small LOBs in the base table), XML multi-versioning, and advanced compression techniques (Huffman), are only supported within UTS.

- **Ease of Management:** UTS enables automatic partition growth (PBG), simplified data reorganization by partition(s), and improved space management for variable-length rows

Why Convert to UTS?

Advanced Schema Flexibility (Pending DDL): UTS allows for complex deferred ALTER statements to be executed as "pending DDL". These changes do not require immediate downtime and can be applied during an online REORG, reducing maintenance windows. The ALTERs can be "stacked" so they are applied within a single REORG TABLESPACE execution.

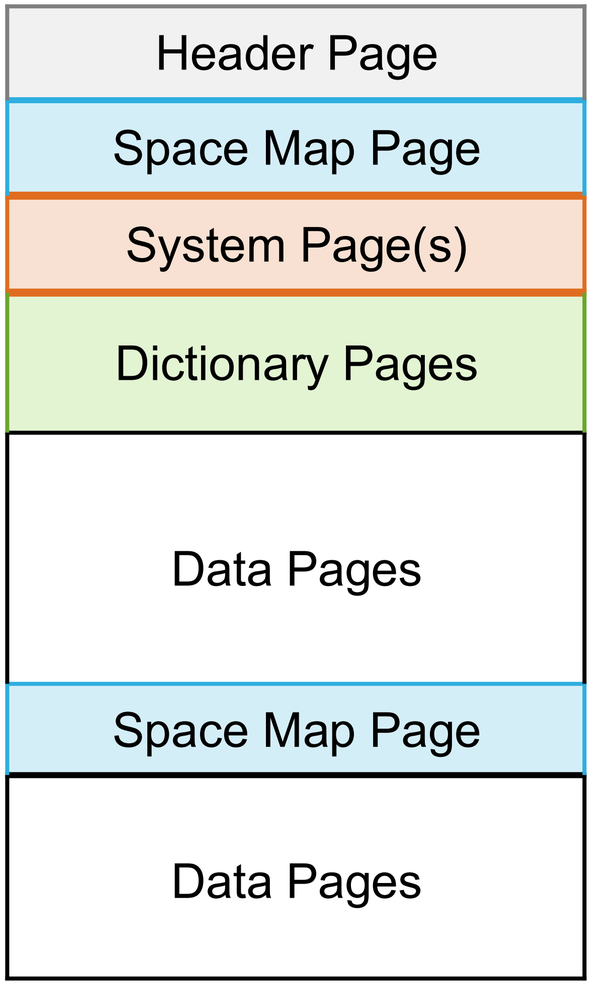
Consider the following features supported:

- **Adding a column** (immediate or pending)
- **Dropping a column**
- **Changing a partition limit key**
- **Inserting a partition** in the middle of a range-partitioned table
- **Converting PBG to PBR**
- **Converting PBR to PBG to PBR**
  - Allows dropping a partition
  - Allows changing the partitioning columns

Universal Table Spaces (UTS)

- UTS table spaces can be PBG or PBR
  - **P**artition **B**y **G**rowth adds partitions, when needed, with a MAXPARTITIONS limit
  - **P**artition **B**y **R**ange fixed number of partitions with limit keys

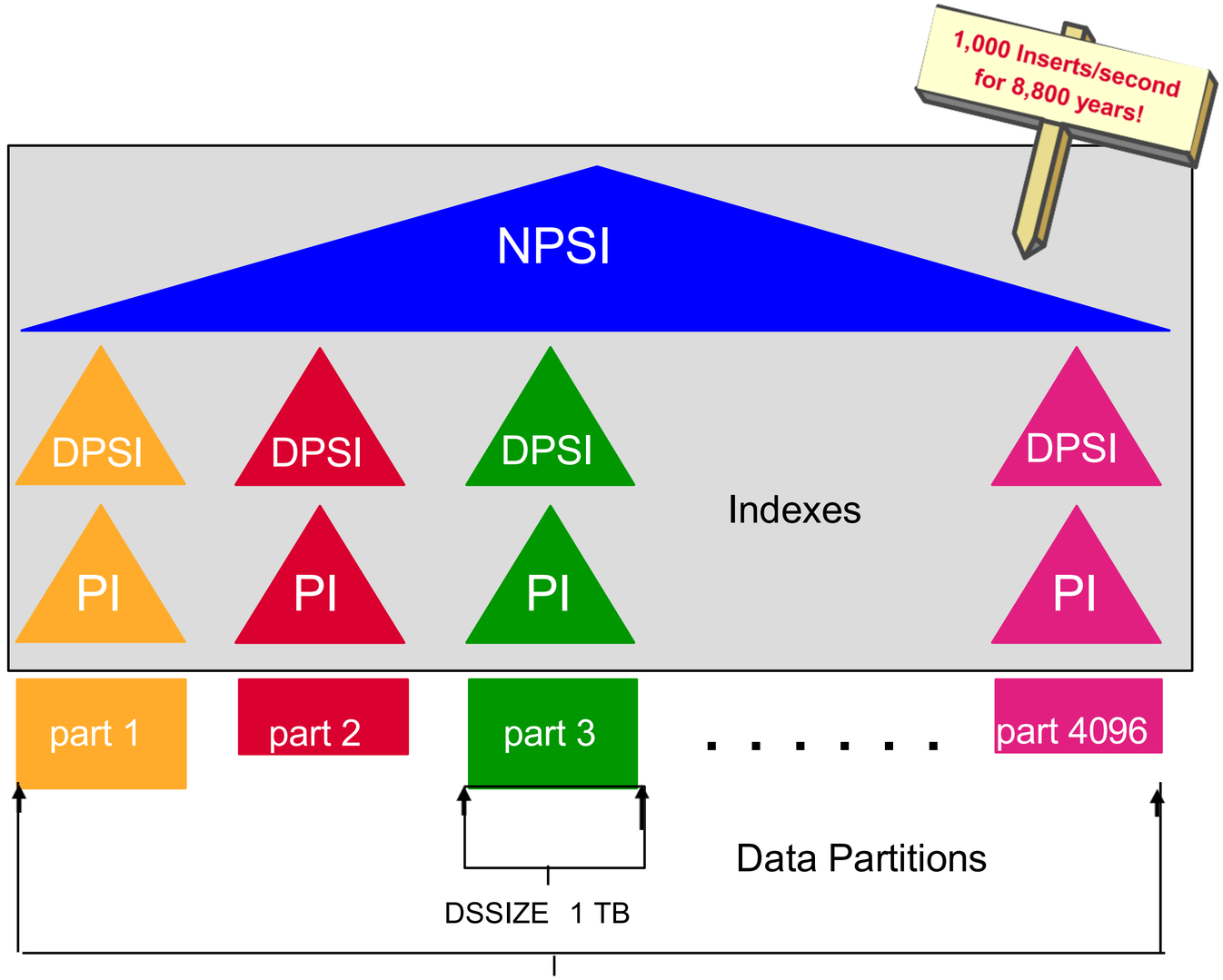
- Both are partitioned AND segmented
  - Header Page per partition – TS attributes
  - Space Map Pages – quick row processing
    - “segments” of pages
  - System Page – self describing
  - Dictionary – Ziv/Lempl or Huffman
  - Data pages
  - Single table per table space



Db2 9
2007

PBR Relative Page Numbering

PBR RPN (aka PBR2)



Db2 12
2016

Max Size = 4 Petabytes or 280 Trillion Rows

V12R1M504 CREATE Deprecation

Table space type	APPLCOMPAT(V12R1M504) and higher	APPLCOMPAT(V12R1M503) and lower
Partition-by-growth	Any of the following combinations: – MAXPARTITIONS and NUMPARTS – MAXPARTITIONS – Omit both	Any of the following combinations: – MAXPARTITIONS and NUMPARTS – MAXPARTITIONS and SEGSIZE *n*[1] – MAXPARTITIONS
Partition-by-range	NUMPARTS only	NUMPARTS and SEGSIZE *n*[1]
Segmented (non-UTS)	Not supported[2]	One of the following combinations: – SEGSIZE *n*[1] – Omit MAXPARTITIONS, NUMPARTS, and SEGSIZE
Partitioned (non-UTS)	Not supported[2]	NUMPARTS and SEGSIZE 0

(link)

Converting Simple or Segmented to PBG

- SYSTABLESPACE column TYPE
  - (‘O’- Lob, ‘P’ – XML, ‘G’ – PBG, ‘R’ – PBR, ‘L’ – Classic Partitioned “LARGE”))

- Find non-UTS simple and segmented table spaces

```
SELECT DBNAME, NAME, TYPE, SEGSIZE
   FROM SYSIBM.SYSTABLESPACE
   WHERE PARTITIONS = 0                Non-partitioned
   AND (TYPE = ‘ ’)                    Classic Partitioned, Simple, or Segmented
```

- Convert with two steps to make it a partitioned PBG
  1. ALTER TS MAXPARTITIONS n — Max partitions 1 up to 4096
  2. REORG TABLESPACE name — For each table space found (not needed for DEFINE NO objects)

Converting Classic Partitioned to PBR

- SYSTABLESPACE column TYPE (‘O’- Lob, ‘P’ – XML, ‘G’ – PBG, ‘R’ – PBR)

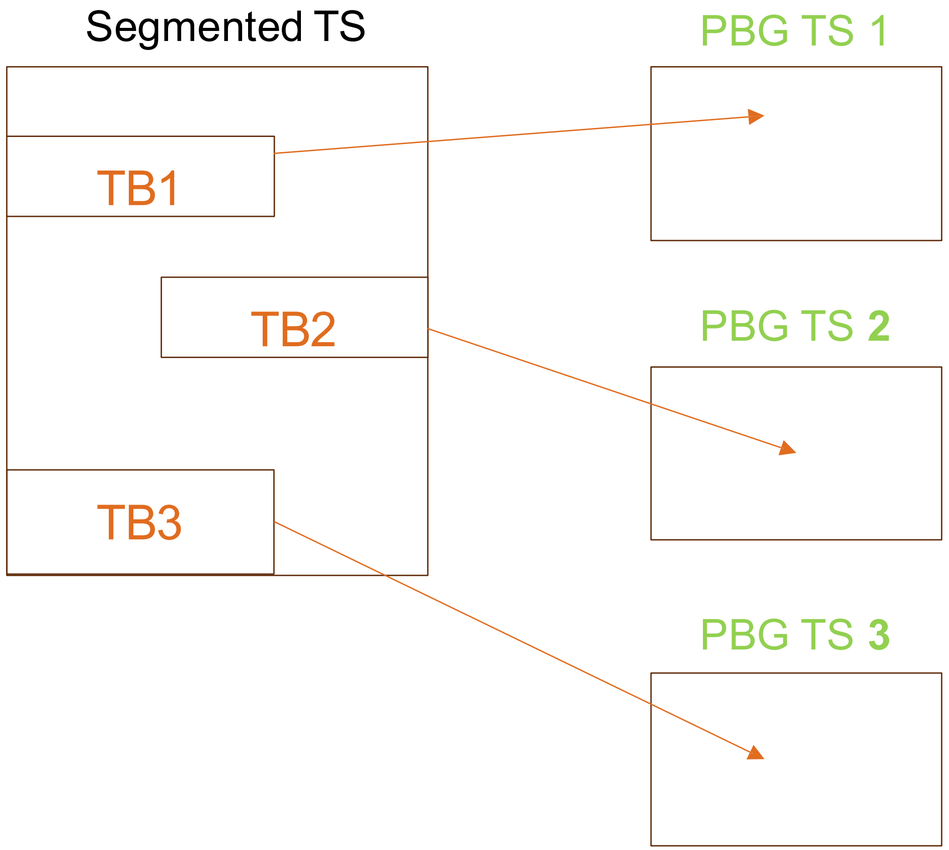
- Find non-UTS simple or segmented table spaces

```
SELECT DBNAME, NAME, TYPE, SEGSIZE
  FROM SYSIBM.SYSTABLESPACE
  WHERE PARTITIONS <> 0          Partitioned
  AND SEGSIZE = 0                Not segmented
  AND (TYPE = ' '                Classic Partitioned, Simple, or Segmented
    OR TYPE = 'L')               Classic Partitioned Large (254 partitions, 5 byte RIDs)
```

- Convert with two steps using pending DDL to make segmented
  1. ALTER TS SEGSIZE n — Specify number of pages per segment
  2. REORG TABLESPACE **NAME** — For each table space found (not needed for DEFINE NO objects)

Converting Multi-Table Segmented to PBG

1. Create a new PBG table space
2. ALTER TABLE <tb in multi-table> MOVE TABLESPACE <new PBG ts>
3. REORG segmented table space
4. Repeat for each table



Moving tables from multi-table table spaces to partition-by-growth table spaces
https://www.ibm.com/docs/en/db2-for-zos/13.0.0?topic=ats-moving-tables-from-multi-table-table-spaces-pbg-table-spaces

Eliminating Hash Tables

- Why? Fast Traversal Block (FTB) mitigates hash performance advantage!

- Identify all table spaces

```
SELECT DISTINCT B.DBNAME, B.TSNAME, B.NAME, B.STATUS
    FROM SYSIBM.SYSTABLESPACE A, SYSIBM.SYSTABLES B
  WHERE A.DBNAME=B.DBNAME
   AND A.NAME=B.TSNAME
   AND A.ORGANIZATIONTYPE = 'H'
```

Find all table spaces with hash organization

- Identify unique constraints to drop;

```
SELECT CONSTNAME, TBNAME FROM SYSIBM.SYSCONST
  WHERE TBNAME = B.NAME;
```

When B.STATUS = ‘X’

- Convert with 2 or 3 steps as follows:
  1. ALTER TABLE B.NAME DROP PRIMARY KEY;
  2. ALTER TABLE B.NAME DROP ORGANIZATION;
  3. REORG TABLESPACE B.TSNAME

When B.STATUS = ‘X’..
Sets the table space in REORG (unavailable)
REORG makes the table available again

Converting Synonyms to Aliases

- Why? Portability, flexibility, accessibility (authorization)
  - Synonyms already replaced with Aliases in V12R1FL504

- Find Synonyms

```
SELECT NAME, TBNAME
    FROM SYSIBM.SYSSYNONYMS
```

- Eliminate by dropping

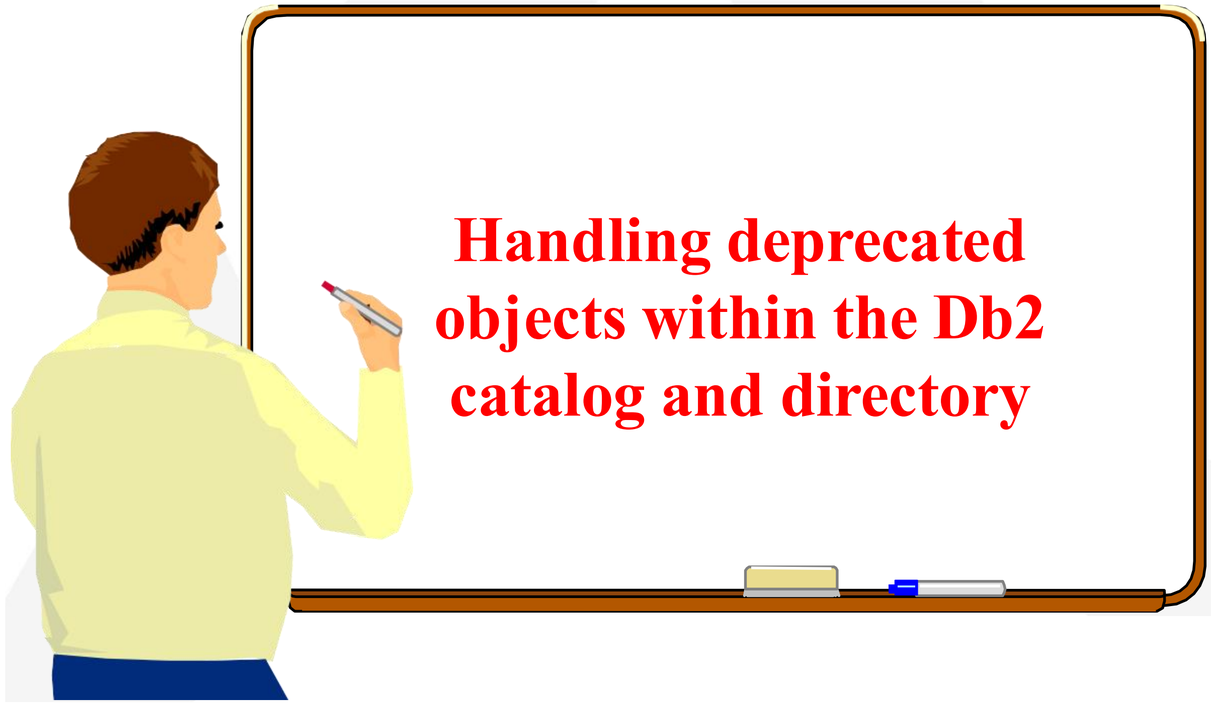
```
DROP SYNONYM NAME
```

For each synonym returned

- Replace with Aliases

```
CREATE ALIAS NAME ….
```

Alias supported for many different items referenced


Handling deprecated objects within the Db2 catalog and directory

UTS Conversion of Catalog and Directory

- UTS Conversion of catalog and directory objects began in D**B**2 10 and continued in D**b**2 11

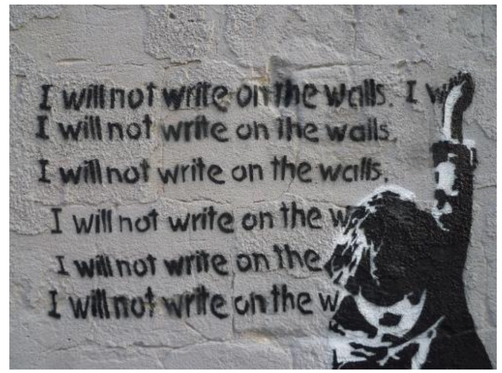
  - Conversions done via ENFM processes

- Conversion process not (yet) completed/finished

  - 21 catalog and directory table spaces currently non-UTS

UTS Conversion of Catalog and Directory

- Why?
  - Adopting universal table spaces across the board for catalog and directory objects means that Db2 can then take advantage of new performance, availability and usability features that have been added exclusively for universal table space objects.
  - Everyone needs to stop using deprecated objects.
  - And…..



UTS Conversion of Catalog and Directory

- What should we all do and when?
  - Finish converting user objects to UTS!
    - Some customers have not completed converting their non-UTS objects to UTS because they see that the Db2 catalog and directory still has deprecated objects
    - The day will come when deprecated object support will be removed 

    - The sooner the UTS conversions are done, the better

UTS Conversion of Catalog and Directory

- **Actively** working to convert remaining catalog and directory objects to UTS (PBG)
  - Flexible
  - No ENFM type process for conversions
  - FL508 creates new objects to prep for conversion
    - Multi-table table spaces must each get their own PBG TS
    - -DIS GROUP UTSCONVERSION
  - FL509 allows conversion of the remaining objects

Identify Catalog/Directory Objects

- -DIS GROUP UTSCONVERSION

New keyword for DIS GROUP

```
---------------------------------------
UTS CONVERSION REPORT
---------------------------
TABLE                UTS
SPACE                CONVERSION
-----------          ----------------------
SCT02                INELIGIBLE
SYSUTILX             INELIGIBLE
SYSALTER             INELIGIBLE
SYSCONTX             INELIGIBLE
SYSDDF               INELIGIBLE
SYSEBCDC             INELIGIBLE
…
SYSSEQ               INELIGIBLE
SYSSEQ2              INELIGIBLE
SYSSTATS             INELIGIBLE
…
SYSUSER              INELIGIBLE
SYSXML               INELIGIBLE
---------------------------------------
```

<= FL508

21 objects (not all shown)
Before FL509, they are shown as “ineligible” for conversion

Run CATMAINT for FL508 or FL509

- CATMAINT UPDATE LEVEL V13R1M509
- -ACTIVATE FUNCTION LEVEL(V13R1M509)
- -DIS GROUP DETAIL

Get appropriate catalog level needed
Activate function level needed

```
*** BEGIN DISPLAY OF GROUP(DSNMSG  ) CATALOG LEVEL(V13R1M508)
                  CURRENT FUNCTION LEVEL(V13R1M509)
                  HIGHEST ACTIVATED FUNCTION LEVEL(V13R1M509)
                  HIGHEST POSSIBLE FUNCTION LEVEL(V13R1M509)
                  PROTOCOL LEVEL(2)
                  GROUP ATTACH NAME(DSNM)
-------------------------------------------------------------------
DB2          SUB                       DB2    SYSTEM    IRLM
MEMBER   ID  SYS  CMDPREF   STATUS     LVL    NAME      SUBSYS IRLMPROC
-------- --- ---- --------  -------- ------ --------  ----   --------
DB1M       1 DB1M DB1M      ACTIVE   131509 MVS1      IR1M   DB1MIRLM
-------------------------------------------------------------------
MIGRATION READINESS REPORT
--------------------------
DB2       CODE       SPE      MIGRATION
MEMBER    LEVEL      APAR     ELIGIBLE
--------  ---------  -------  ---
DB1M      V13R1M509  .......  NO
MIGRATION READINESS STATUS: GROUP IS NOT READY FOR DB2 NEXT
REASON: HIGHEST ACTIVATED FUNCTION LEVEL NOT V13R1M511
***
```

Catalog level 508 has PBG objects needed

Identify Catalog/Directory Objects

- -DIS GROUP UTSCONVERSION

```
---------------------------------------
UTS CONVERSION REPORT
---------------------------
TABLE                UTS
SPACE                CONVERSION
-----------          ----------------------
SCT02                ELIGIBLE
SYSUTILX             ELIGIBLE
SYSALTER             ELIGIBLE
SYSCONTX             ELIGIBLE
SYSDDF               ELIGIBLE
SYSEBCDC             ELIGIBLE
…
SYSSEQ               ELIGIBLE
SYSSEQ2              ELIGIBLE
SYSSTATS             ELIGIBLE
…
SYSUSER              ELIGIBLE
SYSXML               ELIGIBLE
---------------------------------------
```

>= FL509

21 objects (not all shown)
At FL509 they are shown as “eligible” for conversion

REORG Each Object to Convert

- REORG with new CONVERTUTS option
  - REORG TABLESPACE db.ts CONVERTUTS
  - Changes to RRF and PBG with MAXPARTITIONS 1

- Recommendations
  - REORG SHRLEVEL CHANGE
  - Only REORG one object at a time
  - Pick a period of low activity on the system

- Example

```
REORG TABLESPACE DSNDB06.SYSDDF CONVERTUTS
REORG TABLESPACE DSNDB06.SYSCONTX CONVERTUTS
REORG TABLESPACE DSNDB01.SYSUTILX CONVERUTS
```

Identify Catalog/Directory Objects

- -DIS GROUP UTSCONVERSION

```
---------------------------
UTS CONVERSION REPORT
---------------------------
TABLE                UTS
SPACE                CONVERSION
----------           --------------------
SYSUTILX             COMPLETE
SYSCONTX             COMPLETE
SYSDDF               COMPLETE
SCT02                ELIGIBLE
SYSALTER             ELIGIBLE
SYSEBCDC             ELIGIBLE
…
SYSSEQ               ELIGIBLE
SYSSEQ2              ELIGIBLE
SYSSTATS             ELIGIBLE
…
SYSUSER              ELIGIBLE
SYSXML               ELIGIBLE
---------------------------
```

21 objects (not all shown)
The 3 objects reorganized are shown as “complete”

Directory and Catalog Tables Being Converted

Database	Tablespace	Number of Tables	Page Size
DSNDB01	SYSUTILX	2	32K
DSNDB01	SCT02	1	4K
DSNDB06	SYSHIST	11	8K
DSNDB06	SYSSTATS	9	16K
DSNDB06	SYSDDF	8	4K
DSNDB06	SYSJAVA	4	4K
DSNDB06	SYSCONTX	3	16K
DSNDB06	SYSEBCDC	2	4K
DSNDB06	SYSGRTNS	2	8K
DSNDB06	SYSROLES	2	16K
DSNDB06	SYSSEQ2	2	4K
DSNDB06	SYSXML	2	8K

Directory and Catalog Tables Being Converted

Database	Tablespace	Number of Tables	Page Size
DSNDB06	SYSALTER	1	32K
DSNDB06	SYSGPALT	1	4K
DSNDB06	SYSSEQ	1	16K
DSNDB06	SYSTARG	1	4K
DSNDB06	SYSTSASC	1	4K
DSNDB06	SYSTSUNI	1	4K
DSNDB06	SYSTSXTM	1	4K
DSNDB06	SYSTSXTS	1	4K
DSNDB06	SYSUSER	1	4K

Catalog TS Conversions

Original TS Name	Table Name	Maximum Row Length	New Table Space Name	Buffer Pool
SYSHIST	SYSCOLDIST_HIST	6692	SYSTSCDH	BP8K0
SYSHIST	SYSCOLUMNS_HIST	4439	SYSTSCLH	BP8K0
SYSHIST	SYSINDEXES_HIST	590	SYSTSIXH	BP8K0
SYSHIST	SYSINDEXPART_HIST	349	SYSTSIPH	BP8K0
SYSHIST	SYSINDEXSTATS_HIST	327	SYSTSINH	BP8K0
SYSHIST	SYSKEYTARGETS_HIST	6564	SYSTSKTH	BP8K0
SYSHIST	SYSKEYTGDIST_HIST	4571	SYSTSKDH	BP8K0
SYSHIST	SYSLOBSTATS_HIST	78	SYSTSLSH	BP8K0
SYSHIST	SYSTABLEPART_HIST	123	SYSTSTPH	BP8K0
SYSHIST	SYSTABLES_HIST	365	SYSTSTHI	BP8K0

Catalog TS Conversions

Original TS Name	Table Name	Maximum Row Length	New Table Space Name	Buffer Poll
SYSHIST	SYSTABSTATS_HIST	345	SYSTSSTH	BP8K0
SYSSTATS	SYSCOLDIST	6694	SYSTSDIS	BP16K0
SYSSTATS	SYSCOLDISTSTATS	7698	SYSTSDST	BP16K0
SYSSTATS	SYSCOLSTATS	9426	SYSTSCST	BP16K0
SYSSTATS	SYSINDEXSTATS	1347	SYSTSXST	BP16K0
SYSSTATS	SYSKEYTARGETSTATS	9302	SYSTSKTS	BP16k0
SYSSTATS	SYSKEYTGTDIST	6564	SYSTSKTD	BP16K0
SYSSTATS	SYSKEYTGDISTSTATS	7568	SYSTSKDS	BP16K0
SYSSTATS	SYSLOBSTATS	82	SYSTSLOB	BP16K0
SYSSTATS	SYSTABSTATS	257	SYSTSTST	BP16K0

Catalog TS Conversions

Original TS Name	Table Name	Maximum Row Length	New Table Space Name	Buffer Poll
SYSDDF	IPLIST	291	SYSTSIPL	BP0
SYSDDF	IPNAMES	293	SYSTSIPN	BP0
SYSDDF	LOCATIONS	589	SYSTSLOC	BP0
SYSDDF	LULIST	61	SYSTSLUL	BP0
SYSDDF	LUMODES	63	SYSTSLUM	BP0
SYSDDF	LUNAMES	67	SYSTSLUN	BP0
SYSDDF	MODESELECT	217	SYSTSMOD	BP0
SYSDDF	USERNAMES	553	SYSTSUSN	BP0
SYSJAVA	SYSJARCONTENTS	680	SYSTSJRC	BP0
SYSJAVA	SYSJAROBJECTS	2495	SYSTSJRO	BP0

Catalog TS Conversions

Original TS Name	Table Name	Maximum Row Length	New Table Space Name	Buffer Poll
SYSJAVA	SYSJAVAOPTS	3739	SYSTSJVO	BP0
SYSJAVA	SYSJAVAPATHS	2711	SYSTSJPH	BP0
SYSCONTX	SYSCONTEXT	1388	SYSTSCTX	BP16K0
SYSCONTX	SYSCONTEXTAUTHIDS	349	SYSTSCAH	BP16K0
SYSCONTX	SYSCTXTTRUSTATTRS	448	SYSTSTRA	BP16K0
SYSCDC	SYSDUMMYE	9	SYSTSLDME	BP0
SYSCDC	SYSDUMMY1	9	SYSTSDUM	BP0
SYSTNS	SYSROUTINES_OPTS	7577	SYSTSGRO	BP8K0
SYSTNS	SYSROUTINES_SRC	7783	SYSTSGRS	BP8K0
SYSROLES	SYSOBJROLEDEP	531	SYSTSRDP	BP16K0

Catalog TS Conversions

Original TS Name	Table Name	Maximum Row Length	New Table Space Name	Buffer Poll
SYSROLES	SYSROLES	1215	SYSTSROL	BP16K0
SYSSEQ2	SYSSEQUENCEAUTH	721	SYSTSSQA	BP0
SYSSEQ2	SYSSEQUENCESDEP	925	SYSTSSQD	BP0
SYSXML	SYSXMLRELS	672	SYSTSXRL	BP8K0
SYSXML	SYSSMLSTRINGS	1015	SYSTSXSG	BP8K0
SYSALTER	SYSOBDS	30326	SYSTSOBD	BP32K
SYSGPAUT	SYSRESAUTH	609	SYSTSRES	BP0
SYSSEQ	SYSSEQUENCES	1694	SYSTSSEQ	BP0
SYSTARG	SYSTARGETS	9106	SYSTSKYT	BP16K0
SYSTSASC	SYSDUMMYA	9	SYSDUMMYA	BP0

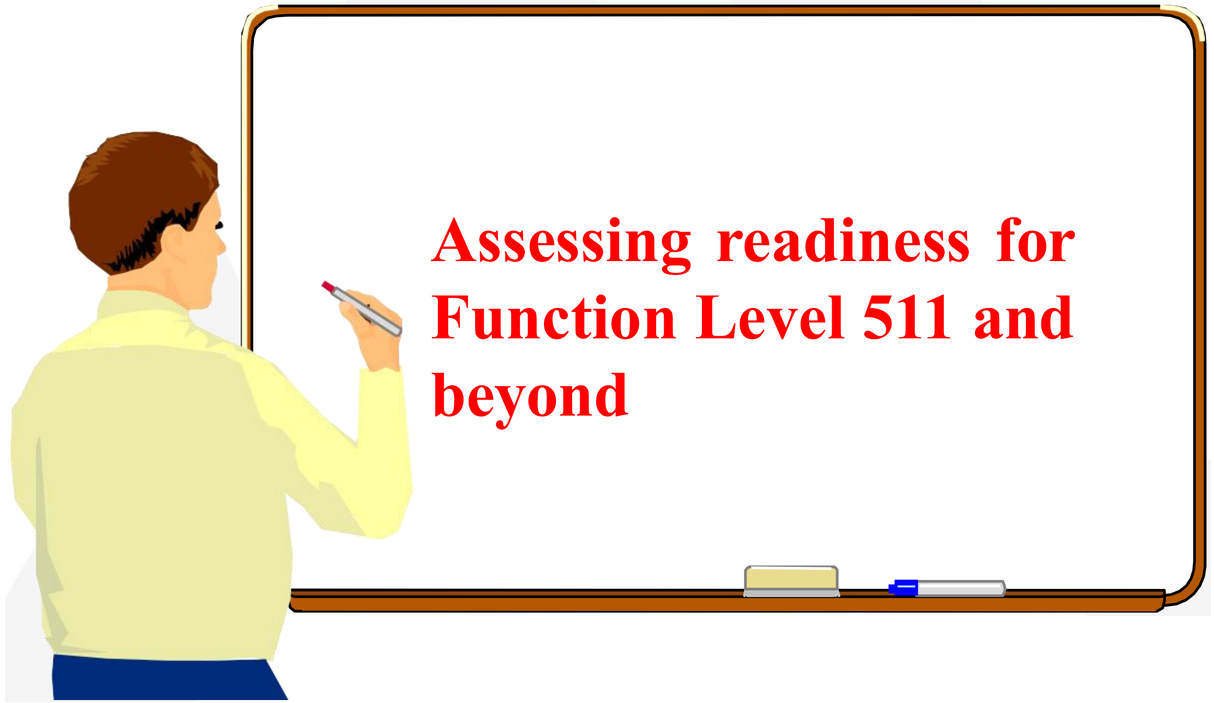
Catalog & Directory TS Conversions

Original TS Name	Table Name	Maximum Row Length	New Table Space Name	Buffer Poll
SYSTSUNI	SYSDUMMYU	9	SYSTSUNI	BP0
SYSTSXTM	SYSXMLTYPMOD	163	SYSTSXTM	BP0
SYSSXTS	SYSXMLTYPMSCHEMA	46	SYSTSXTS	BP0
SYSUSER	SYSUSERAUTH	507	SYSTSUSR	BP0
SYSUTILX	SYSUTIL	27418	SYSTSUTM	BP32K
SYSUTILX	SYSUTILX	32044	SYSTSUTX	BP32K
SCT02	SCTR	4053	SYSTSSCT	BP8K

SYSCOPY Entries

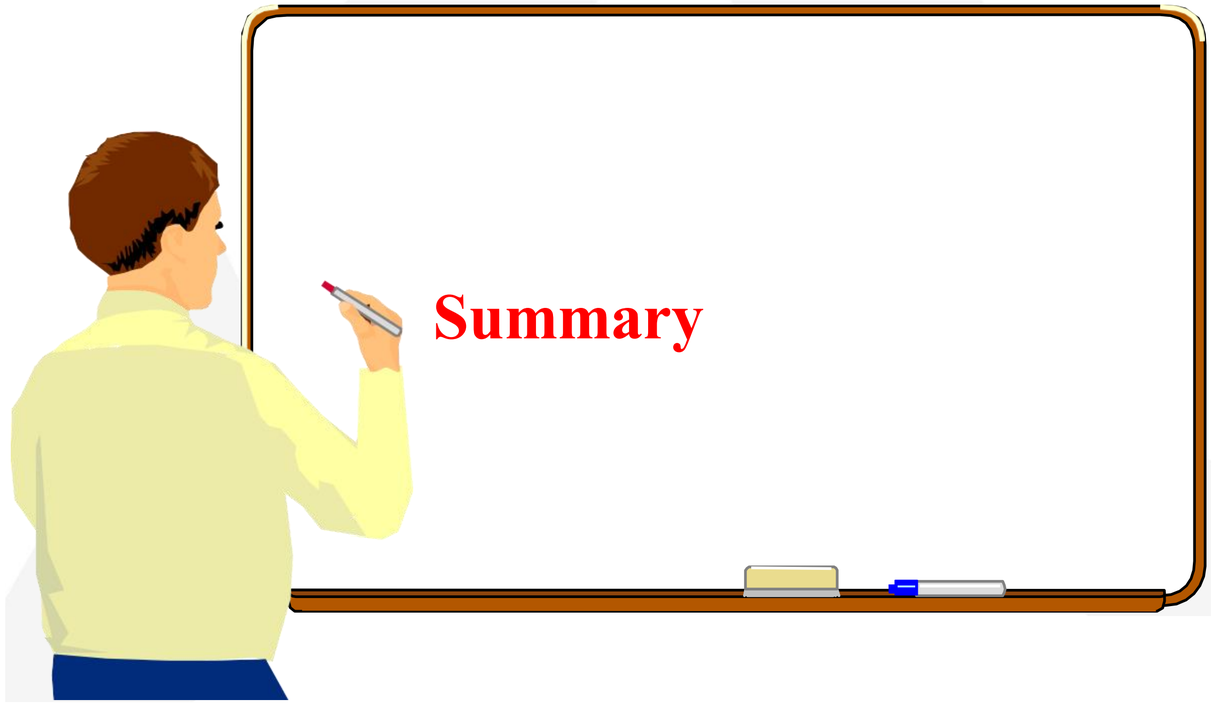
- When a table is moved from a multi-table segmented to PBG
  - E.G. DSNDB06.SYSHIST

ICTYPE	STYPE	DSNAME	Explanation
W	M	Original db.ts names	For UTS conversion to a new catalog or directory table space name. REORG moved a table in from source table space. This record will be deleted at the completion of the UTS conversion process.
C	L	New db.ts names	Target table space
F	W	IC data set	Inline image copy created by REORG


Assessing readiness for Function Level 511 and beyond

Assessing readiness of FL 511 and beyond

- Function Level V13R1M511 (aka FL511) cannot be activated if there are any deprecated objects in the system.
  - The **ACTIVATE** command with the **TEST** option can be used to see if your subsystem or group is ready and enabled for FL511.
    - Same as we had for the activation of FL V12R1M510 in Db2 12.
  - The DSNTIJP* job can be used to determine if a subsystem or group has any deprecated objects.


Summary

Thank you

© 2026 International Business Machines Corporation
IBM and the IBM logo are trademarks of IBM Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on ibm.com/legal/copyright-trademark.

This document is current as of the initial date of publication and may be changed by IBM at any time.

Statements regarding IBM’s future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

THIS DOCUMENT IS DISTRIBUTED “AS IS” WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IN NO EVENT, SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY.

Client examples are presented as illustrations of how those clients have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

Not all offerings are available in every country in which IBM operates.

It is the user’s responsibility to evaluate and verify the operation of any other products or programs with IBM products and programs.

The client is responsible for ensuring compliance with laws and regulations applicable to it. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the client is in compliance with any law or regulation.