

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 

Disclaimer and Trademarks



© 2026 International Business Machines Corporation.
All rights reserved.

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.

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

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM.

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

Any statements regarding IBM’s future direction, intent or product plans are subject to change or withdrawal without notice.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines 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 the Web at “Copyright and trademark information” at: www.ibm.com/legal/copytrade.shtml.

Certain comments made in this presentation may be characterized as forward looking under the Private Securities Litigation Reform Act of 1995.

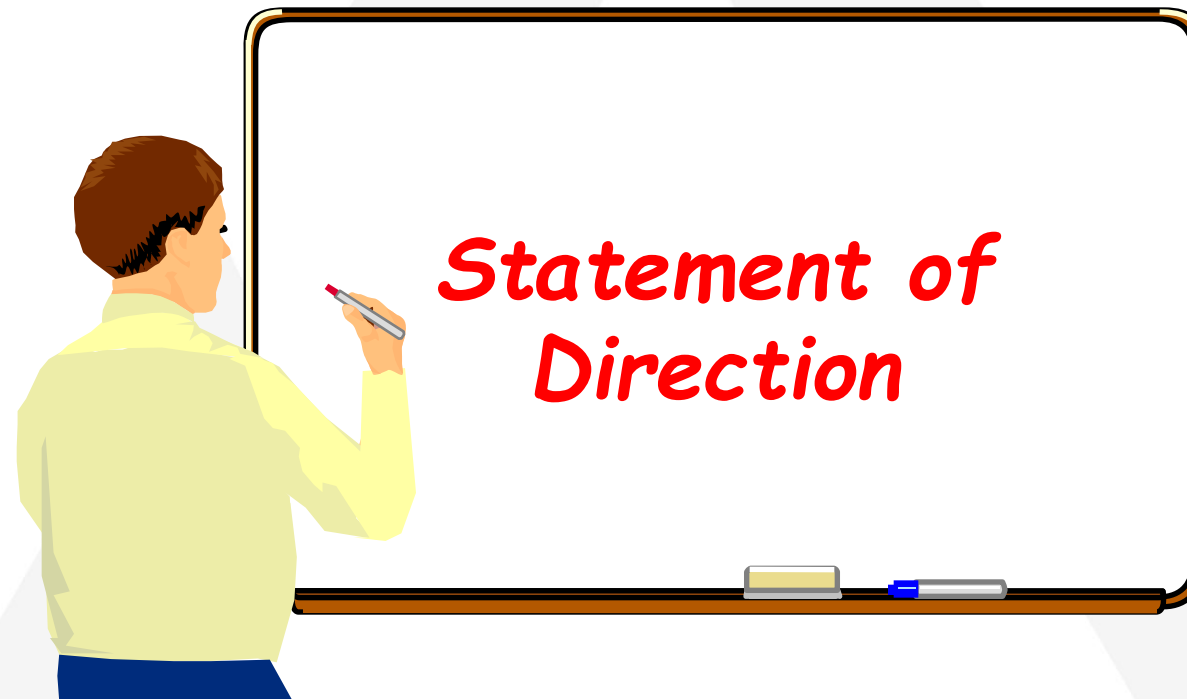
Forward-looking statements are based on the company’s current assumptions regarding future business and financial performance. Those statements by their nature address matters that are uncertain to different degrees and involve a number of factors that could cause actual results to differ materially. Additional information concerning these factors is contained in the Company’s filings with the SEC.

Copies are available from the SEC, from the IBM website, or from IBM Investor Relations.

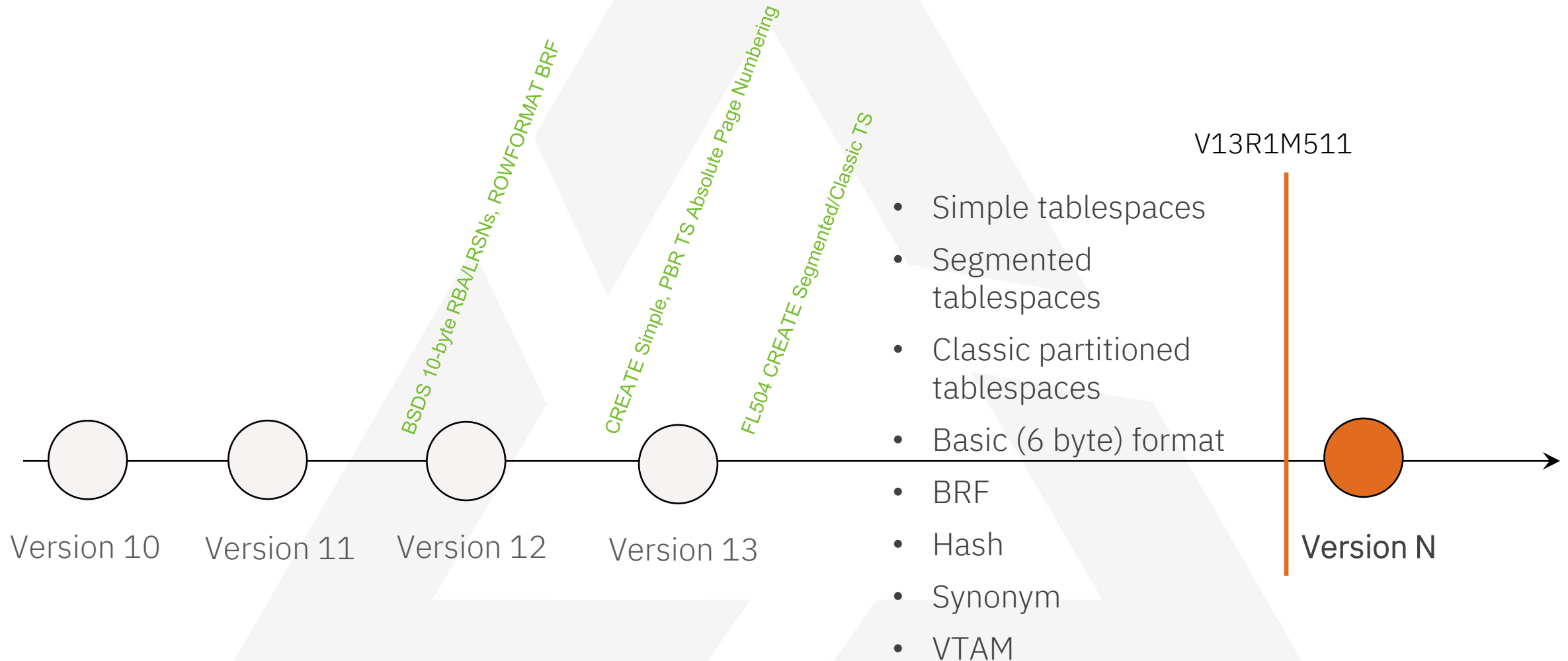
Any forward-looking statement made during this presentation speaks only as of the date on which it is made. The company assumes no obligation to update or revise any forward-looking statements except as required by law; these charts and the associated remarks and comments are integrally related and are intended to be presented and understood together.

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

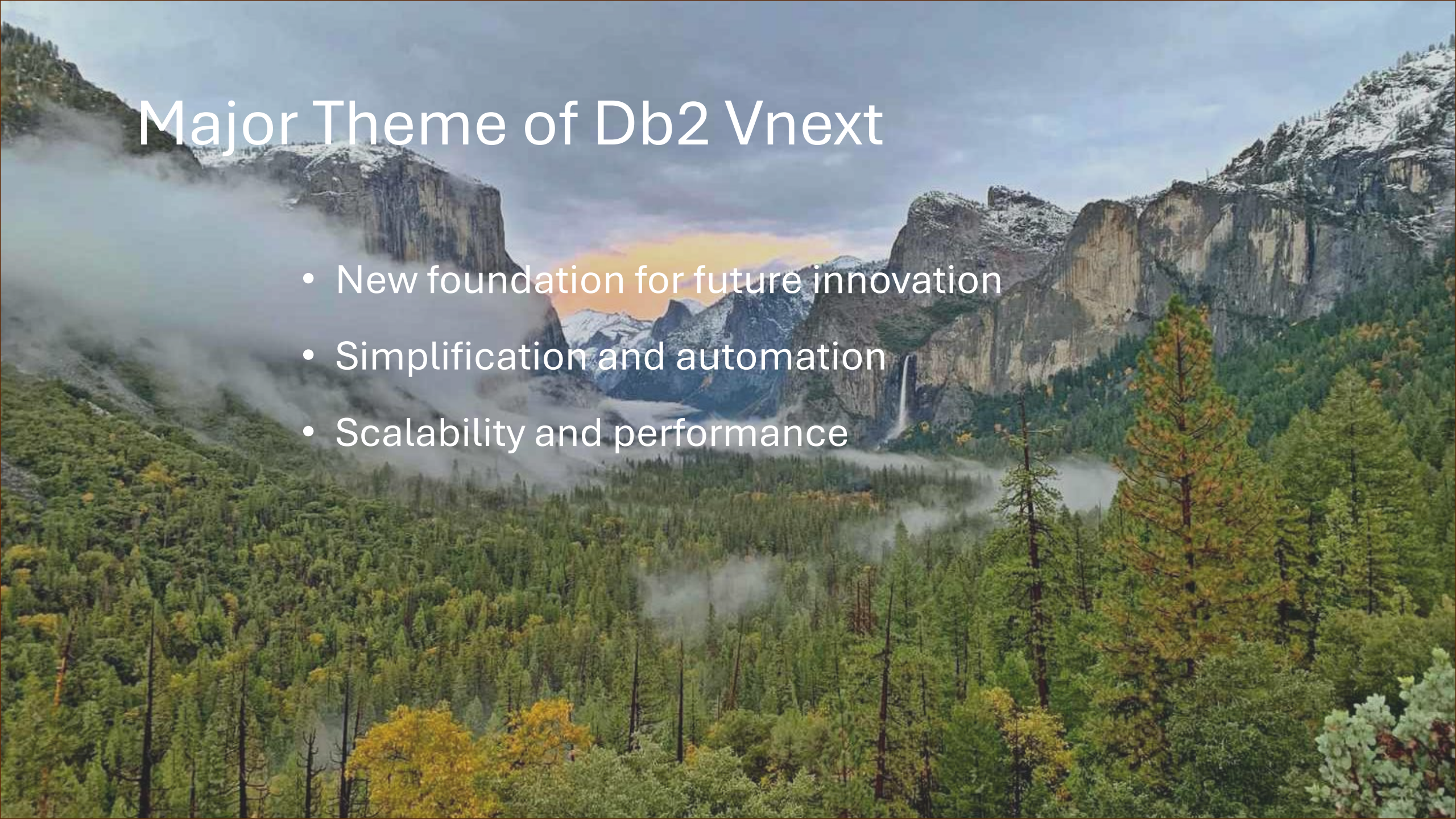


Db2 for z/OS Past, Present and Future

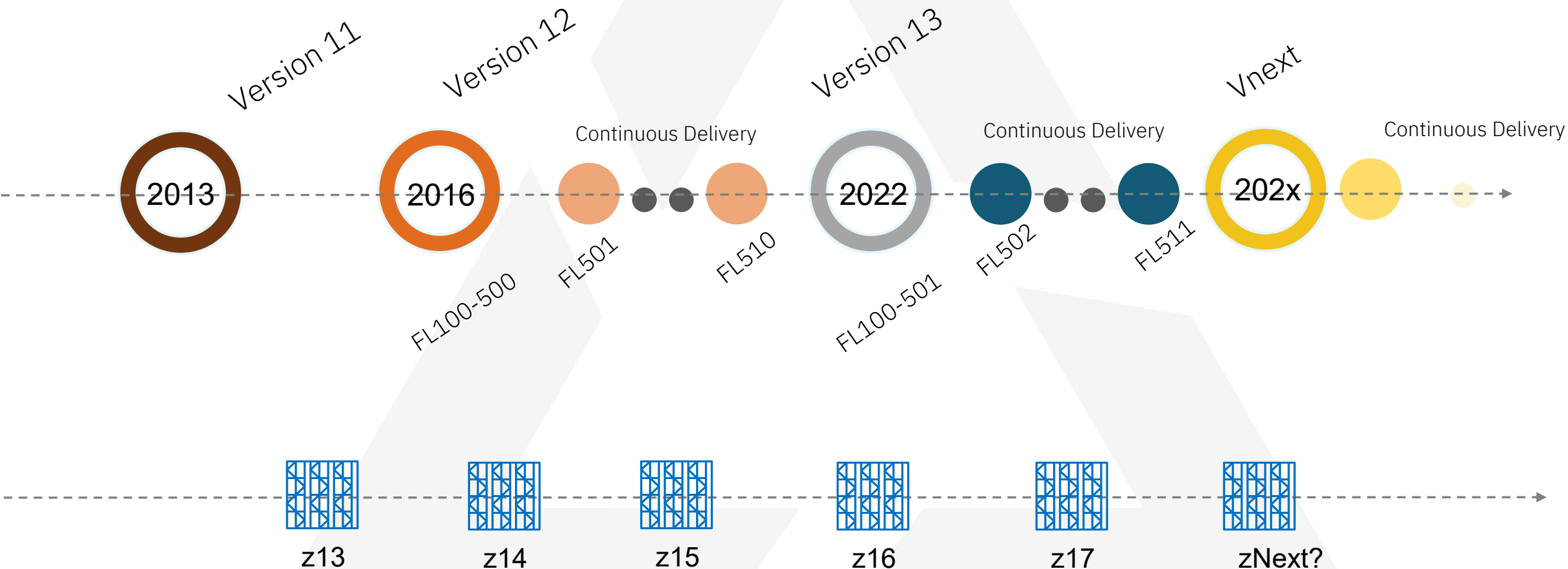


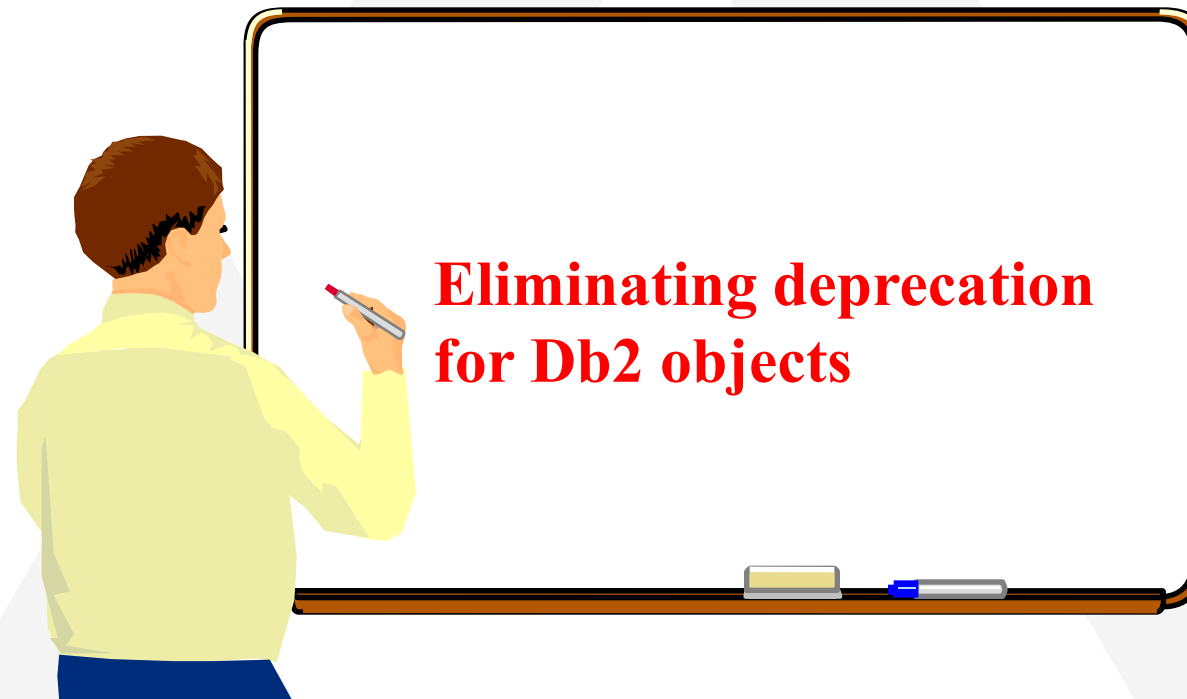
Major Theme of Db2 Vnext

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



Db2 for z/OS Release Plan





Deprecation Overview

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

Converting BRF to Reordered Row Format (RRF)

- Why convert? **Performance and logging reduction**
 - SELECT with WHERE and column comparison
- Identify all user table spaces
 - SYTABLEPART column FORMAT (' ' – BRF, 'R' – RRF)
 - Not for LOB or XML table spaces
 - EDITPROCS and VALIDPROCS must be dropped
- REORG or LOAD REPLACE without syntax option **ROWFORMAT BRF**
 - Specify ROWFORMAT RRF or take the default
 - No zparm RRF 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 = ' ')
AND (A.TYPE <> 'O')
AND (A.TYPE <> 'P')
```

Format is BRF (i.e. not 'R')

Not a LOB table space

Not an XML table space

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

Edit Proc

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 C2 FROM TB1 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 and index spaces
 - SYSTABLEPART/SYSINDEXPART column RBA_FORMAT
 - 'E' (10 bytes), 'B' (6 bytes), 'U' (undefined), or ' ' (6 bytes before V10)
- REORG or LOAD REPLACE or REBUILD INDEX without syntax option
RBALRSN_CONVERSION
 - Specify RBALRSN_CONVERSION EXTENDED (or omit for the default)
 - No Zparm UTILITY_OBJECT_CONVERSION Db12 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')
      OR (A.RBA_FORMAT = ' '))
```

Format is 6-byte RBA/LRSN

Format is not 10-byte RBA/LRSN value of 'E'

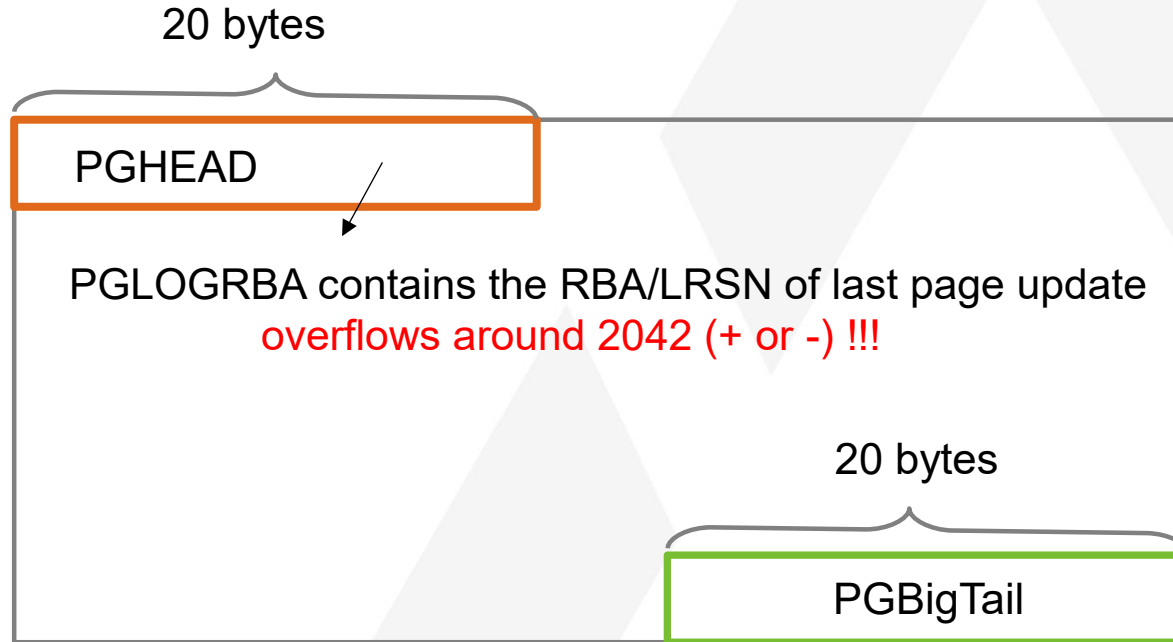
Identify all user index spaces

```
SELECT DISTINCT IXNAME, RBA_FORMAT
FROM SYSIBM.SYSINDEXPART
WHERE RBA_FORMAT = 'B'
OR RBA_FORMAT = ' '
```

Format is 6-byte RBA/LRSN

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 Universal Table Spaces (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), and simplified data reorganization by partition(s).

Why Convert to UTS?

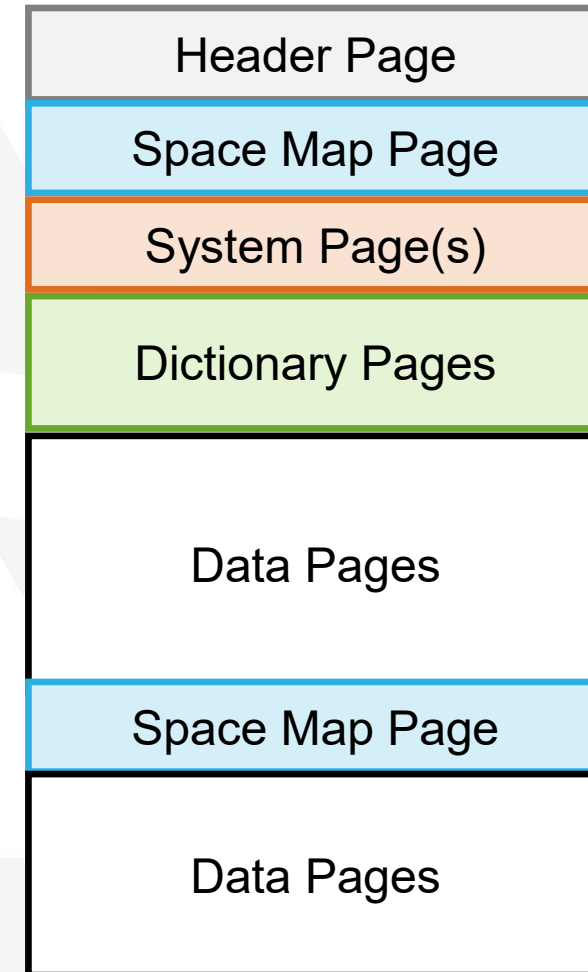
Advanced Schema Flexibility (Pending DDL): UTS allows for complex deferred ALTER statements to be executed as "pending DDL". These changes do not require immediate downtime and can be applied during an online REORG, reducing maintenance windows. The ALTERs can be “stacked” so they are applied within 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 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/Lempel or Huffman
 - Data pages
 - Single table per table space



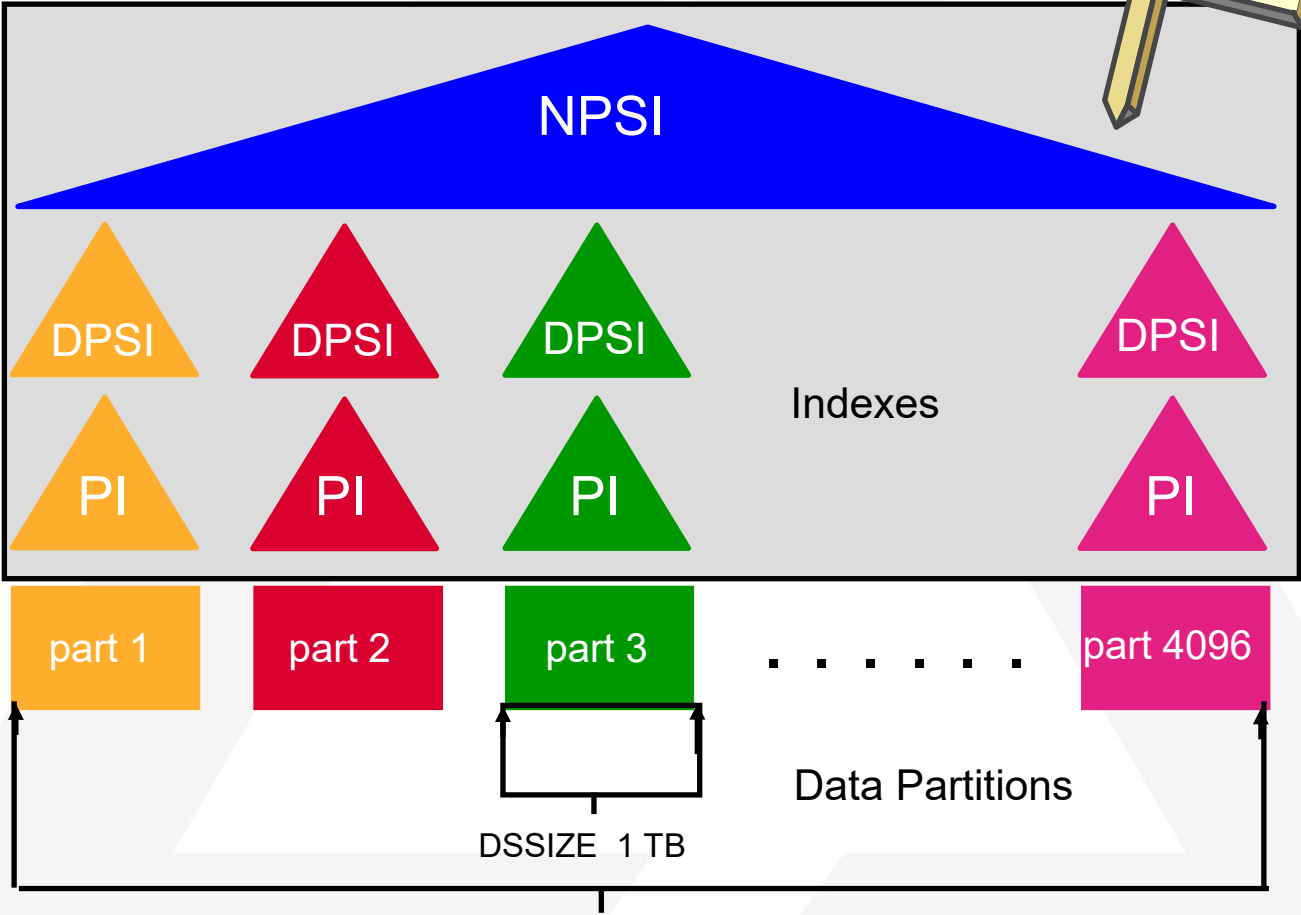
Db2 9
2007

PBR Relative Page Numbering

1,000 Inserts/second
for 8,800 years!

Db2 12
2016

PBR RPN (aka PBR2)



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 ⁿ¹ – MAXPARTITIONS
Partition-by-range	Numparts only	Numparts and SEGsize ⁿ¹
Segmented (non-UTS)	Not supported ²	One of the following combinations: – SEGsize ⁿ¹ – Omit MAXPARTITIONS, Numparts, and SEGsize
Partitioned (non-UTS)	Not supported ²	Numparts and SEGsize 0

[\(link\)](#)

Converting Simple or Segmented to PBG

- SYTABLESPACE 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
AND TYPE = ' '
```

Non-partitioned
Classic Partitioned, Simple, or Segmented

- Convert with two steps to make it a partitioned PBG

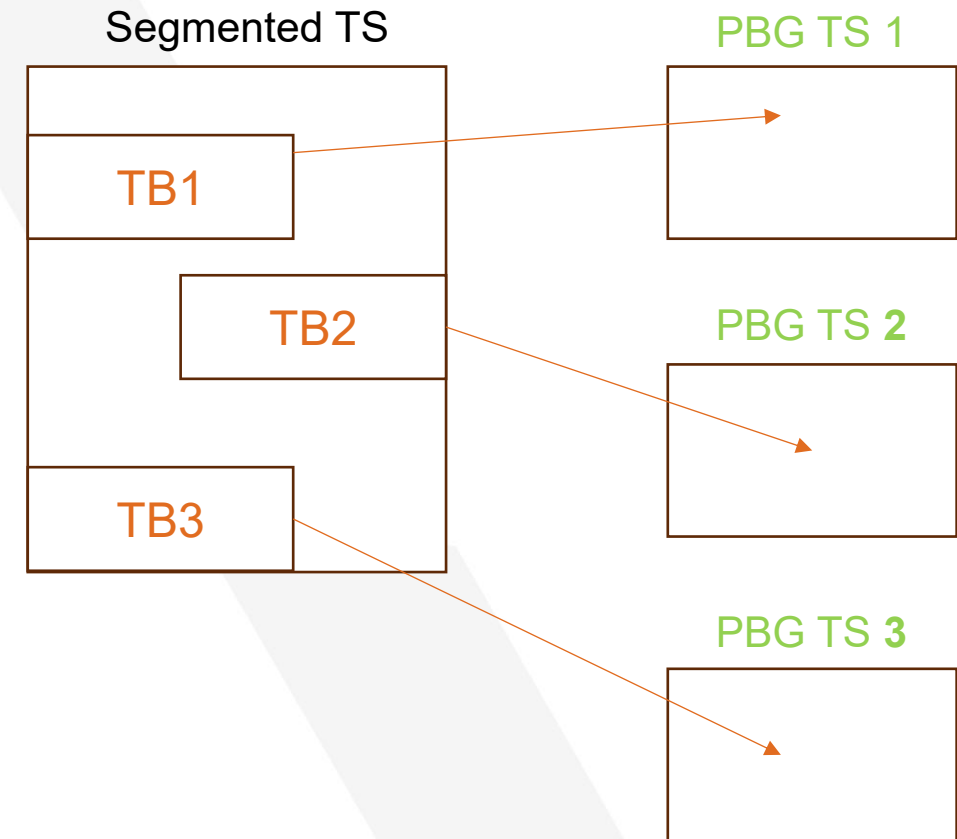
1. ALTER TS MAXPARTITIONS n
2. REORG TABLESPACE name

Max partitions 1 up to 4096

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



Converting Classic Partitioned to PBR

- SYTABLESPACE column TYPE (‘ ’, ‘L’ - Large classic, ‘G’ – PBG, ‘R’ – PBR)
- Find non-UTS classic partitioned (large or non large)

```
SELECT DBNAME, NAME, TYPE, SEGSIZE
FROM SYSIBM.SYSTABLESPACE
WHERE PARTITIONS <> 0
AND SEGSIZE = 0
```

Partitioned
Not segmented

- 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)

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

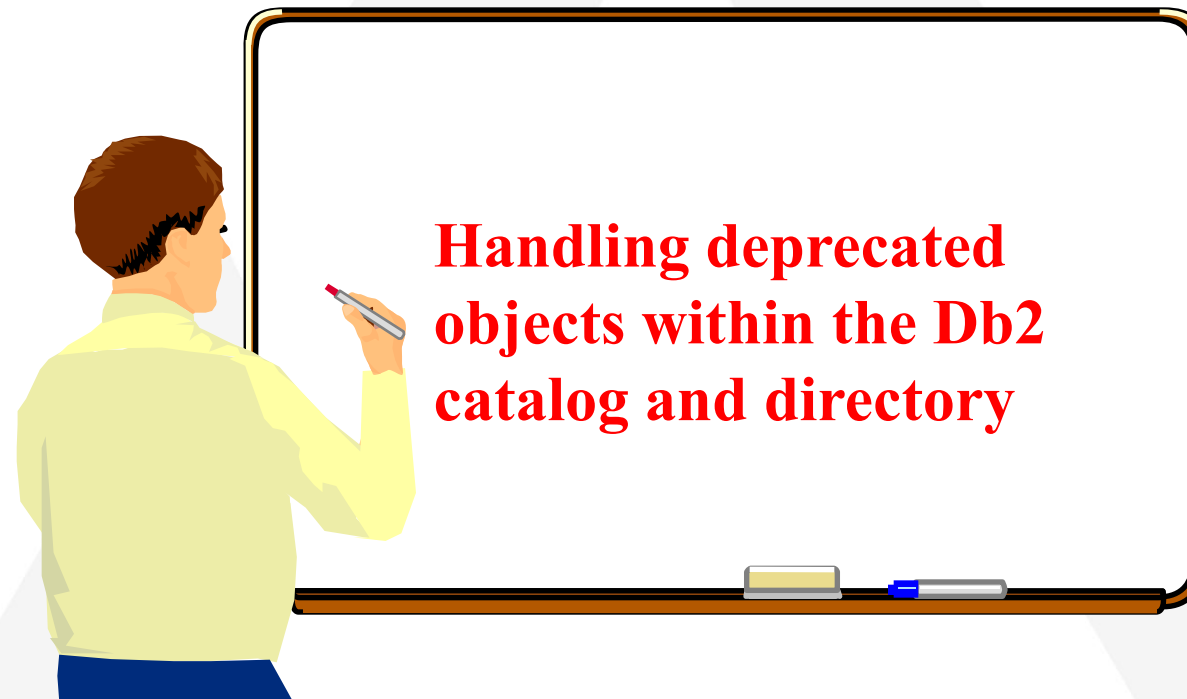
```
DROP SYNONYM NAME
```

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

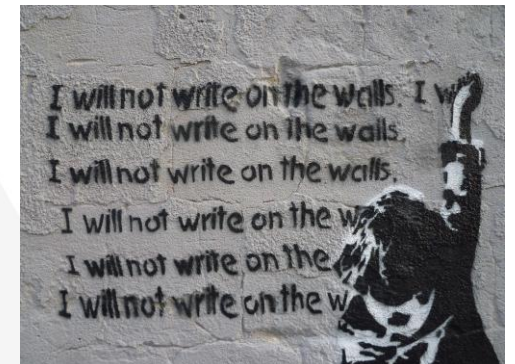
- UTS Conversion of catalog and directory objects began in D**B**2 10 and continued in D**b**2 11
 - Conversions done via ENFM processes
- Conversion process not (yet) completed/finished
 - 21 catalog and directory table spaces currently non-UTS



UTS Conversion of Catalog and Directory

- Why?
 - Adopting universal table spaces (UTS) 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 provide UTS (PBG) conversion process for remaining catalog and directory objects
 - 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 **UTS**CONVERSION

New keyword for DIS GROUP

UTS CONVERSION REPORT

TABLE SPACE	UTS CONVERSION	ACTION REQUIRED
SCT02	INELIGIBLE	
SYSUTILX	INELIGIBLE	
SYSALTER	INELIGIBLE	
SYSCONTX	INELIGIBLE	
SYSDDF	INELIGIBLE	
SYSEBCDC	INELIGIBLE	
...		
SYSSEQ	INELIGIBLE	
SYSSEQ2	INELIGIBLE	
SYSSTATS	INELIGIBLE	
...		
SYSUSER	INELIGIBLE	
SYSXML	INELIGIBLE	

FL <= FL508

21 table spaces (not all shown)

Before FL509, they are shown as “ineligible” for conversion.

After FL 509, they will all show ELIGIBLE or COMPLETE.

Run CATMAINT for FL509

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

Get appropriate catalog level needed
Activate function level needed

-DIS GROUP DETAIL

DSN7100I -DB2A DSN7GCMD

*** BEGIN DISPLAY OF GROUP(DSNCAT) CATALOG LEVEL(V13R1M509)
CURRENT FUNCTION LEVEL(V13R1M509)
HIGHEST ACTIVATED FUNCTION LEVEL(V13R1M509)
HIGHEST POSSIBLE FUNCTION LEVEL(V13R1M509)
PROTOCOL LEVEL(2)
GROUP ATTACH NAME(DSNG)

Catalog level 509 has PBG objects needed

DB2 MEMBER	ID	SUB SYS	CMDPREF	STATUS	DB2 LVL	SYSTEM NAME	IRLM SUBSYS	IRLMPROC
DB2A	1	DB2A	-DB2A	ACTIVE	131509	UTEC5	PR21	PRLM21
DB2B	2	DB2B	-DB2B	ACTIVE	131509	UTEC5	QR21	QRLM21

MIGRATION READINESS REPORT

DB2 MEMBER	CODE LEVEL	SPE APAR	MIGRATION ELIGIBLE
DB2A	V13R1M509		NO
DB2B	V13R1M509		NO

MIGRATION READINESS STATUS: GROUP IS NOT READY FOR DB2 NEXT
REASON: NO NEXT RELEASE OF DB2 IS CURRENTLY AVAILABLE

Identify Catalog/Directory Objects

- -DIS GROUP **UTS**CONVERSION

UTS CONVERSION REPORT

TABLE SPACE	UTS CONVERSION	ACTION REQUIRED
-----	-----	-----
SCT02	ELIGIBLE	
SYSUTILX	ELIGIBLE	
SYSALTER	ELIGIBLE	
SYSCONTX	ELIGIBLE	
SYSDDF	ELIGIBLE	
SYSEBCDC	ELIGIBLE	
...		
SYSSEQ	ELIGIBLE	
SYSSEQ2	ELIGIBLE	
SYSSTATS	ELIGIBLE	
...		
SYSUSER	ELIGIBLE	
SYSXML	ELIGIBLE	

FL >= FL509
HAFL < FL511

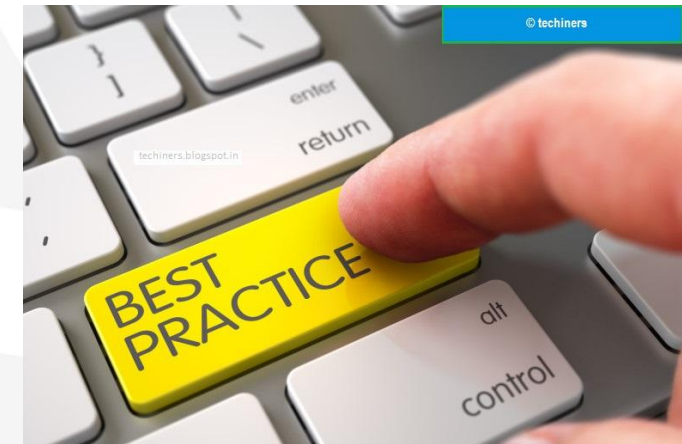
21 objects (not all shown)

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
- Examples:

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



Identify Catalog/Directory Objects

- -DIS GROUP **UTS CONVERSION**

UTS CONVERSION REPORT

TABLE SPACE	UTS CONVERSION	ACTION REQUIRED
-----	-----	-----
SYSUTILX	COMPLETE	
SYSCONTX	COMPLETE	
SYSDDF	COMPLETE	
SCT02	ELIGIBLE	
SYSALTER	ELIGIBLE	
SYSEBCDC	ELIGIBLE	
...		
SYSSEQ	ELIGIBLE	
SYSSEQ2	ELIGIBLE	
SYSSTATS	ELIGIBLE	
...		
SYSUSER	ELIGIBLE	
SYSXML	ELIGIBLE	

FL >= FL509
HAFL < FL511

21 objects (not all shown)

The 3 table spaces reorganized are shown as “complete”.
The other 18 table spaces are ELIGIBLE for UTS conversion.

**What would be displayed for this UTS CONVERSION status
in FL 508* or below?**

Identify Catalog/Directory Objects

- **-DIS GROUP** **UTS CONVERSION** (when in a star FL FL508* or below)

UTS CONVERSION REPORT

TABLE SPACE	UTS CONVERSION	ACTION REQUIRED
SYSUTILX	COMPLETE	}
SYSCONTX	COMPLETE	
SYSDDF	COMPLETE	
SCT02	INELIGIBLE	
SYSALTER	INELIGIBLE	
SYSEBCDC	INELIGIBLE	
...		
SYSSEQ	INELIGIBLE	
SYSSEQ2	INELIGIBLE	
SYSSTATS	INELIGIBLE	
...		
SYSUSER	INELIGIBLE	
SYSXML	INELIGIBLE	

FL* < FL509
HAFL < FL511

21 objects (not all shown)

The 3 table spaces reorganized remain “complete”.

The other 18 table spaces are now ‘INELIGIBLE’ for UTS conversion.

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	SYSCTX	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	SYSTSC LH	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	SYTSKTS	BP16k0
SYSSTATS	SYSKEYTGTDIST	6564	SYTSKTD	BP16K0
SYSSTATS	SYSKEYTGDISTSTATS	7568	SYTSKDS	BP16K0
SYSSTATS	SYSLOBSTATS	82	SYTSLOB	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	SYSJAVA_OPTS	3739	SYSTSJVO	BP0
SYSJAVA	SYSJAVAPATHS	2711	SYSTSJPH	BP0
SYSCTX	SYSCONTEXT	1388	SYSTSCTX	BP16K0
SYSCTX	SYSCONTEXTAUTHIDS	349	SYSTSCAH	BP16K0
SYSCTX	SYSCTXTRUSTATTRS	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	SYTSOBD	BP32K
SYSGPAUT	SYSRESAUTH	609	SYTSRES	BP0
SYSSEQ	SYSSEQUENCES	1694	SYTSSEQ	BP0
SYSTARG	SYSTARGETS	9106	SYTSKYT	BP16K0
SYTSASC	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	SYSXMLTYPMSHEMA	46	SYSTSXTS	BP0
SYSUSER	SYSUSERAUTH	507	SYSTSUSR	BP0
SYSUTILX	SYSUTIL	27418	SYSTSUTM	BP32K, LOCKSIZE PAGE
SYSUTILX	SYSUTILX	32044	SYSTSUTX	BP32K, LOCKSIZE PAGE
SCT02	SCTR	4053	SYSTSSCT	BP0, LOCKSIZE PAGE

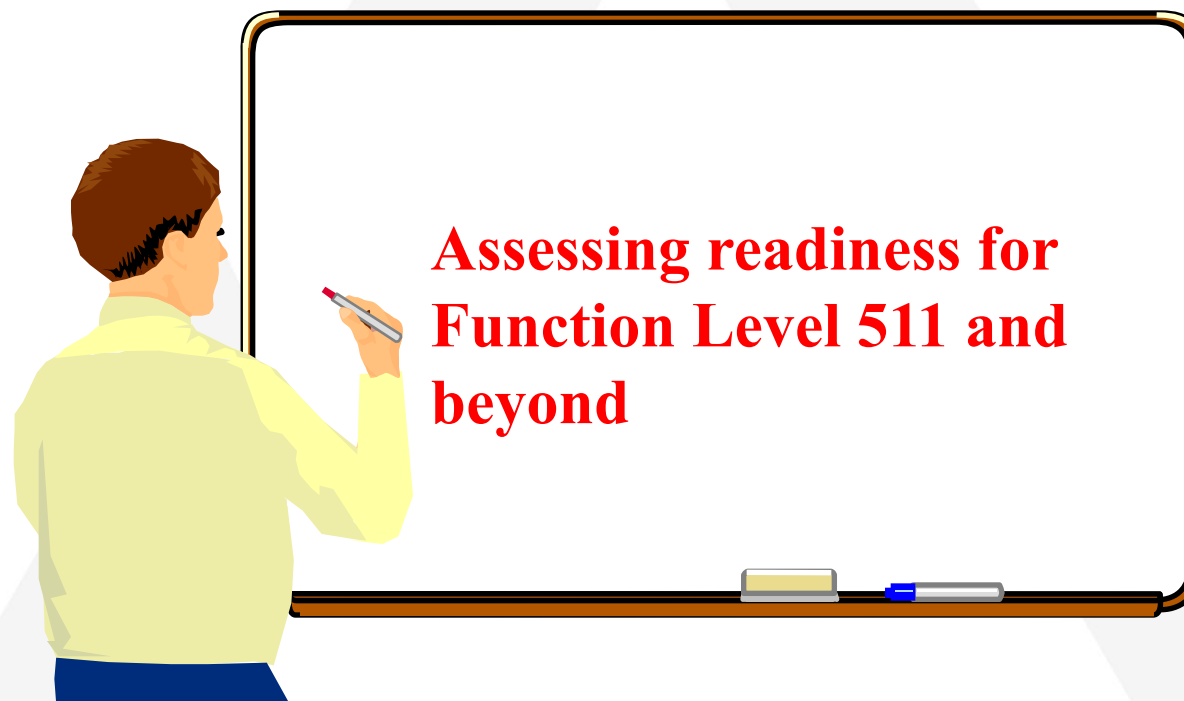
SYSCOPY Entries

- 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

SYSUTILX UTS CONVERSION

- REORG on directory table space SYSUTILX is not allowed but.....
- REORG **IS** being used to convert SYSUTILX to UTS (PBG)
 - Results in two UTS table spaces:
 - SYSTSUTX – SYSUTILX table
 - SYSTSUTM – SYSUTIL table
- **A one time thing.** REORG cannot be used against either of the two new directory table spaces above.
- Not **'real'** REORG processing.

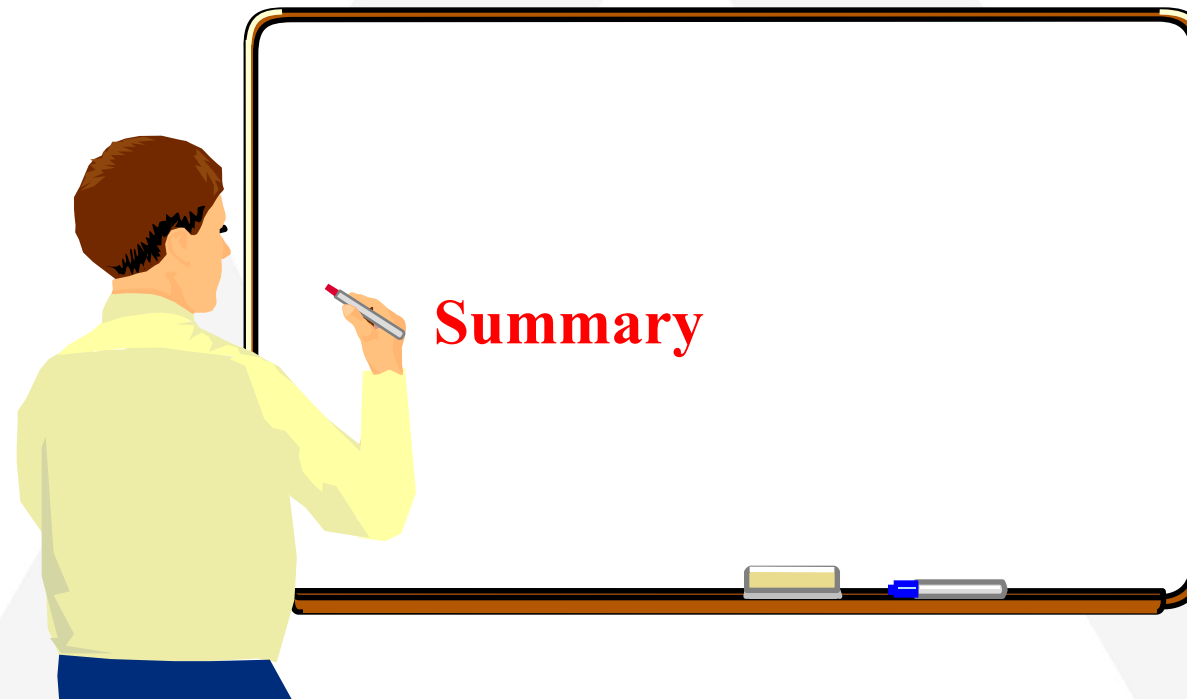


Assessing readiness for Function Level 511 and beyond

Assessing readiness of FL 511 and beyond

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





Summary

- UTS conversion of catalog and directory objects can be done in function levels V13R1M509 – V13R1M510
- Deprecated object removal should be a priority
 - Use a planned approach
 - Don't wait until it's time to migrate to VNext

Your feedback is important!

Submit a session evaluation for each session you attend:

www.share.org/evaluation



Thank
You



Embrace The Future Today By Eliminating Deprecated Objects

Thank you!

2026 SHARE Conference – Orlando, FL
Feb 25th, 2026

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

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