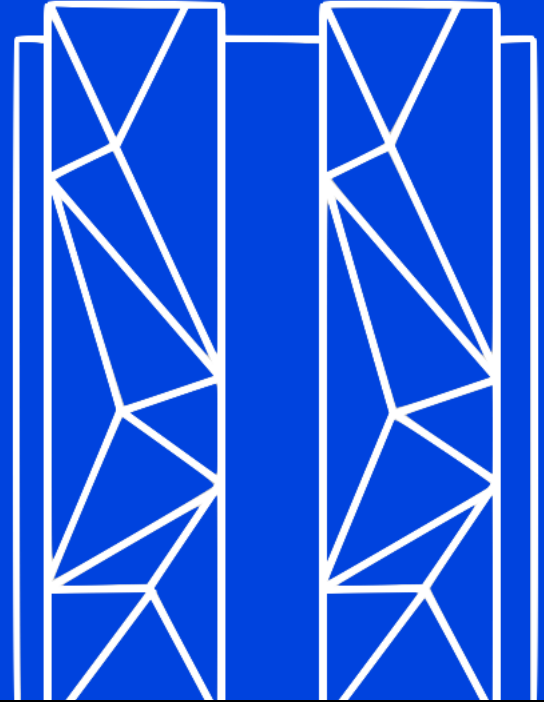


# Are you compatible?

Db2 12 for z/OS, Continuous Delivery,  
Function Levels and Application  
Compatibility (APPLCOMPAT)

March 25, 2021

*Mark Rader*



# Agenda

## Review of continuous delivery

### Function level dependent **highlights**

- FL501 – May 2017
- FL502 – May 2018
- FL503 – October 2018
- FL504 – April 2019
- FL505 – June 2019
- FL506 – October 2019
- FL507 – June 2020
- FL508 – October 2020
- FL509 – February 2021

### Non-function level dependent **highlights**

- Distributed data facility (DDF)
- Utilities
- Analytics
- Z synergy
- Data sharing
- Usability
- Security

# Continuous delivery – feature categories

## Module level or code level dependent

Infrastructure changes, e.g.:

- Utility enhancements
- Parallel Sysplex exploitation
- Db2ZAI enablement

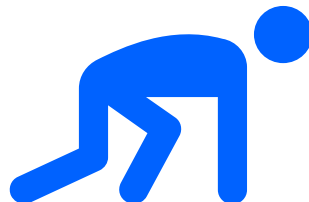


## Function level dependent

Significant system enhancements, e.g.:

- ENCRYPTION\_KEYLABEL in DSNZPARM
- HUFFMAN compression algorithm
- REBIND Phase-in

## Some function levels are catalog level dependent



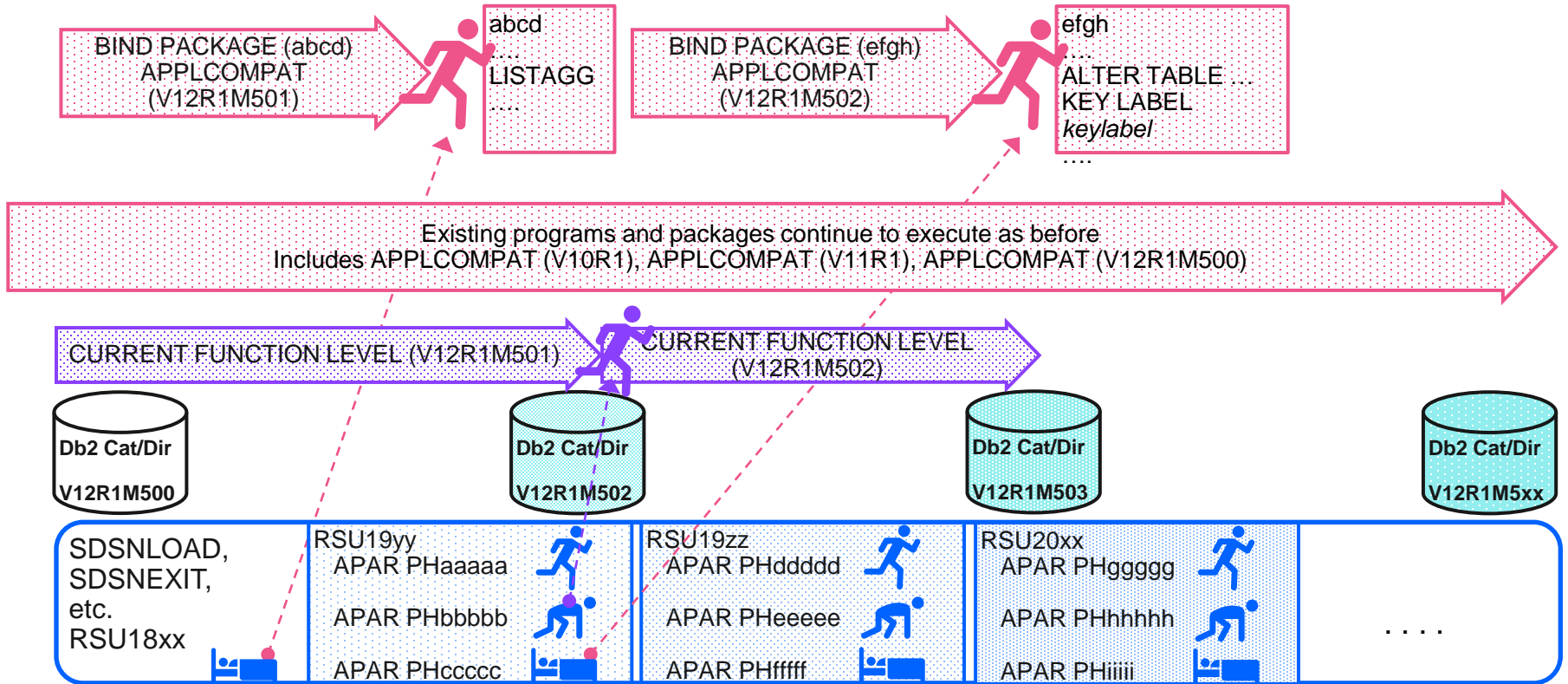
## APPLCOMPAT dependent

SQL-related changes: new or changed SQL or results, e.g.:

- LISTAGG
- KEY LABEL for STOGROUP, TABLE or TABLESPACE
- ENCRYPT\_DATAKEY, DECRYPT\_DATAKEY
- DECFLOAT columns in indexes



# Exploiting function levels



# What does DSNZPARM APPLCOMPAT do?

- This is the default APPLCOMPAT setting for packages that have not set APPLCOMPAT yet
  - BIND new package and 'forget' to specify APPLCOMPAT
  - REBIND pre-Db2 11 package without setting APPLCOMPAT
- Once a package has an APPLCOMPAT setting:
  - Changing the DSNZPARM value will not affect that package
- When should you change the DSNZPARM?
  - Recommendation: after you have issued -ACTIVATE FUNCTION LEVEL (V12R1M500) and are sure you are not going back to V12R1M100\* change to V12R1M500
  - Specific packages that want to take advantage of certain functions will specify the required APPLCOMPAT setting
  - Consider separate collections for IBM Data Server Driver packages (aka 'NULLID')
    - Each collection can have a different APPLCOMPAT setting
    - Can choose collection with SET CURRENT PACKAGE PATH explicitly or via System Profile Monitoring

# Function level 501

## PI70535

- Built-in function LISTAGG

One example - given this data in a table:

EMPNO	LASTNAME	WORKDEPT
0001	THOMAS	A01
0002	ROGERS	B01
0003	HONG	A01
0004	BARKER	B01
0005	KOHL	B01

```
SELECT WORKDEPT,  
LISTAGG(LASTNAME, ' , ' )  
WITHIN GROUP (ORDER BY LASTNAME)  
AS EMPLOYEES  
FROM EMPLOYEE  
GROUP BY WORKDEPT;
```

```
WORKDEPT    EMPLOYEES  
-----  
A01         HONG, THOMAS  
B01         BARKER, KOHL, ROGERS
```

# Function level 502

## PI95511

- Db2 interfaces to specify and report key labels for data set encryption
- System parameter for catalog and directory objects and archive logs on disk
- SQL data definition language (DDL) for tables or storage groups
- Db2 report and display options
- Explicit casting of numeric data types to strings with GRAPHIC or VARGRAPHIC functions

# Encrypting Db2 system objects



Security / System Admin / Storage Admin

In RACF, alter DFP segment in data set profile - `DATAKEY()`

– OR –

In Db2, set key label using system parameter  
OR  
`IDCAMS DEFINE`, etc.

– OR –

In DFSMS, assign to data class

1. Security admin sets key label via RACF
2. System admin can set a key label using Db2 system parameter (DSNZPARM)
  - `ENCRYPTION_KEYLABEL`
  - Command `–SET SYSPARM` to enable parameter change
  - Applies to:
    - Catalog and directory objects
    - Disk archive logs
3. Storage admin can set key label in DFSMS data class

Numbers represent precedence order



# Encrypting Db2 user objects



Security / Database Admin / Storage Admin

In RACF, alter DFP segment in data set profile - `DATAKEY()`

– OR –

In Db2, set key label using SQL DDL interfaces  
OR  
IDCAMS DEFINE, etc.

– OR –

In DFSMS, assign to data class

1. Security admin sets key label via RACF
2. Application database admin sets key label using SQL data definition language (DDL) interfaces
  - `CREATE/ALTER STOGROUP`
  - `CREATE/ALTER TABLE`
  - Note: requires APPLCOMPAT(V12R1M502)
3. Storage admin can set key label in DFSMS data class

Numbers represent precedence order

# Cast numeric values to GRAPHIC, VARGRAPHIC

In Db2 11 and prior, two step process

1. CAST (*numeric column* as CHAR(10) )
2. CAST (*character*) as GRAPHIC

Or

1. apply VARCHAR built-in function to *numeric column*, followed by
2. Apply VARGRAPHIC built-in function to resultant VARCHAR

Db2 12 solution, with APPLCOMPAT V12R1M502

- CAST(*numeric column* AS GRAPHIC(10) )

Or

- VARGRAPHIC(*numeric column*)

One step!



# Function level 503

## PH00506

- APAR PH00506 provides the code level which is the basis for Db2 AI for z/OS (Db2ZAI)
  - Db2 AI for z/OS – product that applies machine learning capabilities to customize Db2 for z/OS for a client's workload
  - Only FL500 required for Db2ZAI

### Temporal enhancements

- System time temporal: query correction
- System time temporal: replication enhancements

Because these change the effect of SQL operations, they require an application compatibility (APPLCOMPAT) bind/rebind setting change

# Function level 504

## PH07672

- Deprecated objects
- Huffman compression (with z14 or later)
- Built-in function (BIF) pass-through to Db2 Analytics Accelerator
- Syntax flexibility for special registers, NULL predicates

Deprecated objects cannot be created with package bound with APPLCOMPAT (V12R1M504) or higher

e.g. CREATE TABLESPACE	issued by application with APPLCOMPAT < V12R1M504	issued by application with APPLCOMPAT >= V12R1M504
SEGSIZE n	<b>Segmented Table Space</b>	<b>UTS – PBG</b>
SEGSIZE 0 & Numparts p	<b>Classic Partitioned Table Space</b>	<b>UTS – PBR</b>
SEGSIZE n & MAXPARTITIONS p	UTS – PBG	UTS – PBG
SEGSIZE n & Numparts p	UTS – PBR	UTS – PBR

# Huffman compression

New type of compression dictionary

Requirements:

- z14 or beyond
- Universal table space (UTS) only
- DSNZPARM: **TS\_COMPRESSION\_TYPE**
  - **FIXED\_LENGTH** (previous compression logic)
  - **HUFFMAN** (new compression; entropy encoding)
- Table space has COMPRESS YES attribute

Next dictionary build (REORG, LOAD REPLACE or INSERT) generates Huffman dictionary

Which compression algorithm was used?

- Use DSN1PRNT
- Check header page field HPGZLD
  - L or F indicates fixed-length dictionary
  - H indicates Huffman dictionary

# Built-in functions pass-through to Db2 Analytics Accelerator

- Db2 verifies the data types of the parameters are valid for the functions
- Accelerator engine does all other function resolution and validation
- Db2 Analytics Accelerator V7 or above

CUME\_DIST

FIRST\_VALUE

LAG

LAST\_VALUE

LEAD

NTH\_VALUE

NTILE

PERCENT\_RANK

RATIO\_TO\_REPORT

REGEXP\_COUNT

REGEXP\_INSTR

REGEXP\_LIKE

REGEXP\_REPLACE

## V12R1M504: New BIFs pass-through (reference)

**CUME\_DIST** : Returns a cumulative distribution of a row within an OLAP window

**FIRST\_VALUE** : Returns the expression value for the first row in an OLAP window

**LAG** : Returns the expression value for the row at offset rows before the current row

**LAST\_VALUE** : Returns the expression value for the last row in an OLAP window

**LEAD** : Returns the expression value for the row at offset rows after the current row

**NTH\_VALUE**: Returns the expression value for the nth-row row in an OLAP window

**NTILE** : Returns the quantile rank of a row

**PERCENT\_RANK** : Returns a relative percentile rank of a row within an OLAP window

**RATIO\_TO\_REPORT** : Returns the ratio of an argument to the sum of the arguments in an OLAP partition

**REGEXP\_COUNT** : Returns a count of the number of times that a regular expression pattern is matched in a string

**REGEXP\_INSTR** : Returns the starting or ending position of the matched substring, depending on the value of the return\_option argument

**REGEXP\_LIKE** : Returns an INTEGER value of 0 or 1 indicating if the regular expression pattern is found in a string

**REGEXP\_REPLACE** : Returns a modified version of the source string where occurrences of the regular expression pattern found in the source string are replaced with the specified replacement string

# Syntax flexibility for special registers, NULL predicates

- New syntax options make it easier to port applications and data to Db2 for z/OS from other platforms

- Example

- SET :WS\_NAME = CLIENT WRKSTNNAME

## Special registers

Existing syntax	New syntax option
CURRENT CLIENT_ACCTNG	CLIENT ACCTNG
CURRENT CLIENT_APPLNAME	CLIENT APPLNAME
CURRENT CLIENT_USERID	CLIENT USERID
CURRENT CLIENT_WRKSTNNAME	CLIENT WRKSTNNAME
CURRENT SERVER	CURRENT_SERVER
CURRENT TIME_ZONE CURRENT TIMEZONE	CURRENT_TIMEZONE

## NULL predicates

Existing syntax	New syntax option
IS NULL	ISNULL
IS NOT NULL	NOTNULL



# Function level 505

## PH09191

- REBIND phase-in
  - Db2 Analytics Accelerator hybrid transactional analytical processing (HTAP) phase 2
  - Built-in function (BIF) for column encryption
  - Indexes for DECFLOAT columns
  - Removal of trigger restrictions for temporal tables and for transparent archiving
  - RUNSTATS sampling simplification
- Note: PH28693 (January 2021) for REBIND phase-in

# REBIND phase-in

## Problem statement

- DBA cannot bind packages that are in use, for example
  - Cannot reoptimize
  - Cannot change bind options
  - Cannot switch to previous access path in case of regression
- 'in use' means use count is  $> 0$ 
  - Applies to RELEASE COMMIT until commit
  - Applies to RELEASE DEALLOCATE until thread termination

## Solution: REBIND phase-in

- Existing threads use current package copy
- DBA issues REBIND; Db2 creates new package copy
  - Db2 marks current copy 'do not use' in EDM pools
- New threads use new copy
  - Old copy deleted when no longer used

# REBIND phase-in example

Thread 1 – copy ID 0



Thread 2 – copy ID 0



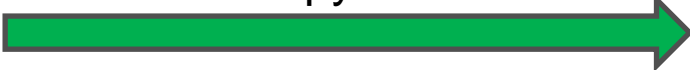
REBIND – copy ID 4: CURRENT



Thread 3 – copy ID 0



Thread 4 – copy ID 4



# REBIND phase-in sequence of events

- Threads executing CURRENT package-copy (copy ID n)
- REBIND
  - Create a next CURRENT copy of package with new copy ID (n+1)
  - Replicate old CURRENT copy (n) to PREVIOUS (copy ID 1) and ORIGINAL (copy ID 2)
    - If needed
  - Move copy ID n to SYSPACKCOPY as phased-out copy
- New threads load and execute new CURRENT copy (n+1) once the REBIND completes

# REBIND phase-in COPYID

COPYID column	in SYSIBM...	Remark
0, 4, 5, ... 16	SYSPACKAGE	CURRENT, used in wrap-around mode
1	SYSPACKCOPY	PREVIOUS
2	SYSPACKCOPY	ORIGINAL
3	-	reserved
0, 4, 5, ... 16	SYSPACKCOPY	Phased-out copies, until deleted

# REBIND phase-in details

- Current copy
  - 1 row in SYSPACKAGE, including COPYID column
  - Copy ID generated as 0, 4, 5, 6, ... 16 then wrap back
    - 1 = PREVIOUS, 2 = ORIGINAL, 3 is reserved
- Phased-out copies
  - In SYSPACKCOPY (with COPYID other than 1, 2)
  - Cleaned up on subsequent REBIND
- EXPLAIN PACKAGE COPY copy-id
  - CURRENT: copy ID in SYSPACKAGE.COPYID column (0 or non-0)
  - Omit COPY clause: includes CURRENT, PREVIOUS, and ORIGINAL
  - Changed value in PLAN\_TABLE.HINT\_USED
- New copy ID field in IFCID 239 package accounting
  - Increased QPAC mapping size

# REBIND SWITCH phase-in

- Threads are executing CURRENT copy ID (n)
- REBIND SWITCH:
  - Copy PREVIOUS or ORIGINAL to new CURRENT copy ID (n+1)
  - Copy n becomes phased-out copy
  - Replicate phased-out copy (n) into PREVIOUS and ORIGINAL if needed
- New threads can execute new CURRENT (n+1)

# REBIND phase-in eligibility

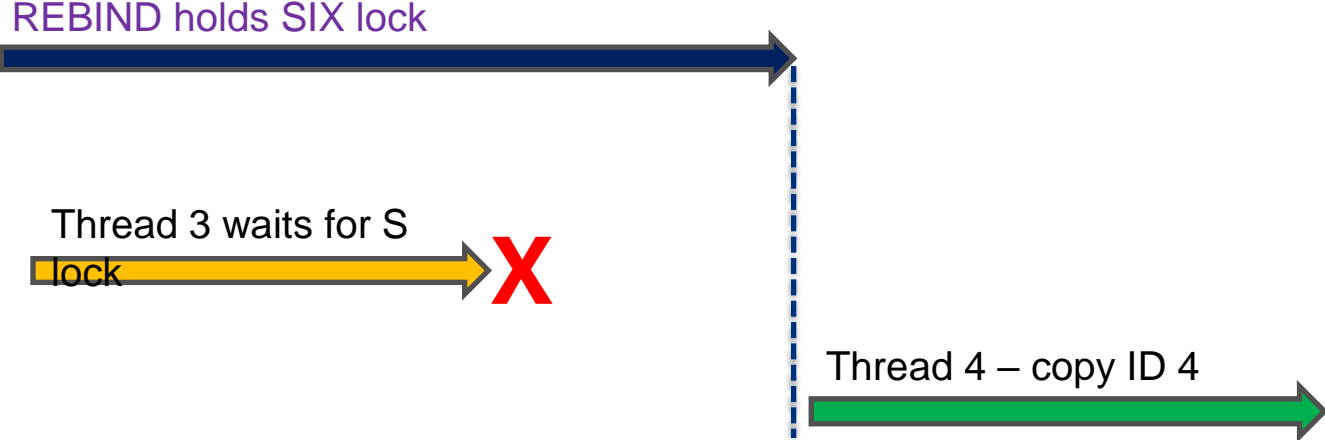
- Phase-in occurs only when current copy is in use
- Initial support:
  - PLANMGMT(EXTENDED) APREUSE(NO)
  - PLANMGMT(EXTENDED) APREUSE(WARN|ERROR) APREUSESOURCE(CURRENT)
- Packages for native SQL routines or advanced triggers do not currently support REBIND phase-in
- REBIND phase-in is an always-on feature in function level 505
  - No APPLCOMPAT requirement



# FREE phased-out copy

- Phased-out copy clean up on REBIND
  - Query package lock holders
  - Compare oldest thread's package allocation time to the time a copy becomes phased-out
    - DSNT500I message with reason code 00E30307 to show thread blocking FREE
  - New IFCID 393 to indicate long running thread RELEASE(DEALLOCATE), uncommitted thread

# REBIND phase-in holds SIX lock, existing scenario 1



# REBIND phase-in wait for SIX lock, existing scenario 2

Thread 1 – copy ID 0 holds S lock



Thread 2 – copy ID 0



REBIND – copy ID 4: CURRENT



REBIND waits for SIX lock **X** holds U lock

Thread 3 waits for S



Thread 4 – copy ID 4



# REBIND phase-in without SIX lock (PH28693)

Thread 1 holds S lock, executing copy ID 0



Thread 2 – copy ID 0



REBIND – copy ID 4: CURRENT



REBIND holds U lock, generates copy ID 4

Thread 3 holds S lock, executing copy ID 0

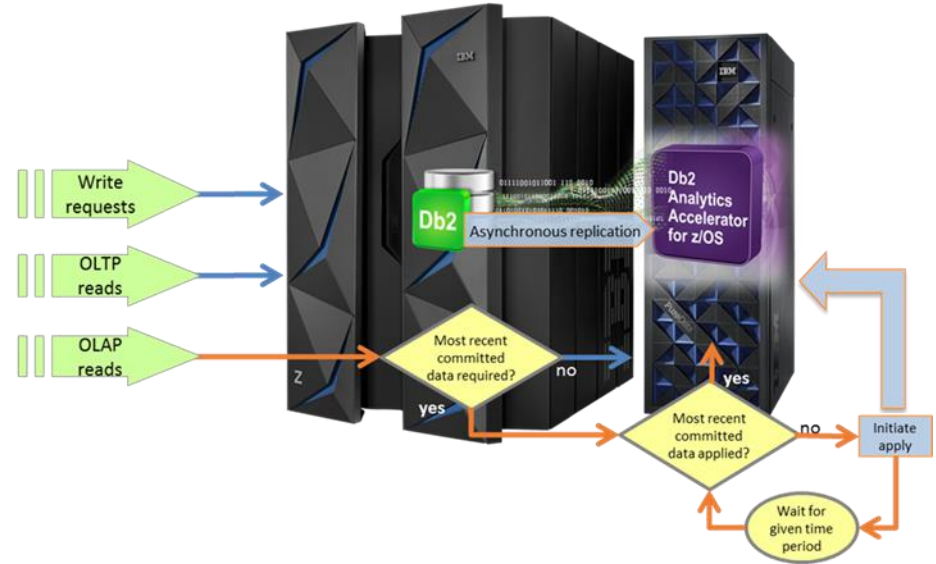


Thread 4 – copy ID 4



# Db2 Analytics Accelerator HTAP phase 2

- Data coherency between Db2 and Accelerator
  - No concern about latency of committed data
    - Queries run when data available
  - No longer need to explain to users why committed data not seen
    - Data used by SELECT contains corresponding commits
- Exploits integrated synchronization
  - Special log processing technology



# BIF for column encryption



## ENCRYPT\_DATAKEY (data string, keylabel, algorithm)

- Column-based encryption of security-sensitive data
- String and numeric datatypes are supported
  - Resulting datatype is VARBINARY for any non-LOB and BLOB for any LOB input value
  - Schema change required for cipher text column
- Use of ICSF protected keys and RACF keylabel protection
  - Primary authid requires permit for keylabel defined in CSFKEYS class
- **AES256DJR** – 256-bit AES CBC mode encryption algorithm
  - 'D' – fixed initialization vector to generate a cipher text
  - 'R' – random initialization vector to generate a cipher text

## DECRYPT\_DATAKEY\_xxxx (encrypted data)

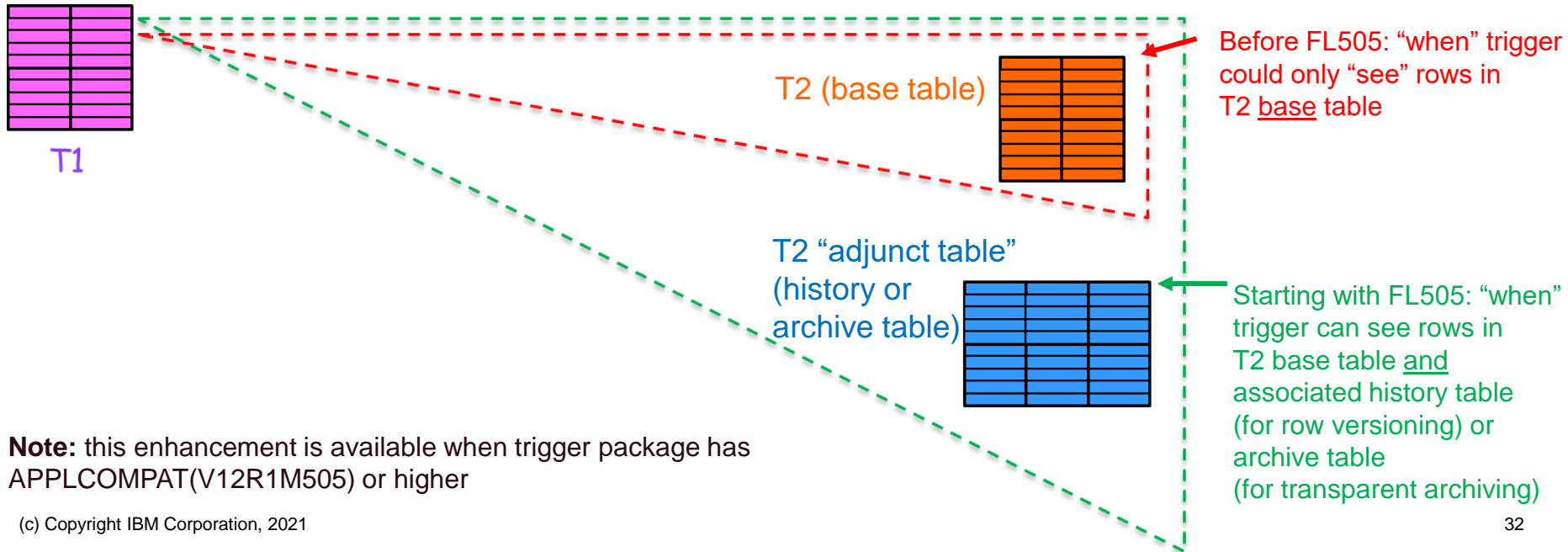
- Datatype dependent BIFs: xxxx = BIGINT or CHAR or INTEGER, etc.

# Indexes for DECFLOAT columns

- DECFLOAT columns can now:
  - be part of an index (unique or non-unique)
  - be specified in a unique constraint
  - be specified in a primary key constraint
- Other restrictions remain
  - DECFLOAT columns can neither be specified in referential constraints nor be part of a partitioning key
  - No support for DECFLOAT columns in INCLUDE indexes nor in indexes on expression
- Note: implicit casting to DECFLOAT data type is still stage 2 in Db2 12, regardless of index support

# Removal of trigger restrictions for temporal tables and for transparent archiving

Scenario: “when” trigger on update to T1 when condition true for T2  
(T2 is archive-enabled table in this example)



**Note:** this enhancement is available when trigger package has APPLCOMPAT(V12R1M505) or higher



# Removal of trigger restrictions for temporal tables and for transparent archiving

- Problem: WHEN clause of trigger could not be applied to data in history table (system-time temporal) or archive table
  - In more technical terms: WHEN clause of trigger could not reference...
    - ...table for which row versioning has been enabled, if trigger package bound with SYSTIMESENSITIVE(YES)
    - ...archive-enabled table, if trigger package bound with ARCHIVESENSITIVE(YES)
  - Trigger packages could be bound or rebound with SYSTIMESENSITIVE(NO) or ARCHIVESENSITIVE(NO), but that means no trigger access to data in history or archive table
- Db2 12 function level M505 removes these restrictions when trigger package bound with APPLCOMPAT(V12R1M505)

# Function level 506

## PH16829

- DROP TABLE: automatic DROP of explicit table spaces
- SQL syntax compatibility for scalar functions
- Explicitly created table spaces are dropped as part of DROP TABLE processing
  - For UTS and LOB table spaces
  - DROP TABLE no longer fails with -669

# SQL syntax compatibility for scalar functions

Current name	Alternate name
COVARIANCE or COVAR	COVAR_POP
CHARACTER_LENGTH	CHAR_LENGTH
CLOB	TO_CLOB
HASH_MD5, HASH_SHA1, HASH_SHA256	HASH with different integer constants as second argument
LEFT	STRLEFT
POSSTR	STRPOS
POWER	POW
RAND	RANDOM
RIGHT	STRRIGHT
TIMESTAMP_FORMAT or TO_DATE	TO_TIMESTAMP

# Function level 507

## PH24371

- Application granularity for locking limits
  - NUMLKUS and NUMLKTS
  - New global variables
    - `SYSIBMADM.MAX_LOCKS_PER_TABLESPACE`
    - `SYSIBMADM.MAX_LOCKS_PER_USER`
    - Can be set for DDF applications with system profile monitoring tables
- Deletion of old statistics when using statistics profiles (PH16345)
- `CREATE OR REPLACE` for procedures
- New pass-through expressions with IBM Db2 Analytics Accelerator

# Application granularity for locking limits

- Before FL507, lock limits for application processes set at Db2 level via two ZPARMs
  - **NUMLKTS**: max locks (page, row or LOB) that a single application process can hold at one time for one table or table space
  - **NUMLKUS**: max locks (page, row or LOB) that a single application process can hold at one time on all table spaces
- Different application processes have different locking requirements
  - If ZPARM values too low, processes that need to hold many locks at once can fail
  - If ZPARM values too high, poor application design can be tolerated unknowingly
- New built-in global variables
  - **SYSIBMADM.MAX\_LOCKS\_PER\_TABLESPACE**
  - **SYSIBMADM.MAX\_LOCKS\_PER\_USER**
- Example: TS1 has LOCKSIZE ROW and LOCKMAX SYSTEM and NUMLKTS = 2000; process inserts 4000 rows into TS1 in a single unit of work
  - Result: lock escalation occurs at insert of 2000<sup>th</sup> row, and no other processes can access the table space until X-lock released (generally at commit)
- Same situation, but app issues this before inserts:
  - SET SYSIBM.MAX\_LOCKS\_PER\_TABLESPACE = 4000;
  - Result: app successfully insert s4000 rows without triggering lock escalation

# CREATE OR REPLACE for procedures

- Goal: consistent deployment of Db2 objects, including stored procedures
  - BIND subcommand with DEPLOY option is not ideal
  - DROP and re-CREATE is not ideal
- **CREATE OR REPLACE PROCEDURE**
  - Native SQL or external stored procedures
    - For native SQL procedure, can specify VERSION
      - If exists, replaced
      - If does not exist, new version added
  - Unlike DROP and re-CREATE:
    - Existing authorizations preserved
    - Not blocked if procedure called by SQL procedure language (SQLPL) routine

# CREATE OR REPLACE PROCEDURE examples


## Create procedure MYPROC1

```
CREATE PROCEDURE MYPROC1
  ( IN  P1 CHAR(5) ,
    OUT P2 DECIMAL(15,2) )
BEGIN
  SELECT AVG(SALARY) INTO P2
  FROM DSN8C10.EMP
  WHERE WORKDEPT = P1;
END
```

## Replace MYPROC1 with new definition

```
CREATE OR REPLACE PROCEDURE MYPROC1
  ( IN  P1 CHAR(5) ,
    OUT P2 DECIMAL (15,2) )
BEGIN
  SELECT AVG(SALARY + 1000) INTO P2
  FROM DSN8C10.EMP
  WHERE WORKDEPT = P1;
END
```

Change  
body of  
procedure



## Add version V2 of MYPROC1

```
CREATE OR REPLACE PROCEDURE MYPROC1
  ( IN  P1 CHAR(5) ,
    OUT P2 DECIMAL (15,2) )
VERSION V2
BEGIN
  SELECT AVG(SALARY + 5000) INTO P2
  FROM DSN8C10.EMP
  WHERE WORKDEPT = P1;
END
```

## Replace version V2 of MYPROC1

```
CREATE OR REPLACE PROCEDURE MYPROC1
  ( IN  P1 CHAR(5) ,
    OUT P2 DECIMAL (15,2) )
VERSION V2
BEGIN
  SELECT AVG(SALARY + 9000) INTO P2
  FROM DSN8C10.EMP
  WHERE WORKDEPT = P1;
END
```

# New pass-through expressions for accelerator

- ADD\_DAYS
- BTRIM
- DAYS\_BETWEEN
- NEXT\_MONTH
- Regression functions (REGR\_AVGX, REGR\_AVGY, REGR\_COUNT, ...)
- ROUND\_TIMESTAMP if invoked with a date expression



# Function level 508

## PH29392

- Migration of multi-table table space to partition by growth (PBG) universal table space (UTS)
- Prior to FL 508, an outage required to migrate to UTS PBG from multi-table table space
- For each table in the table space
  - Create target PBG UTS
  - ALTER table as pending change
- REORG TABLESPACE to materialize pending ALTER statements
  - Possibly REBIND
- ALTER TABLESPACE now has **MOVE TABLE** clause

# Function level 509

## PH33015

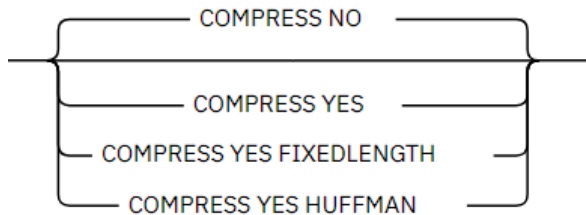
- Huffman compression: specify compression algorithm at table, table space, or partition level
- Tamper-proof audit policies
- Temporal RI allows UPDATE or DELETE on parent table
- High availability for accelerator-only tables (IDAA: HA for IOT)

# Huffman Compression (M509)



## What is different?

- Choose compression type at the table space or partition level for UTS table spaces



- For COMPRESS YES default is TS\_COMPRESSION\_TYPE



## How do you specify?

- CREATE TABLESPACE
- ALTER TABLESPACE
- CREATE TABLE
  - For implicitly created table spaces
  - Default specified by DSNZPARM IMPTSCMP (NO, YES)



## V12R1M509 Catalog level

- SYSTABLEPART columns
- **COMPRESS\_USED** (new)
  - F, H, ' ' or NULL
- **COMPRESS** (changed)
  - Y, F, H, ' '
- SYSTABLESPACE column
- **COMPRESS** (changed)
  - Y, F, H, ' ' or NULL

# Non-function level dependent highlights

- **Db2 distributed data facility (DDF)**
  - System profile monitoring
    - Set limit of queued or inactive threads
    - MONITOR ALL CONNECTIONS, MONITOR ALL THREADS
  - Data server drivers – JDBC, ODBC, .net
    - Continuous delivery enhancements
    - Enhanced messaging
  - REST services command, versioning
  - Transport layer security-only access
- **Utilities**
  - Redirected recovery
  - Usability
  - Performance
  - zIIP exploitation
  - FlashCopy
- **DevOps**
  - Developer Extension for VS Code
- **Security**
  - Dump diagnostics
- **Analytics**
  - Accelerator related enhancements
  - Db2 AI for z/OS (Db2ZAI)
- **Z synergy**
  - zHyperLink: database read, log write
  - Compression performance
  - Encryption performance
- **Data sharing**
  - Asynchronous group buffer pool cross invalidation
  - Messaging for auto recovery of LPL/GRECP
- **Usability**
  - Transparent archiving parameter setting
  - System parameter (DSNZPARM) simplification

# Summary: Db2 12 for z/OS latest features

Continuous delivery is here

Function level enhancements

- Numerous benefits
- Control with ACTIVATE command, APPLCOMPAT setting on package
- Application behaviors

Non-function level enhancements

- Numerous benefits (over 100 so far)
- Control with maintenance stream

# Thank you!

Mark Rader  
*ATG Db2 for z/OS Specialist*  
[mrader@us.ibm.com](mailto:mrader@us.ibm.com)

# Notices and disclaimers

© 2021 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

## **U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.**

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **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.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”

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

Performance data contained herein was generally obtained in a controlled, isolated environments. 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.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

# Notices and disclaimers

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] 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](http://www.ibm.com/legal/copytrade.shtml)



