# Db2 12 for z/OS

# Recipe for Successful Db2 12 Migration

**Anthony Ciabattoni**
*Db2 for z/OS Development*

IBM | IBM z Analytics

# Disclaimer/Trademarks



THE INFORMATION CONTAINED IN THIS DOCUMENT HAS NOT BEEN SUBMITTED TO ANY FORMAL IBM TEST AND IS DISTRIBUTED AS IS. THE USE OF THIS INFORMATION OR THE IMPLEMENTATION OF ANY OF THESE TECHNIQUES IS A CUSTOMER RESPONSIBILITY AND DEPENDS ON THE CUSTOMER'S ABILITY TO EVALUATE AND INTEGRATE THEM INTO THE CUSTOMER'S OPERATIONAL ENVIRONMENT. WHILE IBM MAY HAVE REVIEWED EACH ITEM FOR ACCURACY IN A SPECIFIC SITUATION, THERE IS NO GUARANTEE THAT THE SAME OR SIMILAR RESULTS WILL BE OBTAINED ELSEWHERE. ANYONE ATTEMPTING TO ADAPT THESE TECHNIQUES TO THEIR OWN ENVIRONMENTS DO SO AT THEIR OWN RISK.

ANY PERFORMANCE DATA CONTAINED IN THIS DOCUMENT WERE DETERMINED IN VARIOUS CONTROLLED LABORATORY ENVIRONMENTS AND ARE FOR REFERENCE PURPOSES ONLY. CUSTOMERS SHOULD NOT ADAPT THESE PERFORMANCE NUMBERS TO THEIR OWN ENVIRONMENTS AS SYSTEM PERFORMANCE STANDARDS. THE RESULTS THAT MAY BE OBTAINED IN OTHER OPERATING ENVIRONMENTS MAY VARY SIGNIFICANTLY. USERS OF THIS DOCUMENT SHOULD VERIFY THE APPLICABLE DATA FOR THEIR SPECIFIC ENVIRONMENT.

Trademarks
IBM, the IBM logo, ibm.com, Db2, IBM Z and z/OS are trademarks of International Business Machines Corp., 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.

# Agenda

- Continuous Delivery
- Db2 Maintenance Strategy
- Db2 Conservative Migration Recommendations
- Db2 12 Single Step Migration
- APREUS
- REBIND Strategy
- Function Level V12R1M500 +
- Universal Tablespace Strategy
- Questions

# Db2 for z/OS Into the Future


Continuous delivery brings new functions to market 2-4 times per year instead of every 3 years

As of 4Q2019, six function levels have been delivered, providing 20 gated features, plus nearly 100 new capabilities since Db2 12 GA

# New Db2 for z/OS strategy for delivering new function

- We are dedicating ourselves to going forward on a continuous delivery model
  - Radical internal changes are required within Db2 for z/OS Development to do this
- Db2 12 is the starting point after GA
  - There will be significantly higher volume of continuously delivered items
- Customers will see a single maintenance stream for Db2 12, with the new function delivered into that
  - The function will be designed to be easily consumable
- Point releases or versions will be a rare exception but will eventually occur
  - There are reasons why we might want to have a point release or new version
    - Adopt a new compiler, extend control structures, enable an architecture level set
- Db2 for z/OS Development will have relentless focus on maintaining continuous production level reliability in the service stream
- We are dedicated to doing this
  - We will control the input to “the pipe”, the size and risk of the items
  - Increased internal focus on function and performance regression testing
  - We will deliver new function when the quality is right, not on a delivery date

# New Db2 for z/OS strategy for delivering new function …

- With Continuous Delivery, there is a single delivery mechanism for defect fixes and enhancements
  - PTFs (and collections of PTFs like PUTLEVEL and RSU) → same as today
- With Continuous Delivery, there are four + 1 Db2 for z/OS levels
  - **Maintenance level (ML) - lifted by applying maintenance**
    - Also known as code level - contains defect and new enhancement fixes
    - Most new functions are shipped disabled until the appropriate new function level is activated
  - **Catalog level (CL) - vehicle to enable new FL - cumulative (skip level possible)**
    - Db2 Catalog changes that are needed for some FLs
  - **Function level (FL) - needs to be activated - cumulative (skip level possible)**
    - Introduces new Db2 features and functionality
    - No impact or change in existing application behavior
  - **APPLCOMPAT level (AC) - set by application - provides an "island of stability" for a given application**
    - AC level in BIND/REBIND of package must be <= FL and overrides zParm
    - AC must be advanced to exploit new SQL function
    - Even if FL is regressed to an earlier level *new function continues to be used by that application*
  - **clientApplCompat level - driver configuration keyword allows remote apps access to new functions**
- Minimum starting point for Continuous Delivery is Db2 12 FL=V12R1M500

# Continuous delivery …

- New function/enhancements will be delivered via the maintenance stream in the form of new Function Levels
  - The intent is to enable new Db2 features and functions an easy consumable method at the appropriate time for each customer
  - **New SQL functionality available in V12R1M500…501, etc cannot be used until package is bound with APPLCOMPAT value of V12R1M100
    - Remember the Zparm is always bringing you ‘down’-level, so coordinate Zparm and application changes on an aggressive schedule to up-level them
  - CURRENT APPLICATION COMPATIBILITY special register for dynamic SQL
  - clientApplCompat (in driver) will be a prereq for applications to begin using future new function – a future feature similar to: Global XA transactions and enhanced session data
  - **If the required PTFs for a specific CL/FL are backed off for some reason, Db2 will not restart**
    - Proper preventative maintenance strategy should be developed to take into consideration the continuous delivery model
  - Best practice (not a requirement) would dictate an SMP/E APPLY and ACCEPT before activating a new function level

# Function levels

- The Function Level (FL) that a Db2 member is capable of running is directly related to the maintenance level of that member
  - SYSIBM.SYSLEVLUPDATES track when function levels were implemented
- All members must be capable of supporting a FL before that FL can be activated
  - Maintenance is member scope – members can run code capable of supporting different FLs
  - FL activation is a group scope
  - **ACTIVATE FL TEST** allows you to simulate the command to ensure catalog and member level is all correct (DSNU757I)
- It is possible to return to a previous FL
  - E.G. if you are a FL V12R1M506, you can return to V12R1M504* (wherever I came from). The * will indicate that the group has previously been at a higher FL
  - All existing usage of a feature of a FL will continue to be allowed, but new exploitation of the previous FL will be prevented
  - Returning to a previous FL will still require that all members joining a group must support the highest activated FL ever set for the group

# APPLCOMPAT

- APPLCOMPAT level (AC) - set by application - provides an “island of stability” for a given application
  - AC level in BIND/REBIND of package must be <= FL and overrides zParm
  - AC must be advanced to exploit new SQL function
  - Even if FL is regressed to an earlier level *new function continues to be used by the application* bound on AC Level
- Impact of falling back to older Function Level
  - Function Level will be marked with * …. Example
    - Db2 at FL504
    - Package is bound with APPLCOMPAT 504
    - DB2 function level is regressed to FL502*
    - Package bound with APPLCOMPAT 504 continues to function
    - All rebinds going forward can only be bound with APPLCOMPAT 502
- Db2 Driver Packages & APPLCOMPAT
  - Minimum of 2 collection ids, can be more
  - Example …
    - NULLID
    - NULLID50x
- Db2 System Profiles can be exploited to manage DDF threads

# Db2 Continuous Delivery history -> IBM Knowledge Center

Goals: faster delivery, easier to consume for customers

## FL 501 – 1st post-GA deliver

- Quality, stability is priority #1
- Function levels (FLs) are the mechanism to activate new features on V12
  - System level and application level controls
  - FL 500 is base V12 "NFM". FLs 501, 502, … beyond that

## FL 502 – April, 2018. APAR PI95511

- LISTAG
- Transparent Dataset Encryption: Db2 DBA controls
- Casting numeric to GRAPHIC/VARGRAPHIC

**Nearly 100 new Db2 12 features delivered since GA are not tied to specific Function Level**

## FL 503 – Sept, 2018. APAR PH00506

- Db2 AI for z/OS (Db2ZAI)
- Migration support on DATA CHANGE OPERATION for temporal auditing
- Enablement for replication of system-period temporal tables and generated expression columns

https://www.ibm.com/support/knowledgecenter/SSEPEK_12.0.0/wnew/src/tpc/db2z_db2functionlevels.html

Attention: Many other features delivered in this timeframe (V11 &V12) that were not tied to a Function Level

# Db2 Continuous Delivery history -> IBM Knowledge Center

reinvention in 4Q2019, no new FL in Feb 2020.

## FL 504 – Mar, 2019. APAR PH07672

- **Huffman data compression**
- **New SQL syntax** alternatives
- Prevent new **deprecated** objects
- **Passthru of Built- In Functions** (OLAP and REGEX) to IDAA

## FL 505 – June, 2019. APAR PH09191

- **Rebind phase-in** for packages that are being used for execution
- **Improved RUNSTATS** performance with automatic sampling by default
- **New built in functions** for encrypting and decryption with key labels

## FL 506 – October 2019. APAR 16829

- Newly supported **names for existing built-in functions.**
- **DROP statement enhancements** for implicitly dropping explicitly created table spaces. Avoiding SQLCODE -669
  - **Might introduce incompatibility**
  - **Dropping** a table that resides in an explicitly created **Universal Table Space** no longer returns an error. Instead, the table space is implicitly dropped.
  - **Dropping** an **Auxiliary Table** that resides in an explicitly created LOB table space no longer leaves the LOB table space in the database. Instead, the table space is implicitly dropped.

FL507 – mid 2020
FL508 - 4TH Quarter 2020

https://www.ibm.com/support/knowledgecenter/SSEPEK_12.0.0/wnew/src/tpc/db2z_db2functionlevels.html

Attention: Many other features delivered in this timeframe (V11 & V12), that were not tied to a Function Level

# Db2 Recent Deliveries Not Tied to Function Levels

- REPAIR WRITELOG option to allow vendors to write a dialog record for replication refresh
- REORG NOCHECKPEND option
- REORG DISCARD optimization for IDAA
- Online LOAD compatibility for online REORG
- BIND SERVICE COPY to allow copying of REST services
- Improved IFCID 401 information for IDAA
- Improved IFCID 402 profile statistics
- Ability to choose how many threads to queue when an application-based profile exception is reached
- Additional QUERY_ACCEL_OPTIONS option to improve support for IDAA V7 and above
- Install CLIST improvements (PH12117)
- SYSPROC.ADMIN_INFO_IFCID stored procedure
- SYSPROC.ADMIN_INFO_SQL improvements for case id’s and hidden zparms
- New UTILS_BLOK_FOR_CDC zparm to better control replication refreshes
- Enhanced DISPLAY CLAIMERS of blocking threads(PH13850)
- Enables concurrent REORG/LOAD RESUME with SHRLEVEL CHANGE on object (PH11255)
- Enables RLF in-memory auto Refresh START without Stop/Start (PH06082)
- Multi-Row Insert for Accelerator Only Tables
- Enhanced robustness and monitoring capabilities for Insert Algorithm 2

# Sampling of Db2 for z/OS 2020 items

- Estimate Db2z space savings for Huffman Compression
- zHyperLink dual logging performance improvements
- Client Information propagation from Db2 REST call
- Profile Table Remove Implicit Queueing
- RUNSTATS dynamic profile support
- zPARM Simplification (best practices)
- LOAD PRESORT
- Ability to specify number of tape units for REORG inline copies
- Enhance FORCE option for REORG utility
- Improved reliability & contention relief for online migration
- UNI_60 locale for UPPER/LOWER/TRANSLATE scalar functions
- LOAD REPLACE defer-define AUX DSs
- Function Level 507 (skipped February)
- Function Level 508 (target 4th quarter 2020)
- Application granularity for NUMLKUS and NUMLKTS
- Expand In-Memory Database Support to Non-Unique indexes: Improve INCLUDE index
- Enhancements for Insert Algorithm 2 capabilities
- Create or Replace Procedure
- Eliminate need to bind locally for remote stored procedure
- Contiguous bufferpool (PGSTEAL(NON)) with online reorg
- Increased flexibility for package ownership
- Redirected recovery – recover to another target
- Migration of multi-table tablespace to UTS PBG
- Accounting & statistics for IDAA Insync Remote Logreader tasks

# Db2 Maintenance Strategy

- Recommended maintenance strategy
  - 4 preventive maintenance drops per year
    - 2 major drops
      - ✓ RSU's, HIPERs, PEs and customer impacting APARs
    - 2 minor drops
      - ✓ HIPERs, PEs and customer impacting APARs
    - On a continuous basis perform weekly review of Enhanced HOLDDATA looking at new HIPERs and new PEs
      - ✓ Add into existing maintenance package or expedite into production for 1-2 weeks in test if new HIPER problem is vicious or where apply of PE resolution fix unblocks important missing HIPERs that could not be applied previously
    - Consider having multiple SMPE target zones or multiple release level SMPE environments
  - Design SMPE to have multiple environments (3) at different preventative maintenance levels
    - 1 SMPE environment that contains multiple target zones
    - Multiple pseudo independent SMPE environments

# Db2 Maintenance Strategy …

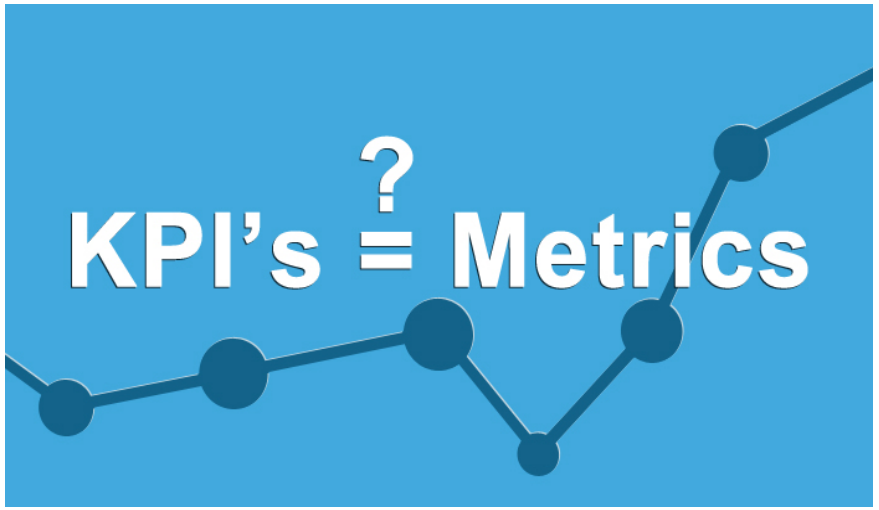
- Db2 maintenance migration consideration
  - Take as much preventative maintenance as possible on Db2 12 including an aggressive base RSU level, HIPERs and PEs
    - A common strategy is to anticipate the migration weekend and choose a base RSU approximately 3-4 months prior to that weekend
    - Continue to review enhanced HOLDATA and take action on high priority impacting HIPERs while moving Db2 12 through your environments
  - It is highly recommended not to have a large time gap between Db2 11 and Db2 12 preventative maintenance at the time of migration
  - In order to keep the level of Db2 11 reasonably close apply one last preventative maintenance package prior to migrating to Db2 12
    - Reduce the shear amount of change that is introduced over time via the maintenance stream
    - Reduce the risk in the event of Db2 release fallback
    - Minimize the preventative maintenance time gap while in mixed release coexistence (Db2 data sharing)
  - Develop a proper preventative maintenance strategy to take into consideration the Db2 12 Continuous Delivery model
    - When starting a new preventative maintenance roll, ACCEPT the previous maintenance and entertain activating the previously applied function level

# Conservative Migration Approach

- Predicated on protecting production stability  for a conservative customer
- Have well defined technical program planning, management, and contain scope
  - Function level V12R1M100 ← Similar to CM mode (single mode migration)
  - Start all inactive Db2 members of data sharing group before activating function level V12R1M500
  - Function level V12R1M500 ← similar to NFM (system wide event)
- Perform thorough testing
  - Run pre-migration queries early and often, and act on incompatibilities and removed functions
  - Perform “Premigration Catalog Migration Testing” on clone
    - Perform consistency checking of Db2 11 Cat/Dir (REPAIR DBD TEST/DIAGNOSE + CHECK DATA/LOB/INDEX + DSNTESQ + ….)
    - Practice conversion of BSDS conversion to 10-byte extended format (DSNJCNVT)
    - Practice DSNTIJTC (CATMAINT)
    - Under Db2 12 REBIND static SQL packages, explain SQL VIEWs, etc and perform impact analysis
  - Design for release fallback, mixed release coexistence, test and practice
  - Perform as much application testing as possible to “keep fires away” from production
  - Migrate to production only when Db2 12 is proven to be running stable in pre-production, and not be date driven

# Key Performance Indicators

- Build a performance 'profile' based on KPIs under Db2 11 as a base line for comparison against Db2 12
  - Subsystem level using Statistics Trace
  - Key Performance Indicators
    - Number or occurrences
      - ✓ Transactions/Second
      - ✓ Commits
    - CPU/Elapsed Time
      - ✓ CPU/Transactions & Elapsed/Transaction
      - ✓ CPU/Commit & Elapsed/Commit
    - Additional Indicators
      - ✓ DML/Transaction and type of DML – Select, Insert, Update, Delete
      - ✓ Getpages/Transaction
      - ✓ Average Sync I/O per Transaction
  - Application process level using Accounting Trace
    - High volume transactions
    - Complex transactions at reasonable volume
    - Critical batch processes



# Simplify Initial Migration into Db2 12

- Minimize number of moving parts and changes in functionality on the initial migration
  - For example, perform the conversion of the BSDS to 10-byte extended format ahead of the migration window
  - Pro-actively plan for increased REAL memory and zIIP capacity requirements
  - Increase size of EDM pool under Db2 11 before leaving Db2 11
    - EDMDBDC & EDMSTMTC
  - Increase size of RID pools under Db2 11 before leaving Db2 11
  - Analyze Db2 active log sizes
    - Pro-actively increase if necessary
    - Minimum goal to keep 6 hours in active log
    - Increase number or size --> Maximum size 4gb – 1 byte
  - Analyze archive log sizes and pools
    - Goal is at least 24 hours of at least archive log 1 or 2 on DASD
  - Rebind deprecated plans and packages with APREUSE(ERROR|WARN) before leaving Db2 11
  - Turn off new features which are on by default
    - Fast Traversal Blocks
    - Insert Algorithm 2

# Db2 12 migrations are a single step

- The standard DSNTIJTC migration job is all you need!
  - Creates new catalog and directory objects, adds columns to existing catalog tables, and creates and updates indexes on the catalog tables to accommodate new Db2 12 objects
    - No REORG, therefore no switch phase
    - DBD locks (minimal)
  - RTS externalization disabled during migration
  - CATMAINT UPDATE LEVEL V12R1M500
  - Once DSNTIJTC processing completes the subsystem/group is now V12!
  - It is now in function level V12R1M100.
  - New function cannot yet be used

# Db2 12 migrations are a single step …

- Online migration is acceptable
- But activity which locks catalog objects can *"potentially"* interfere with the CATMAINT utility operations, including the following:
  - Running utilities
  - Binds, rebinds, and automatic binds
  - DDL statements
  - GRANT or REVOKE statements
  - Running monitors such as Omegamon, TMON, and certain vendor packages
  - Queries against the Db2 catalog

# Db2 12 migrations are a single step …

- If job DSNTIJTC fails …
  - Save the output and correct the problem.
  - Terminate the CATMAINT utility if it is pending
  - Run job DSNTIJTC again without modification
  - If job DSNTIJTC fails again, return to your previous Db2 release
  - CATMAINT failures roll back all Db2 12 changes, the catalog and directory are in the format of your previous Db2 release
  - Altered indexes are not rolled back. Determine if any index is in a pending status by using the CHECK INDEX utility
    - ALTER INDEX SYSIBM.DSNOTX01 ADD COLUMN(VERSION) - SYSTRIGGERS
    - ALTER INDEX SYSIBM.DSNATX02 ADD COLUMN(UNLOADAUTH) – SYSTABAUTH

# Db2 12 migrations are a single step …

- Db2 12 online migration testing methodology
  - In a non-production environment the goal would be disruptive testing with the intent of producing the migration job to fail
  - In a non-production environment test the Db2 12 online migration job at the most disruptive time
    - If more than one non-production environment choose different disruptive scenarios to migrate against
  - If you can produce a failure with normal activity think about crashing Db2 during the migration to simulate a restart scenario
  - The ultimate objective is to acquire confidence in the process
    - Restarting a failed migration step
    - Supplying as much noise and disruptive testing in worst case scenarios
  - **Db2 12 online migration should still be executed at a quiet time and taking all necessary precautions**

# APREUSE

- APREUSE option specifies whether Db2® tries to reuse previous access paths for SQL statements in a package. DB2 uses information about the previous access paths from the directory to create a hint
  - APREUSE(ERRROR)
    - Db2 tries to reuse the previous access paths for SQL statements in the package. If statements in the package cannot reuse the previous access path, the bind operation for the package that contains the statement ends, and processing continues for the next package. Db2 indicates the number of statements that cannot be reused in any package in a message. New and changed statements in a package never prevent the completion of the bind or rebind operation, even though no previous access path is available for reuse
  - APREUSE(WARN)
    - Db2 tries to reuse the previous access paths for SQL statements in the package. Db2 ignores the previous access path and generates a new access path for any statement in the package that cannot reuse the previous access. The bind operation for the package completes, regardless of whether some access paths cannot be reused. Db2 indicates the number of statements that cannot be reused in any package in a message
    - Db2 reports the number of statements that were processed, the number of statements that reused the previous access path and the number of statements that could not reuse the previous access path in a DSNT286I message

# APREUSE ...

- APREUSE option specifies whether Db2® tries to reuse previous access paths for SQL statements in a package. DB2 uses information about the previous access paths from the directory to create a hint
  - APCOMPARE(WARN)
    - Db2 compares the old and new access path for each matching statement. The Db2 optimizer will choose the best access path that is available at bind time. If the access paths are structurally dissimilar, Db2 sends message DSNT285I and continues processing the package.  Db2 indicates the number of statements that cannot be reused in any package in a message
  - APREUSE(NONE/NO)
    - Db2 does not try to reuse previous access paths for statements in the package and a new access path will be created
    - APREUSE(NONE/NO) is the default

# REBIND Strategy

- REBIND high use plans and packages with APREUSE(ERROR|WARN) after migrating to Db2 12 and after proving the stability of Db2 12 in production
  - Initially take a conservative REBIND approach APREUSE(ERROR/WARN), Benefits:
    - Re-enable fast column processing
    - Avoid performance of overhead “puffing” code at runtime
    - Pickup latest runtime performance enhancements
    - Reduce exposure to latent issues seeded previously in earlier releases
  - Gain exposure to new access path selection improvements later
  - DO NOT forget about pre Db2 10 bound plans!
- Collect Db2 12 KPIs and start to compare against baseline KPIs collected before leaving Db2 11
  - Subsystem level using Statistics Trace
  - Key Performance Indicators
  - Application process level using Accounting Trace
    - High volume transactions
    - Complex transactions at reasonable volume
    - Critical batch processes

# Function Level V12R1M500 +

- Consider rolling out an additional preventive maintenance package after successful migration to V12R1M100
- After activating function level V12R1M50+ fallback to Db2 11 is no longer an option
- Set Db2 zPARM APPLCOMPAT=V12R1M500 and leave it
  - Elevating zPARM to a higher APPLCOMPAT after migration to V12R1M501+ needs to have proper considerations
  - Additional considerations Data Server Driver/Db2 Connect packages when incrementing APPLCOMPAT via zPARMs
    - ”Driver type” bind process that rebinds the NULLID.**
      - ✓ APPLCOMPAT isn’t or is unable to be specified in REBIND statement, defaults to zPARM value. There is no additional CATMAINT for function level V12R1500 or V12R1501
- In DSNHDECP Set the SQLLEVEL equal to the Db2 function level
  - A package can be recompiled exploiting new Db2 features and the APPLCOMPAT in the BIND statement can override that lowest level set in zPARMs.
- Explore adding APPLCOMPAT to your organization default BIND parameters in the application life cycle  so that BINDs and not dependent on your zPARMs

# Function Level V12R1M500 +

- After successfully activating function level V12R1M50+ consider exploiting new features and functions
  - Perform due diligence on researching the reliability and robustness
  - Aggressively research, identify and apply supporting Db2 APARs
  - Develop and act on a robust test plan
    - Perform realistic use case test conditions with stress
    - Test anomalies and failure scenarios

# Function Level V12R1M500 + …

- CATMAINT and activating Function Level options
  - Designed to be executed without a “pure maintenance window” but still at a quiet time
  - To date each new function level CATMAINT has been very small and extremely low risk
    - FL502
    - FL503
    - FL505
    - FL507 (coming soon)
  - CATMAINT options
    - Can run each singularly, one catalog change a time
    - Can execute in cumulative manner via one job, one CATMAINT implementing many catalog levels
  - If jumping function levels test falling back and re-activating
  - Follow same testing methodology for CATMAINT entering Db2 12 V12R1M100
- **What is the recommended Db2 Function Level?**
  - Feature and functionality
  - Evergreening periodically, e.g. every 18 – 24 months

# What Function Level to Apply ?

- Function level 500 (activated at Db2 12 installation or after migration – October 2016)
  - Function level 500 (V12R1M500) represents the first opportunity for applications to take advantage of most new capabilities in the Db2 12 initial release, including new SQL capabilities.
- Function level 501 (activation enabled by APAR PI70535 – May 2017)
  - LISTAGG built-in function, which produces a list of all values in a group.
- Function level 502 (activation enabled by APAR PI95511 – May 2018)

  Management of key labels of z/OS DFSMS data set encryption and explicit casting of numeric values to GRAPHIC or VARGRAPHIC.
- Function level 503 (activation enabled by APAR PH00506 – October 2018)
  - IBM Db2 AI for z/OS (Now FL500, ML503), replication of system-period temporal tables and generated expression columns, a change to temporal auditing support for temporal data.
- Function level 504 (activation enabled by APAR PH07672 – April 2019)
  - IBM Z hardware-based Huffman compression of Db2 data, the ability to prevent the creation of certain new deprecated objects types, new support for certain built-in functions by pass-through to IBM Db2 Analytics Accelerator.
- Function level of 505 (activation enabled by APAR PH09191 – June 2019)
  - REBIND Phase In support, automatic page sample by default for RUNSTATS
- Function level of 506 (activation enabled by APAR PH16829 – October 2019)
  - Support for new alternative names for existing built-in functions and support for implicitly dropping explicitly created table spaces.

# Universal Tablespace Strategy and Execution

- Universal Tablespaces (UTS) is the current and future direction of the Db2 for z/OS Development Lab
  - Non-UTS tablespaces deprecated in Db2 Function Level 504
  - Non-UTS Tablespaces will be retired in the future as part of a future function level or VNEXT
  - Multi-table tablespace conversion to PBG
    - Capabilities will be supplied in future function level, point in time targeting FL508
    - Increase in the total number of open data sets in Db2
- Customers need a regimented cook-book for creating new tablespaces
  - UTS - PBG – Partitioned by Growth  is a great replacement for the classic segmented tablespaces
  - UTS - PBR – Partitioned by Range is a natural evolution from moving from classic partitioned
  - UTS - PBR RPN – Partitioned by Range Relative Page Number for supersized UTS PBR
- Customers need a plan to take action for existing tablespaces
  - Develop a plan and a strategy to migrate from classic segmented & table controlled partitioned non-UTS
  - Develop a plan and a strategy to correct previous incorrect decisions to move to UTS PBG
    - Enable partition level independence and parallelism
    - Eliminate technical debt, protect against CPU burn

# Universal Tablespace Strategy and Execution ...

- Primary driver for the developing UTS PBG tablespace was the removal of the 64GB limit for classic segmented tablespace and avoid the disruptive migration to classic partitioned tablespace
- Some considerations
  - All indexes are going to be NPIs
  - Limited partition independence for utilities (REORG, LOAD)
  - Partitioning not used for query parallelism
  - Degraded insert performance (free space search) as the number of partitions grow
  - REORG Considerations
    - REORG PART will fail for a full UTS PBG partition if FREEPAGE or PCTFREE are non-zero
    - Setting system parameter REORG_DROP_PBG_PARTS = ENABLE could lead to operational issues if the number of PARTs are pruned back
      - No point-in-time recovery prior to the REORG that prunes partitions
      - Cannot use DSN1COPY to move data between Db2 systems
- **Should not be using UTS PBG as the design default for all tables (with large number of partitions)**

# Universal Tablespace Strategy and Execution ...

- General recommendations for use of UTS PBG tablespace
  - Only use UTS PBG tablespace as the alternative and replacement for classic segmented tablespace
  - A table greater than ~64GB in size should be created as a UTS PBR tablespace
  - Good reasons to limit number of partitions - should have as few partitions as possible - ideally only 1
  - DSSIZE and SEGSIZE should be consistent with the target size of the object e.g.
    - Small size object: DSSIZE = 2GB and SEGSIZE = 4
    - Medium size object: DSSIZE = 4GB and SEGSIZE = 32
    - Large size object: DSSIZE = 64GB and SEGSIZE = 64
  - REORG at the table space level unless do not have sufficient DASD space for sort
  - Setting system parameter REORG_DROP_PBG_PARTS = DISABLE?
    - If required to prune back the number of partitions
      - ✓Use online system parameter to temporarily enable for controlled use
    - Better still, in Db2 12, use the DROP_PART YES option of REORG

# Hidden ROWID support to partition

- ROWID can be used as a partitioning column
- Application impact if ROWID cannot be hidden
  - APARs to support to define a hidden ROWID
    - PI76972, PI77310, PI77302 (Db2 12)
    - PI77718, PI77719, PI77360 (Db2 11)
- Benefits
  - Allows table to be partitioned where no natural partitioning key exists or candidate partitioning keys
    do not provide a good spread across partitions
  - Transparent to the application
  - Improved insert throughput
    - Less lock/latch contention on index and data

```
CREATE TABLE  PRDA.ZJSCNTP0
      (  CLIENT    VARGRAPHIC(3) NOT NULL,
         WI_ID     VARGRAPHIC(12) NOT NULL,
         LENGTH    SMALLINT,
         DATA      VARCHAR(1000),
         ROW_ID    ROWID NOT NULL
         IMPLICITLY  HIDDEN generated always
      )  PARTITION BY (ROW_ID)
          (PARTITION 1 ENDING AT (X'0FFF'),
           PARTITION 2 ENDING AT (X'1FFF'),
           PARTITION 3 ENDING AT (X'2FFF'),
           PARTITION 4 ENDING AT (X'3FFF'),
           :
           PARTITION 14 ENDING AT (X'DFFF'),
           PARTITION 15 ENDING AT (X'EFFF'),
           PARTITION 16 ENDING AT (MAXVALUE))
```

# Questions

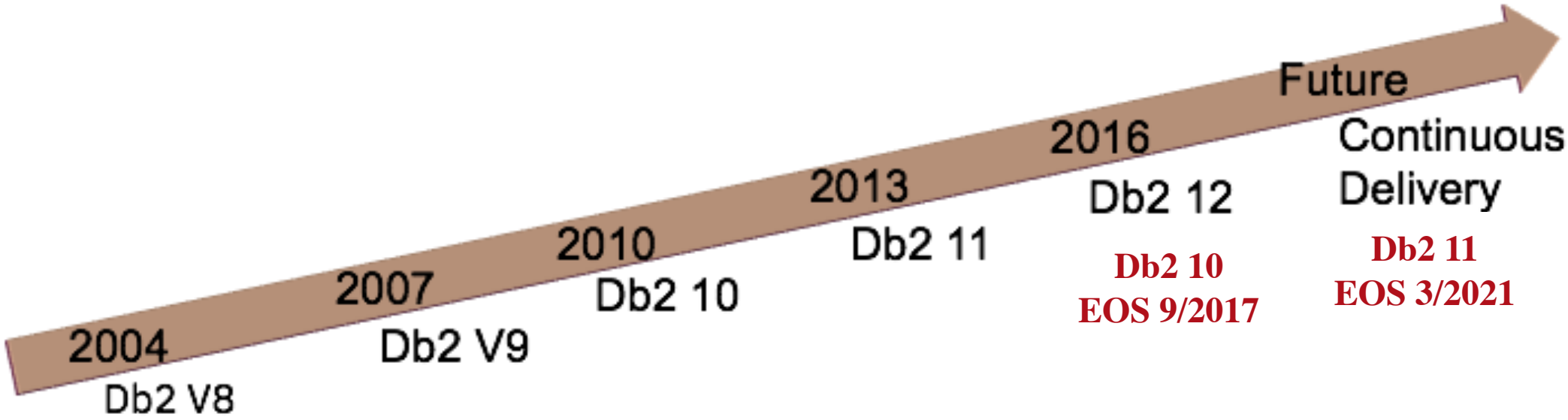
# Thank You

IBM®

# “Recipe for Success … Db2 12 migration“

IBM | IBM z Analytics

# Db2 for z/OS Support Timeline




Db2 12 GA was October, 2016

No single version charging – same price for *any* Db2 version

Quality metrics, continuous improvement: V12 better than V11 which is better than V10

# Db2 Recent Deliveries Not Tied to Function Levels

- IDAA
  - Special Register and Bind Option for User-Specified Accelerator
  - CAST of GRAPHIC/VARGRAPHIC support for IDAA
  - Multi-Row Insert for Accelerator Only Tables
  - IDAA Federation
  - IDAA V6/V7
  - IDAA HTAP Dynamic Query Support
- z14 Synergy
  - HyperLink Support for Random Database Reads – (V11, V12)
  - Transparent Dataset Encryption Support – (V11, V12)
- Performance Enhancements
  - zIIP enablement for RELOAD phase of LOAD and REORG Utilities
  - IMS Attach Connection Pooling Support
  - Partition by hidden ROWID columns
  - zIIP enablement for LOAD PARALLEL RELOAD phase
  - RUNSTATS Performance Improvement for Single Colgroups – (V11, V12)
  - CHECK LOB Utility Performance Improvement
  - Access Path Improvement To Encourage The Tables With Good Filtering To Be Joined Earlier
  - REORG Performance Improvement during UTS conversions

# Db2 Recent Deliveries Not Tied to Function Levels…

- Utility Enhancements
  - Support new INVALIDATECACHE option in M100
  - Table Schema Checking Enhancement for Repair Catalog
  - Additional LOAD IGNORE Options for Ignoring Rejected Records
  - Inline image copy for LOAD RESUME (v11, v12)
- Serviceability, Availability, Usability Enhancements
  - Enhanced Monitoring for in Index In-Memory Optimization
  - Improved Reporting of Real Storage Statistics
  - Set Partition Key Columns as Updateable for Tables Created Prior to V5
  - Enhanced Monitoring for Insert Algorithm 2 Capabilities
  - Insert Algorithm 2 Robustness improvements, and Zparm option
  - Enhanced Metadata Self-Description Capability (Storing Version 0 Info) – (V11, V12)
- GDPS Active /Active with zero data loss
  - CDDS Online Recovery and Cleanup
  - CDDS print utility and recovery enhancement

# Db2 Recent Deliveries Not Tied to Function Levels…

- Some Sponsor User favorites:
  - Enhanced DISPLAY CLAIMERS of blocking threads (PH13850)
  - Enables concurrent REORG / LOAD RESUME with SHRLEVEL CHANGE on object (PH11255)
  - Enables RLF in-memory auto Refresh START without Stop/Start (PH06082)
  - New messages DSNU2919 /DSNU2920 LOB comp. info (PI80351)
- Application Developer self-service and productivity
  - DB2aaS improvements: Provision Schema With z/OSMF Workflows - (V11, V12)
  - Native REST Client Certificate Support – (V11, V12)
  - Native REST Trusted Context Support - (V11, V12)
  - Native REST Persistent Connection Support – (V11, V12)
  - Native REST TSO BIND/FREE Service Support – (V11, V12)
  - COBOL PL/I Co-processor from HFS
  - Query performance improvements for join predicate pushdown
    - V12 APAR PI89564, 40 SAP Core Banking queries
  - Allows MODIFIES SQL DATA function to be invoked in a a full select – March 2018

| | V12 Base (As-is) | Epic 47830 (To-be) | Delta |
|---|---|---|---|
| Average elapsed time | 625.285 | 0.521 | -99.9% |
| Average cpu time | 341.389 | 0.446 | -99.9% |

After enhancement

# Hidden ROWID support to partition

- Epic 1506 - RUNSTAT Dynamic Profile Support - Koshy John
- Epic 2011 - More accelerator support built-in-functions (IDAA) -- Marie Sumabat
- Epic 75672 - Application granularity for NUMLKUS and NUMLKTS - - Sarbindar
- Epic 72 - CREATE OR REPLACE -- Long Tu
- Epic1925 - FL507 activation - John Lyle
- Epic 39301 - Online Migration - John Lyle
- Epic 111 - Db2AI V1.3
- SQL Optimization - Tom Beavin
  System Assessment - Bharat Verma
  DCC - Sharon Reoder
  UI - Martin Dinh