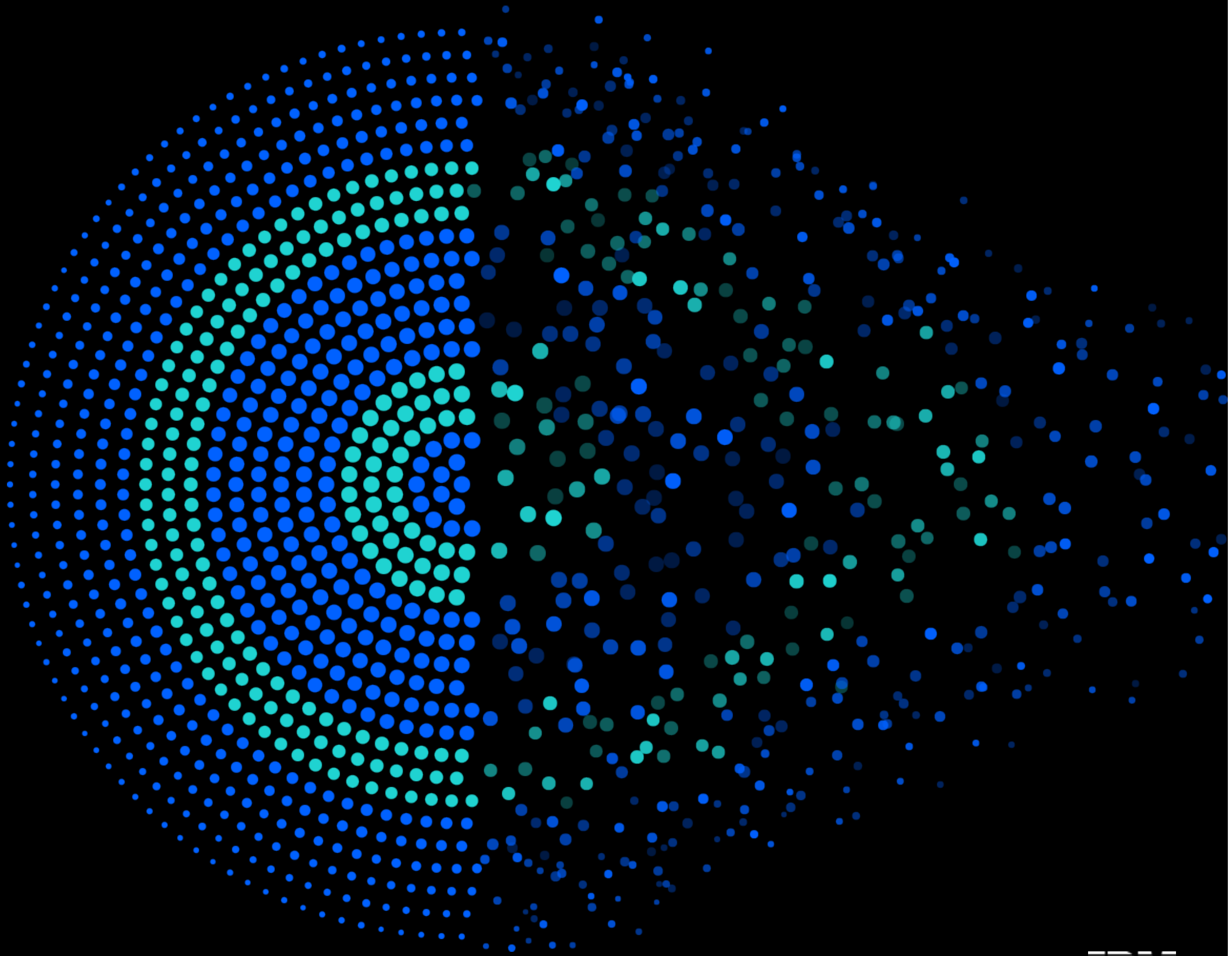
Db2 12 for z/OS Migration Planning and Experiences Part 1

John Campbell
Distinguished Engineer
Db2 for z/OS Development
Email: campbelj@uk.ibm.com



IBM Data and AI

Objectives

- Share lessons learned, surprises, pitfalls
- Provide hints and tips
- Address some myths
- Provide additional planning information
- Provide usage guidelines and positioning on new enhancements
- Help customers migrate as fast as possible, but safely

Agenda

- Part 1
  - Db2 11 for z/OS prerequisites for migration to Db2 12 for z/OS
  - Db2 12 for z/OS Migration – Quick Hits
  - Maintenance recommendations for early adopters of Db2 12 for z/OS
  - Db2 12 for z/OS Risk Mitigation
  - Understand Continuous Delivery starting with Db2 12 for z/OS
  - Understanding new function levels
  - JC Recipe for successful migration
- Summary

Agenda …

- Part 2
  - Db2 12 for z/OS Greatest Hits
  - Fast Un-clustered INSERT
  - RTS enhancements
  - Fast Index Traversal
  - Data dependent vs. numeric based pagination syntax
  - Increase in log record size after converting BSDS in Db2 11 and entry to Db2 12
  - Dynamic Plan Stability
  - More granular global commit LSN and global read LSN
  - SQLCODE -109 Issue
  - Enhanced SQL MERGE
  - UTS Relative Page Number (RPN)
  - INSERT Partition
  - Asynchronous CF Lock structure duplexing
  - Setting initial Statistics Profile
- Summary

Db2 11 for z/OS prerequisites for migration to Db2 12 for z/OS

- Ensure catalog consistency
  - REPAIR DBD TEST/DIAGNOSE + CHECK DATA/LOB/INDEX + DSNTESQ + ...
- Run pre-migration check queries and act on the reported findings
  - DSNTIJPM (Db2 12 for z/OS) or DSNTIJPC (APAR PI58254 for Db2 11 for z/OS)
- Apply fallback SPE PTF to all data sharing members
  - APAR PI33871 / II14794
- Make sure Db2 11 for z/OS PTF level is reasonably current especially if exploiting mixed release coexistence with data sharing and all maintenance is applied related to Db2 12 for z/OS migration
  - Use SMP/E Fix categories
    - IBM.Migrate-Fallback.DB2.V12 and
    - IBM.Coexistence.DB2.SYSPLEXDataSharing

Db2 11 for z/OS prerequisites for migration to Db2 12 for z/OS …

- Convert BSDS to 10 byte log RBA before leaving Db2 11 for z/OS NFM
  - For data sharing, convert single member at a time
  - Things to consider before converting the BSDS (DSNJCNVT)
    - Stop the Db2 for z/OS subsystem that owns the subject bootstrap data set pair
    - Any utility (e.g., RECOVER, REORG) running on alternate Db2 member in data sharing that reads from peer BSDS must be terminated
      - Special considerations for Data Replication
        - Stop any data replication process to ensure BSDS is successfully renamed and replaced
        - Best practice is to stop data replication process first, then stop the Db2 subsystem
    - RACF user ID running DSNJCNVT must have read/write access on the new BSDSs, and read access on the old BSDSs
  - After converting the BSDS, will see increased logging volume (3 <-> 40%)
  - There will be further increase in log record size after entry in Db2 12 for z/OS because of 7-byte RID values and independent of using UTS PBR RPN
  - Need to consider increasing size/number of active log pairs to maintain recommended 6 hours of recovery log data across active log configuration
  - Need to reevaluate the size of the archive log DASD pool to ensure 48 hours’ worth of recovery log data can be kept

Db2 11 for z/OS prerequisites for migration to Db2 12 for z/OS ...

- Avoid autobind on pre-DB2 10 for z/OS plans and packages under Db2 12 for z/OS
  - Avoid painful lesson related to plans
  - Thread break-in capability delivered in Db2 11 for z/OS may help for packages when performing rolling migration
  - Recommend explicit rebind under Db2 11 for z/OS NFM before leaving for Db2 12 for z/OS
    - Resolve any potential authorization issues that may exist
    - Use plan management for packages to keep a backup copy
- Remember to set and keep ZPARM ABIND=COEXIST for mixed release coexistence (Db2 11, Db2 12)
- APAR PI87675 - Re-migration autobinds are disabled
  - Applies to both for plans and packages
  - Applies to both ABIND=COEXIST and ABIND=YES which now behave the same behavior
    - Will no longer perform re-migration autobinds
  - Delivers complete solution for Db2 11 for z/OS and Db2 12 for z/OS

Db2 11 for z/OS prerequisites for migration to Db2 12 for z/OS ...

- FREE inactive package copies (access plan management) before first REBIND under Db2 12 for z/OS
- Upgrade EXPLAIN tables to Db2 12 for z/OS format (should be at least Db2 11 for z/OS version)
  - Can be done in Db2 11 for z/OS NFM with fallback SPE applied
  - Use of sample batch job DSNTIJXA with REXX DSNTXTA can help
- Apply PTFs for APARs PI69589 (Db2 11 for z/OS) & PI69584 (Db2 12 for z/OS)
  - Reduce catalog contention during “online” migration to Db2 12 for z/OS
- Plan for activation of Db2 12 for z/OS EARLY code
  - Activation via IPL or Command –REFRESH DB2,EARLY

Db2 12 for z/OS Migration – Quick Hits

- Minimum OS level lifted from z/OS V1R13 to V2R1
- Minimum hardware level lifted from z10 to z196/z114
- Replication
  - Db2 12 for z/OS (with APAR PI70998) and Db2 11 for z/OS require the Q Capture and Capture programs from IBM InfoSphere Data Replication for Db2 for z/OS Version 10.2.1
  - Q Apply and Apply programs at architecture level 1001 will work with both Db2 11 for z/OS and Db2 12 for z/OS
    - APAR PI70998 for Db2 for z/OS
    - APAR PI66768 for IIDR 10.2.1 Q and SQL
    - APAR PI61562 for CDC

Db2 12 for z/OS Migration – Quick Hits ...

- Db2 Connect – Situation until recently prior to applying PTF for APAR PH08482
  - Any level of Db2 Connect drivers should work with Db2 12 for z/OS, both before and after new function is activated with no behavior change
  - Data server clients and drivers must be at the following levels to exploit Db2 for z/OS function-level application compatibility (APPLCOMPAT) of V12R1M**501** or greater:
    - IBM® Data Server Driver for JDBC and SQLJ: Versions 3.72 and 4.22, or later
    - Other IBM data server clients and drivers: Db2 for Linux, UNIX, and Windows Version 11.1 Modification 2 Fix Pack 1, or later
  - New ClientApplCompat (ODBC) and clientApplcompat (JDBC) property setting allows you to control the capability of the client when updated drivers ship changes to enable new server capability
    - You might want specific control of driver capability when:
      - Db2 client driver introduces new behavior currently not controlled by Db2 application compatibility
      - Change needs to be controlled at the application level to ensure compatibility with new behavior
  - ClientApplCompat/clientApplcompat setting of V12R1M500 is required to access Db2 12 for z/OS Server capability shipped after GA at function levels beyond Db2 12 for z/OS FL=V12R1M**500**
  - Db2 Connect Server gateway does NOT support the ClientApplCompat/clientApplcompat

Db2 12 for z/OS Migration – Quick Hits …

- Db2 Connect – new behavior after applying PTF for APAR PH08482
  - Makes setting of ClientApplCompat/clientApplcompat property optional
    - Customer no longer forced to have the setting
  - No changes are required in Db2 Connect level and configuration
  - As before all customers must upgrade to at least Db2 Connect V11.1 FP1 or higher in order to run DRDA applications where APPCOMPAT > FL500
  - Db2 Connect Server gateways will need to be upgraded to at least Db2 Connect V11.1 FP1 to access packages using an APPLCOMPAT > FL500
  - When ClientApplCompat/clientApplcompat is set, Db2 will perform validation checking where there are changes in DRDA message flows i.e., check the underlying infrastructure and avoid application incompatibilities

Db2 12 for z/OS Migration – Quick Hits …

- Db2 Connect – new behavior after applying PTF for APAR PH08482 …
  - Going out into the future when not setting ClientApplCompat/clientApplcompat there are consequences in terms of risk when DRDA flows changes as existing applications
    - So far in Db2 12 there are no changes in DRDA flows, no changes are in plan, but at some point it will likely happen
    - Applications may break in the future when DRDA transactions run with package where APPLCOMPAT > FL500
    - If an application breaks then Db2 will not provide server support to allow these broken applications to run i.e., no more new DDF_COMPATABILITY zparm settings
  - So what are the your options
    1. Rebind driver packages in the NULLID collection and back level the APPLCOMPAT setting
       This is a "one size fits all" solution to fallback to an earlier APPLCOMPAT
    2. “Penalty box” the problem applications
       a. Switch the problem applications out to use the driver packages in a different collection which has a back levelled APPLCOMPAT setting, or
       b. Switch all the good applications out into a new collection using driver packages with the new APPLCOMPAT setting and leave the problem applications still using the driver packages in the NULLID or different collection but with the driver packages running a back levelled APPLCOMPAT setting

Db2 12 for z/OS Migration – Quick Hits …

- Db2 Connect – new behavior after applying PTF for APAR PH08482 …
  - General best practice recommendation
    - When migrating to Db2 12 - all DRDA applications should continue to use the driver packages in the NULLID collection
    - These packages can have an APPLCOMPAT setting of V10R1, V11R1, V12R1M100 or V12R1M500 depending on where you are in the migration process
    - The APPLCOMPAT setting for the driver packages in the NULLID collection should not advance beyond V12R1M500
    - When specific applications and their application servers want to use new function requiring APPLCOMPAT setting > FL500, these application servers should switch away from using the driver packages in the NULLID collection to a new collection (e.g., V12R1M503) where the driver packages are bound with a higher APPLCOMPAT setting (e.g., driver packages bound with APPLCOMPAT V12R1M503 in collection V12R1M503)

Db2 12 for z/OS Migration – Quick Hits ...

- Changes to Utilities Suite installation
  - Requires registration in SYS1.PARMLIB(IFAPRDxx)
    - CBPDO is being sunset, and SystemPac is the strategic direction
    - Any separately orderable product using only F or J FMIDs has to be changed to use an E or H base FMID
  - Documented in Db2 Utilities Suite program directory

```
PRODUCT OWNER('IBM CORP') NAME('DB2 UTIL SUITE') ID('577-AF4')
VERSION(12) RELEASE(1) MOD() FEATURENAME('V12R1') STATE(ENABLED)
```

  - Failure to register Utilities Suite results in utility errors

```
DSNU3333I 012 14:35:50.01 DSNUGPRS - THE DB2 UTILITIES SUITE FOR Z/OS HAS NOT BEEN ENABLED
DSNU3330I 012 14:35:50.09 DSNUGPTS - THE xxxxxxxx UTILITY HAS RESTRICTED FUNCTION
IT IS PART OF THE DB2 UTILITIES SUITE FOR Z/OS WHICH HAS NOT BEEN ENABLED
```

Db2 12 for z/OS Migration – Quick Hits …

- REORG MAPPING TABLE format must allow for 7-byte RID values
  - SQL DDL changes
    - Column 'SOURCE_RID' CHAR(5) -> CHAR(7)
    - Column 'TARGET_XRID' CHAR(9) -> CHAR(11)
  - No toleration logic in Db2 11 for z/OS NFM
    - Db2 11 for z/OS NFM REORG running with the Db2 12 for z/OS mapping table format will fail
  - Db2 12 for z/OS REORG in FL=V12R1M100 tolerates Db2 11 for z/OS format mapping table format
  - Db2 12 for z/OS REORG in FL=V12R1M5nn only supports the Db2 12 for z/OS mapping table format
- Incompatible changes
  - BIF_COMPATIBILITY system parameter is still supported
  - IFCID 366 has been removed in Db2 12 as unattractive to use because of the volume of records
  - Typical complaint: collect IFCID 376, but cannot match with dump contents of dynamic statement cache
  - Recommendation: turn on IFCID 376 for 10 minutes and then dump contents of dynamic statement cache

Db2 12 for z/OS Migration – Quick Hits …

- zparm REORG_IGNORE_FREESPACE removed
- RACF changes related to IDAA
  - RACF ACCESS(CONTROL) on MVS.VARY.TCPIP.DROP(OPERCMDS)
- HVSHARE should be 510 TB (default)
  - Db2 12 for z/OS requires 1 TB of 64-bit shared (private) storage in z/OS (same as Db2 11 for z/OS)
  - Virtual, not real
  - Monitor use with IFCIDs 217 and 225
- Plan for real memory increase
  - Trend continues … using larger size REAL memory to deliver performance improvements
  - Expect ~ 15% increase
  - Expect up to 30% increase if taking advantage of new in-memory function
    - Largest percentage from use of Fast Traverse Block (FTB) area – default with AUTO is 20% increase on allocated VPSIZE
- Consider current zIIP utilization, upgrade zIIP capacity if needed - trend to extend zIIP offload continues
  - Fast Traverse Block monitoring
  - Insert Algorithm 2 Pipe management
  - REORG and LOAD RELOAD phase
  - SQL query parallelism (child task eligibility 80% -> 100%)

Db2 12 for z/OS Migration – Quick Hits …

- Increased space requirement for RID Pool as a result of RID value size increase 5 -> 8-byte
  - Internally Db2 for z/OS uses a normalized 8-byte RID value to allow for future expansion
  - More RID blocks will be used for the same query because each RIDLIST holds fewer RIDs
  - RID Pool memory usage will be roughly 60% higher (for smaller lists it will be up to 2x higher)
  - May have to increase MAXRBLK (RID Pool size) by up to 60%
  - Data Manager logical limit (RIDMAP/RIDLIST) reduced from 26M (26,602,376) RIDs to 16M (16,625,976) RIDs
  - More RID Pool overflow to workfile is to be expected
- Increased space requirement for RID Pool as a result of potentially more use of list prefetch
  - Enhancement to the Optimizer cost model to more closely reflect the true cost (and benefit) of list prefetch
  - Expected to see an increase in list prefetch (and potentially hybrid join)
  - But not necessarily changes in the access plan where Db2 would previously have chosen a sort avoidance plan
  - Db2 for z/OS trying to be careful not to select list prefetch (with sort) as an access path when there was an alternative access path that could use an index to avoid a sort i.e., for pagination type SQL

Db2 12 for z/OS Migration – Quick Hits …

- Increased EDM Pool memory usage for DBDs
  - Starting with Db2 12 Function Level V12R1M100, there will be increased demand for EDM pool memory for DBDs
    - How much depends on the size and number of OBDRECs (tables) and the number of OBDPSETs (table spaces)
  - Extra memory comes from “puffing” of table obdrec and tablespace obdpset to new DBD format in Db2 12
    - Pre-Db2 12 “puffed” size under Db2 12 > Db2 12 size (after ALTER/REPAIR) > Db2 11 size
  - In V12R1M100 mode, the “puffed” OBD (OBDREC and OBDPSET) will not be written out to DBD01 which means that “puffing” to Db2 12 format will occur every time the DBD is read from DBD01
  - For V12R1M5xx, before APAR PH05624, the “puffing” code is invoked even when the OBD is already in Db2 12 format
    - No further additional memory requirement since OBD (OBDREC and OBDPSET) is already in Db2 12 format
  - Some customers experienced significant performance impact after migration to Db2 12
  - Solutions
    - Double size of EDM DBDC pool
    - Apply PTF for APAR PH05624
    - Any DDL or REPAIR DBD REBUILD will result in update of OBD to Db2 12 format and new format will be persisted in DBD01

Db2 12 for z/OS Migration – Quick Hits ...

- Goal is to reduce CPU resource consumption by avoiding scheduling a (dynamic) prefetch engine when sequential detection triggers dynamic prefetch and where all pages are in bufferpool
  - Reduces the risk of running out of prefetch engines
  - Reduces DBM1 SRB CPU time
  - Reduces LC24 latch contention
- Solution
  - Dynamic prefetch now uses buffer pool statistics - # dynamic prefetch I/Os and # sequential sync I/Os
  - When about to schedule dynamic prefetch requests, Db2
    - Maintains a counter about # of times dynamic prefetch I/O occurred
    - Checks whether the dynamic prefetch I/O count has not changed
    - If this happens 3 times, dynamic prefetch scheduling is disabled and starts recording # sequential sync I/Os for the bufferpool
    - Once dynamic prefetch scheduling is disabled, but the # of sequential sync I/Os has changed, then scheduling of dynamic prefetch is re-enabled

Db2 12 for z/OS Migration – Quick Hits …

- Deprecation of Basic Row Format (BRF)
  - zparm SPRMRRF is now hidden in Db2 12 for z/OS
  - ROWFORMAT keyword option for REORG/LOAD to convert a pageset between BRF/RRF has been removed from the documentation
    - Still supported from a utility syntax perspective
  - New objects created will always be RRF
  - Existing pagesets in BRF will continue to be supported for the time being

Db2 12 for z/OS Migration – Quick Hits …

- Invalidation of prepared SQL statements in dynamic statement cache
  - Prior to Db2 12 for z/OS, RUNSTATS would always invalidate prepared statements dependent on the object that the utility was run against
  - In Db2 12 for z/OS, RUNSTATS by default will not invalidate the prepared statements (incompatible change)
    - Use new INVALIDATECACHE YES option to force the invalidation of prepared statements
  - Invalidation of prepared statements will still occur when
    - RUNSTATS … INVALIDATECACHE YES
    - RUNSTATS after SQL DDL (CREATE/DROP INDEX) and statistics profile updated
    - RUNSTATS … UPDATE(NONE) REPORT(NO)
    - For other utilities, if the object was in an invalid state before the utility began e.g., rebuild pending or reorg pending

Db2 12 for z/OS Migration – Quick Hits …

- PGSTEAL(NONE) buffer pool and FRAMESIZE(2G)
  - Incompatible change when migrating from Db2 11 to Db2 12
  - For Db2 11, Db2 will use 2G size frames for PGSTEAL(NONE) buffer pools
  - For Db2 12, contiguous buffer pools (PGSTEAL(NONE)) will NOT use 2G size frames
    - Request to use 2G size frames is not honoured
    - Buffer pool will still be allocated, but in 4K size frames
    - DSNB548I message will be issued when
      - Allocating buffer pool which has PGSTEAL(NONE) and FRAMESIZE(2G) specified
      - ALTER BUFFERPOOL command changes either attribute with the result being PGSTEAL(NONE) FRAMESIZE(2G)
  - Why be concerned?
    - If the size contiguous buffer pools are very large, this can lead to shortage of 4K frames on the LPAR, with consequences
      - Penalty of page movement or paging I/O overhead with corresponding CPU burn in RASP address space
      - Worst case the LPAR will crash out!
  - Recommendation
    - If using 2G size frames under Db2 11, then switch to using 1M size frames before leaving Db2 11

Maintenance recommendations for Db2 12 for z/OS and Continuous Delivery

- Apply preventative maintenance every 3 months based on rolling calendar
  - Use RSU instead of PUT to be less aggressive on applying non-HIPER maintenance
  - Sample strategy based on two 'major' and two 'minor' releases
    - Refresh of the base every 6 months ('major')
    - Each base upgrade should be based on latest quarterly RSU
      - Ensure that RSU-only service is installed by adding the SOURCEID (RSU*) option in the supplied APPLY and ACCEPT jobs
    - In addition, apply two mini packages covering HIPERs and PEs in between ('minor’)
- Continuous program to review Enhanced HOLDDATA on a weekly basis
  - When rolling out maintenance package to production and after post production cutover
  - For vicious HIPER problem where no operational bypass/workaround, expedite fix into production after 2 weeks in test
  - Others can be deferred until the rollout of the next maintenance drop
    - By applying operational bypass/workaround in production
    - Only “low grade” problem in production
    - Not applicable

Different APAR numbers across Db2 11 and Db2 12 for z/OS

- Short term solution was to make the connection visible, bi-directional, either in the SYSROUTED FROM or SYSROUTED TO section of the Outline/Overview of the respective APAR

```
APAR    PI76204    OUTLINE

ENTER    DISPLAY-ITEM    PAGES    ENTER    DISPLAY-ITEM    PAGES
 S     SUMMARY              3      X     SUBMITTER TEXT      5
 E     RESPONDER TEXT       0      P     PIN ITEM            0
 I     INTRSTD PARTIES      1      K     TRACKING            2
 C     CONSTANT             1      Z     SCRATCH-PAD         1
 F     FEEDBACK             0            STRUCTURE           2
 DUP   FIRST DUPLICATE APAR   (OR DUP/NNN)   ORG  ORIGINAL APAR
 PTF   FIRST PTF REQUESTED.   (OR PTF/NNN)
                                         LAST PAGE


PTF RQSTD:
APAR FIXED BY:
DUPLICATES:
SYSROUTED FROM:
SYSROUTED TO: PI76206
APPLICABLE PE-PTF'S:
```

Different APAR numbers across Db2 11 and Db2 12 for z/OS ...

- Single APAR solution for a single problem for new APARs has now been delivered (June 2017)
  - Default now is a single APAR for the same problem across Db2 11 for z/OS and Db2 12 for z/OS
  - Assumes the same “basing” – so there will still be some exceptions
  - Will not be applied respectively

Db2 12 for z/OS Risk Mitigation

- Regression testing is critical piece to keep “fires away from production”
  - Test all critical and custom processes, and scale them up
  - Run performance measurements and establish Db2 11 for z/OS baseline for comparison
  - Go / No Go decision for Db2 12 for z/OS migration of production system should be based on positive results from proper testing
    - Be prepared to postpone migration as opposed to forcing in Db2 12 for z/OS
  - Practice migration fallback from Db2 12 for z/OS to Db2 11 for z/OS and back to Db2 12 for z/OS
  - Design fallback strategy and practice it in pre-production environments
- Minimize change and use of new function in and around when Db2 12 for z/OS is first introduced into production
- For production systems, stay on FL=V12R1M100 for at least a month to prove running smoothly
  - Leaves back door open to go back to Db2 11 for z/OS NFM in an emergency
- Make sure very current on preventative service for these specific Db2 12 for z/OS functions
  - Fast Index Traversal for in-memory index performance optimisation
  - INSERT ALGORITHM 2 for fast un-clustered insert
  - UTS PBR RPN
  - Improved EDM Pool management
  - Increased active log dataset size

New Db2 for z/OS Strategy for delivering new function

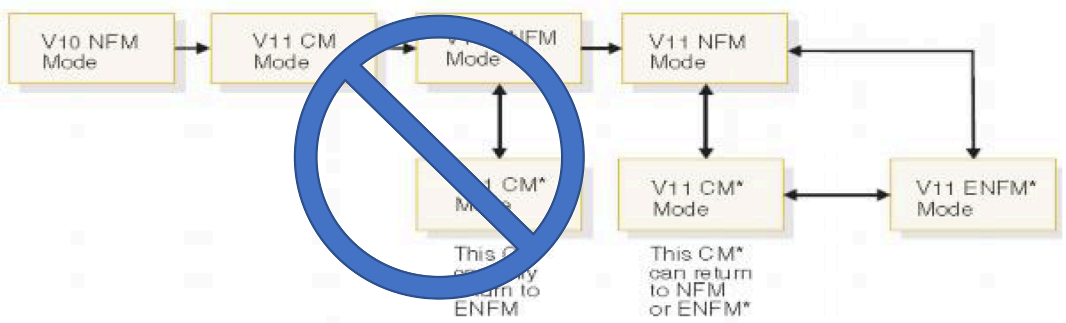
- We are dedicating ourselves to going forward on a continuous delivery model
  - Radical internal changes are required within Db2 for z/OS Development to do this
- Db2 12 for z/OS is the starting point after GA
  - There will be significantly higher volume of continuously delivered items
- Customers will see a single maintenance stream for Db2 12 for z/OS, with the new function delivered into that
  - The function will be designed to be easily consumable
- Point releases or versions will be a very rare exception
  - There are reasons why we might want to have a point release or new version
  - e.g., 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 for you 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, and not based on a promised date for delivery

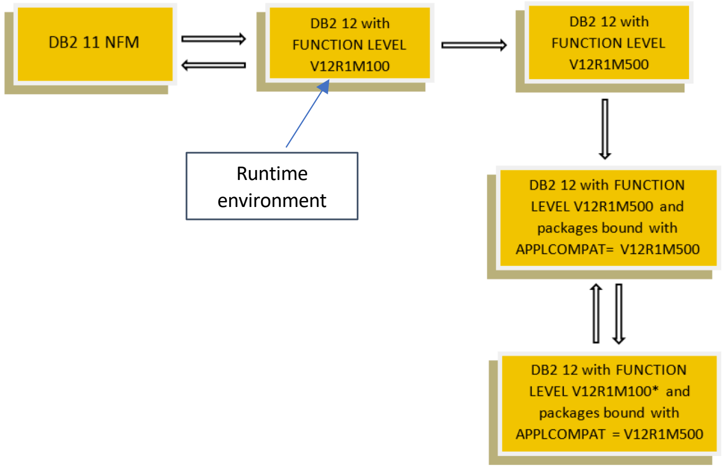
Understand Continuous Delivery starting with Db2 12 for z/OS

- 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 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 - accumulative (skip level possible)
    - Db2 Catalog changes that are needed for some FLs
  - Function level (FL) – needs to be activated - accumulative (skip level possible)
    - Introduces new Db2 for z/OS features and functionality
    - No impact or change in existing application behaviour
  - APPLCOMPAT level (AC) – set by application - provides an "island of stability" for a given application
    - Determines SQL function level of applications – can increase FL of the application (and fallback)
    - AC must be advanced to exploit new SQL function
    - AC level in BIND/REBIND of package must be <= FL and rules over FL
    - Freezes new SQL syntax even if FL is later moved back to earlier level
- Minimum starting point for Continuous Delivery is Db2 12 for z/OS FL=V12R1M500

Understanding new function levels

- CM / ENFM / NFM no longer used
- Function Level V12R1M100
  - Similar to CM / BNFA
  - Db2 12 for z/OS engine and catalog / directory
  - DSNTIJTC (CATMAINT) to get there
  - Fallback to Db2 11 for z/OS NFM possible
- Function Level V12R1M500
  - Similar to NFM /ANFA
  - New functionality available
  - Command –ACTIVATE FUNCTION LEVEL(V12R1M5nn) to get there
  - Fallback to Db2 11 for z/OS NFM no longer possible (PIT recovery be required)





DB2 11 NFM

DB2 12 with FUNCTION LEVEL V12R1M100

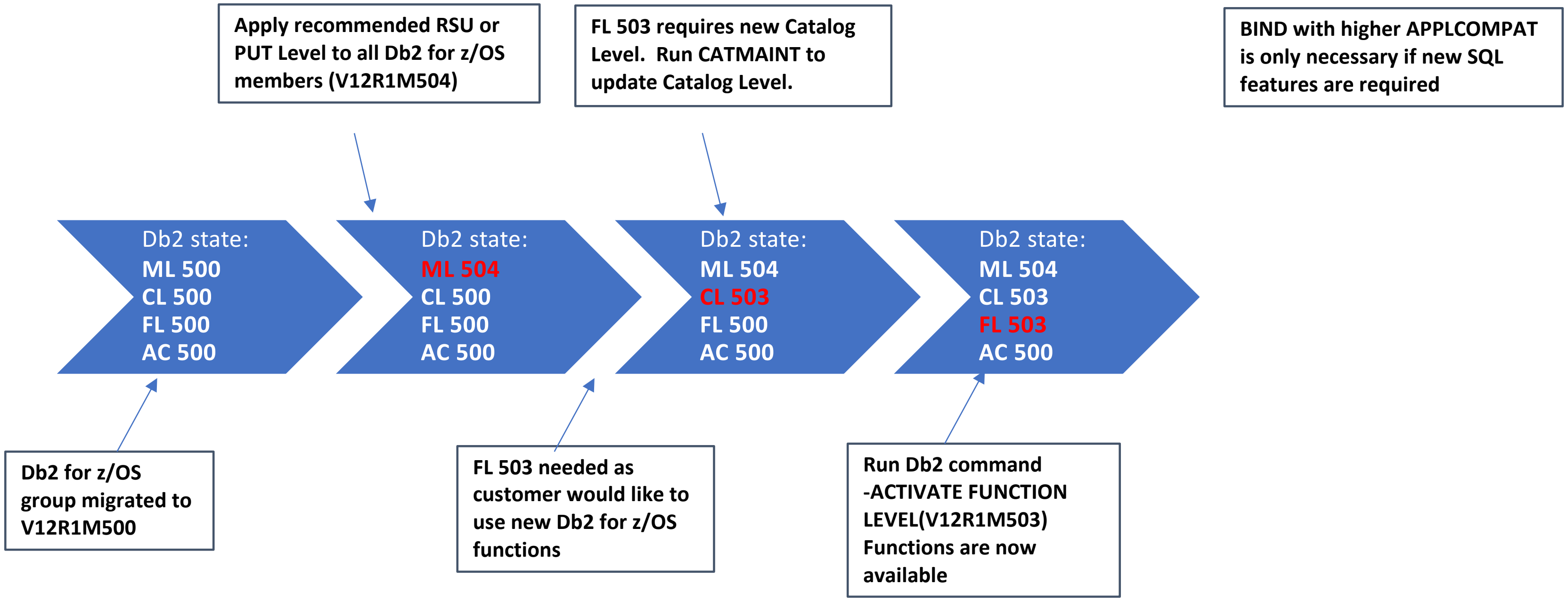
DB2 12 with FUNCTION LEVEL V12R1M500

Runtime environment

DB2 12 with FUNCTION LEVEL V12R1M500 and packages bound with APPLCOMPAT= V12R1M500

DB2 12 with FUNCTION LEVEL V12R1M100* and packages bound with APPLCOMPAT = V12R1M500

Example of how to get to a new function level



Db2 for z/OS group migrated to V12R1M500

Db2 state:
ML 500
CL 500
FL 500
AC 500

Apply recommended RSU or PUT Level to all Db2 for z/OS members (V12R1M504)

Db2 state:
ML 504
CL 500
FL 500
AC 500

FL 503 needed as customer would like to use new Db2 for z/OS functions

FL 503 requires new Catalog Level. Run CATMAINT to update Catalog Level.

Db2 state:
ML 504
CL 503
FL 500
AC 500

Run Db2 command -ACTIVATE FUNCTION LEVEL(V12R1M503) Functions are now available

Db2 state:
ML 504
CL 503
FL 503
AC 500

BIND with higher APPLCOMPAT is only necessary if new SQL features are required

Change in strategy for APPLCOMPAT

- No need to force the rebind all packages with a new, higher APPLCOMPAT level
- APPLCOMPAT will now have many more versions to support many Function Levels
- Must still rebind a package with a higher APPLCOMPAT level in order to exploit new SQL DML, SQL DDL, SQL DCL, and XML function
  - Applications can only use new SQL if the packages are bound with the necessary and required Application Compatibility (APPLCOMPAT)
    - Packages can only be bound with an APPLCOMPAT less or equal to the current FL

- Still recommended best practice to regularly rebind all packages with APREUSE(WARN) and plans
  - Benefit from latest run time performance improvements
  - Gain exposure to new access path selection improvements
  - Benefit from defect fixes
  - Reduce exposure to latent issues seeded previously

Develop a strategy for APPLCOMPAT

- Use IFCID 376 tracing to identify specific packages which are potentially exposed to incompatible changes, and investigate and upgrade application programs where necessary
- Do not increase ZPARM APPLCOMPAT until such time as all incompatibilities have been identified and addressed
  - Avoid breaking existing applications which have latent compatibility issues
- Set APPLCOMPAT explicitly as a BIND option and manage it throughout the application deployment
- Any application wanting to use new SQL DML or XML functions will need to be explicitly rebound with a specific APPLCOMPAT, after cleaning up any residual incompatibilities
  - Include system packages such as SPUFI, DSNTEP2, client packages
- Going forward, need direction from application providers about APPLCOMPAT requirements

Is APPLCOMPAT a ‘sticky’ option on BIND/REBIND?

- BIND REPLACE does not reuse any bind option from the existing package if the option is not explicitly specified
- SQL statements can be totally different so BIND REPLACE is considered a new bind
- REBIND and BIND COPY are the only subcommands that reuse the existing/source package's options
  - But only if at least one BIND/REBIND of the package since introduction of Db2 11
  - APPLCOMPAT value for respective package is set in SYSIBM.SYSPACKAGE
- This is true in all Db2 for z/OS releases and not just Db2 12 for z/OS

Setting CURRENT APPLICATION COMPATIBILITY special register

- Db2 11 for z/OS
  - Value can be >= APPLCOMPAT level of the executing package but not > current Db2 version
- Db2 12 for z/OS
  - Value has to be <= APPLCOMPAT level of the executing package, independent of the current Db2 Function Level
  - Why?
    - To be able to control application use of new functionality
    - Avoid breaking applications when factivating a lower function level i.e., fallback

Function Level Adoption – Best Practices 1/2

- Prioritise maintaining production stability over new function adoption
- If applying preventative maintenance only once or twice per year, then apply more frequently
  - Every 3 months using a rolling calendar
- PTFs (RSUs…) are applied that may increase the Maintenance Level (ML) of a Db2 for z/OS subsystem
- After system is stable on maintenance – may be after a complete maintenance cycle
  - Execute (If Any) Catmaint
    - **After execution of Catmaint, the subsystem can only be started with a ML that supports the catal**og
  - Activate Function Level (FL)
    - **After activating a new FL, the subsystem can only be started with a ML that supports the FL**
    - New function not related to SQL DML, DDL and DCL syntax is available
    - REBIND of packages with any APPLCOMPAT would pick up optimizer enhancements
    - Non-stabilized dynamic SQL would pick up optimizer / other non-APPLCOMPAT related enhancements

Function Level Adoption – Best Practices 2/2

- After Function Level is considered stable - allow new application feature rollout
  - REBIND DBA packages to allow new DDL to be utilized
  - REBIND application static packages with higher APPLCOMPAT to exploit SQL DDL/DML new functions/behaviors
  - REBIND dynamic packages with higher APPLCOMPAT to allow new SQL functions to be used
  - REBIND distributed packages (***in separate collection) with higher APPLCOMPAT to allow new SQL functions to be used
    - Switch applications to use new distributed package collection
  - Leverage PLANMGMT extended
    - Use REBIND SWITCH (PREVIOUS) to restore static SQL packages to prior runtime structures
    - Use REBIND SWITCH (PREVIOUS) for dynamic SQL packages would restore prior APPLCOMPAT
    - ***switching to prior collid for distributed dynamic would restore APPLCOMPAT

JC Recipe for successful migration

- 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

JC Recipe for successful migration ...

- Be aggressive on taking preventative maintenance
  - Take as much preventative maintenance as possible on Db2 12 and keep level of Db2 11 reasonably close
    - Important for mixed release coexistence (data sharing) and release fallback toleration
    - Recommendation every 3 months per year with Continuous Delivery and Db2 12
  - 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
  - Especially important for new functionality

- Minimise 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 and RID pools under Db2 11 before leaving Db2 11
  - Rebind deprecated plans and packages with APREUSE(ERROR|WARN) before leaving Db2 11
  - Turn off new features which are on by default

JC Recipe for successful migration …

- Build a performance 'profile' based on KPIs under Db2 11 as base line for comparison against Db2 12
  - Subsystem level using Statistics Trace
  - Application process level using Accounting Trace
    - High volume transactions
    - Complex transactions at reasonable volume
    - Critical batch processes
- 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 target high volume transactions and complex transactions at reasonable volume
  - Benefits
    - Re-enable xPROCs for migrated packages
    - Avoid overhead of any run time "puffing" code under Db2 12 for migrated packages (PTs) and plans (CTs)
    - Accrue Db2 12 run time performance improvements
    - Reduce exposure to latent issues seeded previously in earlier releases
  - Gain exposure to new access path selection improvements later

Summary

- Share lessons learned, surprises, pitfalls
- Provide hints and tips
- Address some myths
- Provide additional planning information
- Provide usage guidelines and positioning on new enhancements
- Help customers migrate as fast as possible, but safely

Top DB2z Social Media Channels #DB2z

- Join the World of DB2 www.worldofdb2.com
- Follow @IBMDB2 on Twitter https://twitter.com/IBMDB2
- Join DB2z LinkedIn Group
- https://www.youtube.com/user/IBMDB2forzOS
- DB2z on Facebook
  - https://www.facebook.com/IBMDB2forzOS/

Now … Live Q&A with John Campbell