DB2 for z/OS – Ultimate Database for Cloud, Analytics and Mobile

Industry-leading performance, security, scale and reliability

DB2 11 for z/OS Migration Planning and Early Experiences – Part 1

John Campbell

DB2 for z/OS Development

Twitter: @GURUDB2

Disclaimer/Trademarks

© Copyright IBM Corporation 2015. All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

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, 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.

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

- Introduction
- ESP Highlights
- Migration Considerations
- Pre-Migration Catalog Migration Testing
- Availability
- Utilities
- Performance and Scalability
- Other Enhancements
- Summary

Introduction

DB2 11 Major Themes

- **Out-of-the-box CPU Savings**
  - Improving efficiency, reducing costs, no application changes
  - Up to 10% for complex OLTP
  - Up to 10% for update intensive batch
  - Up to 40% for queries
  - Additional performance improvements through use of new DB2 11 features
- **Enhanced Resiliency and Continuous Availability**
  - Improved autonomics which reduces costs and improves availability
  - Making more online changes without affecting applications
  - Online REORG improvements, less disruption
  - DROP COLUMN, online change of partition limit keys
  - Extended log record addressing capacity - 1 yottabyte (or 1B petabytes)
  - BIND/REBIND, DDL, Online REORG to break into persistent threads
- **Enhanced business analytics**
  - Expanded SQL, XML, and analytics capabilities
  - Temporal and SQLPL enhancements
  - Hadoop integration, NoSQL and JSON support
  - Transparent archiving
- **Simpler, faster DB2 version upgrades**
  - Improved product quality/reliability – through iterative approach on 3 monthly cycle (1:N rallies) to FVT, SVT, Performance testing, and stabilization phase ahead of start of ESP
  - Application changes divorced from DB2 system upgrade (APPLCOMPAT)
  - Access path stability improvements

Announce
Oct. 1, 2013

GA
Oct. 25
2013

ESP Highlights

DB2 11 ESP Client Feedback

- **Very much improved quality and reliability at this early stage in the release cycle**
- **Good performance and CPU savings**
  - DRDA workload up to 20% CPU reduction
  - CICS workload up to 18% CPU reduction
  - Batch workload up to 20% CPU reduction
- **Greatest hits**
  - BIND, REBIND, DDL, Online REORG break in
  - Transparent archiving
  - IFCID 306 filtering by object (Qreplication)
  - Online schema change
  - Utility improvements particularly Online REORG
  - Extended LRBA/LRSN
  - Optimizer and migration improvements
  - GROUP BY Grouping Sets

Migration Considerations

Prerequisites – Hardware & Operating System

- **Processor requirements:**
  - zEC12, z196, z10 processors supporting z/Architecture
  - Will probably require increased real storage for a workload compared to DB2 10 for z/OS (up to 15%)
- **Software Requirements:**
  - z/OS V1.13 Base Services (5694-A01) at minimum
  - DFSMS V1 R13 – DB2 Catalog is SMS managed
  - Language Environment Base Services
  - z/OS Version 1 Release 13 Security Server (RACF)
  - IRLM Version 2 Release 3 (Shipped with DB2 11 for z/OS)
  - z/OS Unicode Services and appropriate conversion definitions are required
  - IBM InfoSphere Data Replication (IIDR) 10.2.1
  - For DB2 Connect – please the next slides

- **V9 Early Code not compatible with DB2 10 and 11**
  - See APAR PI23400 will generate warning message and terminate

Prerequisites – DB2 Connect

- **DB2 11 for z/OS in all modes should operate with existing versions of DB2 Connect in place, even back to DB2 Connect V8**
  - DB2 for z/OS Development will investigate any connectivity related issues with existing applications using older versions of DB2 Connect and try to provide a fix
  - If any issues cannot be resolved within the DB2 for z/OS server, DB2 Connect will have to be upgraded to an in-service level to obtain a fix
- **For continuous availability during the migration process the minimum recommended level before leaving DB2 10 is V9.7 FP6 or V10.1 FP2**
  - This is the level that provides continuous availability for a given application server as a customer goes from V10 NFM base -> V11 CM -> V11 NFM
- **Most DB2 for z/OS engine features in NFM are supported with any version of DB2 Connect**
- **The minimum level for full DB2 11 for z/OS exploitation is currently V10.5 FP4**
  - Required for specific new function: array support for stored procedures, WLB support with global variables, autocommit performance improvements, improved client info
  - This recommended level could and probably will change and go up over time as we gain more customer experiences, roll through best practices, and provide defect fixes into newer driver levels

Prerequisites – DB2 Connect ...

- **DB2 for z/OS Development are being proactive in recommending customers to move from the client or runtime client packages towards using the data server (ds) driver instead**
- **For "evergreen" and/or new function the general upgrade path is the following:**
  - DB2 for z/OS Server
  - DB2 Connect Server (if present – we are encouraging direct connect)
  - Drivers installed on application servers (push from client, runtime client -> ds driver)
  - End user workstations (also push from client, runtime client -> ds driver)
- **We do have customers that will push out the drivers first - those are generally driven by the need for specific application enhancements e.g.,**
  - The most common example is in the .NET arena - wanting the latest tooling and driver support in the MS arena

Pre-migration planning

- **Run DSNTIJPM (DSNTIJPB) pre-migration job**
  - Check for situations needing attention before migration
    - Take the actions recommended by the report headers

- **Run DSNTIJPM or DSNTIJPB, to identify them**
  - DSNTIJPM ships with DB2 11 and should be run on DB2 10 to identify pre-migration catalog clean-up requirements
    - DSNTIJPM may provide DDL or utility statements for the clean-up

  - DSNTIJPB is the same job and is shipped for DB2 10 to maximize prepare time

Important preparation

- **Old plans and packages before V9 -> REBIND**
- **Views, MQTs, and Table functions with Period Specification -> DROP**
  - Those created in V10 are not supported
  - Period Specification must be on base table

Items deprecated in earlier versions – Now eliminated

- **Password protection for active log and archive log data sets**
- **DSNH CLIST NEWFUN values of V8 and V9 – Use V10 or V11**
- **Some DB2 supplied routines**
  - SYSPROC.DSNAEXP –> Use the EXPLAIN Privilege and issue EXPLAIN directly
  - AMI-based DB2 MQ (DB2MQ) functions –> use the MQI-based functions in schema (see APAR PK37290 for guidance)
    - DB2MQ1C.*,   DB2MQ2C.*
    - DB2MQ1N.*,   DB2MQ2N.*
- **CHARSET application programming default value (KATAKANA) – use CCSIDs**
- **BIND PACKAGE options ENABLE and DISABLE (REMOTE) REMOTE (location-name,...,<luname>,...) -- specific names cannot be specified**
- **Sysplex Query Parallelism – Single member parallelism is still supported**
- **DSN1CHKR – There are no longer any links in the Catalog or Directory**

APPLCOMPAT – Application Compatibility

- **Requirements**
  - De-couple the need for application program changes to deal with incompatible SQL DML and XML changes from the actual DB2 system migration to the new DB2 release which introduced the incompatible SQL DML and XML changes
  - Provide a mechanism to identify application programs affected by incompatible SQL DML and XML changes
  - Provide a mechanism to introduce changes at an individual application program (package) level
    - Enable support so that application program changes can be phased in over much longer time
    - Enable support for mixed DB2 release co-existence in data sharing
    - Enable support for up to two back level releases of DB2 (N-2)
- **Solution**
  - APPLCOMPAT which separates DB2 system migration to the new DB2 release from application program migration to deal with incompatible SQL DML and XML introduced by the new release

APPLCOMPAT – Application Compatibility ...

- **APPLCOMPAT zparm provides default for BIND/REBIND**
  - V10R1 for DB2 10 SQL DML behaviour
  - V11R1 for DB2 11 SQL DML behaviour
  - Default is V11R1 for new installs, V10R1 for migration
- **APPLCOMPAT option on BIND/REBIND to override zparm default**
- **CURRENT APPLICATION COMPATIBILITY special register and DSN_PROFILE_ATTRIBUTES for DDF**
  - For dynamic SQL
- **Does not address issues with new reserved words or other incompatibilities that could only be resolved by having multiple levels of the DB2 parser**
- **BIF_COMPATIBILITY zparm is independent of APPLCOMPAT**
- **New SQL functionality available in in V11 NFM cannot be used until package is bound with APPLCOMPAT value of V11R1**

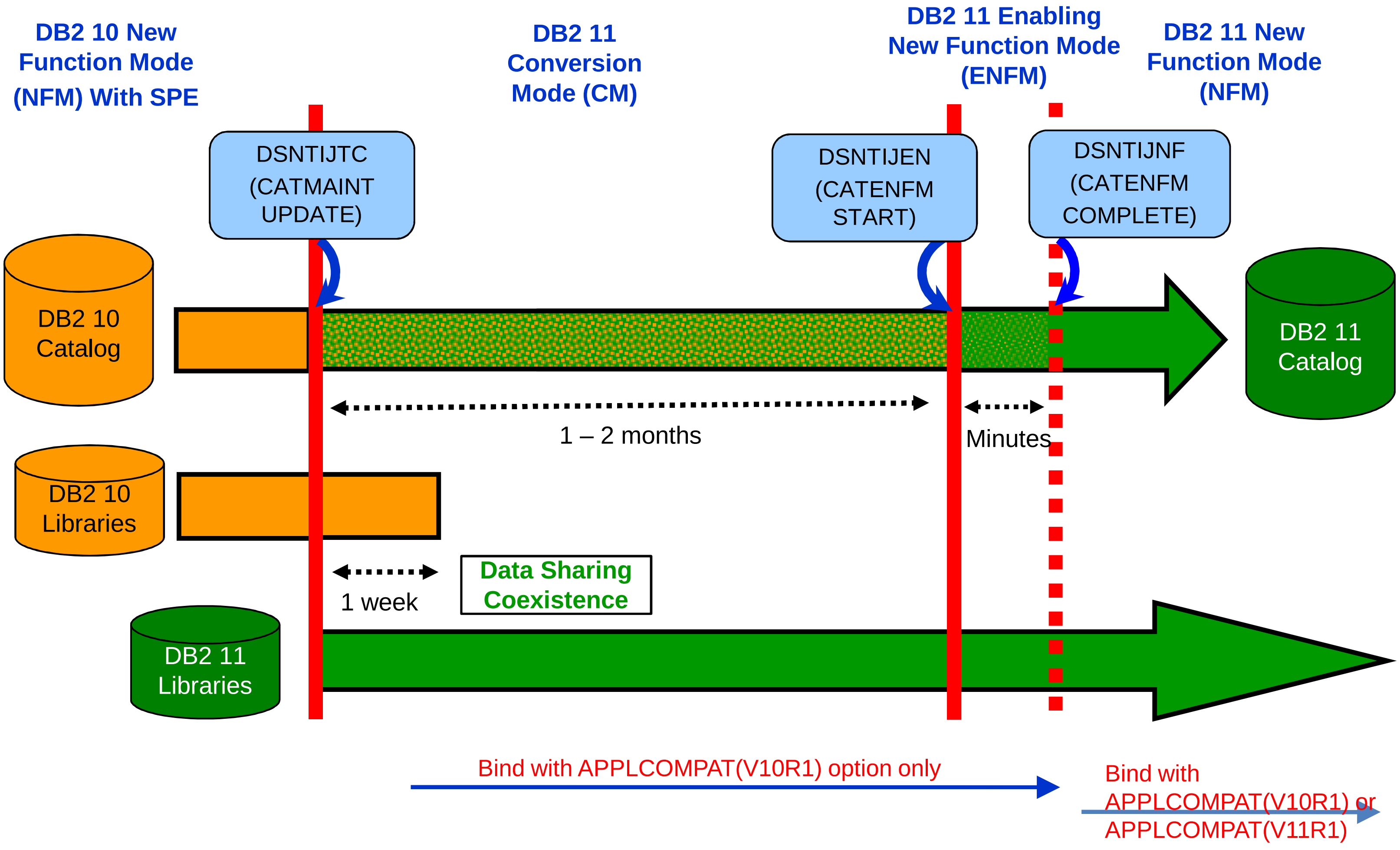
APPLCOMPAT – Application Compatibility ...

- **Migration automatically sets V10R1 prior to NFM … otherwise**
  - DSNT225I -DSN BIND ERROR FOR PACKAGE location.collid.member APPLCOMPAT(V11R1) OPTION IS NOT SUPPORTED
  - IFCID376 – Summary of V10 function usage
  - IFCID366 – Detail of V10 function usage, identifies packages
  - We expect changes necessary to avoid V10R1 usage to happen after reaching NFM
- **Workaround to distinguish packages which have to absolutely run as V10R1 until they are corrected**
  - Annotate the package using SQL COMMENT ON PACKAGE colid.name.”version” IS ‘V10R1’
    - If version is a pre-compiler timestamp then the double quotes are necessary
  - Stored in the REMARKS column in SYSIBM.SYSPACKAGE table
    - Can be queried and be exploited by housekeeping

APPLCOMPAT vs. BIF_COMPATIBILITY

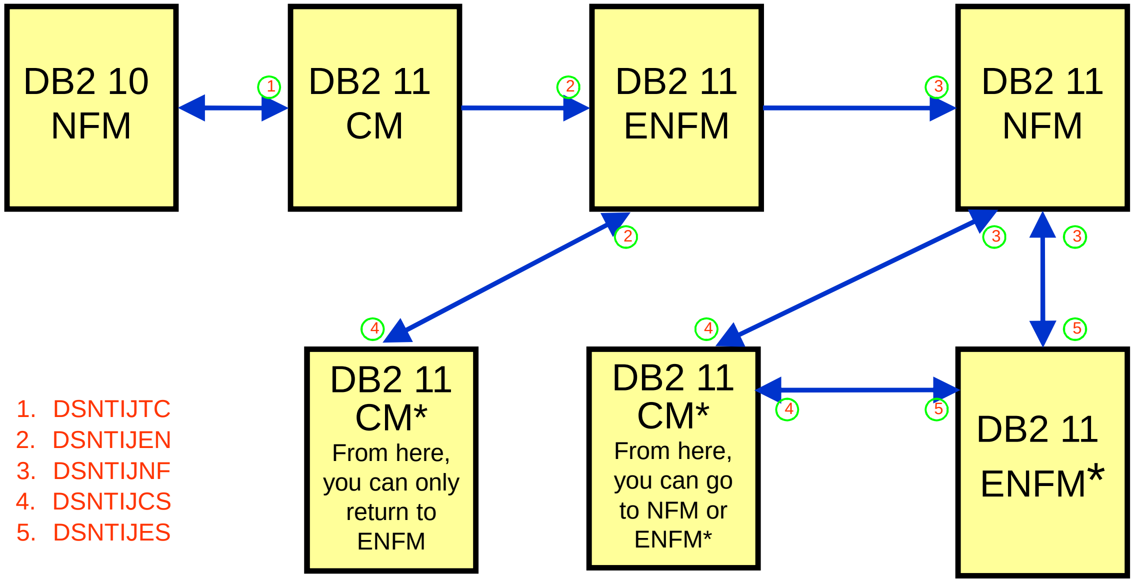
- **BIF_COMPATIBILITY=V9|V9_DECIMAL_VARCHAR is still honored in all modes of V11**
  - The 2 timestamp formats that are with invalid with BIF_COMPATIBILITY=CURRENT are
    - yyyy-mm-dd-hh:mm:ss.tttttt
    - dd.mm.yyyyhh.mm.ss.tttttt
  - Support for these ‘undocumented’ timestamp formats is back again, but only with APPLCOMPAT(V10R1)
  - Rogue applications must be cleaned up before use of APPLCOMPAT(V11R1)

Migration Overview DB2 10 -> DB2 11


DB2 10 New Function Mode (NFM) With SPE
DB2 11 Conversion Mode (CM)
DB2 11 Enabling New Function Mode (ENFM)
DB2 11 New Function Mode (NFM)
DSNTIJTC (CATMAINT UPDATE)
DSNTIJEN (CATENFM START)
DSNTIJNF (CATENFM COMPLETE)
DB2 10 Catalog
DB2 11 Catalog
1 – 2 months
Minutes
DB2 10 Libraries
1 week
Data Sharing Coexistence
DB2 11 Libraries
Bind with APPLCOMPAT(V10R1) option only
Bind with APPLCOMPAT(V10R1) or APPLCOMPAT(V11R1)

Migration and Fallback Paths

- **With DB2 11, you can always drop back to the previous stage**
- **Cannot fallback to DB2 10 after entry to DB2 11 (ENFM), but can return to DB2 11 (CM*)**
- **DSNTIJTC and DSNTIJEN will not execute if entries found in SYSUTILX**
  - DB2 will no longer blindly re-initialize it



DB2 10 NFM

DB2 11 CM

DB2 11 ENFM

DB2 11 NFM

DB2 11 CM* From here, you can only return to ENFM

DB2 11 CM* From here, you can go to NFM or ENFM*

DB2 11 ENFM*

1. DSNTIJTC
2. DSNTIJEN
3. DSNTIJNF
4. DSNTIJCS
5. DSNTIJES

Preparing your current DB2 10 NFM for Migration to DB2 11 CM

- **Apply the Fallback SPE APAR, PM31841 and any prerequisite fixes**
  - Your DB2 10 system MUST be at the proper service level
  - See Info APARs II14660

- **Non-Data Sharing**
  - Current DB2 10 must be started with the SPE applied, or migration to DB2 11 will terminate
- **Data Sharing**
  - Before migrating a member to DB2 11, all other started DB2 10 members must have the fallback SPE applied
  - The fallback SPE must be on all active DB2 10 group members for DB2 11 to start

Important – Apply SPE to ALL Data Sharing Members Before Starting Migration!

Pre-Migration Catalog Migration Testing

Pre-migration catalog migration testing

- **Objective is to mitigate risk**
- **Strongly recommended for a successful migration of production DB2 subsystem**
- **Clone the production DB2 Catalog and Directory into a separate environment to practice**
  - Options to use:
    - COPY SHRLEVEL CHANGE during a quiet catalog period (no BINDs, no DDL, …)
    - FlashCopy Image Copy
    - DFSMS Dump
    - Concurrent Copy
    - GDPS/PPRC techniques
    - etc
- **Or use DR/BCP environment**

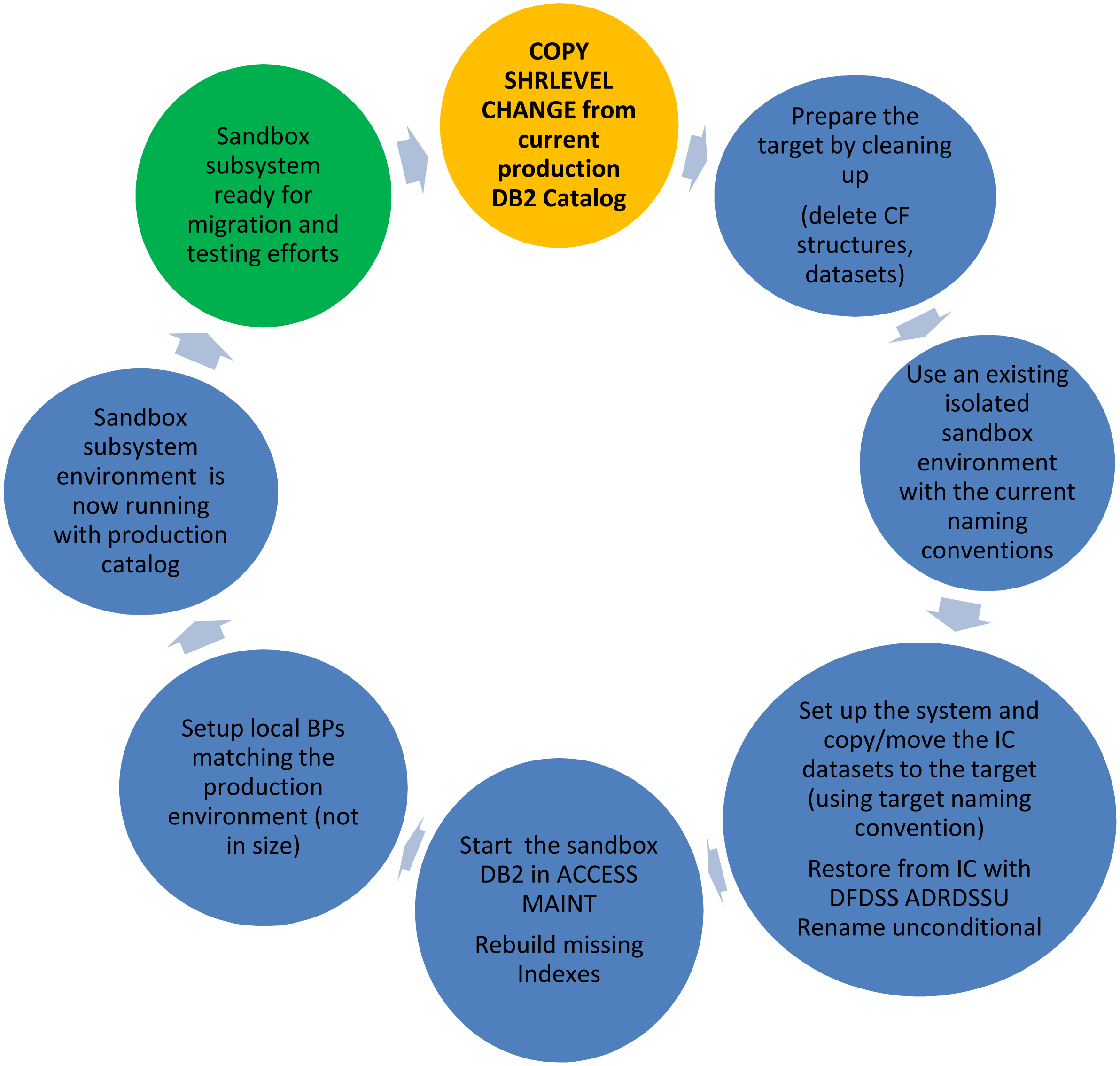
Pre-migration catalog migration testing …

- **Run health checks on DB2 Catalog and Directory to flush out any latent inconsistency and clean up before the migration**
  - Queries from migration job DSNTESQ - should always return zero rows
  - REPAIR DBD TEST or DIAGNOSE
  - COPY, or DSN1COPY with CHECK option
  - Basic RUNSTATS on all objects
  - For objects with LOB columns:
    - CHECK LOB
    - CHECK INDEX on the AUX index
    - CHECK DATA on the base tablespace using SCOPE AUXONLY AUXERROR REPORT
  - CHECK INDEX on all indexes
  - CHECK DATA for DSNDB01.SYSUTILX, DSNDB06.SYSPKAGE, DSNDB06.SYSSTATS, DSNDB06.SYSSTR

Pre-migration catalog migration testing …

- **Practice migration on the cloned copy without risk and before migrating the catalog of any production DB2 subsystem**
  - V10 -> V11 CM -> V11 ENFM -> V11 NFM
  - Rebind plans and packages
  - Explain all views
- **Engage other parties in the pre migration process, migration process and testing**
  - DBAs, Application Developers
- **Possible to expand this testing to become an extended regression test system**

Pre-migration catalog migration testing …



COPY SHRLEVEL CHANGE from current production DB2 Catalog

Prepare the target by cleaning up

(delete CF structures, datasets)

Use an existing isolated sandbox environment with the current naming conventions

Set up the system and copy/move the IC datasets to the target (using target naming convention)

Restore from IC with DFDSS ADRDSSU Rename unconditional

Start the sandbox DB2 in ACCESS MAINT

Rebuild missing Indexes

Setup local BPs matching the production environment (not in size)

Sandbox subsystem environment is now running with production catalog

Sandbox subsystem ready for migration and testing efforts

Availability

BIND/REBIND/DDL/Online REORG breaking into persistent thread running packages bound with RELEASE(DEALLOCATE)

- **Persistent threads with RELEASE(DEALLOCATE) which were previous blocking**
  - e.g., IMS Pseudo WFI, CICS Protected ENTRY threads, etc
- **Types of REORGs which invalidated packages were previously blocked**
  - REORG REBALANCE
  - Materializing REORG
- **The 'break-in' behavior is ON by default (zparm PKGREL_COMMIT =YES)**
- **Break-in is performed on a “best efforts” basis**
- **Break-in mechanism can handle idling threads at a transaction boundary (i.e., where commit or abort is the last thing performed)**

BIND/REBIND/DDL/Online REORG breaking into persistent thread running packages bound with RELEASE(DEALLOCATE) …

- **Several factors come into play for a successful break-in**
  - Persistent thread must COMMIT
  - The timing of the COMMIT and the frequency of the COMMITs are both key
  - Increasing the zparm for IRLM resource timeout (IRLMRWT) helps to keep the BIND/REBIND/DDL/Online REORG operation waiting to increase the chances of a successful break-in
- **The break-in mechanism does not apply when**
  - Running packages bound KEEPDYNAMIC(YES) which issue dynamic SQL, or
  - OPEN cursors defined WITH HOLD at the time of COMMIT, or
  - If the COMMIT happens inside a stored procedure
- **RELEASE(COMMIT) would also not break-in for the above conditions**

BIND/REBIND/DDL/Online REORG break in - How does it work

1. **BIND/REBIND/DDL/Online REORG is initiated and waits on a package lock**
   – Will timeout after 3x IRLM timeout limit (IRLMRWT)
2. **At 1/2 of the IRLM timeout limit, DB2 will get notified by IRLM that someone is stuck on a package lock**
   – If DB2 has an S-holder, DB2 will post a system task to take further action
3. **DB2 system task is awakened and checks to see if a 'recycle' of locally attached threads has been done in the last 10 seconds**
   – If not, the break-in operation will proceed
   – DB2 is trying to avoid a battering of the system via BIND/REBIND/DDL/Online REORG
4. **Send broadcast to all DB2 members to perform a recycle of locally attached threads**
5. **If task proceeds, it will loop through all locally attached threads (not DIST!) and see if they were last committed/aborted in > 1/2 of the IRLM timeout limit**
   – If so, the BIND/REBIND/DDL/Online REORG is likely waiting on them
6. **The next test is to see if DB2 can do anything about it?**
   – Each thread must be at a transaction boundary (i.e., commit or abort is the last thing)
   – If so, DB2 can process the thread

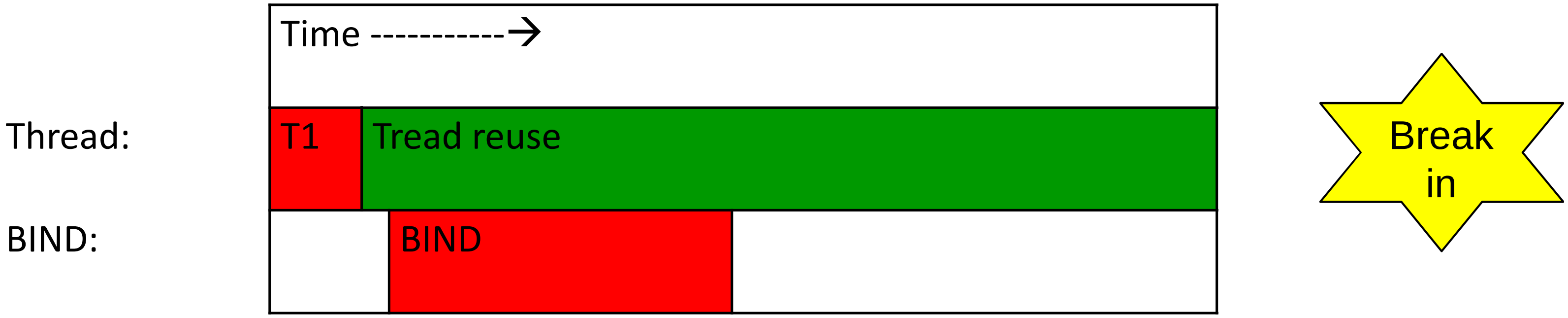
BIND/REBIND/DDL/Online REORG break in - How does it work …

7. **DB2 will fence the API for the thread, grab the agent structure and drive a ‘dummy COMMIT’**
   - The commit is transactionally OK since we are at a transaction boundary
   - DB2 will be the coordinator as this is single-phase commit and get out
   - On the COMMIT, RDS sees that there is a waiter for a package lock held by this agent and will switch to RELEASE(COMMIT) for this commit cycle
   - The lock is freed and DB2 is one step closer to the BIND/REBIND/DDL/Online REORG breaking in
8. **Repeat for all qualifying threads**
9. **BIND/REBIND/DDL/Online REORG should break-in provided there are no blockers that had to be excluded e.g., long running read only application process without a commit**
10. **If the application starts using the thread during the recycle processing, it will be blocked at the API level**
    - DB2 will spin the thread in a timed wait loop until the recycle is done
    - DB2 will wait a millisecond approximately between polls
    - DB2 has also taken care to fence end-of-task (cancel application TCB), end-of-memory (force down the home ASID during recycle), associate, dissociate, etc

BIND Break-in – Simple customer test

One CICS ENTRY Thread

1 transaction



Time -----------→

Thread: T1 | Tread reuse

BIND: BIND

Break in

- BIND waits approximately 30 sec before it breaks into an idle thread

BIND break in – additional customer testing

Action	Threads	Result
BIND	Batch No Commit	No break-in
BIND	Batch frequent Commit	Break-in
BIND	50*CICS ENTRY	Break-in
DDL:		
Create Index	CICS ENTRY	Break-in
Drop Index	CICS ENTRY	Break-in
Alter Table Add Column	CICS ENTRY	Break-in
Alter index (NO) cluster	CICS ENTRY	Break-in
Alter Tablespace to UTS	CICS ENTRY	Break-in
Alter Partition	CICS ENTRY	Break-in

ALTER LIMITKEY enhancement

- **Behavior is different depending on how the table partitioning is controlled**
- **With table-controlled table partitioning, this is a pending alteration**
  - Dropping of these alters can occur at any time
- **With index-controlled table partitioning**
  - If alter is done via ALTER INDEX ALTER PARTITION
    - Partition goes into ‘hard’ reorg pending (REORP)!
    - Tablespace remains index-controlled
    - Alter cannot be withdrawn!
  - If the alter is done by ALTER TABLE ALTER PARTITION
    - If the partition is not empty
      - Partition goes into ‘hard’ reorg pending (REORP)!
      - Tablespace is converted to table-controlled partitioning!
      - Alter cannot be withdrawn
    - If the partition is empty
      - Alter is executed immediately
      - Tablespace is converted to table-controlled partitioning

ALTER LIMITKEY enhancement …

- **Two new zparms introduced**
  - PREVENT_ALTERTB_LIMITKEY=YES
    - ALTER TABLE ALTER PARTITION leads to SQLCODE -876
    - ALTER INDEX ALTER PARTITION is still possible – do not use it because of REORP
  - PREVENT_NEW_IXCTRL_PART=YES
    - Can no longer create new index-controlled partitioned tablespaces
- **Materializing REORG can now break-in to a persistent thread running RELEASE(DEALLOCATE) package**
- **REORG REBALANCE**
  - Not possible for partitions with pending ALTER LIMITKEY changes
  - Will work for the other partitions
  - Will work for partitions which are 'hard' reorg pending (REORP)

DROP COLUMN

- **Works well**
  - Can convert to UTS and concurrently DROP COLUMN
  - Materializing REORG can be run at the partition level – if all partitions are covered
  - All packages touching the table will be invalidated
- **Restrictions**
  - Cannot use DROP COLUMN in classic tablespace type (SQLCODE -650)
  - Cannot drop a column contained in an index or view (SQLCODE -478)
  - Cannot add a dropped column before the materializing REORG (SQLCODE -20385)
  - Cannot create a view with a dropped column (SQLCODE -20385)
  - Cannot drop the same column a second time before the materializing REORG (SQLCODE -205)
  - Cannot unload from an image copy taken before the materializing REORG (DSNU1227I)
  - Cannot recover to a PIT before the materializing REORG (DSNU556I)

Utilities

REORG Enhancements

- **Better chance of achieving drain on part level REORG and faster SWITCH phase**
- **SWITCHTIME option avoids the need for multiple jobs to control the start of the drain**
- **Part level COPY when reorganizing a subset of partitions**
  - Tape support added, but no support yet for STACK YES
  - Changes required to existing jobs
- **REORG SORTDATA NO RECLUSTER YES|NO**
  - RECLUSTER NO will bypass sort (and speed up conversion to extended log RBA/LRSN format)
    - Will not unload via the clustering index and will not sort the data
    - Will pay the additional cost of sorting the clustering index keys, but if the index key is much smaller than the row length can save quite a bit of processing
  - RECLUSTER NO is enforced for SHRLEVEL CHANGE with SORTDATA NO
    - Specify SORTDATA to get reclustering
  - DSNU2904I DATA RECORDS WILL BE UNLOADED VIA unload-method
    - CLUSTERING INDEX
    - TABLE SCAN
    - TABLE SPACE SCAN

REORG Enhancements …

- **Automated building of the mapping table with new 10 byte LRBA/LRSN**
  - V11 CM behavior
    - An existing mapping table in V10 or V11 format will be reused
    - If mapping table does not exist, mapping table will be automatically temporarily created
  - V11 NFM behavior
    - Existing mapping table if in V11 format will be reused
    - If mapping table exists but in V10 format, a new mapping table will be automatically created in the same database as the original mapping table
    - If mapping table does not exist, mapping table will be automatically created in database specified by zparm, or in declared database or in DSNDB04
  - Recommendations
    - Predefine and keep mapping tables around for regularly scheduled REORG jobs to avoid SQL DDL contention on the Catalog
    - Use single specific database as specified by zparm for all mapping tables
    - Modify schema of existing mapping tables to V11 format as part of migration process to NFM i.e., ALTER TABLE TBMAP ALTER COLUMN LRSN SET DATA TYPE CHAR(10) NOT NULL;
    - Wait for APAR PI08339 if you want automated building of mapping tables

REORG Enhancements …

- **Use of DRAIN_ALLPARTS YES option (not default) has the potential to significantly reduce the ‘outage’**
  - Avoid deadlocks between drains and claims across NPIs and partitions when reorganizing subset of partitions
  - Solution is to momentarily drain all partitions (entire tablespace)
  - More likely to be successful in getting successful DRAIN to make the SWITCH
  - Big reductions seen in the elapsed time to complete DRAIN and SWITCH
  - REORGs should run with less problems using this feature
- **REORG message output - DSNU1138I provides drain begin / end information**
- **PARALLEL (maximum number of subtasks) option to control the number of subtasks**
- **Be aware of changed defaults e.g., NOPAD YES for REORG DISCARD**
- **LOGRANGES NO option should only be used when SYSLGRNX is known to be logically corrupted and/or has to be reset**

REORG Enhancements …

- **REORG REBALANCE**
  - Now supports SHRLEVEL(CHANGE) – big step forward for 24*7
  - Can now deal with partitions that were empty (or did not contain enough data for a compression dictionary to be built during the UNLOAD phase) before the REORG
    - Will now build a single compression dictionary that will get applied to all target partitions
    - There is no longer a need for subsequent REORG to gain compression
  - Can now break-in on persistent threads running RELEASE(DEALLOCATE) packages
  - But aware of possible negative impact on parallel application batch processing
- **Partition pruning for UTS PBG tablespaces**
  - Option to physically remove and contract the number of UTS PBG partitions
  - Only performed when zparm REORG_DROP_PBG_PARTS=ENABLE
  - Disabled by default
  - There is no support for PIT recovery to a point in time prior to SWITCH phase for a pruned tablespace

RUNSTATS and RTS Enhancements

- **Inline Statistics are rough estimates and should not be compared against a separate RUNSTATs**
- **Now possible to avoid DSNU602I STATISTICS ARE NOT COLLECTED FOR NONPATITIONED INDEX on REORG PART operation**
  - When SORTNPSI option on REORG job or REORG_PART_SORT_NPSI zparm set to AUTO or YES, and
  - REORG sorted all of the non-partitioned index keys because the amount of data that was being reorganized relative to the size of objects exceeded internal thresholds
- **New RESET ACCESSPATH option**
  - Reset missing and/or conflicting access path statistics in the Catalog
  - Does not affect space statistics in the Catalog or RTS
- **Avoid DSNU1363I THE STATS PROFILE FOR TABLE table-name NOT FOUND**
  - Will use fixed defaults
- **No support for USE PROFILE with inline statistics in REORG and LOAD**
- **Can externalize RTS in-memory blocks via the following command**
  - –ACCESS DATABASE (DB) SP(TS) MODE(STATS) command

RECOVER enhancements

- **Fast Log Apply (FLA) now implemented for RECOVER INDEX**
  - Previously DB2 would wait until a log record was to be applied before reading the associated index page into the local bufferpool where it would then be cached
  - Now DB2 will use list prefetch to read all the index pages that are needed to apply log records for, before applying any log record
  - Potential for significant savings in elapsed time
  - Should now reconsider decision: run RECOVER INDEX in parallel with RECOVER TABLESPACE [PART] vs. wait for RECOVER TABLESPACE [PARTs] to complete and then run REBUILD INDEX
  - Enhancement taken back to V9 and V10 via APAR PI07694
- **Optimization to point-in-time RECOVER list of objects**
  - Recover objects only when necessary when performing PIT recovery when TOLOGPOINT or TORBA are specified
  - It does not apply to log only recoveries, RECOVER BACKOUT, and recovers to current
  - DIAGNOSE TYPE(607) is required to activate this behavior

Thank You!