Best Practices for Utilities on DB2 for z/OS

Haakon Roberts
IBM

Session Code: F13
Wed Nov 16 2011 15:45-16:45 | Platform: z/OS

Agenda

- General recommendations
- COPY & FlashCopy
- RECOVER/QUIESCE/MODIFY
- LOAD/UNLOAD
- REORG
- RUNSTATS
- CHECK
- DSN1COPY
- Summary

General recommendations

Summary

- **Newer releases are better than older releases**
  - Performance
  - Function
  - Availability
  - But be mindful of part level REORG in V9
- **Newer maintenance levels are better than older maintenance levels**
  - Performance
  - Function
  - Availability
- **Attend "Utility Update" talks at conferences to get the latest information**

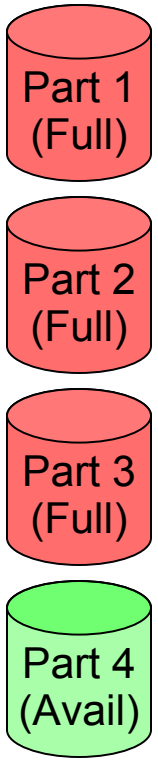
Sort processing

- **Improved utility sort processing**
  - CHECK INDEX, REBUILD INDEX, REORG, RUNSTATS
  - PK45916 (V8) & PK41899 (V9)
  - Better performance, more robust, simpler
- **SORTNUM no longer required**
  - Correct value hard to determine, resulting in utility failure if too low or excessive sort work allocation if too high
- **New zparms UTSORTAL & IGNSORTN (online changeable)**
  - UTSORTAL YES|NO
    - Use RTS data to estimate number of rows to sort
    - DB2 will dynamically allocate sort work datasets
    - If SORTWK DD cards not hard coded
  - IGNSORTN YES|NO
    - Override utility job setting of SORTNUM
- **Recommendation**
  - Turn on UTSORTAL, test it, then consider turning on IGNSORTN

DSNU3340I 168 08:13:52.66 DSNUGLSR - UTILITY PERFORMS DYNAMIC ALLOCATION OF SORT DISK SPACE

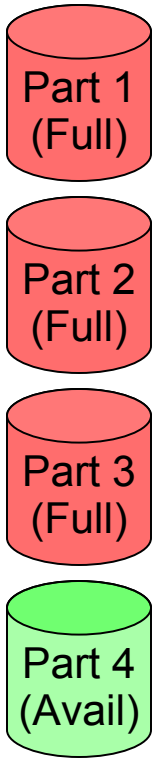
PBG

- **No LOAD or REORG parallelism**
- **REORG of single partition must fit rows back into that partition**
- **REORG of multiple parts can redistribute rows across partitions**
  - Will fill up earlier partitions first
- **Prior to V10, REORG of base with LOB columns cannot redistribute rows across partitions**
- **Empty partitions will not be deleted**
- **REORG cannot grow new parts unless REORG at table space level**
- **REORG at table space level will grow new parts as needed**
  - Not prior to V10 if LOB columns exist
  - REORG of single part or subset of parts will not grow new parts
  - New LOB table spaces will be added, but will be left copy-pending



PBG

- **REORG fails if rows must fit back into partition but cannot**
  - Possible causes:
    - PCTFREE & FREEPAGE
    - Change in compression ratio or alter to COMPRESS NO
    - BRF-RRF conversion
    - MAXROWS change
    - Change to MEMBER CLUSTER
    - Etc
- **What to do if REORG fails because rows won't fit?**
  - Use SHRLEVEL CHANGE to avoid application impact
  - View as single table and REORG whole PBG table space
  - Use zparm to ignore PCTFREE/FREEPAGE on part-level PBG REORG
    - Introduced by PK83397
  - If LOB columns exist then may need to UNLOAD/RELOAD if pre-V10
- **Prior to DB2 10 be careful about using PBG for tables with LOB columns**



Utilities on demand with supplied stored procedures or tools

- **Run utilities only when necessary and not on fixed schedules**
- **Information on the current status of all objects is contained in Real-Time Statistics (RTS) tables**
- **DSNACCOR/DSNACCOX apply our suggested thresholds and formulas against a list of objects and recommend utility actions**
  - DSNACCOX in V9 NFM has improved RTS exploitation and recommendations
  - Use RESTRICT option to ensure restricted/advisory states are also checked
- **Leverage the ability to invoke utilities programmatically via stored procedures**
  - DSNUTILS for EBCDIC parameters
  - DSNUTILU for UNICODE parameters
- **Rich application logic can control what is run and when**
- **Refer to the DB2 Utility Guide and Reference Appendix B and samples**
- **Use LISTDEFs and TEMPLATEs for further simplification**
- **Or use an automation tool for simplified automation**

COPY

COPY

- **SHRLEVEL CHANGE unless consistent copies are essential**
- **Use PARALLEL keyword to exploit parallelism**
- **Consider OPTIONS EVENT(ITEMERROR,SKIP)**
  - Sets UTRW state only for duration of copy of individual page set
  - But increases COPY overhead
    - Serialisation required for each pageset on the fly
- **Consider taking incremental copies and using MERGECOPY**
  - MERGECOPY marks relevant page set UTRW
- **Copy indexes on large, critical tables**
  - Particularly if rarely or never updated
  - Only drawback – increase in SYSLGRNX & SYSCOPY recording
  - Needed for index RECOVER BACKOUT option in DB2 10
  - Automatically included in MODIFY RECOVERY
- **Consider CONCURRENT COPY**
  - Can reduce CPU & elapsed time
  - Uses DFDSS backup/restore
  - Prohibits use of DSN1COPY & UNLOAD from image copy

FlashCopy considerations

- **Can provide significant elapsed & CPU savings**
  - Slightly higher setup cost so may see increased elapsed time for very small datasets
- **RECOVER will work even though background copy not complete**
  - Relationship will be broken & then slow restore of copied tracks
- **No incremental copy permitted after Flashcopy**
  - Incremental copy requests will be converted to full
- **Ensure that data resides on Flashcopy-enabled DASD to avoid slow DFDSS copy**
  - DB2 10 REORG will be changed to use FASTREPLICATION(REQUIRED) in APAR
  - This means Flashcopy failure in REORG can leave object in copy-pending
- **FlashCopies cannot have GDGs as a target**
- **No UNLOAD direct from Flashcopy – can use COPYTOCOPY to create sequential from Flash**
- **HSM migrated datasets not supported by RECOVER**
  - If FlashCopy dataset is migrated then it will be skipped
- **Do not create transaction-consistent image copies in DB2 10 unnecessarily**
  - DB2 10 allows creation of consistent image copies from COPY SHRLEVEL CHANGE
  - Can result in longer recovery time

Backup solutions

- **Multiple options**
- **BACKUP SYSTEM**
  - Volume-level FlashCopy
  - Significant DASD investment required
  - Can be complex to set up & administer, but invocation simple
  - Must understand any limitations that currently exist
    - E.g. DASD mirroring issues, dataset movement issues, etc.
- **Sequential image copies**
  - Tried and trusted solution since V1.1
- **Other external backups, such as volume-level backups, DSN1COPY**
  - Outside of DB2's control
  - Requires careful management and co-ordination
- **FlashCopy in DB2 10**
- **Choice is dependent on environment and requirements, all options will continue to be supported**

RECOVER/QUIESCE/MODIFY RECOVERY

RECOVER

- **Maximise exploitation of parallel restore and Fast Log Apply**
  - Recover multiple objects in a list in parallel but ideally <100
  - Avoid running more than 10 RECOVER jobs per member
- **Copy indexes and include in recovery list, particularly for PIT recovery**
- **Split off page sets that are not updated and recover separately**
- **For PIT recovery, include whole RI set in same RECOVER statement**
- **For PIT recovery, include base and aux objects in same RECOVER statement**
  - V10 can enforce this via zparm
- **RECOVER option BACKOUT YES in DB2 10 for PIT recovery**

QUIESCE & MODIFY RECOVERY

- **QUIESCE**
  - Do you still need it in V9 with PIT recovery with consistency?
    - If you want an LRSN or RBA marked in SYSCOPY, run QUIESCE on DSNDB06.SYSEBCDC
  - Use WRITE NO unless you absolutely must have pages written out
- **MODIFY RECOVERY**
  - Consider running every time a backup is taken or at least weekly
  - REORG SYSLGRNX regularly for optimal performance and minimal MODIFY impact on system
  - DB2 9 has RETAIN LAST n, GDGLIMIT and BSDS options
    - Careful with GDGLIMIT if you use multiple GDGs for a single object
  - Will not clean up “orphan” entries
  - Run MODIFY to delete recovery information from prior to a REORG that materialises row alterations
    - Makes subsequent REORGs more efficient

LOAD/UNLOAD

LOAD/UNLOAD

- **Run LOAD with LOG NO, REUSE, KEEPDICTIONARY if possible**
- **Allocate inline copy data sets to DASD**
- **Split up input dataset and drive LOAD partition parallelism in a single LOAD**
- **Use SORTNUM elimination**
- **Specify NUMRECS if input is on tape or variable length**
- **If loading partitioned table with single input dataset, presort data in partitioning key order**
  - PRESORT option in Utility Enhancement Tool
- **For LOAD REPLACE, consider loading into a "clone" then renaming tables or datasets**
- **Consider using USS named pipes**
  - Refer to PK70269 & PK96023
- **DB2 10: Do not expect LOAD into hashed tables to perform as well as non-hashed**
  - UET provides PRESORT option to significantly improve LOAD-to-hash performance

LOAD/UNLOAD

- **Consider whether you want UNLOAD utility or HPU**
  - High Performance Unload is a separately chargeable tool
    - May provide elapsed & CPU reduction
    - Permits SQL interface
    - Permits unload from page set on DASD

- **FRV processing for LOAD/UNLOAD**
  - Apply PK75216 for significant performance enhancement for PDS FRVs in V9
    - HFS still performs better in elapsed time
  - Note limit of 522,239 members in a PDS/E
  - NULL LOBs are handled better than zero length LOBs
    - No FRV created on UNLOAD for null LOBs
  - V8 – zero length LOBs will result in zero length LOBs put in LOB table space
    - Fixed in V9
  - V9 performance better than V8, but V10 VBS format much better than FRVs

REORG

REORG

- Reorg multiple partitions in a single REORG statement, particularly if NPSIs exist
- If reorging many parts, and NPSIs exist, consider whether REORG of entire table space is quicker
- If NPSIs disorganised, move to V10 for faster REORG INDEX & part-level REORG
- Use DSNACCOX, Automation Tool or other logic to only reorg what needs to be reorged
- Use SORTNUM elimination (UTSORTAL=YES, IGNSORTN=YES)
- Consider REBUILD INDEX instead of REORG INDEX if pre-V10 and index is disorganised

REORG

- **REORG SHRLEVEL CHANGE for maximum availability**
- **Use DRAIN ALL rather than DRAIN WRITERS**
- **Use TIMEOUT TERM to free up objects on timeouts**
- **If minimising application impact is key:**
  - (DRAIN_WAIT + MAXRO) < (IRLMRWT -5 or 10 secs) for minimal application impact
  - Specify high RETRY value (6 or more)
- **If REORG success in a small window is key:**
  - Consider starting REORG early with MAXRO DEFER then -ALTER UTILITY command
  - High DRAIN_WAIT & MAXRO to guarantee REORG success
- **If using REORG DISCARD, use NOPAD for improved performance**
- **LOBs**
  - SHRLEVEL REFERENCE in V9, SHRLEVEL CHANGE in V10
  - Stop using SHRLEVEL NONE before DB2 10 NFM

REORG INDEX vs. REBUILD INDEX

- **REBUILD INDEX SHRLEVEL CHANGE provided in V9**
  - Excellent for create of new non-unique indexes and for indexes that are broken or already in RBDP
  - Does not operate against a shadow, so will set RBDP if not already set
- **REORG INDEX operates against a shadow & unloads keys in order from index**
- **REORG INDEX has better availability than REBUILD INDEX**
- **REBUILD may be faster, particularly if index is disorganized**
  - **REORG INDEX performance improved in V10 due to list prefetch**

RUNSTATS

RUNSTATS

- **Do not use RUNSTATS to gather space statistics – rely on RTS**
- **Do not gather unnecessary stats**
- **Do not run RUNSTATS for LOB table spaces**
  - Use RTS instead
- **Use sampling, and TABLESAMPLE AUTO in DB2 10 for page sampling**
- **Use profiles in DB2 10**
- **Use inline stats where possible rather than RUNSTATS**
  - No PROFILE support or zIIP offload for inline stats in V10
- **Specify KEYCARD**
  - Index cardinality stats are cheap to collect and heavily relied upon by optimizer

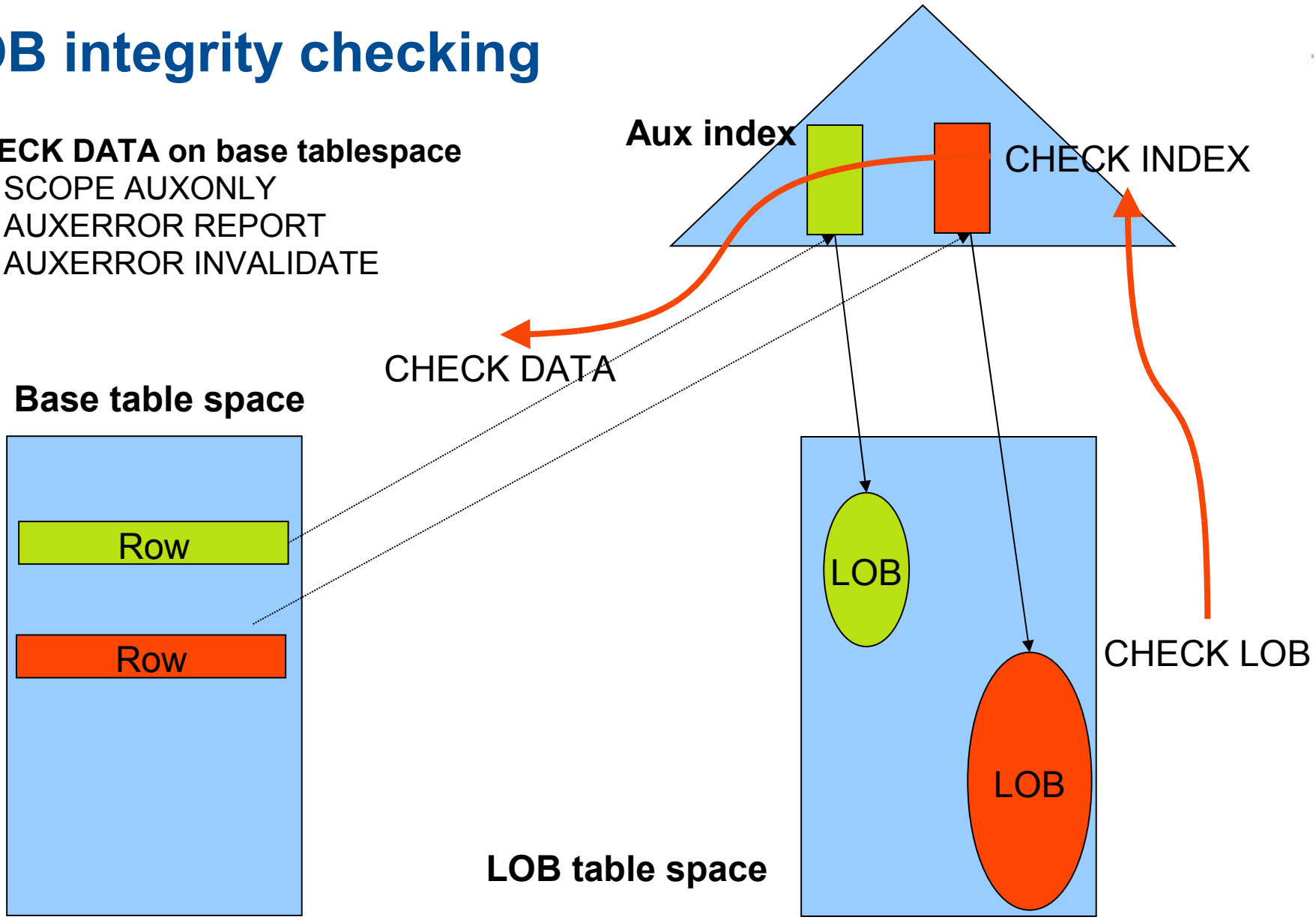
CHECK utilities

CHECK

- **If no applications are impacted and you just want to check consistency, then:**
  - Either run CHECK… SHRLEVEL CHANGE
  - Or run standard CHECK utility but be ready with REPAIR to reset CHKP/ACHKP states
  - Or could consider running SQL ISO(UR) instead
  - CHECK utilities do not set CHKP/ACHKP any more in V10
- **If running SHRLEVEL CHANGE:**
  - It will not reset CHKP or ACHKP states, nor will it set them
  - Make sure the page set is on FlashCopy-enabled DASD
    - If not then you'll get a slow copy with the data in UTRO
  - If using DASD mirroring or BACKUP SYSTEM then use ZPARM UTIL_TEMP_STORCLAS to prevent impacting either
    - Refer to PK41711

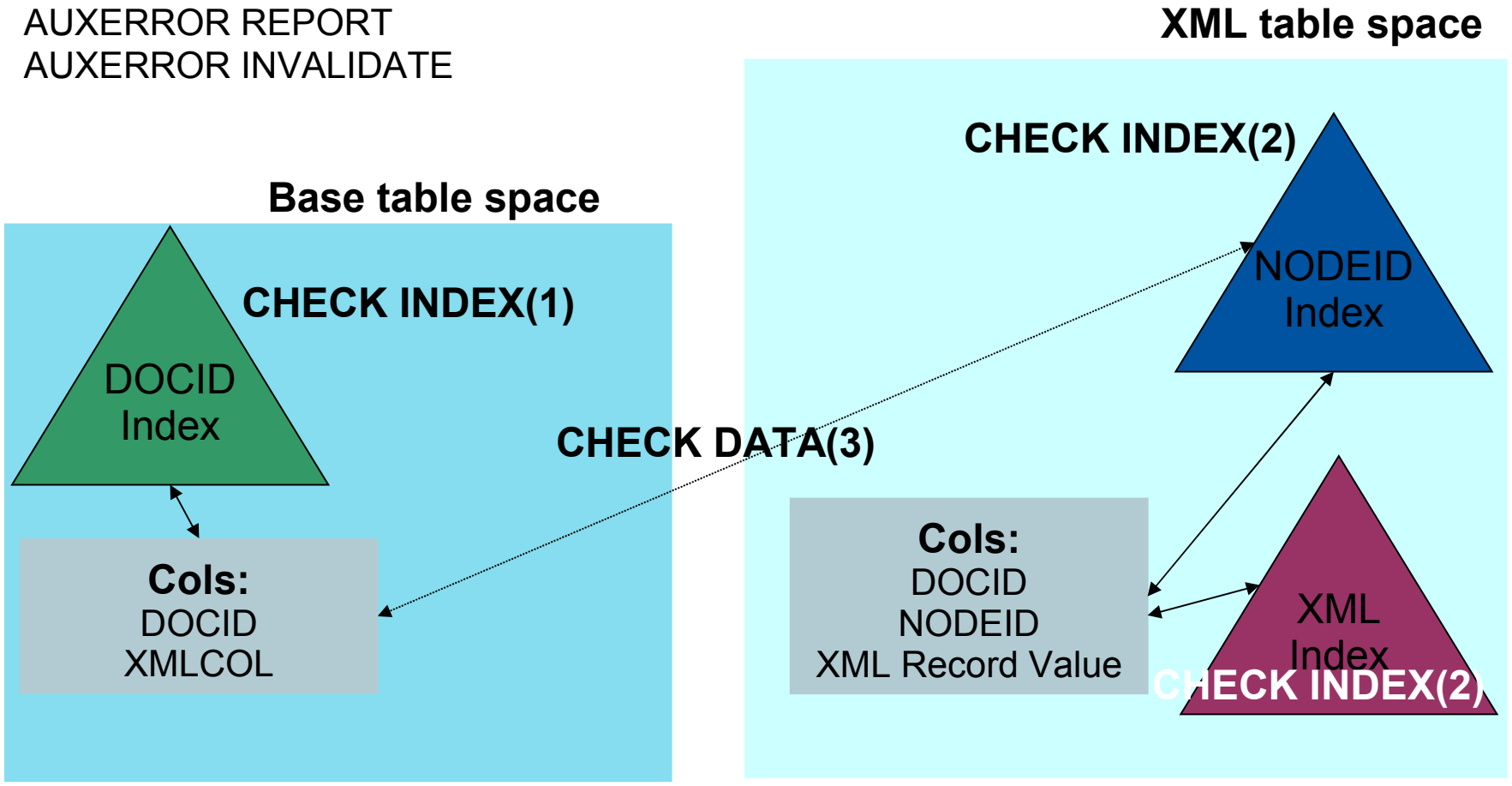
LOB integrity checking

- **CHECK DATA on base tablespace**
  - SCOPE AUXONLY
  - AUXERROR REPORT
  - AUXERROR INVALIDATE



XML integrity checking

- **CHECK INDEX on DOCID, NODEID, XML indexes**
- **CHECK DATA on base tablespace**
  - SCOPE AUXONLY
  - AUXERROR REPORT
  - AUXERROR INVALIDATE



DSN1COPY

DSN1COPY

- **DSN1COPY is an essential part of the utilities portfolio**
- **DSN1COPY runs standalone and cannot ensure that data matches definition at target**
- **All target datasets must be pre-allocated for multi-piece table spaces**
- **Areas to watch out for:**
  - BRF-RRF mismatch
    - Tolerated by SQL, but not REORG
    - Convert page sets to ensure copy is RRF-RRF or BRF-BRF
    - If that isn't possible, e.g. if image copy is BRF, then UNLOAD from BRF image copy and LOAD into RRF page set
  - Data definition changes, e.g. columns added
    - REORG at source before DSN1COPY, particularly if no updates since first ALTER
    - Problem if first alter of table creates new version but no data inserted so no system page information and then DSN1COPY to target
    - Use REPAIR VERSIONS at target site
    - PM27940 enhances REPAIR VERSIONS to extract system page information from any partition and preserve it for use by data in other parts
  - Different table space types or different segsizes
    - Not policed, abends will occur

DSN1COPY

- **Areas to watch out for contd.:**
  - XML
    - Data-dependent information kept in catalog table XMLSTRINGS
      - Cannot DSN1COPY XML table space from one subsystem/group to another
      - DSN1COPY within a subsystem/group is fine
      - Solution is UNLOAD/LOAD
    - DOCID is a sequence generated by DB2 – DSN1COPY to a new target where the DOCID is lower will result in -803 on insert because DB2 generates a value of n but n already exists in the table. ALTER of the sequence isn't allowed for DB2-generated sequences.
      - SELECT NEXT VALUE FOR <seq-for-docid> can be used to increment the number up to the max DOCID in the table

Summary

- Stay reasonably current on versions and maintenance
- Understand what this gives you in terms of utility capability
- Revisit your existing utility jobs to benefit from new options

Haakon Roberts
IBM
haakon@us.ibm.com

F13
Best Practices for Utilities on DB2 for z/OS