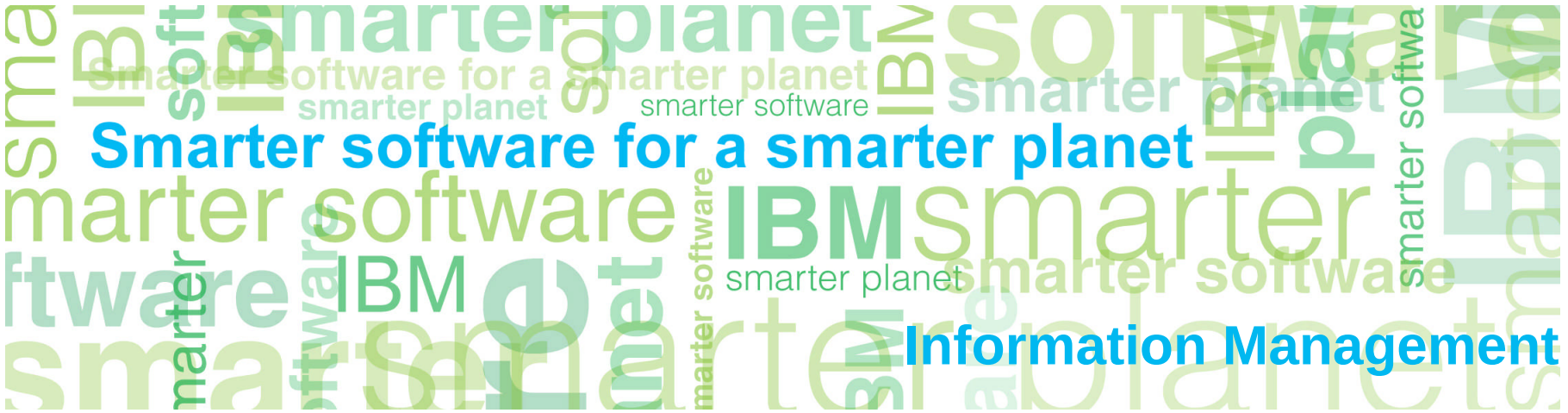
DB2 for z/OS REORG Utility Deep Dive

Ka Chun Ng
DB2 Development
Nov 2012



Acknowledgements and Disclaimers:

Availability. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

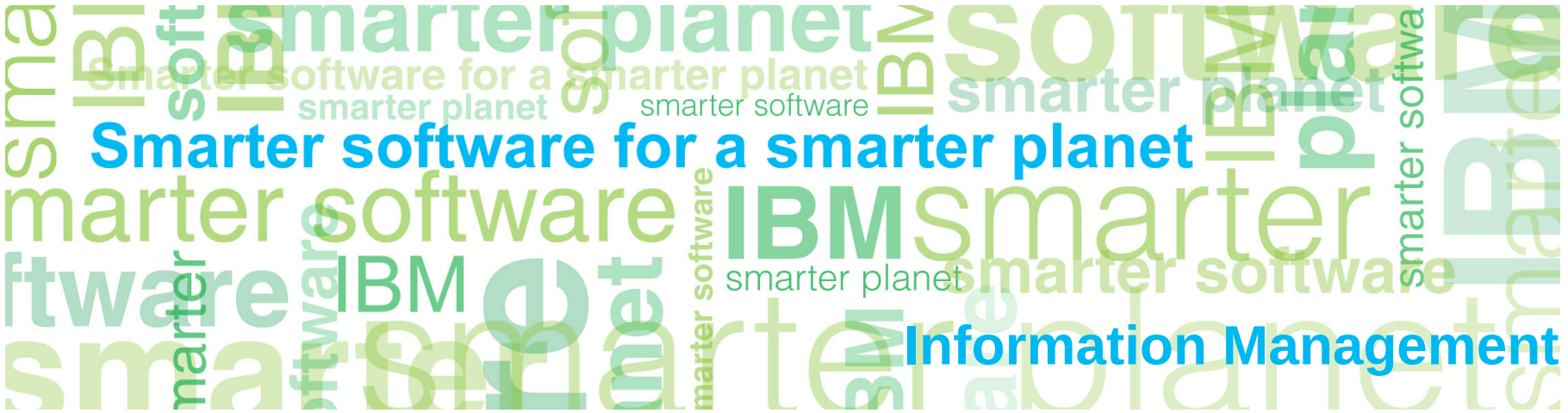
The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Agenda

- **Overview**
- **REORG Processing**
  - Phases
  - NPSIs
  - Drain processing
- **Controlling Availability**
- **PBG**
- **LOBs**
- **What and When to REORG**
- **Summary of New Function in DB2 10**
- **Overview of Recent Maintenance Stream Enhancements**
- **Summary of Recommendations & Best Practices**
- **Summary**

Overview



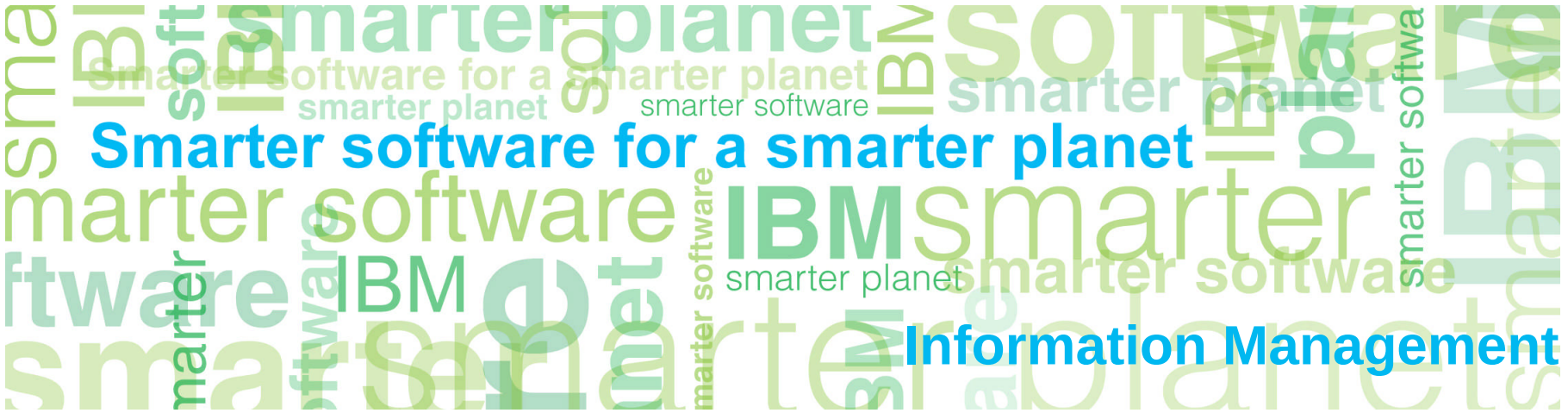
Why REORG?

- **The REORG utility is critical for core function enablement in DB2 for z/OS**
  - Non-disruptive materialization of online schema changes
    - Added columns
    - Altered columns
    - BRF-RRF row format conversion
    - Inlining of existing LOB data
    - Conversion to/from hash pageset format
    - Partition limitkey changes
    - Conversion to UTS
    - Page size, dataset size alteration
  - Redistribute data across partitions, even with LOB columns
  - Discard of data from tables
  - Resize pagesets
  - Resize hash space, including auto-resize
  - Compress data

Why REORG?

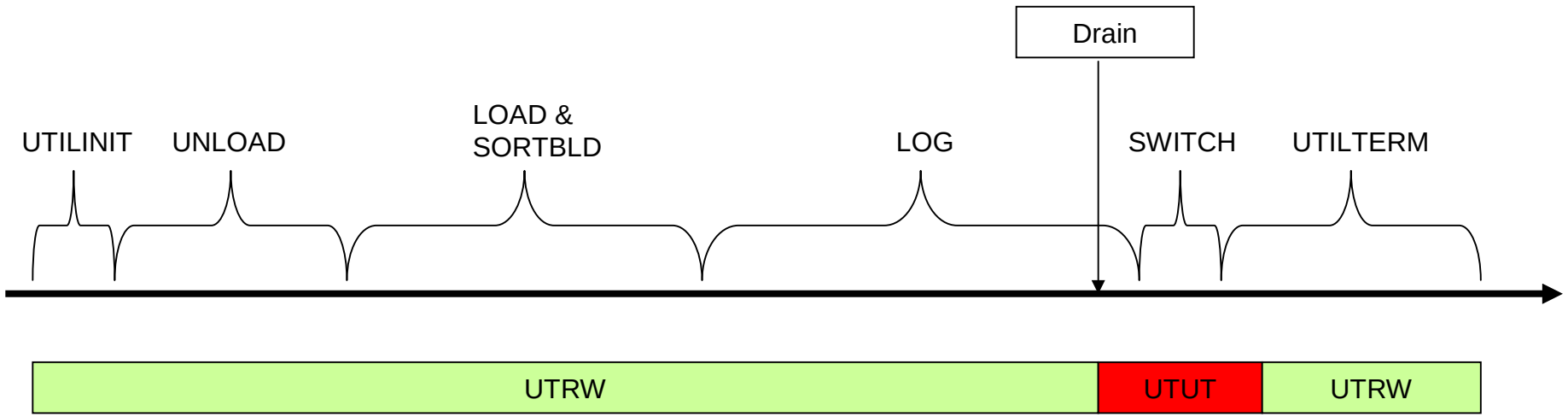
- **And of course… REORG for application performance**
  - Re-chunk LOB data
  - Re-establish data clustering
  - Re-establish free space
  - Consolidate pointer-overflow records
  - Remove deleted & pseudo-deleted records
  - Reorder index leaf pages & remove pseudo-deleted index entries
- **All this, while maintaining application availability**

REORG Processing



REORG SHRLEVEL CHANGE phases & availability

- **SHRLEVEL CHANGE & DRAIN ALL**
- **Not to scale**



Phase processing

- **UTILINIT**
  - Build data structures
  - Serialise objects
  - Create shadow pagesets
- **UNLOAD**
  - Unload pagesets in parallel
  - Pass to sort & start data sort
- **LOAD**
  - Continue sort process
  - Load sorted data into shadow pagesets
  - Build inline image copy
  - Gather statistics if requested
- **SORTBLD**
  - Concurrent with load process
  - Extract index keys, sort if necessary & build shadow index pagesets

Phase processing

- **LOG**
  - Scan log in multiple iterations for missing data updates & apply to shadow pagesets
  - Continue inline image copy build
  - Drain & complete log phase
- **SWITCH**
  - Finish inline image copy
  - Switch shadow & main pagesets
  - Update catalog & directory information
- **UTILTERM**
  - Release drain
  - Update catalog statistics if requested
  - Delete shadow datasets
  - Deserialise & clean up

NPSI processing with partition-level REORG

- **Prior to V9**
  - No shadowing of NPSIs
  - BUILD2 phase at end of REORG leaves logical NPSI partitions unavailable for extended period
- **V9 & beyond**
  - BUILD2 phase removed
  - NPSIs shadowed
- **Implications of V9 removal of BUILD2 phase**
  - Better availability since NPSI processed whilst pagesets are UTRW
  - Shadow of NPSI means cannot REORG parts in separate REORG jobs concurrently
  - Entire NPSI must be drained for switch to shadow
  - Processing of entire NPSI can be costly
- **Enhancements delivered & planned to address these issues**

Support for multiple partition ranges

- **Mitigate inability to reorg separate parts in separate REORG jobs concurrently**
- **Support REORG of multiple part ranges**
- **More efficient, improved availability, exploit parallelism**
  - PK87762 & PM13259 (V9)
- **E.g. REORG PART 1,45:71,500:503,4010**
- **LISTDEF parts will now process in a single REORG**
  - 2 implications to consider:
    - Might not have the disk space for sortwork or shadow pagesets
    - OFFPOSLIMIT/INDREFLIMIT apply to entire set of partitions
      - May find more partitions are reorged because more are processed in single REORG
- **If cannot tolerate the above, then set new zparm to SERIAL**
- **PM25525 (V9) – new PARALLEL keyword on REORG**
- **PM37293 (V9) – new REORG_LIST_PROCESSING zparm to control parallel REORG processing when input is a part-level LISTDEF**

NPSI processing cost

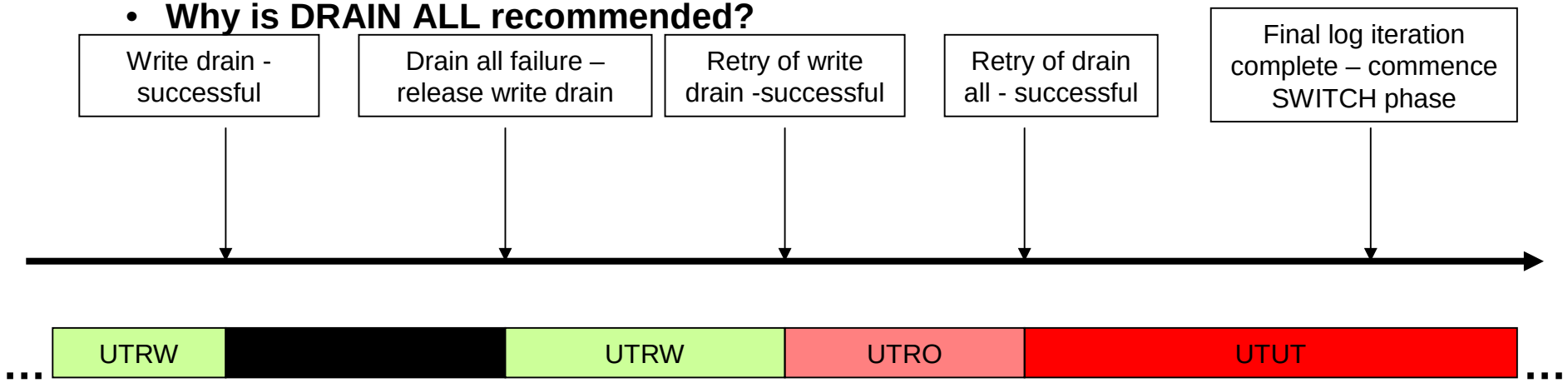
- **2 parts to NPSI cost when reorging subset of partitions**
  - Keys for partitions not included in REORG are unloaded in key order to build shadow
  - Once sorted, keys extracted from data being reorged are inserted into shadow NPSI
- **Cost of 1 is higher if NPSI is disorganized since keys extracted in key order**
- **If small % of data is reorged then cost of 1 is higher and cost of 2 is lower**
- **If high % of data is reorged then cost of 1 is lower and cost of 2 is higher**

NPSI processing cost

- **Relief**
  - DB2 10 provides index leaf page list prefetch to cut down on cost of unload of index keys from disorganized NPSI (scenario 1)
    - One internal measurement showed elapsed time for REORG reduced by up to 60%
  - REORG of entire table space may be cheaper than REORG of subset of parts
  - Consider converting NPSIs to DPSIs – but be aware of potential application performance impact of DPSIs
  - Retrofit Sequoia enhancement to reduce cost of index key insert to shadow in PM55051
    - Additional sort processing required
    - 35% ET reduction for REORG of 10% of data with single NPSI, but additional CPU cost incurred
    - DB2 Sort can eliminate CPU overhead of new algorithm

Drain processing

- **Drain of all claimers required to perform switch**
- **Pagesets drained serially**
  - So REORG of multiple parts can require extended period for drain completion
- **2 options:**
  - DRAIN WRITERS
  - DRAIN ALL
- **Why is DRAIN ALL recommended?**



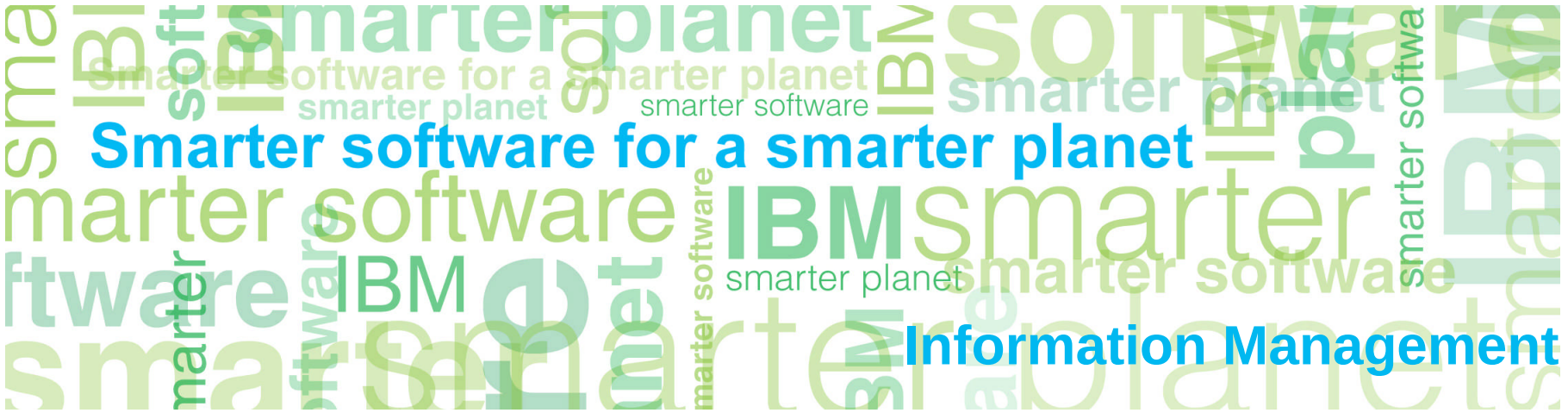
- **With DRAIN WRITERS, failure of subsequent read claims causes unnecessary impact to write claimers that have already been drained**

What parameters govern drain processing?

- **When do I try to drain?**
  - MAXRO
  - Uses previous log iteration to estimate time taken to process to end of log
  - Alterable with –ALTER UTILITY command
  - MAXRO DEFER will keep REORG in LOG phase until further notice
- **How long do I wait for the drain to succeed before giving up?**
  - DRAIN_WAIT
- **If drain fails, should I try again, and if so, when?**
  - RETRY & RETRY_DELAY

- **Log scan duration is not just MAXRO time**
  - It is MAXRO + time taken to drain + switch phase overhead
  - Hence recommendation is MAXRO+DRAIN_WAIT+overhead < IRLMRWT

Controlling Availability



Controlling availability

- **Outage duration is from time drain is requested to the time the drain is released**
- **Application claims will wait and may time out based on IRLMRWT**
- **Duration encompasses**
  - Time required to drain
  - Last log iteration in LOG phase
  - SWITCH phase
  - Start of UTILTERM phase prior to dedrain

How to control outage duration

- **Reduce time required to drain**
  - REORG at a quiet time
  - Have well-behaved applications that commit frequently
  - Have fewer pagesets in single REORG since processed serially
  - FORCE option in DB2 10
- **Reduce duration of last log iteration**
  - Reduce MAXRO
    - But if too short and high logging rate then REORG may never try for drain
  - Use inline Flashcopy instead of inline sequential copy
    - Eliminate cost of final incremental processing during last log iteration
  - Reduce number of pagesets processed in single REORG

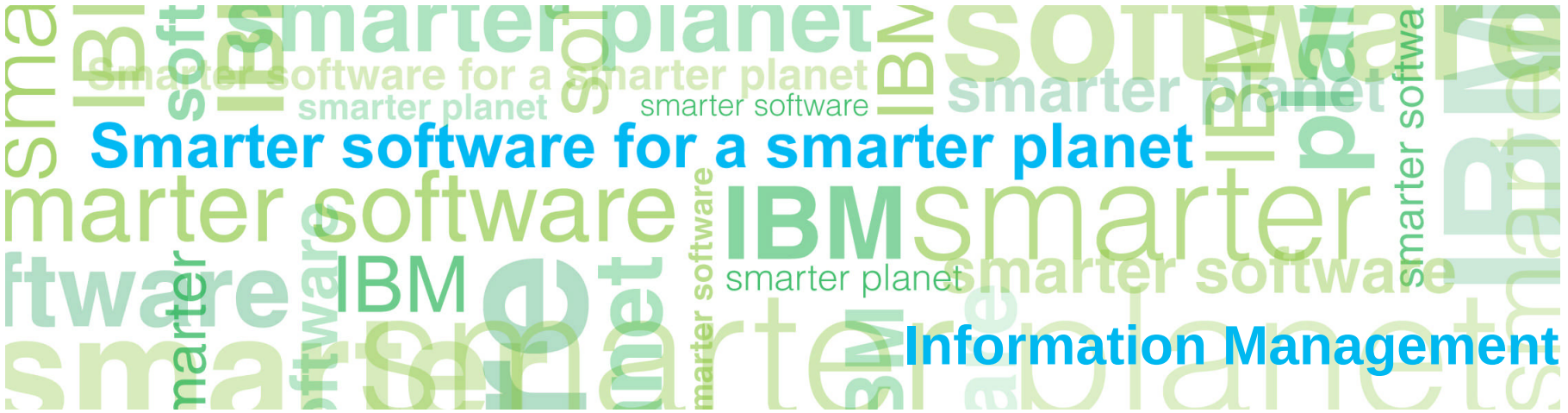
How to control outage duration

- **Reduce SWITCH phase duration**
  - Use inline Flashcopy instead of sequential copy
  - Reduce number of datasets processed by single REORG
    - Switch phase duration is dependent on number of datasets being processed
- **Reduce UTILTERM processing prior to dedrain**
  - Move to DB2 10
    - DB2 10 updates catalog statistics columns after dedraining
    - Or gather less inline statistics
    - Or reduce number of datasets processed by REORG

Controlling REORG completion

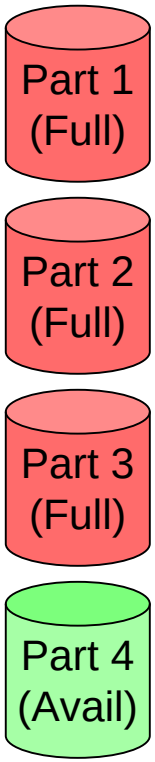
- **Set high MAXRO**
  - “Guarantees” drain will be attempted
- **Set high DRAIN_WAIT**
  - “Guarantees” drain success
- **Once drain is successful, no other parameters can prevent REORG completion**

REORG of PBGs



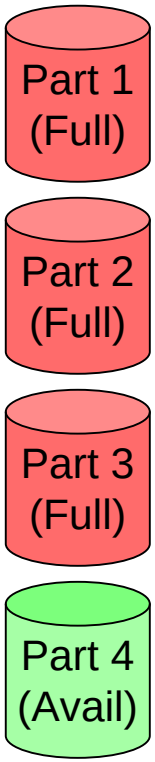
REORG of PBG

- **No 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

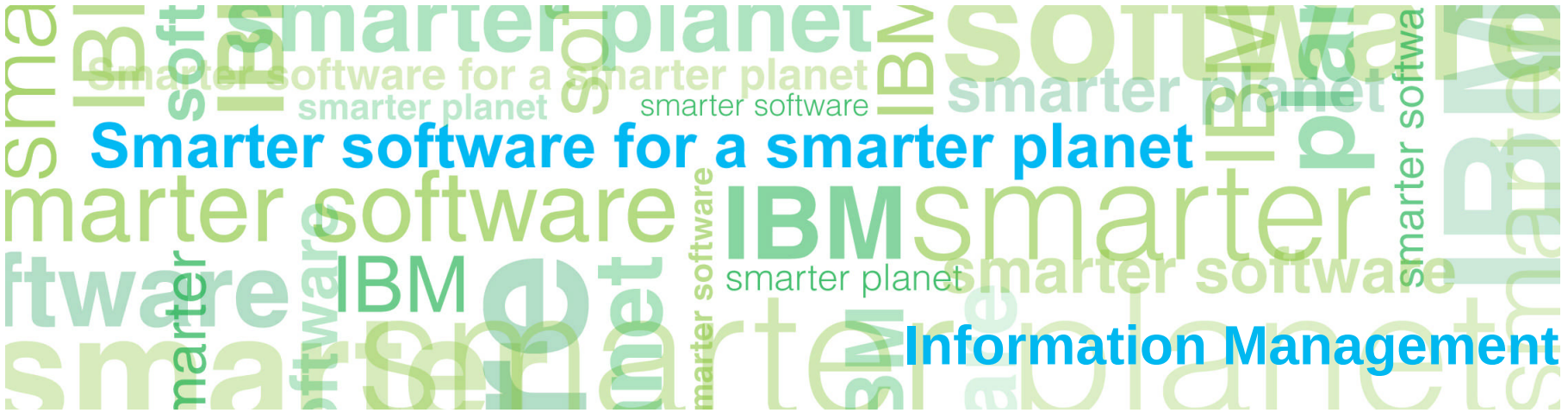


REORG of PBG

- **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 REORG_IGNORE_FREESPACE zparm on part-level PBG REORG
    - Introduced by PK83397 (V9) / PM53254 (V10)
  - 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**



REORG of LOBs

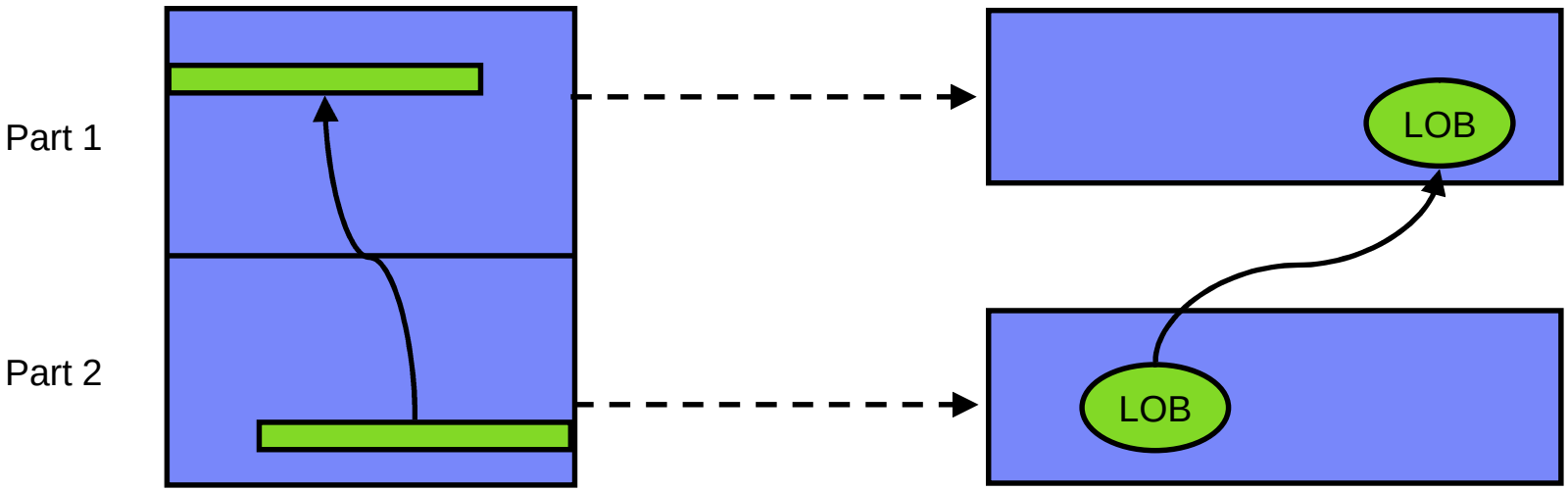


REORG & LOBs

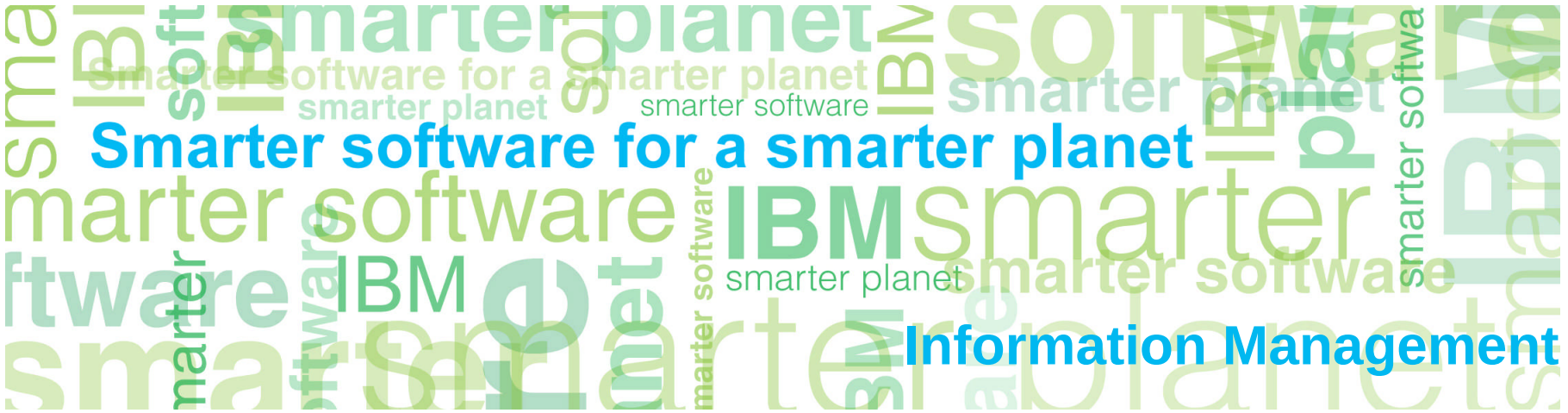
- **REORG of LOB table spaces**
  - V6 – V8
    - In-place REORG
    - SHRLEVEL NONE & LOG YES only
    - Incomplete rechunking of LOBs
    - No reclamation of physical space
    - No sort required
  - V9
    - SHRLEVEL REFERENCE
    - No logging
    - Perfect rechunking of LOB data
    - Reclaims physical space
    - No sort required
  - V10
    - SHRLEVEL CHANGE
    - No mapping table required
    - SHRLEVEL NONE deprecated in 10 NFM – will complete RC0 but no action is taken
    - Convert REORG jobs to SHRLEVEL REFERENCE or CHANGE before 10 NFM

REORG & LOBs

- **New AUX keyword on REORG of partitioned base for improved LOB handling in DB2 10**
  - Permit rows to flow between partitions
  - Supports REORG REBALANCE with LOB columns
  - Supports ALTER of LIMITKEY with LOB columns
  - Permits move of rows between parts on PBG REORG
  - Permits deletion of corresponding LOBs on REORG DISCARD
  - Default is AUX NO unless LOB objects required to complete REORG
  - No XML column support for classic partitioned or PBR
  - No mapping table change



What and when to REORG



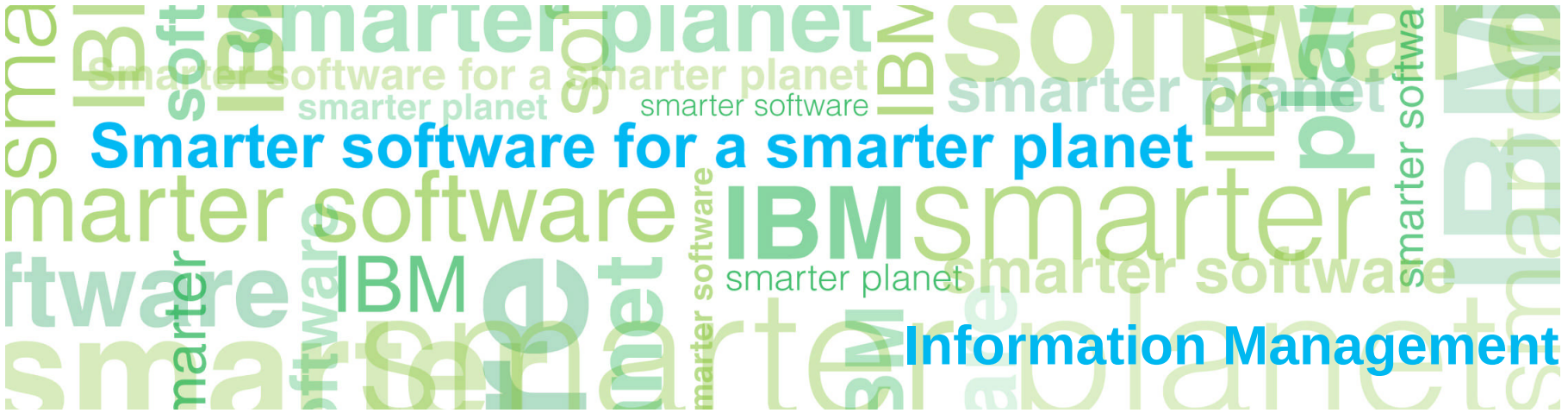
REORG avoidance

- **Why REORG if you don't have to?**
- **Use DSNACCOR/DSNACCOX, own application logic or automation tool to determine what to REORG and when**
- **If REORG purely to gain compression**
  - Consider LOAD COPYDICTIONARY instead
  - DB2 10 builds compression dictionary on the fly on insert
- **New REORGCLUSTERSENS RTS column in V10**
  - Do not REORG to regain clustering if no application benefit
- **Index leaf page list prefetch in V10 can reduce need to reorg indexes**

REORG INDEX vs. REBUILD INDEX

- **REORG is not a substitute for REBUILD, but REBUILD could be a substitute for REORG**
- **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
  - Note SHRLEVEL CHANGE means data is RW, not index – index is RBDP
- **REORG INDEX operates against a shadow – better availability than REBUILD INDEX**
- **REBUILD can be faster in V9, particularly if index is disorganized**
- **REORG INDEX performance improvement in V10 due to prefetch**
- **Summary:**
  - If you need to REBUILD then REBUILD
  - If you need to REORG on V9, and you can tolerate the index being RBDP, then REBUILD may be quicker
  - If you need to REORG on V10 then REORG

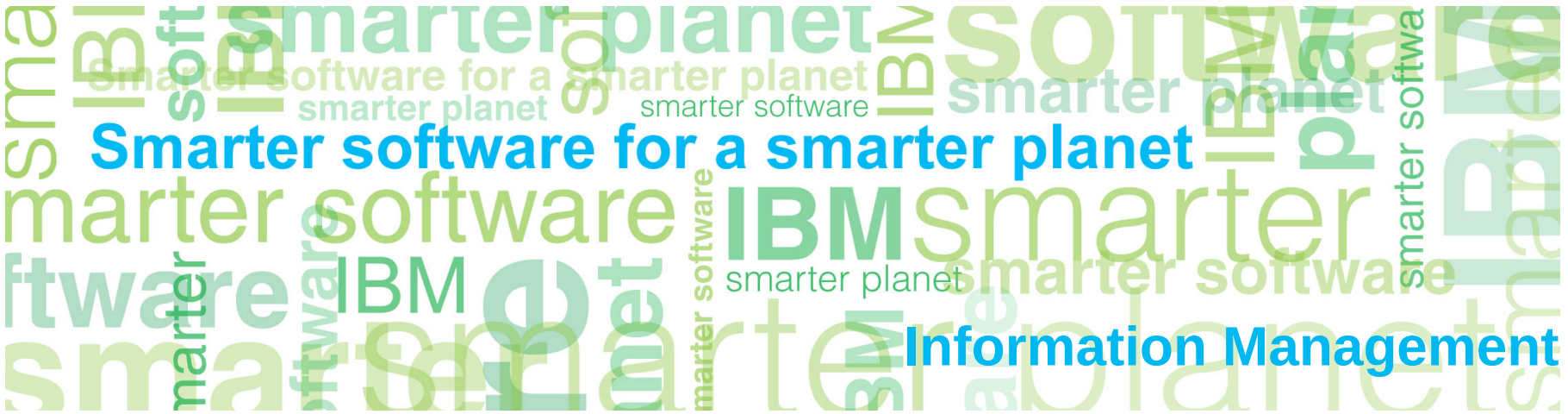
Summary of DB2 10 REORG Enhancements



REORG Enhancements in DB2 10

- **REORG SHRLEVEL CHANGE for entire catalog & directory**
- **REORG SHRLEVEL CHANGE for LOB table spaces**
- **Additional RTS information for intelligent REORG**
- **Partition-level REORG performance with NPSIs due to index leaf page list prefetch**
- **REORG INDEX performance due to index leaf page list prefetch**
- **REORG REBALANCE on partitioned tables with LOB columns**
- **ALTER LIMITKEY support for partitioned tables with LOB columns**
- **Ability to use SHRLEVEL REFERENCE or CHANGE to remove REORP state after alter of limitkey values**

Maintenance Stream Enhancements



Maintenance Stream Enhancements

- **SORTNUM elimination**
- **PK45916 (V8) & PK41899 (V9)**
- **Better performance, more robust, simpler**
- **SORTNUM no longer required**
  - Difficult to estimate: failure if too low, 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

Maintenance Stream Enhancements

- **Support REORG of multiple part ranges**
  - PK87762 & PM13259 (V9)
  - PM25525 (V9) – new PARALLEL keyword on REORG
  - PM37293 (V9) – new REORG_LIST _PROCESSING zparm
- **REORG online materialization of inline LOBs**
  - PM29037 (V10)
- **Utility support for spatial indexes**
  - PM35200 (V9)
- **Faster REORG with no NPIs**
  - PM37112 (V9)
  - Improve SYSLGRNX processing
- **zIIP enablement for REORG UNLOAD processing**
  - PM37622 (V9)

Maintenance Stream Enhancements

- **REORG sort performance improvements**
  - PM37630 (V9)
  - Reduce sort processing
- **Improve performance in last log iteration for outage reduction**
  - PM46632 (V9)
- **Extend limit for variable length sorts from 2bn to 4bn**
  - PM43006 (V9)
  - Applies to single data sorts in REORG with DFSORT
- **Auto-correction of BRF/RRF mismatches**
  - PM40646 (V9)
  - Corrects mismatch between data and catalog/directory after DSN1COPY

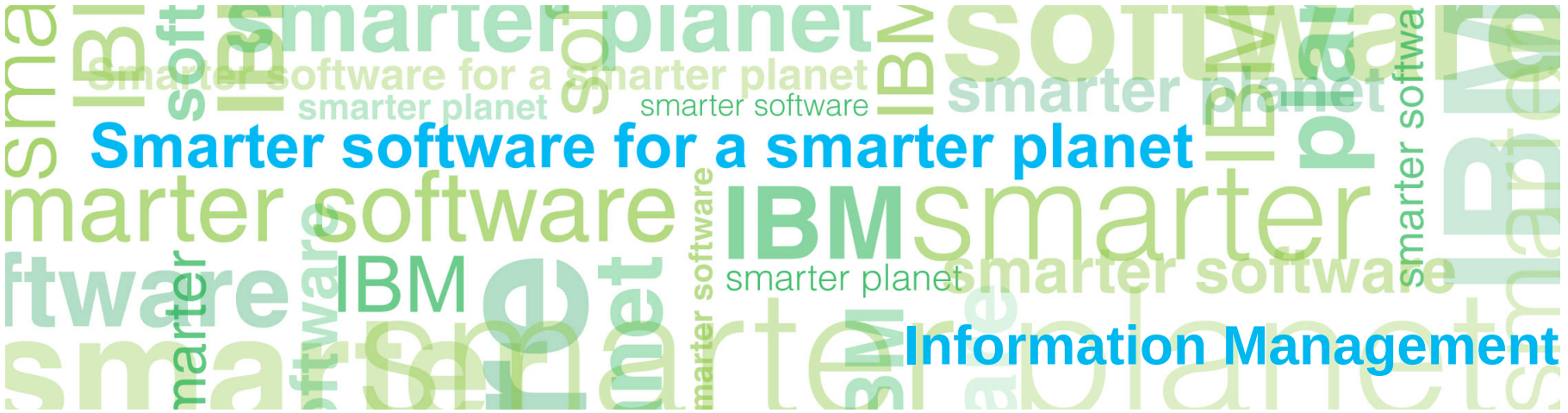
Maintenance Stream Enhancements

- **Remove requirement for sequential inline copy on REORG**
  - PM34776 (V10)
- **Improve REORG concurrency during online schema materialization**
  - PM25648 (V10)
- **Improve REORG performance in SORTBLD when using DPSIs**
  - PK92924 (V9)
- **New REORG_IGNORE_FREESPACE zparm for part-level PBG REORG**
  - PK83397 (V9), PM53254 (V10)

Maintenance Stream Enhancements

- **Part-level REORG performance with NPSIs**
  - PM55051 (V9)
- **CPU & ET reduction for REORG of multi-table tablespace**
  - PM52469 (V9)
- **Improve REORG dedrain processing for partitioned tablespace-level REORG**
  - PM56419
- **Improve scalability – allow mapping tables in PBGs**
  - PM58177 (V9)
- **Improve availability for PBG tablespace REORG in DB2 9**
  - PM61752 (V9)
- **Improve REORG DISCARD performance**
  - PM65550/PM73000 (V10)

Recommendations, Best Practices & Summary



Recommendations & Best Practices

- **Stay current on maintenance & DB2 releases and take advantage of new features**
- **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 disorganized, 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 disorganized**

Recommendations & Best Practices

- **REORG SHRLEVEL CHANGE for maximum availability**
- **Use DRAIN ALL rather than DRAIN WRITERS**
- **Use TIMEOUT TERM to free up objects on timeouts**
- **If minimizing 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**

Summary

- **The REORG utility is essential for data management and maintaining application performance**
- **As data volumes grow, it is critical that customers have confidence in the ability to manage that data**
- **IBM will continue to deliver availability, performance and functional enhancements to meet these needs**

Thank You for Joining Us today!

Go to www.ibm.com/software/systemz/events/calendar to:

- Replay this teleconference
- Replay previously broadcast teleconferences
- Register for upcoming events