DB2 Utilities Update and Best Practices

Speaker Name and Title

Agenda

- Overview
- REORG
- Statistics
- Backup and recovery
- UNLOAD and LOAD
- Compression dictionaries
- General enhancements
- Deprecated items
- Additional information
- Summary

Overview: goals of DB2 Utilities

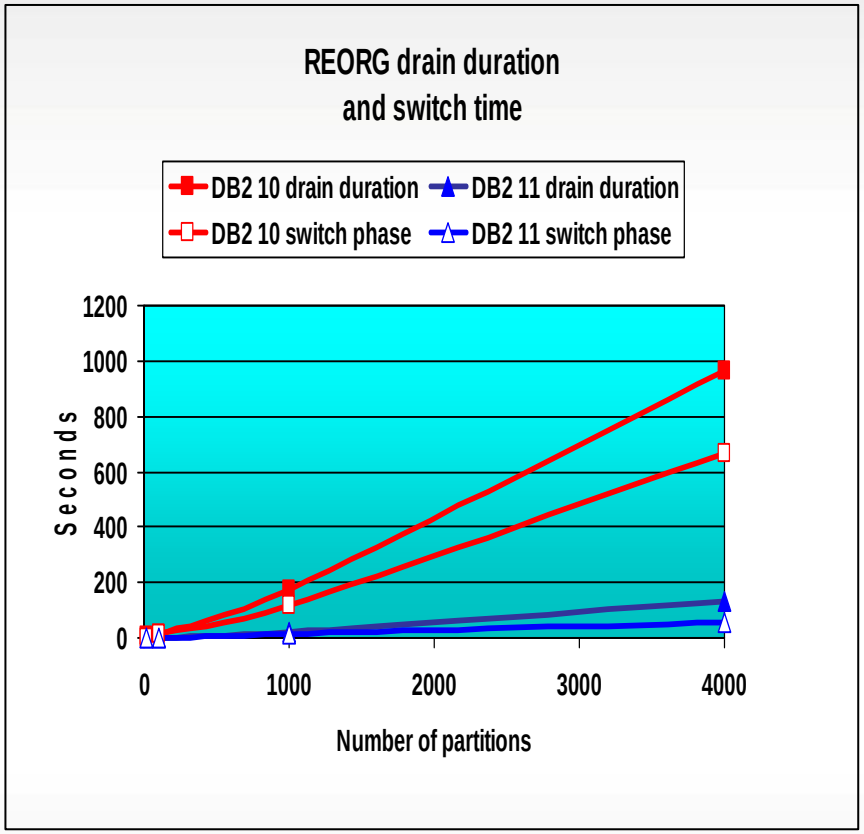
- Support core DB2 functionality
- Reduce CPU, elapsed time and resource consumption
- Maximize availability
- Remove constraints and limitations
- Simplify data management

Improve performance of part-level online REORG with NPSIs

- New option to defer shadow index build until all keys passed through sort
- New utility option and ZPARM to govern
  - AUTO/YES/NO values
- Retrofit to DB2 9 and 10 via fix for APAR PM55051
- Result:
  - Customer test of REORG of 40% of partitions showed 55% elapsed time reduction and 22% CPU increase
  - When used in the same online REORG scenario, DB2 Sort provides additional elapsed time reduction and cuts CPU cost to less than original starting point

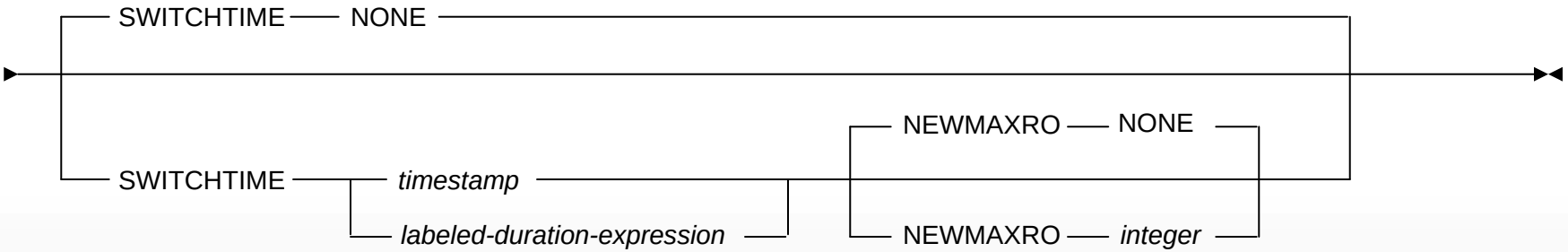
SWITCH phase impact relief – reduced application impact

- Easier drain acquisition
- Prevent new claims on all target partitions while waiting for drains on target partitions
  - Faster drain acquisition for partition-level REORG
- New DRAIN_ALLPARTS option to momentarily drain all data parts
  - Eliminates claim-drain “deadlocks” for partition-level REORG with NPSIs
- SWITCH phase processing restructured for outage reduction
  - SWITCH phase ET reduction of 91% measured when REORGing 20 partitions



Timing of SWITCH phase with MAXRO DEFER

- Govern timing of drain and switch for long-running REORGs without the need to schedule separate -ALTER UTILITY command
- New SWITCHTIME parameter to determine earliest point at which drain processing will be attempted



Physically delete empty PBG partitions

- Ability for REORG to physically delete empty PBG partitions
- New ZPARM: REORG_DROP_PBG_PARTS
  - DISABLE – keep V10 behavior (default)
  - ENABLE – delete empty PBG partitions on table space-level REORG
- Considerations:
  - Cannot be specified on REORG statement
  - If PBG created using NUMPARTS or if ALTER ADD partition used, REORG may prune to a lesser number of partitions
  - No point-in-time recovery to prior to a partition-pruning REORG
    - No facility to resurrect deleted partitions

Automated handling of mapping tables

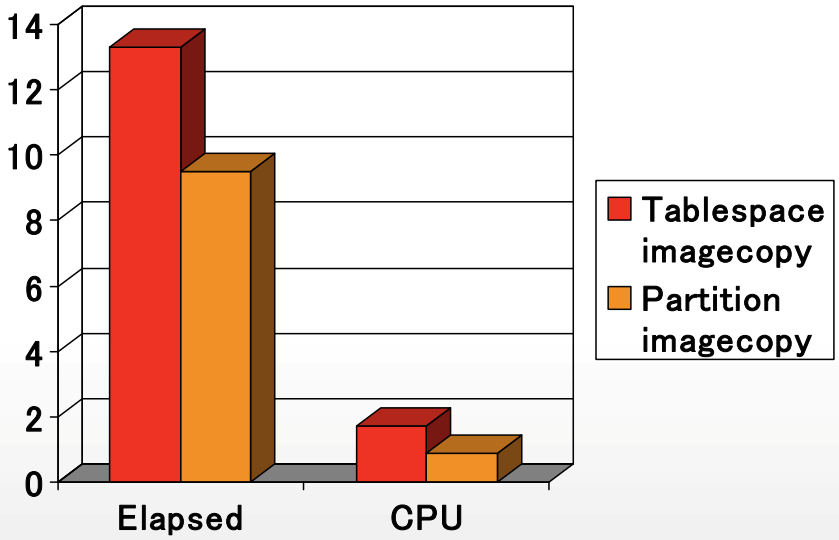
- Prior to DB2 11, scalability constraint when size of mapping table index limits the number of rows that can be REORGed
- Mapping table DDL must change in 11 due to longer RBA/LRSN
- Requirement to automate handling of mapping tables
- Solution: New automated mapping table functionality in REORG
  - Support for mapping tables in partition-by-growth table spaces
    - Increases max size of mapping index from 64GB to 16TB
      - Retrofit to DB2 9 and 10 via APAR PM58177
  - Automatically create new-format mapping table if required
    1. If mapping table specified and correct format then honor specification
    2. Else if specified but incorrect format then create new in same database as original
    3. Else if not specified and ZPARM database specified then create in that database
    4. Else create in implicitly-created database
    5. DROP at end of REORG or end of last REORG if multiple REORGs in job step
  - NFM requires new-format mapping table
    - CM, CM*, ENFM, ENFM* support old- and new-format mapping table

REORG without sorting data

- Increasingly REORGs are performed for reasons other than to regain clustering of data, yet no ability to avoid cost of reclustering
- Before DB2 11, REORG SHRLEVEL CHANGE did not support SORTDATA NO
- DB2 11 allows SORTDATA NO with SHRLEVEL CHANGE
- New RECLUSTER YES/NO option on SORTDATA NO
  - RECLUSTER NO – Do not unload data through clustering index and do not sort data records in clustering order

Partition-level inline image copy

- Faster partition-level recovery from inline image copy
- Create partition-level inline image copies if using TEMPLATE with &PA or &PART
  - No new option or keyword on REORG
  - PM93611:
    - Support substring notation with &PA and &PART as long as it ensures uniqueness
    - Support writing to tape as long as STACK YES not specified
- RECOVER of single partition of a 20-partition table space
  - ET reduced by 28%
  - CPU reduced by 49%



Improved REORG LISTDEF processing

- PARALLEL YES/NO option introduced via APAR in DB2 9
  - NO – Prevent REORG from processing multiple partitions in parallel in a single REORG when input is partition-level LISTDEF
  - ZPARM: REORG_LIST_PROCESSING
- Need compromise option for customers who want to take advantage of REORG parallelism but cannot afford to shadow many partitions at a time
- New option LISTPARTS n to limit number of partitions to be processed in a single REORG if input is a partition-level LISTDEF
- With DB2 11, PARALLEL YES/NO is replaced by PARALLEL(n), with “n” specifying the maximum number of subtasks to be started in parallel for a REORG

REBALANCE enhancements

- Improved availability & failure prevention
- Support REORG SHRLEVEL CHANGE REBALANCE
  - Complements online ALTER of partition limit key value
- Improve resiliency with enhanced distribution algorithm and improved handling of empty partitions
- Build compression dictionary for all partitions
  - Previously, partitions that were empty at the start of REORG would not have a dictionary built, requiring a subsequent REORG to get compression
- New SORTCLUSTER option to sort data in clustering as well as partitioning order to avoid AREO*
  - Occurred when partitioning key not a superset of clustering key

REORG of LOB data

- Support REORG of LOB data even if aux index is unavailable
  - Problem in DB2 10 if LOB tablespace is REORP and index is RBDP
    - LOBs can’t be REORGed and index can’t be rebuilt
- REORG SHRLEVEL NONE for LOBs changed to RC8 from DB2 11 CM onwards
  - Not supported in 10 NFM, but returns RC0 with MSGDSNU126I

Improved REORG serviceability

- Need ability to use online REORG even when SYSLGRNX cannot be relied upon
- Support LOGRANGES NO option for REORG SHRLEVEL CHANGE

REORG change of defaults to match best practices

- Change default options:
  - DRAIN WRITERS to DRAIN ALL
  - DISCARD to DISCARD NOPAD YES
  - UNLOAD EXTERNAL to UNLOAD EXTERNAL NOPAD YES

REORG 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 and MAXRO to guarantee REORG success
- If using REORG DISCARD, use NOPAD for improved performance
- LOBs:
  - SHRLEVEL REFERENCE in DB2 9, SHRLEVEL CHANGE in DB2 10
  - Stop using SHRLEVEL NONE before DB2 10 NFM

Statistics enhancements

- More zIIP offload for RUNSTATS distribution statistics
  - Up to 80% zIIP-eligible
- zIIP offload for inline statistics
  - Up to 30% offload to zIIP
- Enhance inline statistics for RUNSTATS avoidance
  - Inline statistics collection on NPSIs during REORG with SORTNPSI
  - Inline histogram statistics
  - Inline COLGROUP distribution statistics
- New RUNSTATS RESET option to reset existing statistics
- Improved PROFILE usability for LISTDEF processing
  - Gather default statistics if no profile exists for table

Statistics enhancements

- Optimizer determination of missing statistics
  - Optimizer identifies missing statistics and writes information to new catalog table SYSSTATFEEDBACK
  - OQWT modifies statistics profile
  - Automation Tool detects profile change and builds RUNSTATS job
  - DSNACCOX similarly enhanced to recommend RUNSTATS
- -ACCESS DATABASE … MODE(STATS) option to externalize RTS statistics
- RTS overhead reduction

Statistics best practices

- Do not use RUNSTATS to gather space statistics – rely on real-time statistics
- Do not gather unnecessary statistics
- Use sampling, and TABLESAMPLE in DB2 10
- Use profiles in DB2 10 and 11
- Use inline stats where possible rather than RUNSTATS (but no zIIP for inline stats in DB2 10)
- Specify KEYCARD prior to DB2 10
  - Index cardinality stats are cheap to collect and heavily relied upon by optimizer
- Don’t bother running RUNSTATS on LOB table spaces
  - RTS contains all the information you need

Backup and recovery enhancements

- Faster directory recovery
  - SYSLGRNX recording extended to previously not-recorded directory objects
- New VCAT name translation for RESTORE SYSTEM for system cloning
  - Support log apply when RESTORE SYSTEM used for cloning purposes
- Improved recoverability with COPY/REORG concurrency
  - Permit COPY to run concurrent with long-running REORGs
- Avoid allocating empty image copy datasets for incremental or CHANGELIMIT copies
- Lifted many restrictions on PIT recovery prior to materializing REORG
  - PIT recovery restrictions lifted for
    - LOB, XML and PBR table spaces
    - Including when immediate alters have occurred since materializing REORG
  - PIT recovery restrictions still in place
    - Table space conversion
    - PBG table spaces
    - PBG partition pruning
    - Online DROP COLUMN

Backup and recovery best practices

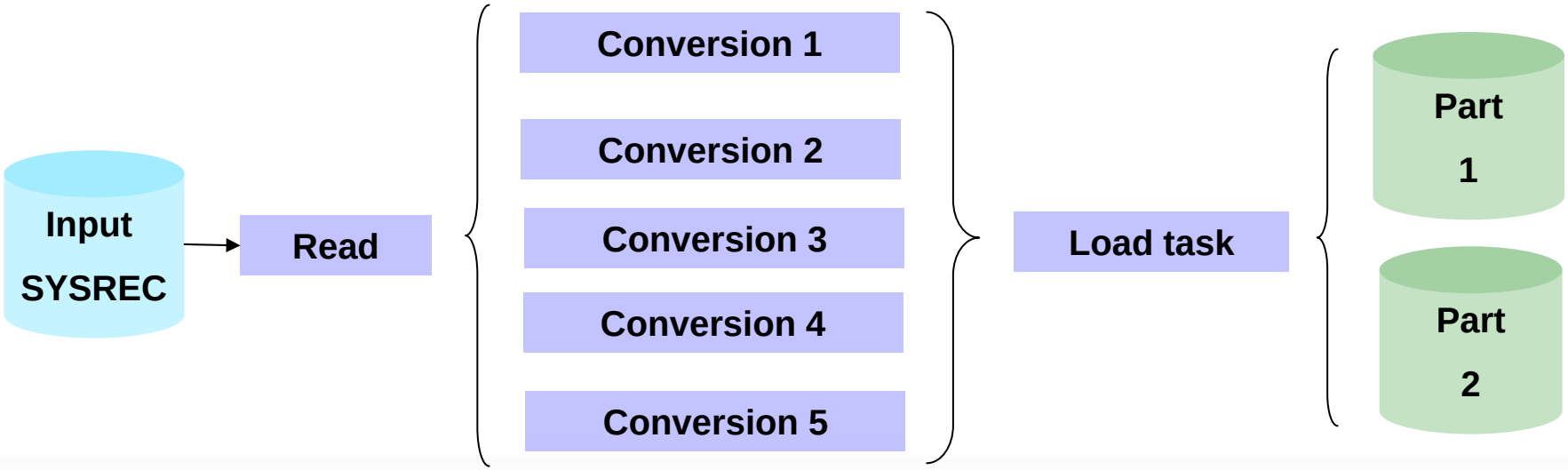
- SHRLEVEL CHANGE unless consistent copies are essential
- Use PARALLEL keyword to exploit parallelism
- Consider OPTIONS EVENT(ITEMERROR,SKIP)
  - Sets UTRW state only for duration of copy of individual page set
  - But increases COPY overhead
  - Serialization required for each pageset on the fly
- 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 and SYSCOPY recording
  - Automatically included in MODIFY RECOVERY

LOAD and UNLOAD enhancements

- Cross-loader support for XML data
- Exploit FETCH CONTINUE for processing large LOBs and XML data with cross-loader
  - Reduce virtual storage requirement
  - Avoid DSNU1178i errors
  - 28% CPU reduction
    - Load of 1Mb LOBs
- zIIP offload for LOAD REPLACE PART clearing of NPSIs
  - 100% offload to zIIP for LOAD REPLACE with dummy input

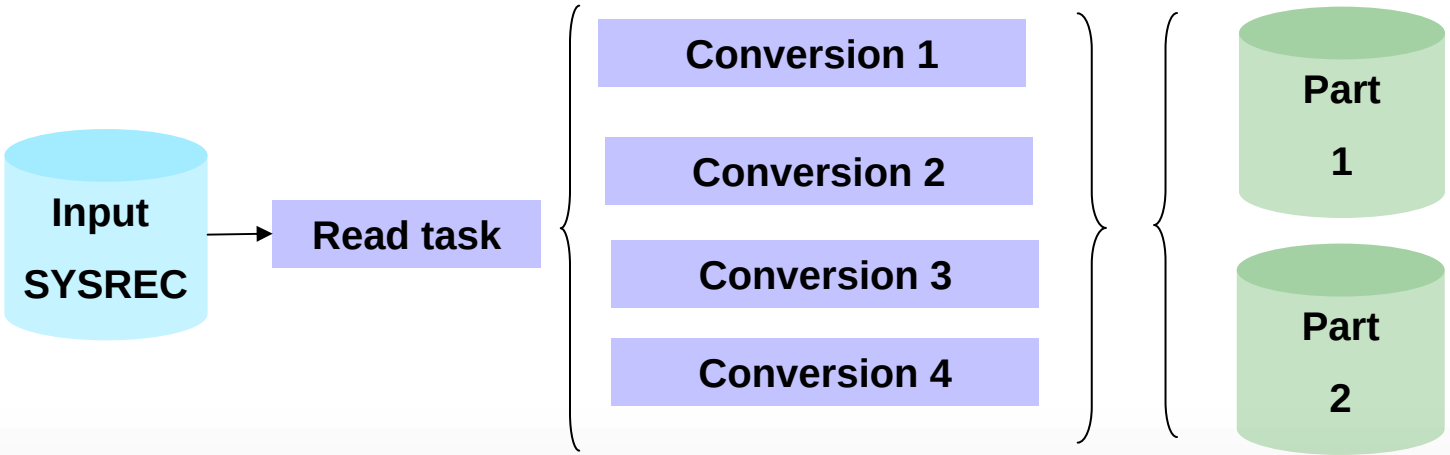
LOAD and UNLOAD enhancements

- LOAD SHRLEVEL NONE PARALLEL with single input dataset
  - Parallel data conversion
  - Not supported for PBGs
  - 50% ET reduction possible on single SYSREC load



LOAD and UNLOAD enhancements

- LOAD SHRLEVEL CHANGE PARALLEL
  - Supports non-partitioned as well as partitioned
  - Single input dataset
  - Not supported for PBGs
  - >80% ET reduction



LOAD and UNLOAD best practices

- Run LOAD with LOG NO, REUSE, KEEPDICTIONARY if possible
- Use SORTDEVT to drive parallel index build
- 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 only a fraction of input records will be loaded
- If loading partitioned table with single input dataset, presort data in clustering (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
- Use FORMAT INTERNAL, PRESORTED or INDEXDEFER if possible

Compression dictionaries

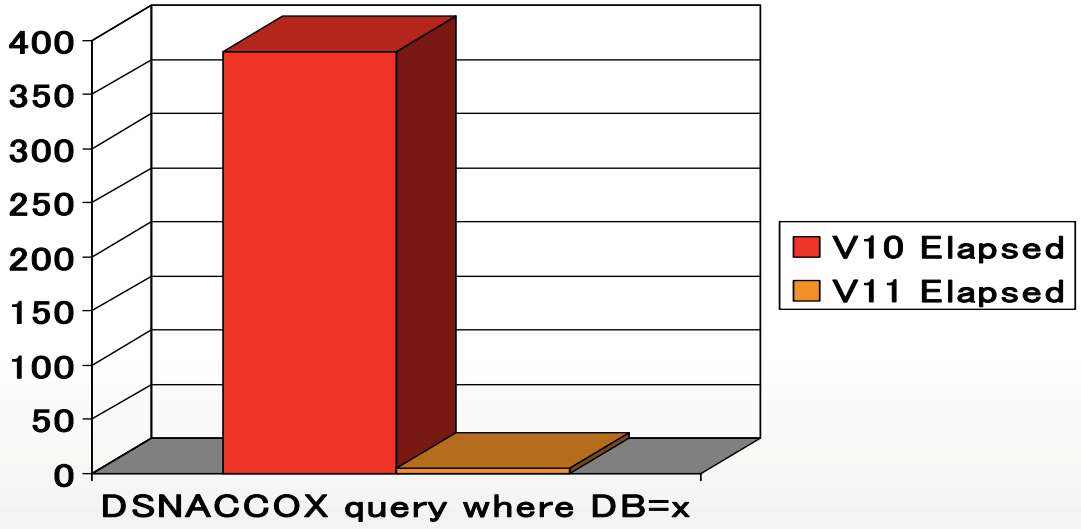
- Avoid decompression failures for IFI 306 readers when new compression dictionary built by REORG/LOAD
- Old compression dictionary stored on log
- New SYSCOPY record written pointing to old compression dictionary for CDC tables
- IFI 306 read automatically retrieves old compression dictionary if necessary
- Avoid need for replication target refresh when dictionary changes

General enhancements

- Greater parallelism for faster utilities
  - 11% elapsed time reduction measured for REORG, LOAD, REBUILD INDEX
- PARALLEL option for parallelism control for LOAD, REORG, REBUILD INDEX, UNLOAD, CHECK INDEX
- -DISPLAY UTILITY enhancements
  - Remove serialization between -DIS UTIL and -TERM UTIL
  - Jobname, start timestamp
  - Late addition: SWITCHTIME and NEWMAXRO
- Utility impact reduction on buffer pools
  - Extend MRU for UNLOAD, REORG TABLESPACE, RUNSTATS TABLESPACE, RUNSTATS INDEX, REBUILD INDEX, CHECK INDEX, CHECK DATA
- Improved dataset cleanup with utility stored procedures
  - Previously, datasets remained allocated on utility failure, preventing cleanup

General enhancements

- Improved TEMPLATE support for large / extended format data sets and local time values
  - DSNTYPE LARGE, EXTREQ, EXTPREF
  - New EATTR option on TEMPLATE to request extended attributes
  - New TIME LOCAL|UTC option
- Enforce NUMTCB=1 for utility stored procedures
- DSNACCOX performance



Deprecated items

- REORG SHRLEVEL NONE for LOBs changed to RC8 from DB2 11 CM onwards
  - Not supported in 10 NFM, but returns RC0 with MSGDSNU126I
- Still supported in 11, but no longer documented:
  - REORG
    - PARALLEL YES|NO
      - Superseded by LISTPARTS
    - INDREFLIMIT
    - OFFPOSLIMIT
    - LEAFDISTLIMIT
    - UNLOAD ONLY
    - UNLOAD PAUSE
    - UNLOAD EXTERNAL
  - COPY
    - CHANGELIMIT

Want to learn more about DB2 utilities?

- Attend a DB2 Utilities Workshop
  - What is it?
    - Complimentary (1/2 day+) session with information to help gain a better understanding of the features delivered in the DB2 Utilities Suite for z/OS and how they can be used to manage your DB2 environment. Topics include:
      - Which utility features can save you time and money
      - Understanding your utility maintenance needs to meet SLAs
  - What's the benefit to you?
    - Useful information, materials and contacts help you achieve the maximum benefits from the features delivered in the DB2 Utilities Suite for z/OS and how they can help manage your utility maintenance to improve your DB2 for z/OS critical applications
  - Who is it designed for?
    - DBAs/App DBAs, DB2 system administrators, and technical management
  - Contact your IBM sales rep for more details and to schedule

Best practice tip for DB2 utilities: the DB2 Utilities Solution Pack

"More data, reduced costs"

- During DB2 utilities sort processing, reduce CPU usage and elapsed time by up to 50%*
- Eliminate ALL CPU and elapsed time by avoiding unnecessary utility processing
  - Set it and forget it, adjust to fit changing needs
- Use the fastest and most flexible unload in the market to modernize your data movement
- Set company-wide DB2 utility standards
  - Enforced and auditable
- ONE change to embedded DB2 Utility JCL can find and update every occurrence in every application with the most current version of DB2 for z/OS

Optimize, control manage & automate

When used with the IBM DB2 Utilities Suite, IBM beats or matches performance with every other 3rd party vendor for:

- *Using less CPU*
- *Reducing ET*
- *More zIIP offload*

DB2 Utilities Solution

**** With zIIP engine***

General best practices

- Plan your move to DB2 11 to take advantage of the latest in DB2 11 enhancements
- Stay current on DB2 maintenance as much as possible
- Attend a DB2 Utilities Workshop
- Check out the DB2 Utilities Solution Pack to make your DB2 utility environment more efficient and easier to manage

Summary

- Day One support for utilities and utility management products with core DB2 versions/functions
- Innovation continuing and delivery pace accelerating
- Continuous delivery of performance enhancements and features with real business value
- Eliminate application impact from utilities
- Reduce elapsed time and CPU consumption
- Reduce resource consumption

- Reduce complexity and improve automation -- Build expert knowledge into the tools, not just tools for the expert!