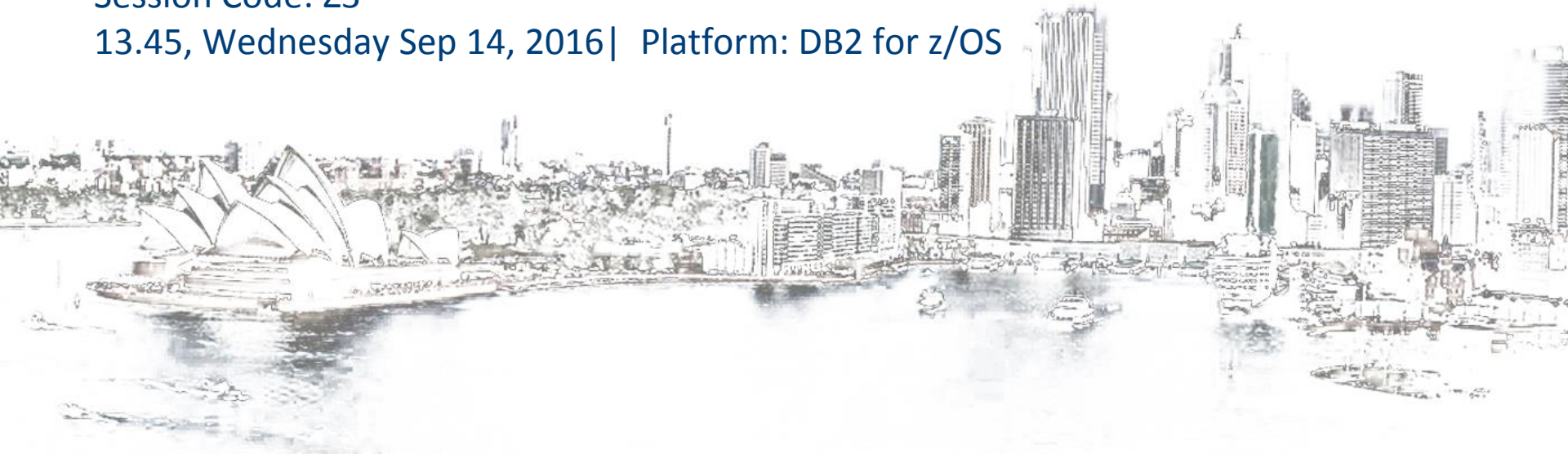


# DB2 12 for z/OS - IDUG User Experiences

**Ron Brown**  
*Ronek Consulting*

Session Code: Z3

13.45, Wednesday Sep 14, 2016 | Platform: DB2 for z/OS



## Topics

- Introduction
- IDUG DB2 12 for z/OS Technical White Paper
- User Experiences
  - Developer
  - DBA
  - Systems Programmer
- Summary

# IDUG DB2 12 for z/OS Technical White Paper

- Objectives
  - Create **technical** white paper to provide detailed examination of DB2 12 features, focusing on how they work and how best to implement them
- Target Audience
  - DBAs, developers and systems programmers within existing DB2 for z/OS customers considering upgrading their DB2 systems to DB2 12
- Approach
  - Run alongside formal ESP program
  - Access to ESP training materials, lab experts and early DB2 12 code
- IDUG volunteers on White Paper Editorial Team
  - Ron Brown
  - Joe Geller
  - Cuneyt Goksu
  - Dan Luksetich
  - Bjarne Nelson
  - John Maenpaa
  - Cristian Molaro
  - Kurt Struyf
  - Julian Stuhler
  - Billy Sundarrajan
  - Isaac Yassin

# Disclaimers

- The IDUG DB2 12 for z/OS Technical White Paper (and this presentation) were developed during the DB2 12 ESP, and therefore all new features and functions covered are subject to change
- This presentation addresses a selection of the features and findings within the white paper
  - We'll concentrate on impact of new features rather than how it works
  - There are many more new features in DB2 12 than we have time to discuss here
  - Go download the full paper to learn more!
- The editorial team concentrated on “real world” use of the new features, so we've frequently concentrated on feature limitations / restrictions
  - Not being deliberately negative, just pragmatic 😊
- All references to “DB2” and “DB2 12” refer to IBM DB2 12 for z/OS

# User Experiences – Developer

- **SQL Enhancements – Enhanced Merge**
  - What?
    - Extend MERGE capability to allow DELETE of existing row
  - Experiences
    - Enhanced MERGE matches DB2 for LUW capability and greatly expands use-cases for MERGE (e.g. to allow staging tables to be merged)
- **SQL Enhancements – XMLMODIFY**
  - What?
    - Enhance XMLMODIFY to allow multiple updates
  - Experiences
    - Significant improvement to both usability and performance compared to individual updates – especially apparent on 5+ updates

# User Experiences – Developer

- SQL Enhancements – Data Dependent Pagination
  - What?
    - New pagination syntax (aka data-dependent pagination)
    - `WHERE (KEY1 = :LAST_KEY1 AND KEY2 > :LAST_KEY2 OR KEY1 > :LAST_KEY1)` becomes `WHERE (KEY1, KEY2) > (:LAST_KEY1, :LAST_KEY2)`
  - Experiences
    - Much easier to code and more intuitive
    - Optimiser will rewrite data-dependent syntax to old OR-based approach so that efficient Range List index access (ACCESSTYPE=NR) can still be used

# User Experiences – Developer

- **SQL Enhancements – Numeric Pagination**
  - What?
    - New OFFSET clause to tell DB2 to skip n rows before fetch
  - Experiences
    - Provides compatibility with other RDBMS (including DB2 for LUW)
    - Simpler code, but performance not as good as data-dependent pagination, especially as number of skipped rows increases (access path similar to pre-NR behaviour)
- **SQL Enhancements – Piecemeal DELETE**
  - What?
    - Simple control of COMMIT scope for DELETE statements via FETCH FIRST n ROWS ONLY
  - Experiences
    - Allows simplicity of a simple searched DELETE while allowing impact of lock/log to be limited
    - Number of rows can be specified via a host variable, lots of flexibility

# User Experiences – Developer

- SQL Query Performance Enhancements – UNION ALL and Outer Join
  - What?
    - Query rewrite enhanced for some types of queries involving outer joins and UNION ALL, may now be able to correlate predicates inside nested table expressions and views
  - Experiences
    - Common use cases include UNION ALL in views/nested table expressions, archive table queries and system-period temporal time travel queries
    - Dramatic potential performance improvement, can improve index access, reduce materialization of nested table expressions and view, and eliminate redundant table access
    - Query rewrite Enhancement is for outer join queries only, so inner joins must either be transformed to outer joins or tuned via other means



# User Experiences – Developer

- SQL Query Performance Enhancements – Improvements to function and expression processing
  - What?
    - Caching of deterministic user-defined function results, performance improvements for CASE statements and the SUBSTR scalar function, sharing of data calculated from expressions embedded in nested table expressions
  - Experiences
    - UDF caching could reduce UDF time up to 99% for low cardinality inputs
    - 20-70% CPU reductions for CASE expression xPROCs
    - 5-20% for SUBSTR xPROCs (depending on data type)

# User Experiences – Developer

- **SQL Query Performance Enhancements – Pushdown of FETCH FIRST with ORDER BY**
  - What?
    - ORDER BY FETCH FIRST clause coded against a result set generated by a view or nested table expression that contains a UNION ALL can be pushed down into the view or nested table expression
  - Experiences
    - Having ORDER BY within each UNION ALL leg allows the optimizer to use an available index to avoid an ORDER BY sort and combined with FETCH FIRST, fetch only those rows from that index
    - If no index exists, DB2 can use the optimized sort processing for FETCH FIRST that has been enhanced since DB2 9 for z/OS
    - Early filtering of the result data and a significant performance improvement over previous releases of DB2 for z/OS

# User Experiences – Developer

- SQL/PL Enhancements – Advanced Triggers
  - What?
    - Advanced Triggers can support many more SQL statements, logic programming capability, SQL variables, the ability to debug triggers, and versioning
  - Experiences
    - Greatly improves portability / compatibility with other RDBMS (including DB2 for LUW)
    - SELECT privilege on underlying table always required (not so with basic triggers)
    - No support for stand-alone full Select and Values statements, need to use SELECT INTO or VALUES INTO statements
    - More robust error handling and retry capabilities are now possible
    - Trigger versioning very similar in approach to SQL/PL stored procedures

# User Experiences – Developer

- SQL/PL Enhancements – Obfuscated Stored Procedures and Triggers
  - What?
    - WRAP function provides a mechanism to ship SQL PL routines in a binary format that can be executed by DB2 without exposing the inner workings of those routines
  - Experiences
    - Ideal for vendors who want to take advantage of power of SQL/PL, but would also like to protect their intellectual property
    - Only the routine logic is obfuscated
    - Obfuscated routines have to work cross-platform so strong encryption not used – deterrent not a 100% secure defence

# User Experiences – DBA

- **Dynamic Plan Stability**
  - **What?**
    - Extend plan stability advantages of static SQL to also include dynamic SQL
    - Access path for “stabilized” SQL stored in SYSIBM.SYSDYNQRY catalog table following START DYNQUERYCAPTURE command
  - **Experiences**
    - Can stabilize based on specific statement ID, or various thresholds
      - If using thresholds then multiple statements may be stabilized and it will not be as obvious which are captured – query SYSIBM.SYSDYNQRY to check
    - Some restrictions apply – no support for concentrated literals or temporal or transparent archive tables

# User Experiences – DBA

- Partition Enhancements
  - What?
    - New relative page numbering option for UTS tablespaces with expanded 7-byte RID
    - Allows for partitions up to 1TB with total size of up to 4PB, with partition-level DSSIZE
    - Allows partitions to be inserted into a PBR tablespace
  - Experiences
    - Data sets for the table space must be associated with a DFSMS data class that has been specified with extended format and extended addressability
    - PIT recovery prior to materializing REORG is restricted
    - For insert partition, the new physical partition is added at the end (e.g. A00n+1) but the new logical partition is added in the middle (following logical partition numbers are renumbered)

# User Experiences – DBA

- Fast Insert Algorithm
  - What?
    - New “insert algorithm 2” for UTS Member Cluster tablespaces promises reduced page contention by using asynchronous page pre-formatting and in-memory pipes
  - Experiences
    - Increase in INSERT throughput with modest CPU reduction
      - May just move bottleneck to another area (e.g., log write, index leaf page latch)
    - Additional real storage will be required for in-memory pipes (approx. 40k per partition per member)
    - New stats/accounting/performance counters for piped insert (wait time, # of events, FIA disabled, etc.)

# User Experiences – DBA

- Temporal Enhancements
  - What?
    - Series of enhancements including: Temporal auditing, Temporal logical transactions, Temporal inclusive/inclusive model, Temporal RI
  - Experiences
    - Inclusive/inclusive model makes it easier for a company to migrate from a home grown solution to DB2 temporal tables if that home grown solution used the inclusive / inclusive model
    - RI support is comprehensive enough to eliminate home-grown alternatives such as key tables, triggers / stored procedures, etc.
    - Temporal logical transactions are powerful, but fraught with potential dangers, use with care!



# User Experiences – DBA

- Utility Enhancements
  - What?
    - Many! Including: new UNLOAD privilege, new DROP\_PART keyword on REORG to remove empty PBG partitions, PBG REORG can overflow to new partition, LOAD RESUME YES...BACKOUT YES, profile support for inline stats on LOAD and REORG, MODIFY...NOCOPYPEND, DSN1COMP LOB support
  - Experiences
    - New privilege required for UNLOAD as default so this is an incompatible change – need to change ZPARM to keep old behaviour
    - DROP\_PART will not remove first partition (A001) even if it's empty
    - If no profile exists when USE PROFILE specified on LOAD or REORG, default stats are still collected
    - zEDC card is needed for DSN1COMP estimation – Catch 22 when building a business case...

# User Experiences – DBA

- Security and regulatory compliance
  - What?
    - Transfer object ownership
    - Install/upgrade using Install SYSOPR (which cannot read application data)
  - Experiences
    - TRANSFER OWNERSHIP will make migration to role-based security much more straightforward / feasible – especially for older DB2 systems
    - Some limitations around use of INSTALL SYSOPR that will need careful planning / consideration (e.g. can't use implicit creator on CREATE TABLE/INDEX, other authids needed for all DDL and BIND OWNER, new plan for DSNTIAD ) – not going to be a “quick win”

# User Experiences – Systems Programmer

- Installation, Migration and Activating New Functionality
  - What?
    - Install dialog same as previously but zOSMF can be used
    - Migration modes only BNFA and ANFA, invoked via the new command:  
–ACTIVATE NEW FUNCTION
    - **Continuous Delivery** of maintenance will enable future new functionality to be activated using APPLCOMPAT levels along with Function Levels (and associated Catalog Levels) within a DB2 version much less disruptively than the traditional 3-year cycle of DB2 version upgrades
  - Experiences
    - Familiar dialog is best option and zOSMF offers no advantage
    - CM, ENFM, NFM gone. Simpler activation of new function for DB2 12

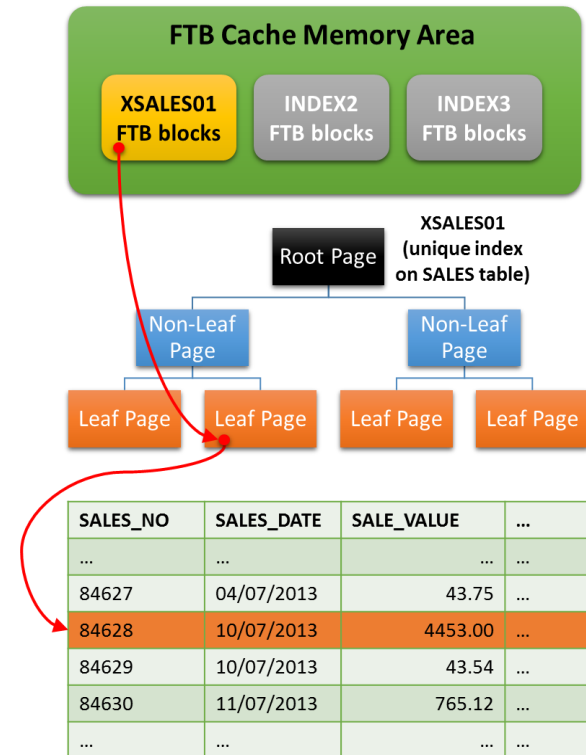
# User Experiences – Systems Programmer

- **Contiguous Buffer Pools**
  - **What?**
    - Extension to PGSTEAL(NONE) that physically lays out pages in memory in the same order as they reside on disk
    - Provides CPU savings via reduction in GETPAGE/RELEASE PAGE overheads and elimination of hash and LRU chain management
  - **Experiences**
    - Sweet spot as for previous PGSTEAL(NONE) candidates – smaller, commonly accessed tables within OLTP workloads
    - Special overflow area defined for pages that won't fit in pool, so need to allocate additional real storage (10% of pool size, up to max of 6400 pages)
    - New DSNB604I issued when overflow area is used – good to trap this message!

# User Experiences – Systems Programmer

- **Fast Traverse Blocks**
  - **What?**
    - New memory area used to cache non-leaf index data for commonly-used random lookups via unique indexes
    - Access to a cached index needs just one GETPAGE for the leaf page
    - Available out of the box (once activated)
  - **Experiences**
    - Useful instrumentation via `-DISPLAY STATS` and IFCID 389/477
    - Sweet spot seems to be indexes with 3 or more levels that support heavy read access
      - More index levels = more benefit
      - Random insert/delete may also benefit
    - Needs more real storage – additional 20% of total BP space as a minimum, 30-40% recommended

```
SELECT SALES_DATE, SALE_VALUE ...
FROM SALES
WHERE SALES_NO = 84628
```



# User Experiences – Systems Programmer

- **Active Log Enhancements**
  - **What?**
    - 4GB restriction for active log files has been removed, and log latch activity has been significantly reduced
  - **Experiences**
    - Need to pay attention to archive log files, which will also increase
    - Total active log size can't exceed 768GB, and maximum of 93 log pairs still enforced so size logs accordingly
    - Possible to mix >4GB and <4GB logs (e.g. during transition to larger logs)
    - Need to be in After New Function Mode (ANFA)

## Summary

- DB2 12 has some great new features
  - Good mix of out-of-the-box savings, scalability improvements and new DBA/developer function
  - CPU/cost savings expected to be a major upgrade incentive once again
- Code quality seems to be very robust, and has been from an early stage in the ESP
- The full IDUG White Paper is available for download from IDUG website <http://idug.org/db2v12whitepaper>
- Thanks again to IBM for allowing IDUG to participate!

# Ron Brown

Ronek Consulting

*Ronek.Consulting@gmail.com*

DB2 12 for z/OS - IDUG User Experiences

*Please fill out your session  
evaluation before leaving!*

