IDUG

IDUG
2022 NA **Db2** Tech Conference

Db2 Advanced Log Space Management - Unleashed

Michael Roecken, IBM

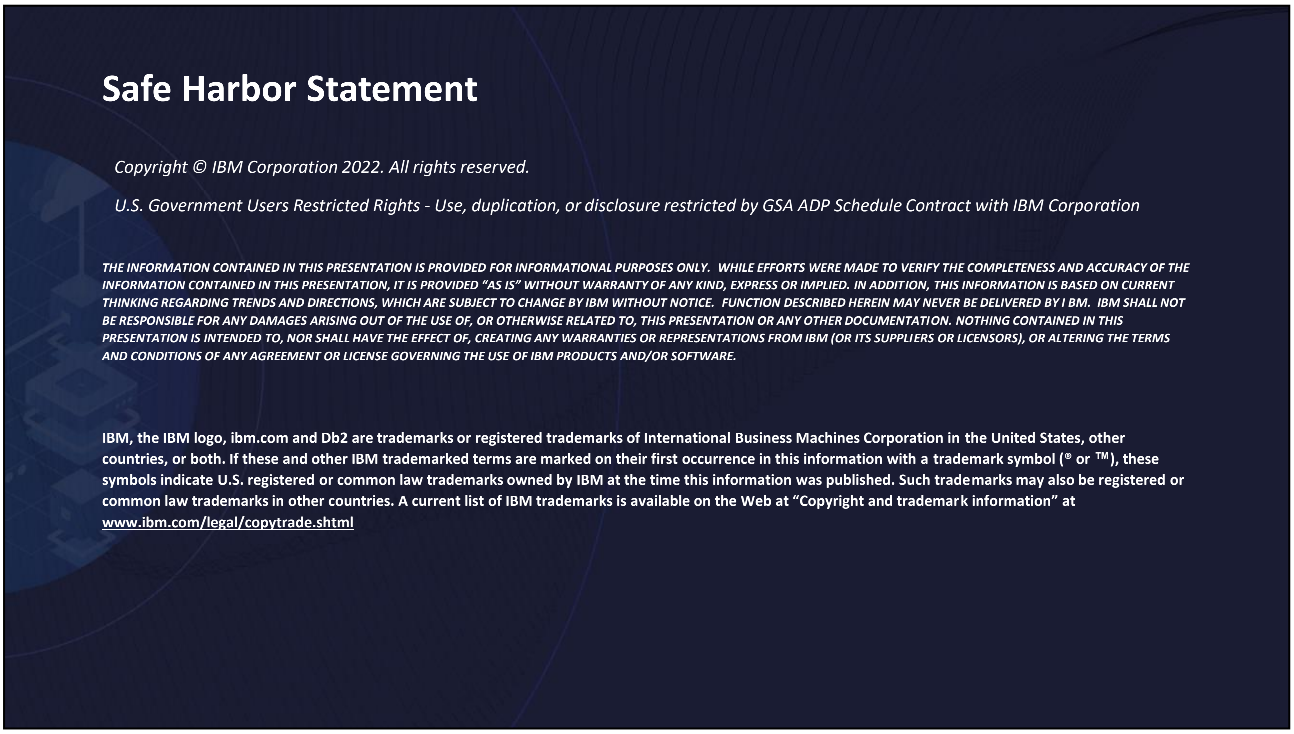
Session Code: C14

Db2 (LUW)

Boston, MA

Michael Roecken is a senior software developer with Db2 for Linux, UNIX, and Windows platforms at the IBM Toronto Lab. Michael has worked since 2000 designing, implementing and supporting various features and capabilities in the areas of: backup/restore, crash/rollforward recovery, high availability/disaster recovery, and logging/transaction management.

Please connect with Michael on Twitter at @roecken and LinkedIn.

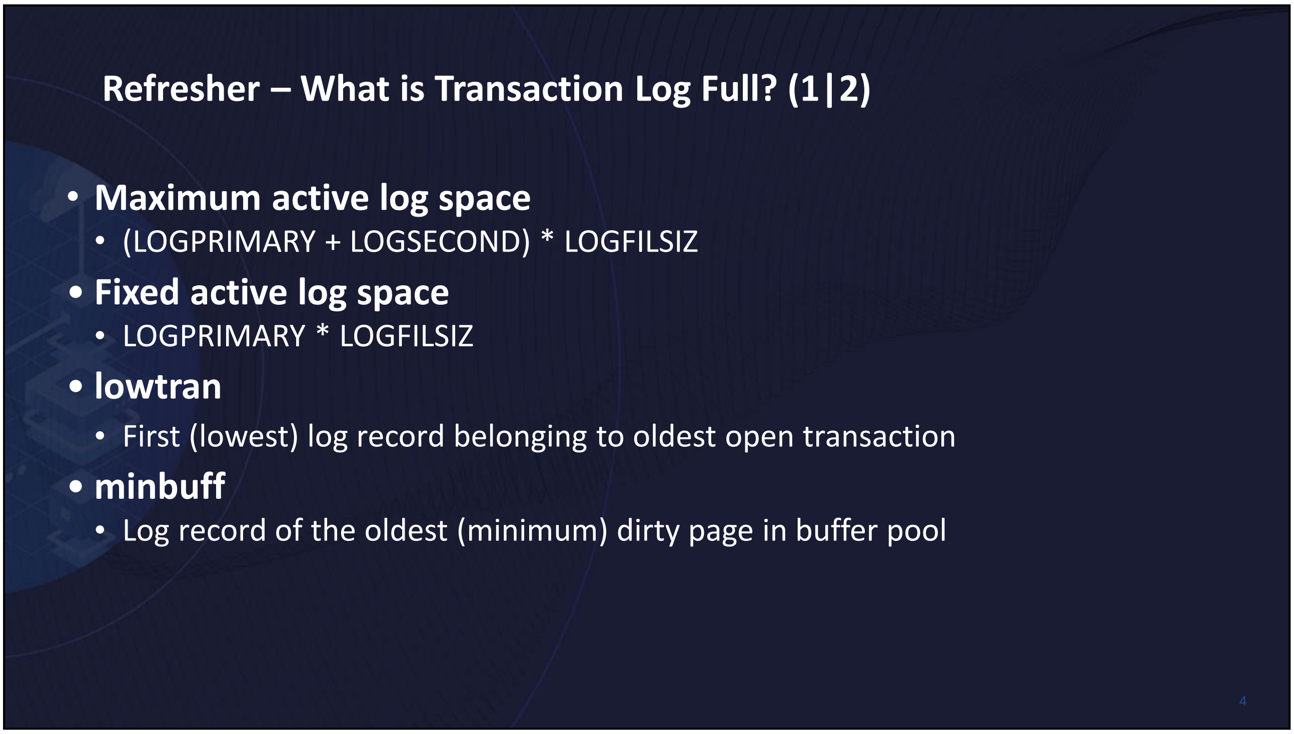


Safe Harbor Statement

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. 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. IN ADDITION, THIS INFORMATION IS BASED ON CURRENT THINKING REGARDING TRENDS AND DIRECTIONS, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. FUNCTION DESCRIBED HEREIN MAY NEVER BE DELIVERED BY I BM. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. 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 ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

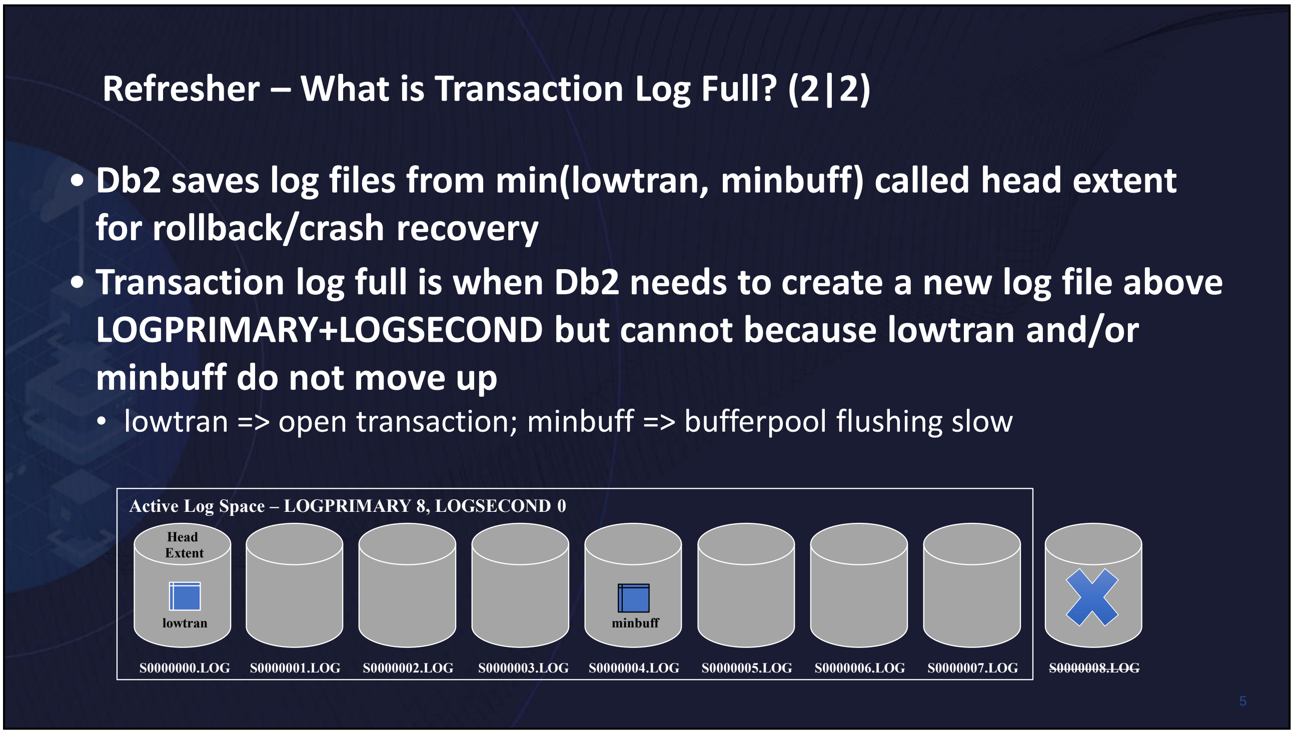
Agenda

- Advanced Log Space Management
  - Overview
  - Monitoring
  - Future



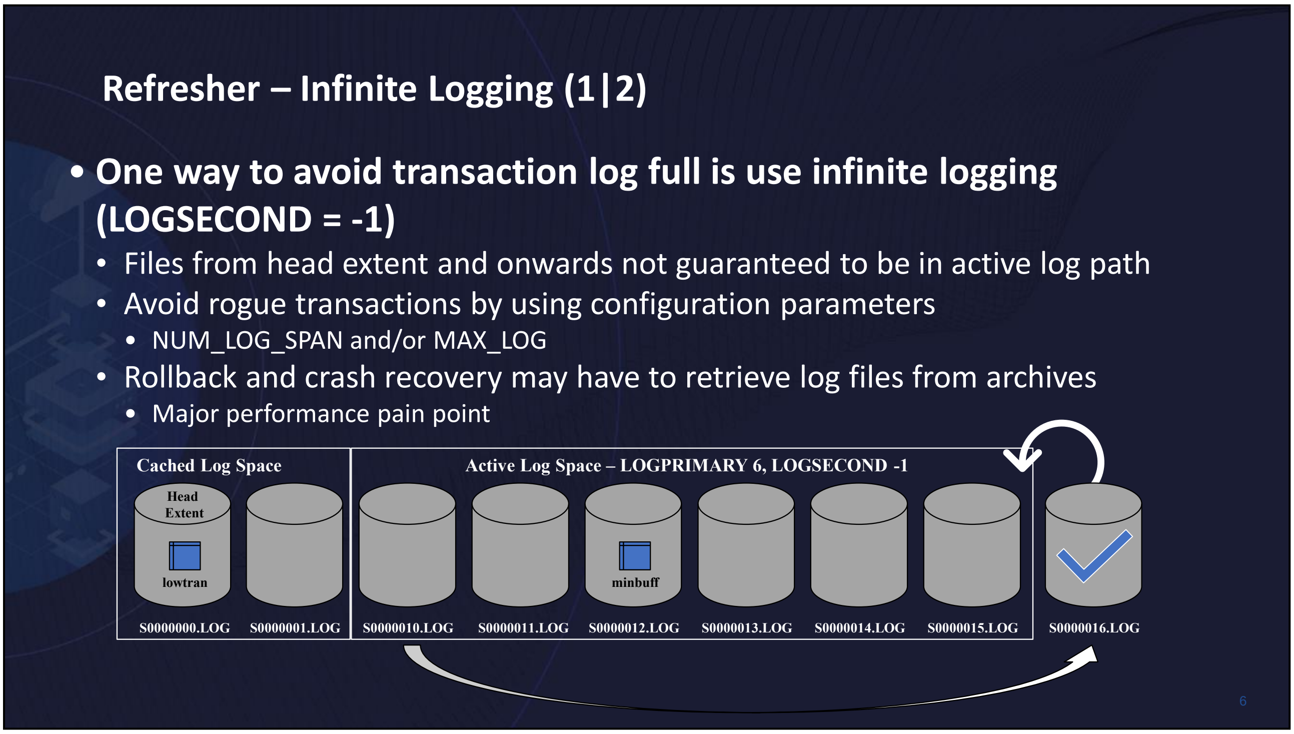
Refresher – What is Transaction Log Full? (1|2)

- **Maximum active log space**
  - (LOGPRIMARY + LOGSECOND) * LOGFILSIZ
- **Fixed active log space**
  - LOGPRIMARY * LOGFILSIZ
- **lowtran**
  - First (lowest) log record belonging to oldest open transaction
- **minbuff**
  - Log record of the oldest (minimum) dirty page in buffer pool



Refresher – What is Transaction Log Full? (2|2)

- Db2 saves log files from min(lowtran, minbuff) called head extent for rollback/crash recovery
- Transaction log full is when Db2 needs to create a new log file above LOGPRIMARY+LOGSECOND but cannot because lowtran and/or minbuff do not move up
  - lowtran => open transaction; minbuff => bufferpool flushing slow



Refresher – Infinite Logging (1|2)

- **One way to avoid transaction log full is use infinite logging (LOGSECOND = -1)**
  - Files from head extent and onwards not guaranteed to be in active log path
  - Avoid rogue transactions by using configuration parameters
    - NUM_LOG_SPAN and/or MAX_LOG
  - Rollback and crash recovery may have to retrieve log files from archives
    - Major performance pain point

Cached Log Space

Active Log Space – LOGPRIMARY 6, LOGSECOND -1

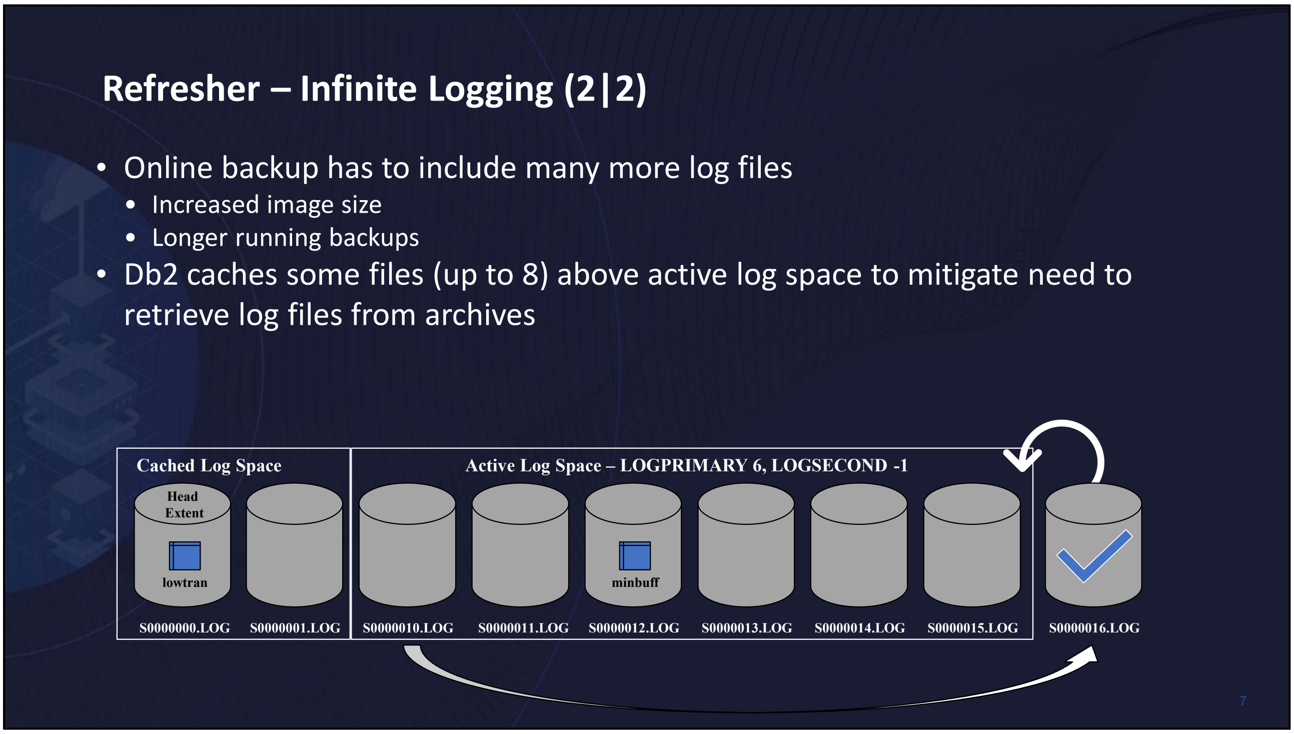
Head Extent

lowtran

minbuff

S0000000.LOG S0000001.LOG S0000010.LOG S0000011.LOG S0000012.LOG S0000013.LOG S0000014.LOG S0000015.LOG S0000016.LOG

6

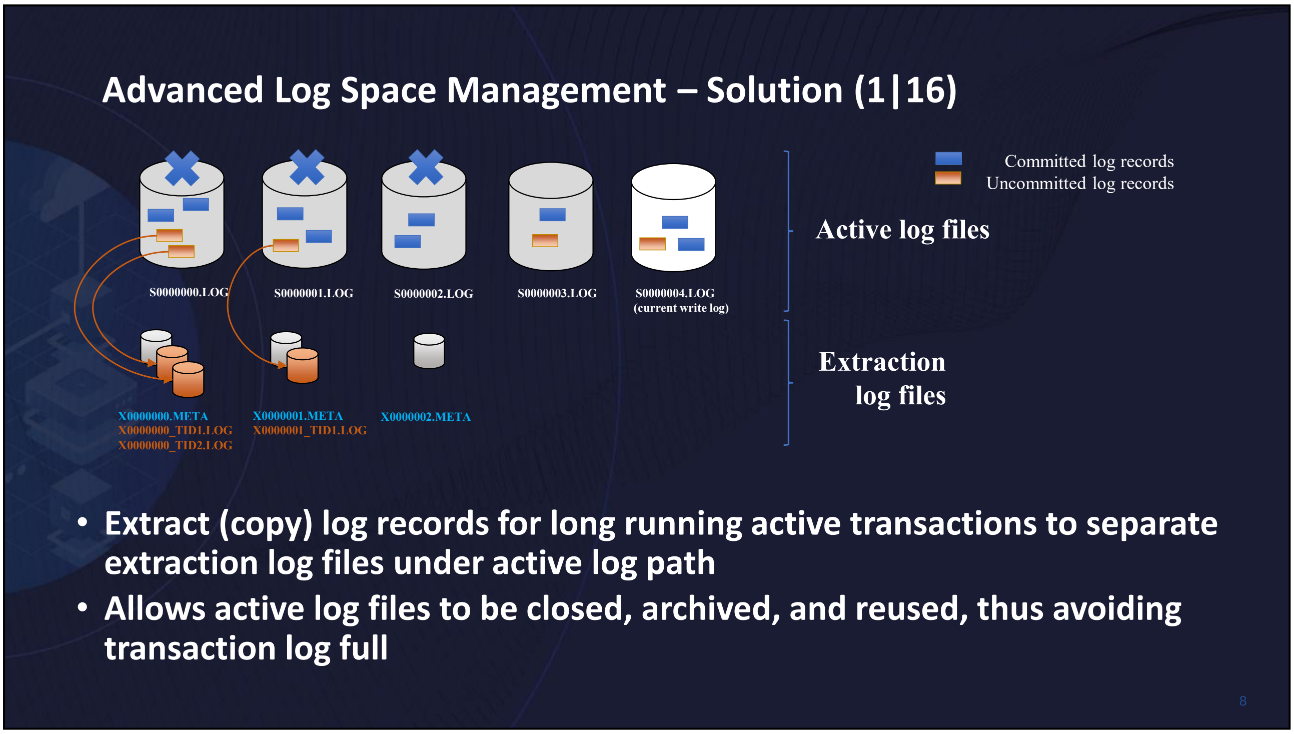


Refresher – Infinite Logging (2|2)

- Online backup has to include many more log files
  - Increased image size
  - Longer running backups
- Db2 caches some files (up to 8) above active log space to mitigate need to retrieve log files from archives

Cached Log Space

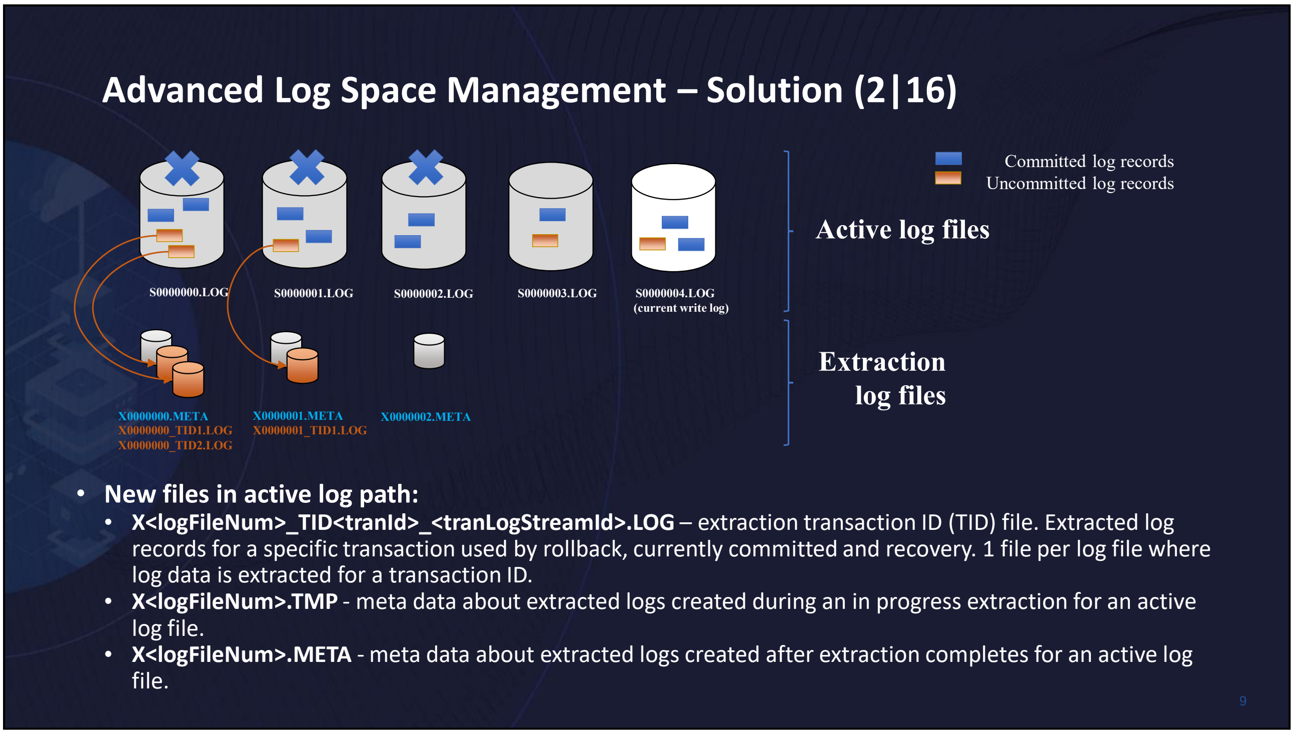
Active Log Space – LOGPRIMARY 6, LOGSECOND -1



Advanced Log Space Management – Solution (1|16)

- Extract (copy) log records for long running active transactions to separate extraction log files under active log path
- Allows active log files to be closed, archived, and reused, thus avoiding transaction log full

Extraction will kick in when a certain percentage of log space has been consumed and will extract log records starting from head extent ID into separate extraction log files stored in the active log path.



Advanced Log Space Management – Solution (2|16)

🟦 Committed log records
🟧 Uncommitted log records

S0000000.LOG
S0000001.LOG
S0000002.LOG
S0000003.LOG
S0000004.LOG
(current write log)

Active log files

X0000000.META
X0000000_TID1.LOG
X0000000_TID2.LOG

X0000001.META
X0000001_TID1.LOG

X0000002.META

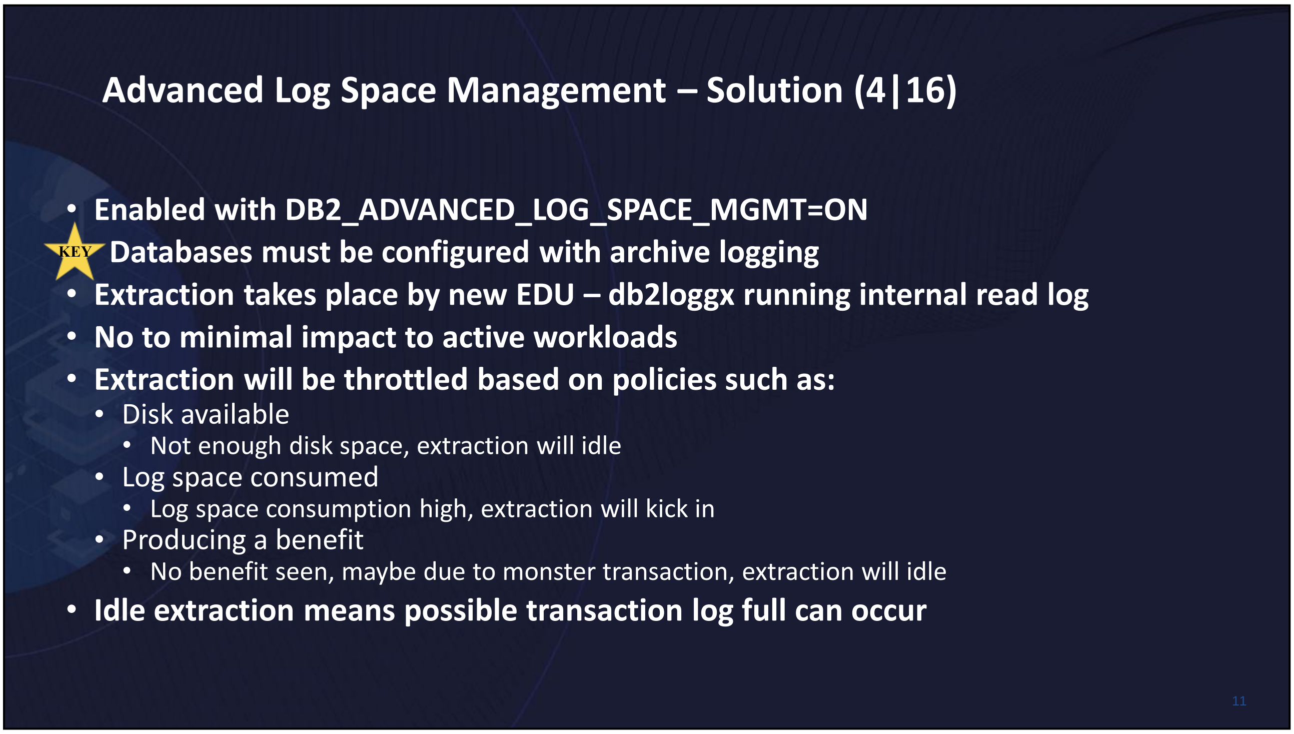
Extraction log files

- **New files in active log path:**
  - **X<logFileNum>_TID<tranId>_<tranLogStreamId>.LOG** – extraction transaction ID (TID) file. Extracted log records for a specific transaction used by rollback, currently committed and recovery. 1 file per log file where log data is extracted for a transaction ID.
  - **X<logFileNum>.TMP** - meta data about extracted logs created during an in progress extraction for an active log file.
  - **X<logFileNum>.META** - meta data about extracted logs created after extraction completes for an active log file.

Advanced Log Space Management – Solution (3|16)

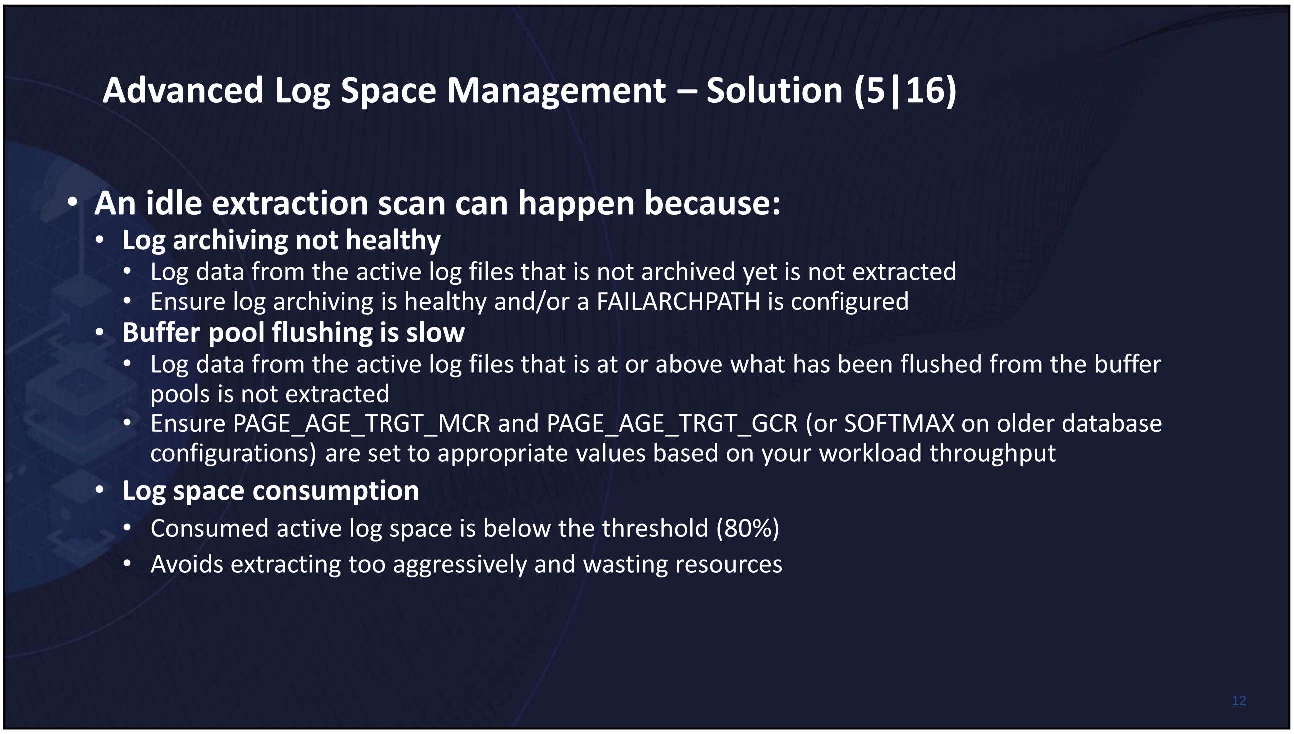
Timeline:

- 11.5.0.0 – Technical Preview (not for production use)
- 11.5.4.0 – Production support
  - No mirror log support
  - No HADR support
  - No online backup support
  - No pureScale support
- 11.5.5.0 – Basic mirror log support
- 11.5.6.0 – HADR and online backup support



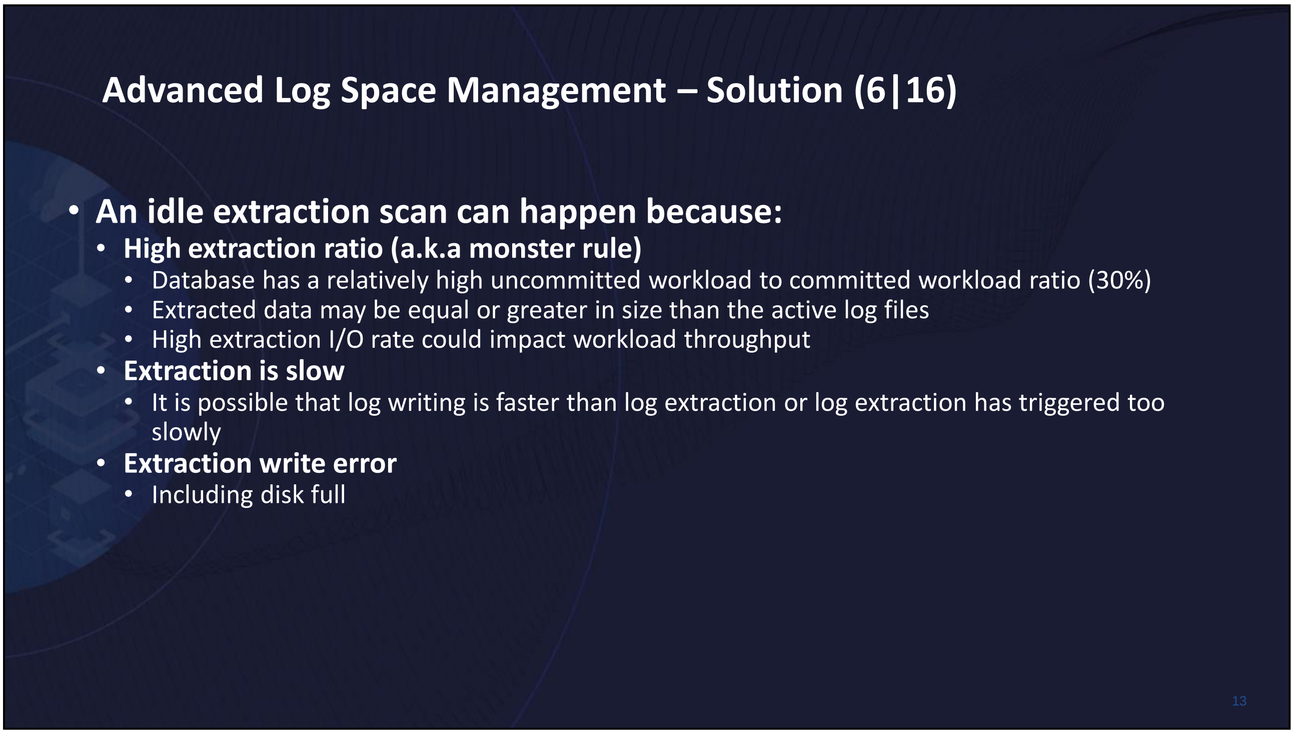
Advanced Log Space Management – Solution (4|16)

- **Enabled with DB2_ADVANCED_LOG_SPACE_MGMT=ON**
- KEY **Databases must be configured with archive logging**
- **Extraction takes place by new EDU – db2loggx running internal read log**
- **No to minimal impact to active workloads**
- **Extraction will be throttled based on policies such as:**
  - Disk available
    - Not enough disk space, extraction will idle
  - Log space consumed
    - Log space consumption high, extraction will kick in
  - Producing a benefit
    - No benefit seen, maybe due to monster transaction, extraction will idle
- **Idle extraction means possible transaction log full can occur**



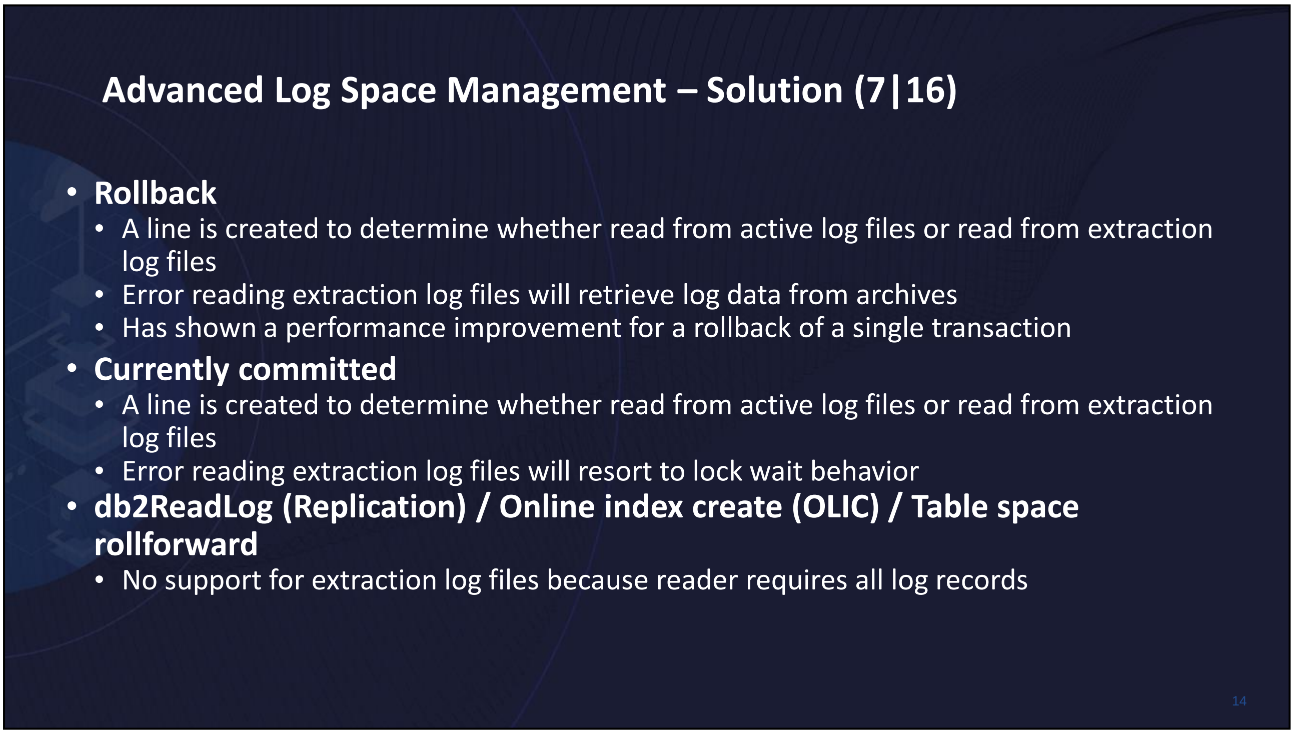
Advanced Log Space Management – Solution (5|16)

- **An idle extraction scan can happen because:**
  - **Log archiving not healthy**
    - Log data from the active log files that is not archived yet is not extracted
    - Ensure log archiving is healthy and/or a FAILARCHPATH is configured
  - **Buffer pool flushing is slow**
    - Log data from the active log files that is at or above what has been flushed from the buffer pools is not extracted
    - Ensure PAGE_AGE_TRGT_MCR and PAGE_AGE_TRGT_GCR (or SOFTMAX on older database configurations) are set to appropriate values based on your workload throughput
  - **Log space consumption**
    - Consumed active log space is below the threshold (80%)
    - Avoids extracting too aggressively and wasting resources



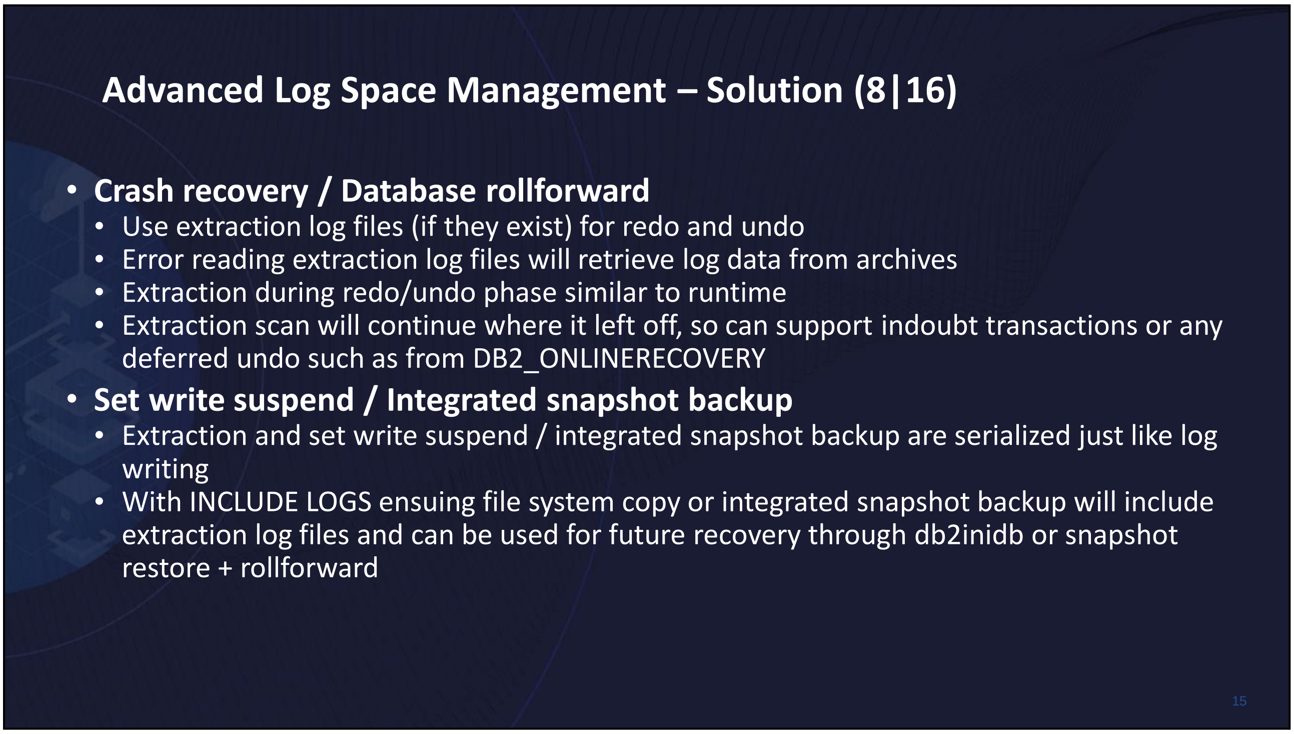
Advanced Log Space Management – Solution (6|16)

- An idle extraction scan can happen because:
  - **High extraction ratio (a.k.a monster rule)**
    - Database has a relatively high uncommitted workload to committed workload ratio (30%)
    - Extracted data may be equal or greater in size than the active log files
    - High extraction I/O rate could impact workload throughput
  - **Extraction is slow**
    - It is possible that log writing is faster than log extraction or log extraction has triggered too slowly
  - **Extraction write error**
    - Including disk full



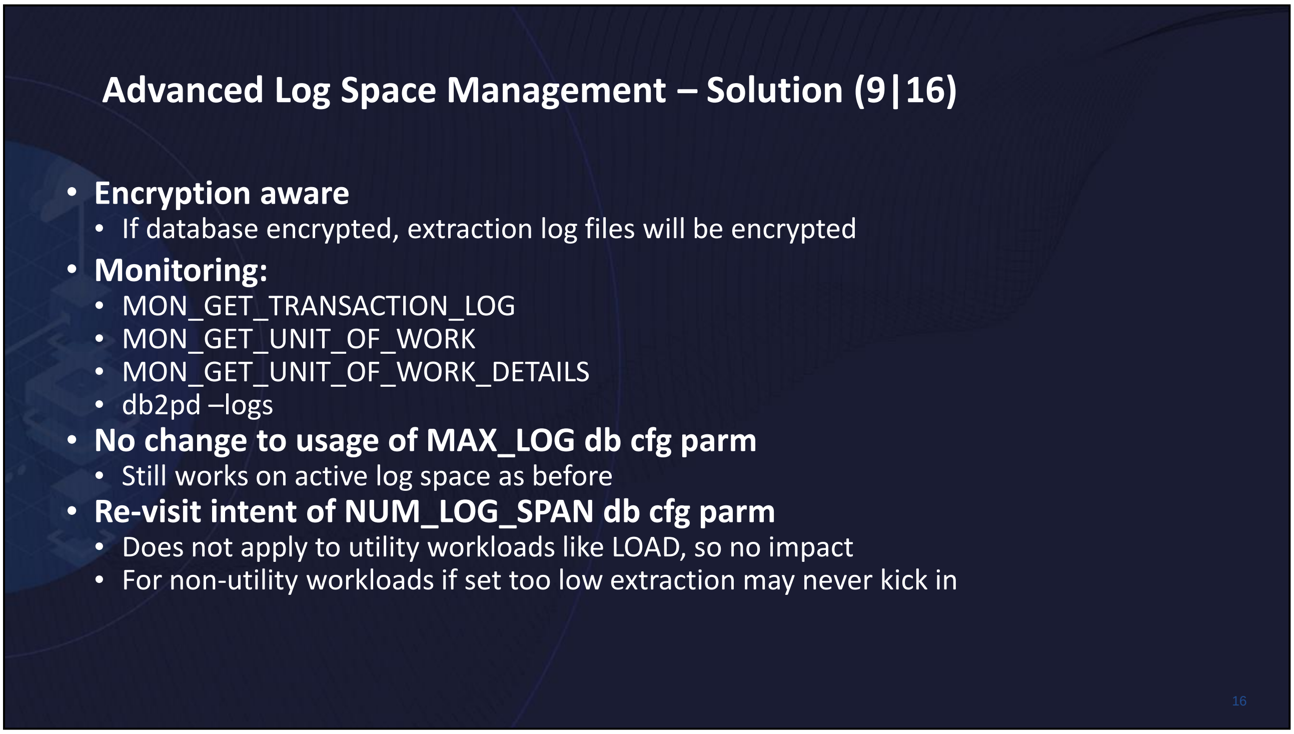
Advanced Log Space Management – Solution (7|16)

- **Rollback**
  - A line is created to determine whether read from active log files or read from extraction log files
  - Error reading extraction log files will retrieve log data from archives
  - Has shown a performance improvement for a rollback of a single transaction
- **Currently committed**
  - A line is created to determine whether read from active log files or read from extraction log files
  - Error reading extraction log files will resort to lock wait behavior
- **db2ReadLog (Replication) / Online index create (OLIC) / Table space rollforward**
  - No support for extraction log files because reader requires all log records



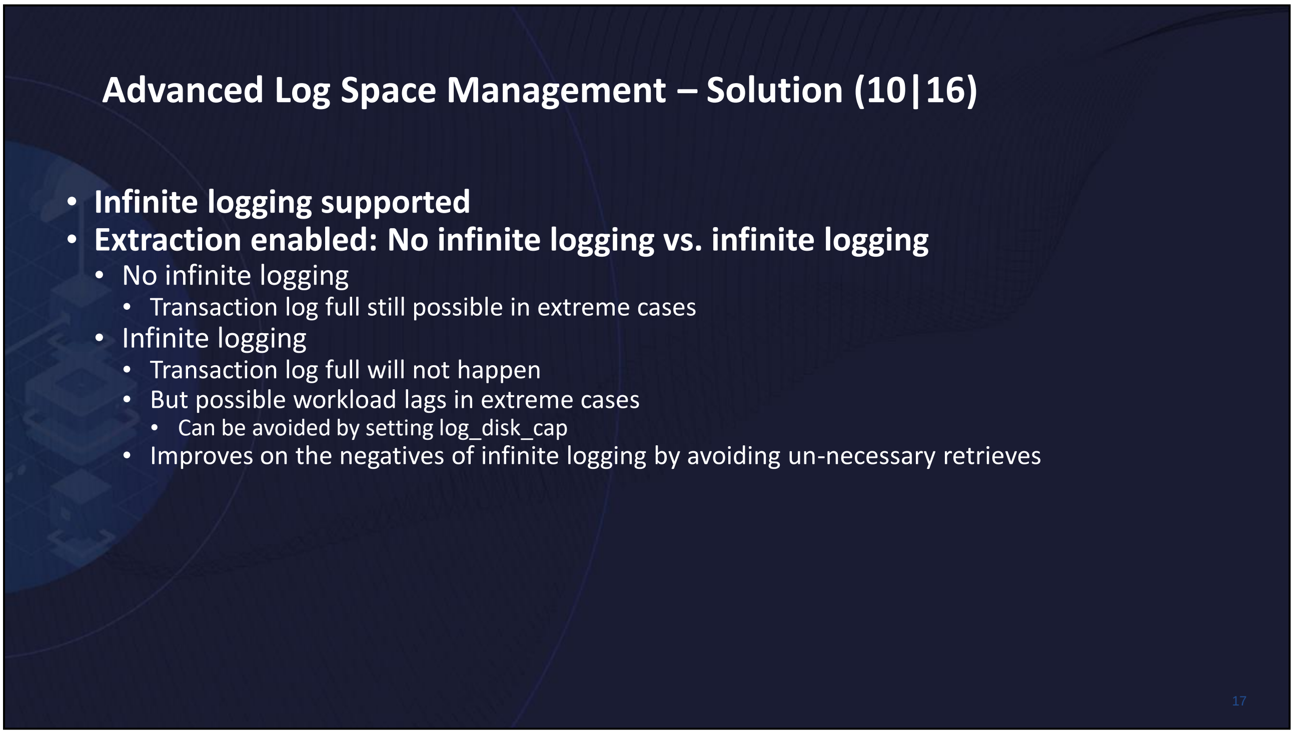
Advanced Log Space Management – Solution (8|16)

- **Crash recovery / Database rollforward**
  - Use extraction log files (if they exist) for redo and undo
  - Error reading extraction log files will retrieve log data from archives
  - Extraction during redo/undo phase similar to runtime
  - Extraction scan will continue where it left off, so can support indoubt transactions or any deferred undo such as from DB2_ONLINERECOVERY
- **Set write suspend / Integrated snapshot backup**
  - Extraction and set write suspend / integrated snapshot backup are serialized just like log writing
  - With INCLUDE LOGS ensuing file system copy or integrated snapshot backup will include extraction log files and can be used for future recovery through db2inidb or snapshot restore + rollforward



Advanced Log Space Management – Solution (9|16)

- **Encryption aware**
  - If database encrypted, extraction log files will be encrypted
- **Monitoring:**
  - MON_GET_TRANSACTION_LOG
  - MON_GET_UNIT_OF_WORK
  - MON_GET_UNIT_OF_WORK_DETAILS
  - db2pd –logs
- **No change to usage of MAX_LOG db cfg parm**
  - Still works on active log space as before
- **Re-visit intent of NUM_LOG_SPAN db cfg parm**
  - Does not apply to utility workloads like LOAD, so no impact
  - For non-utility workloads if set too low extraction may never kick in

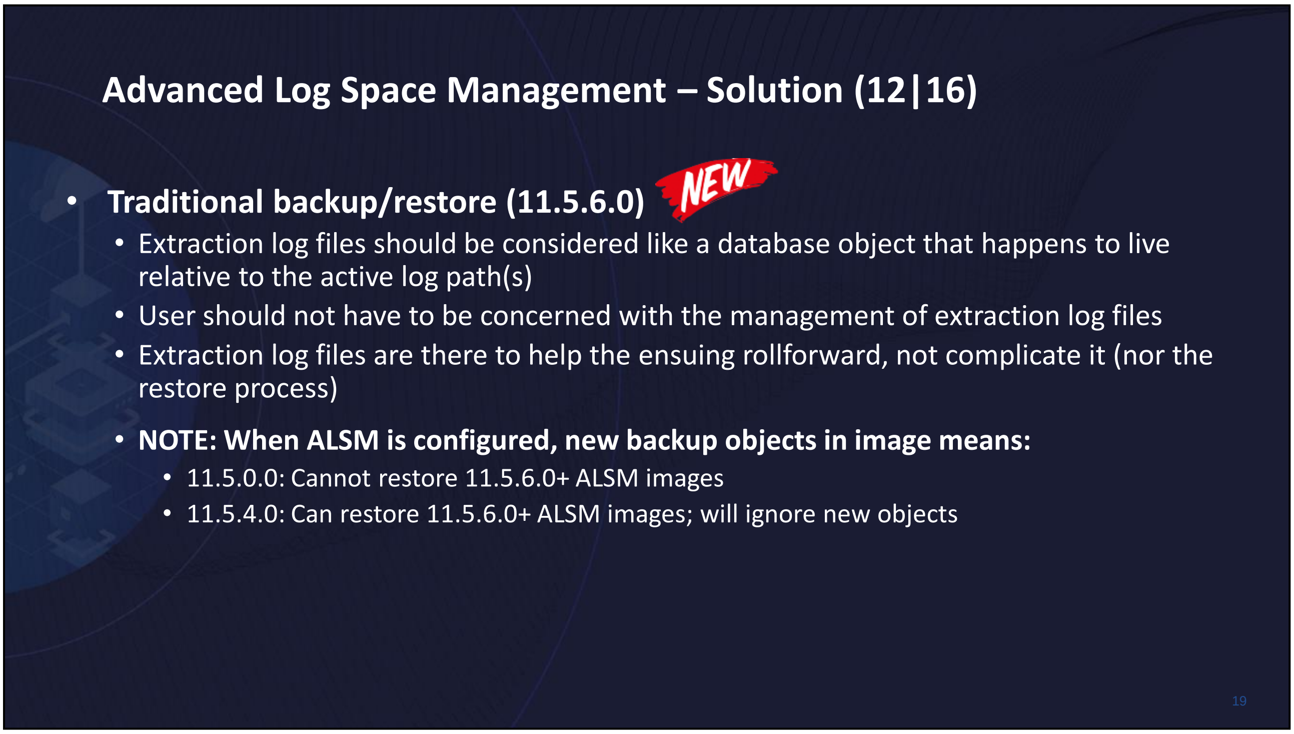


Advanced Log Space Management – Solution (10|16)

- **Infinite logging supported**
- **Extraction enabled: No infinite logging vs. infinite logging**
  - No infinite logging
    - Transaction log full still possible in extreme cases
  - Infinite logging
    - Transaction log full will not happen
    - But possible workload lags in extreme cases
      - Can be avoided by setting log_disk_cap
    - Improves on the negatives of infinite logging by avoiding un-necessary retrieves

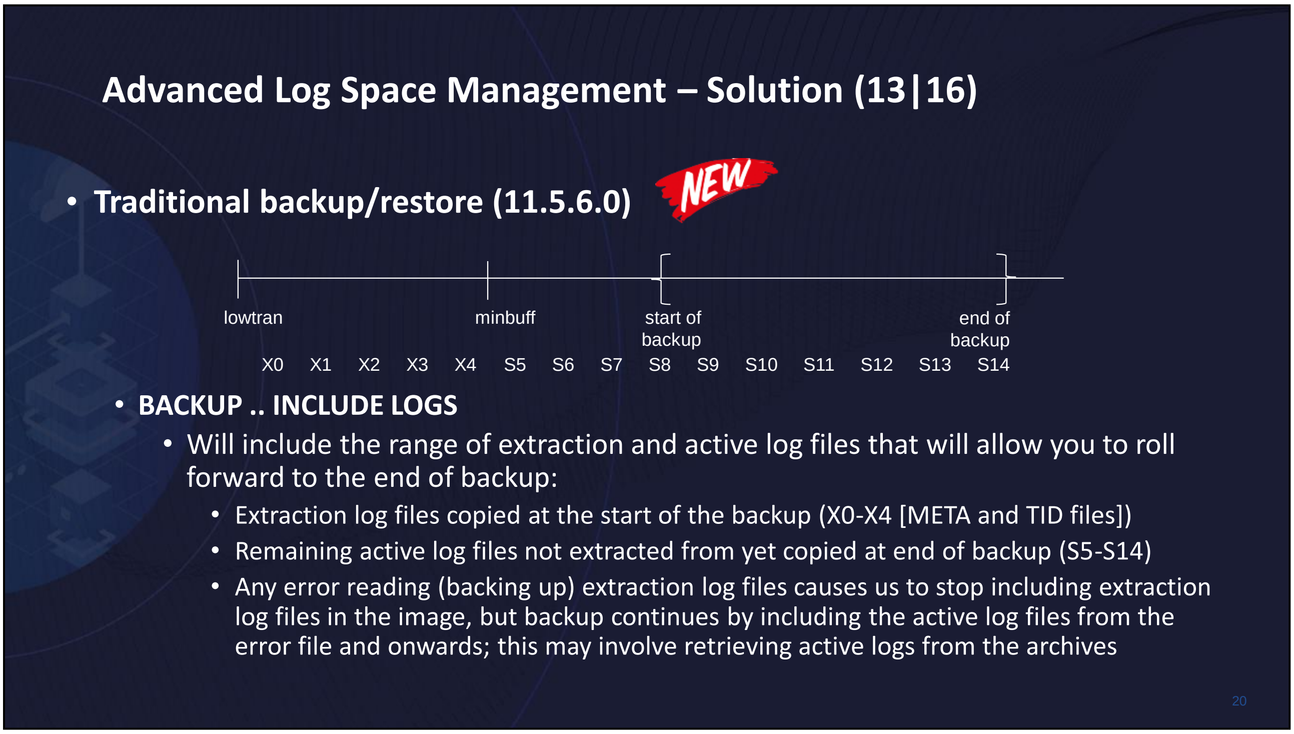
Advanced Log Space Management – Solution (11|16)

- **Databases configured with a MIRRORLOGPATH (11.5.5.0)**
  - 11.5.5.0 introduced basic support
  - Extraction will take place as long as log data can be:
    - Read from the active log files found in either the primary or mirror log path; and
    - Extracted log data can be written to the extraction log files found under the primary log path
  - No extraction log files will be written to the mirror log path (Future support)



Advanced Log Space Management – Solution (12|16)

- **Traditional backup/restore (11.5.6.0)** NEW
  - Extraction log files should be considered like a database object that happens to live relative to the active log path(s)
  - User should not have to be concerned with the management of extraction log files
  - Extraction log files are there to help the ensuing rollforward, not complicate it (nor the restore process)
  - **NOTE: When ALSM is configured, new backup objects in image means:**
    - 11.5.0.0: Cannot restore 11.5.6.0+ ALSM images
    - 11.5.4.0: Can restore 11.5.6.0+ ALSM images; will ignore new objects



Advanced Log Space Management – Solution (13|16)

- **Traditional backup/restore (11.5.6.0)** NEW

lowtran

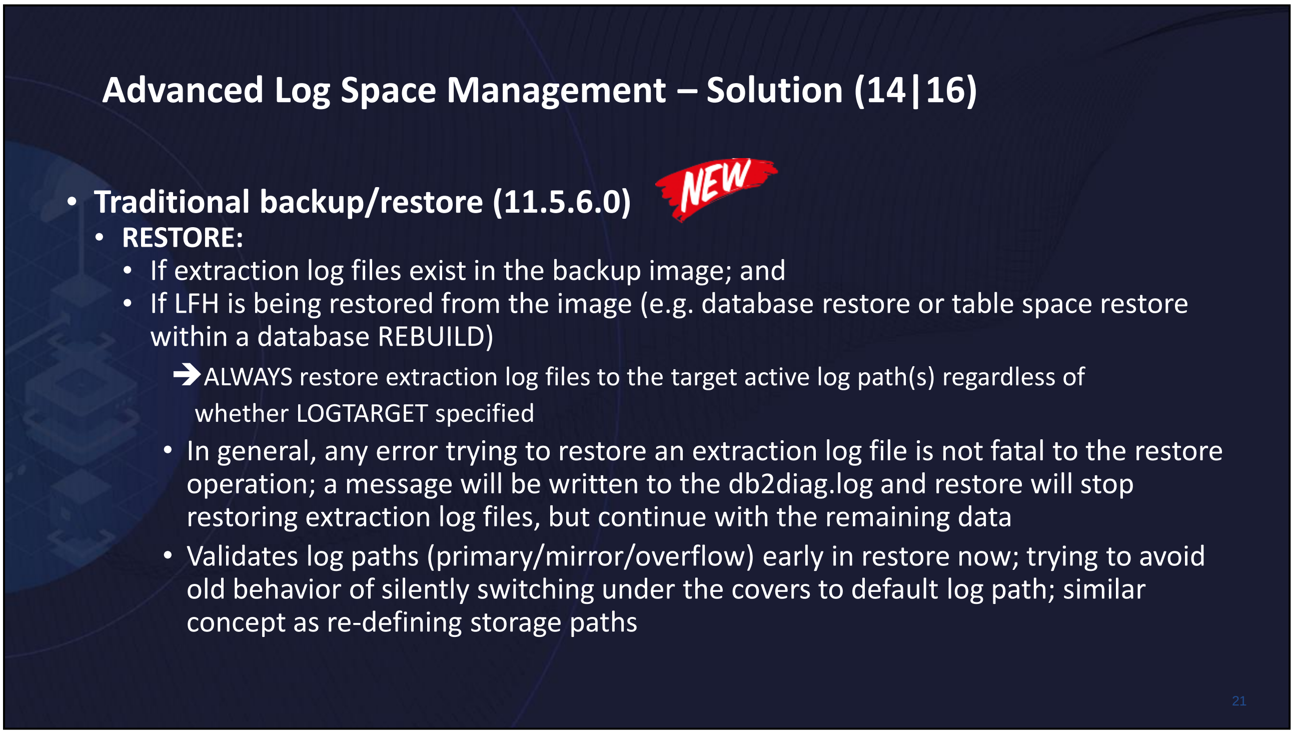
minbuff

start of backup

end of backup

X0 X1 X2 X3 X4 S5 S6 S7 S8 S9 S10 S11 S12 S13 S14

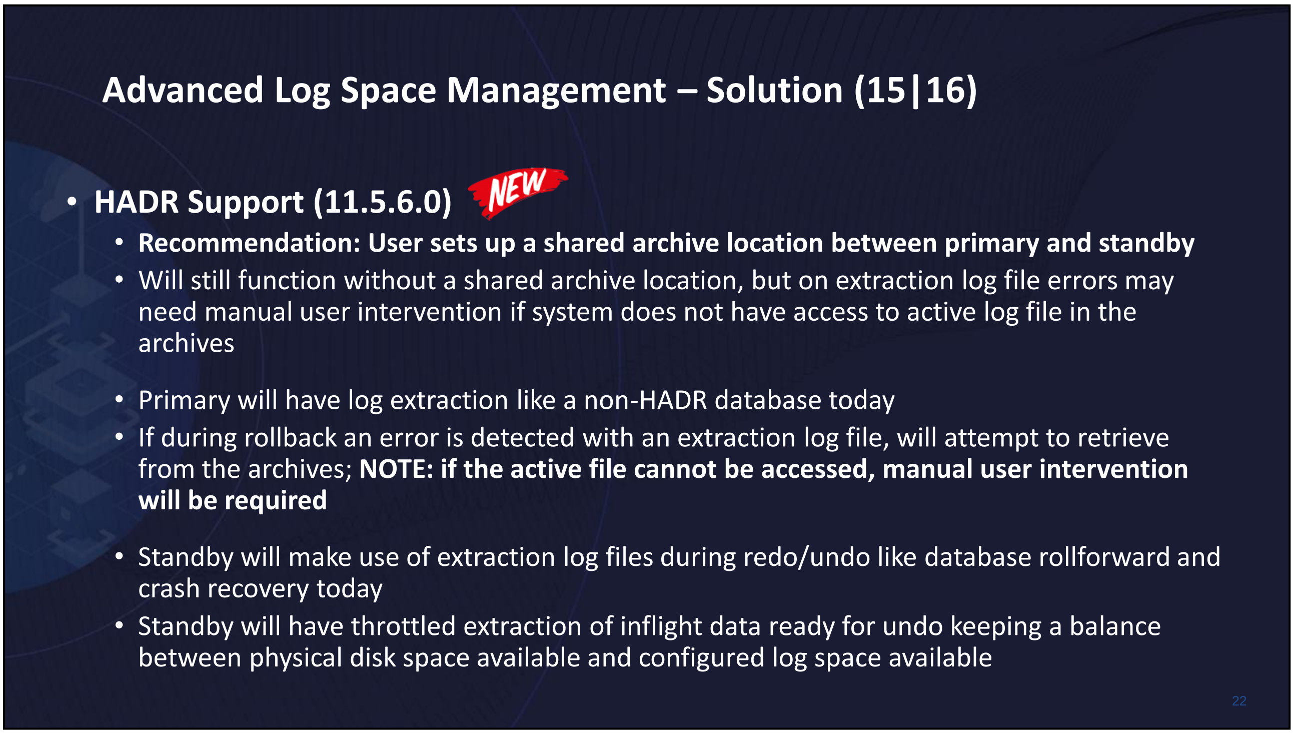
- **BACKUP .. INCLUDE LOGS**
  - Will include the range of extraction and active log files that will allow you to roll forward to the end of backup:
    - Extraction log files copied at the start of the backup (X0-X4 [META and TID files])
    - Remaining active log files not extracted from yet copied at end of backup (S5-S14)
    - Any error reading (backing up) extraction log files causes us to stop including extraction log files in the image, but backup continues by including the active log files from the error file and onwards; this may involve retrieving active logs from the archives



Advanced Log Space Management – Solution (14|16)

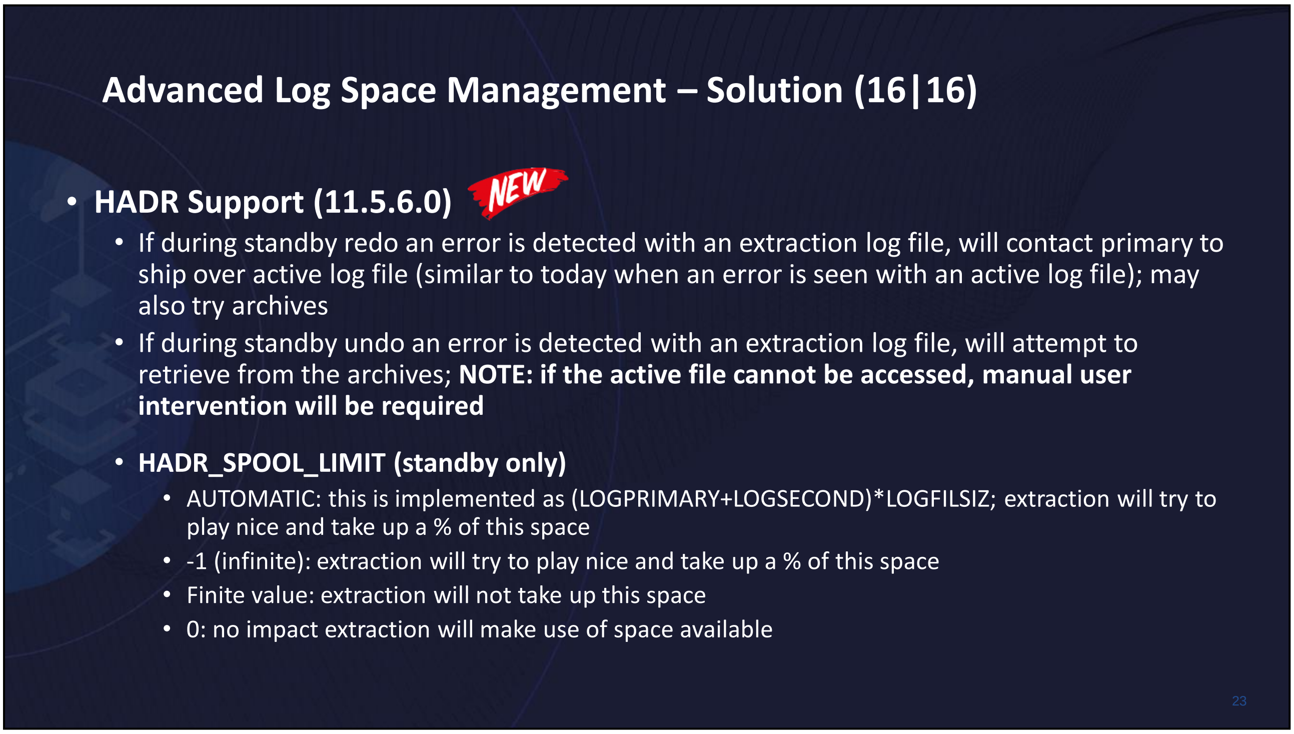
- **Traditional backup/restore (11.5.6.0)** NEW
  - **RESTORE:**
    - If extraction log files exist in the backup image; and
    - If LFH is being restored from the image (e.g. database restore or table space restore within a database REBUILD)

      ➔ALWAYS restore extraction log files to the target active log path(s) regardless of whether LOGTARGET specified
    - In general, any error trying to restore an extraction log file is not fatal to the restore operation; a message will be written to the db2diag.log and restore will stop restoring extraction log files, but continue with the remaining data
    - Validates log paths (primary/mirror/overflow) early in restore now; trying to avoid old behavior of silently switching under the covers to default log path; similar concept as re-defining storage paths



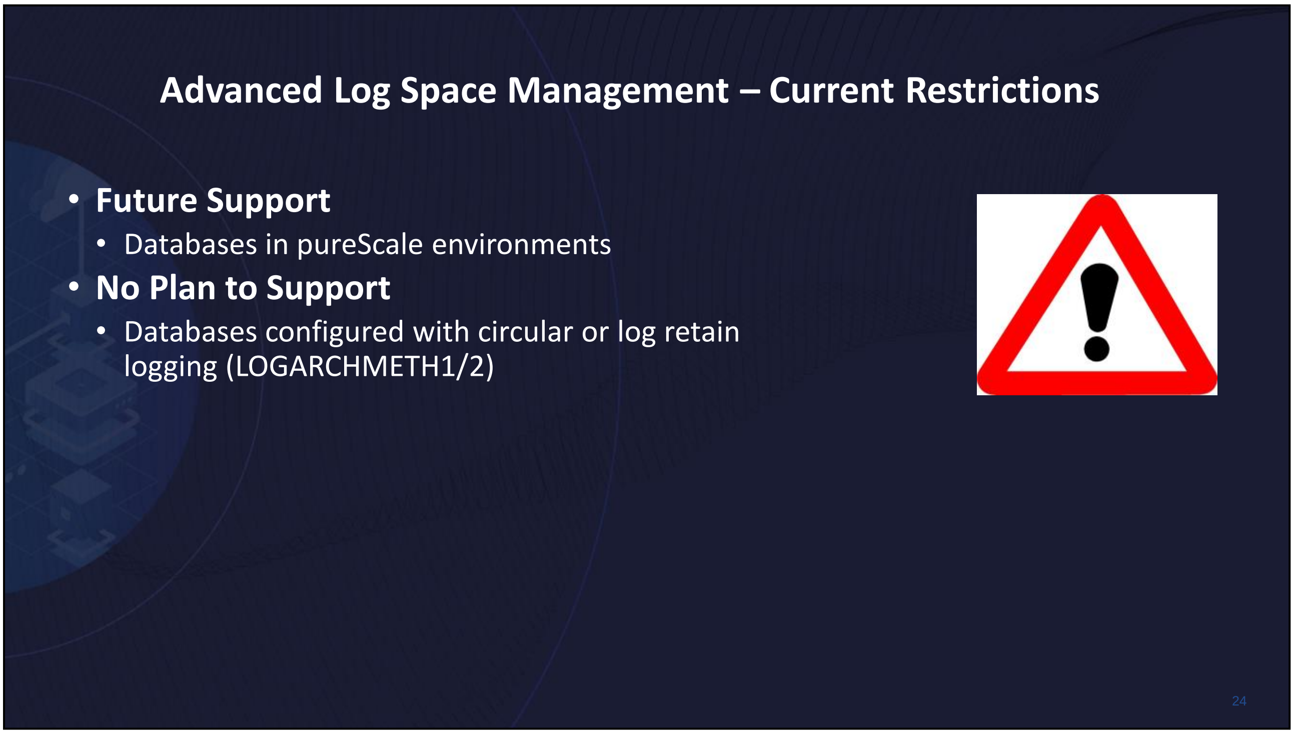
Advanced Log Space Management – Solution (15|16)

- **HADR Support (11.5.6.0)** NEW
  - **Recommendation: User sets up a shared archive location between primary and standby**
  - Will still function without a shared archive location, but on extraction log file errors may need manual user intervention if system does not have access to active log file in the archives
  - Primary will have log extraction like a non-HADR database today
  - If during rollback an error is detected with an extraction log file, will attempt to retrieve from the archives; **NOTE: if the active file cannot be accessed, manual user intervention will be required**
  - Standby will make use of extraction log files during redo/undo like database rollforward and crash recovery today
  - Standby will have throttled extraction of inflight data ready for undo keeping a balance between physical disk space available and configured log space available



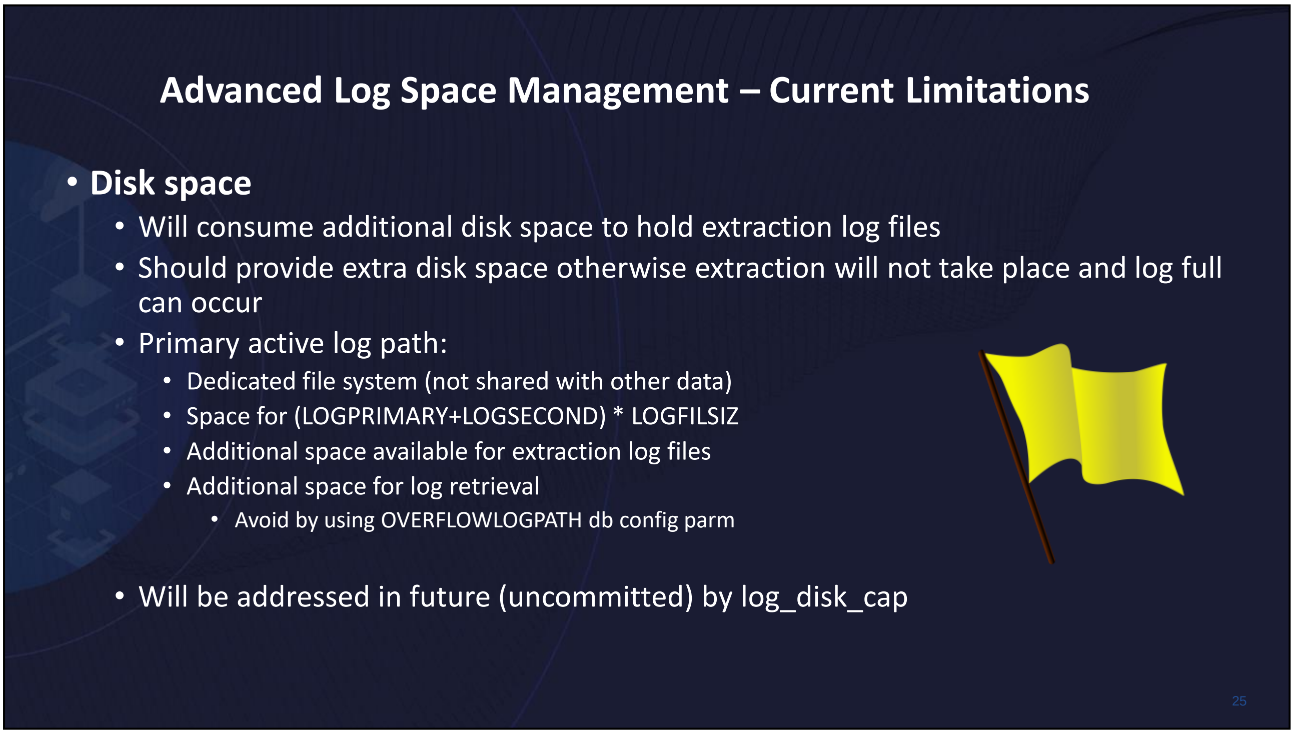
Advanced Log Space Management – Solution (16|16)

- **HADR Support (11.5.6.0)** NEW
  - If during standby redo an error is detected with an extraction log file, will contact primary to ship over active log file (similar to today when an error is seen with an active log file); may also try archives
  - If during standby undo an error is detected with an extraction log file, will attempt to retrieve from the archives; **NOTE: if the active file cannot be accessed, manual user intervention will be required**
  - **HADR_SPOOL_LIMIT (standby only)**
    - AUTOMATIC: this is implemented as (LOGPRIMARY+LOGSECOND)*LOGFILSIZ; extraction will try to play nice and take up a % of this space
    - -1 (infinite): extraction will try to play nice and take up a % of this space
    - Finite value: extraction will not take up this space
    - 0: no impact extraction will make use of space available



Advanced Log Space Management – Current Restrictions

- **Future Support**
  - Databases in pureScale environments
- **No Plan to Support**
  - Databases configured with circular or log retain logging (LOGARCHMETH1/2)

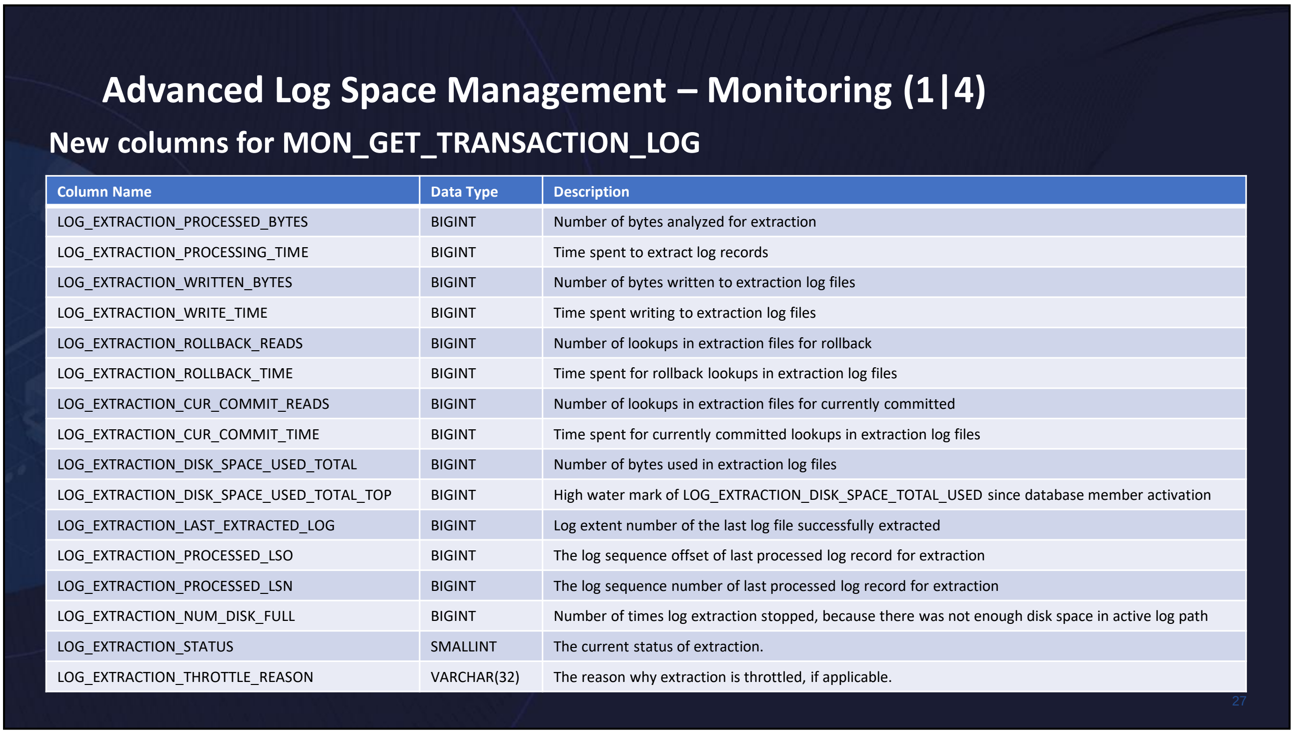


Advanced Log Space Management – Current Limitations

- **Disk space**
  - Will consume additional disk space to hold extraction log files
  - Should provide extra disk space otherwise extraction will not take place and log full can occur
  - Primary active log path:
    - Dedicated file system (not shared with other data)
    - Space for (LOGPRIMARY+LOGSECOND) * LOGFILSIZ
    - Additional space available for extraction log files
    - Additional space for log retrieval
      - Avoid by using OVERFLOWLOGPATH db config parm
  - Will be addressed in future (uncommitted) by log_disk_cap

Agenda

- Advanced Log Space Management
  - Overview
  - Monitoring
  - Future

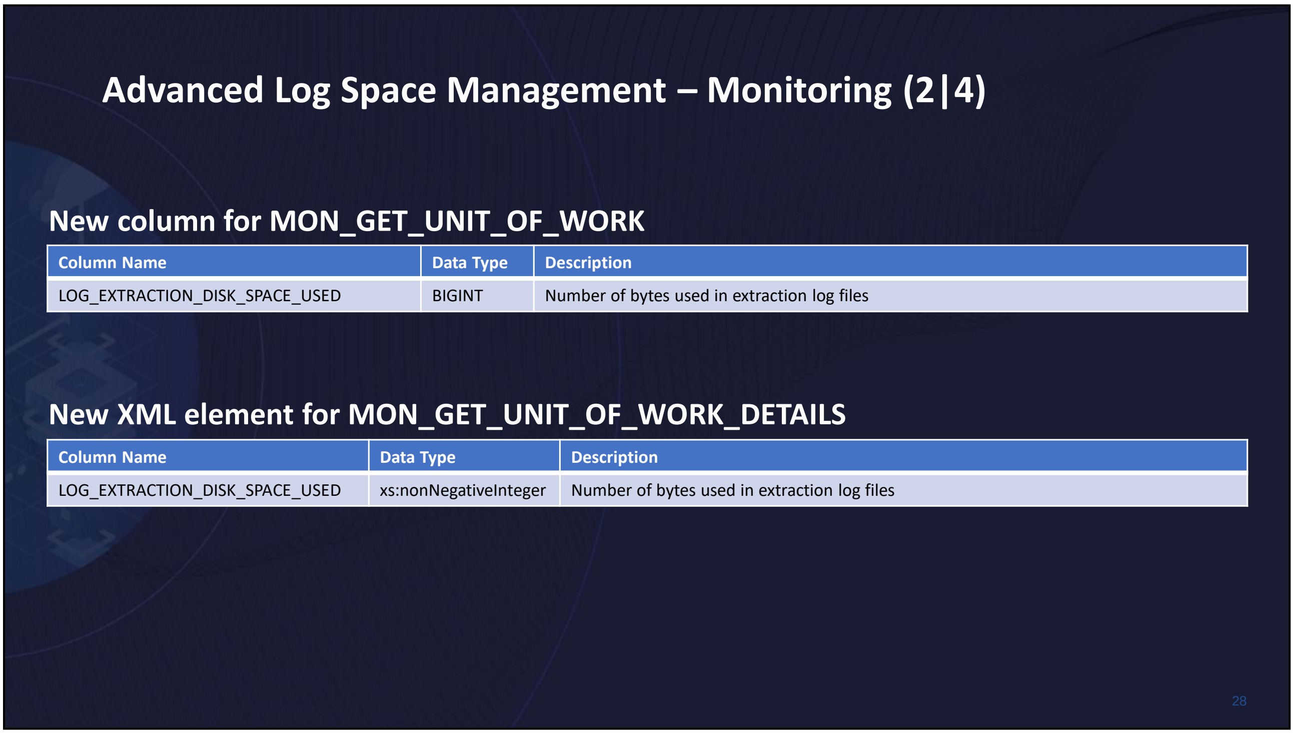


Advanced Log Space Management – Monitoring (1|4)

New columns for MON_GET_TRANSACTION_LOG

Column Name	Data Type	Description
LOG_EXTRACTION_PROCESSED_BYTES	BIGINT	Number of bytes analyzed for extraction
LOG_EXTRACTION_PROCESSING_TIME	BIGINT	Time spent to extract log records
LOG_EXTRACTION_WRITTEN_BYTES	BIGINT	Number of bytes written to extraction log files
LOG_EXTRACTION_WRITE_TIME	BIGINT	Time spent writing to extraction log files
LOG_EXTRACTION_ROLLBACK_READS	BIGINT	Number of lookups in extraction files for rollback
LOG_EXTRACTION_ROLLBACK_TIME	BIGINT	Time spent for rollback lookups in extraction log files
LOG_EXTRACTION_CUR_COMMIT_READS	BIGINT	Number of lookups in extraction files for currently committed
LOG_EXTRACTION_CUR_COMMIT_TIME	BIGINT	Time spent for currently committed lookups in extraction log files
LOG_EXTRACTION_DISK_SPACE_USED_TOTAL	BIGINT	Number of bytes used in extraction log files
LOG_EXTRACTION_DISK_SPACE_USED_TOTAL_TOP	BIGINT	High water mark of LOG_EXTRACTION_DISK_SPACE_TOTAL_USED since database member activation
LOG_EXTRACTION_LAST_EXTRACTED_LOG	BIGINT	Log extent number of the last log file successfully extracted
LOG_EXTRACTION_PROCESSED_LSO	BIGINT	The log sequence offset of last processed log record for extraction
LOG_EXTRACTION_PROCESSED_LSN	BIGINT	The log sequence number of last processed log record for extraction
LOG_EXTRACTION_NUM_DISK_FULL	BIGINT	Number of times log extraction stopped, because there was not enough disk space in active log path
LOG_EXTRACTION_STATUS	SMALLINT	The current status of extraction.
LOG_EXTRACTION_THROTTLE_REASON	VARCHAR(32)	The reason why extraction is throttled, if applicable.

27



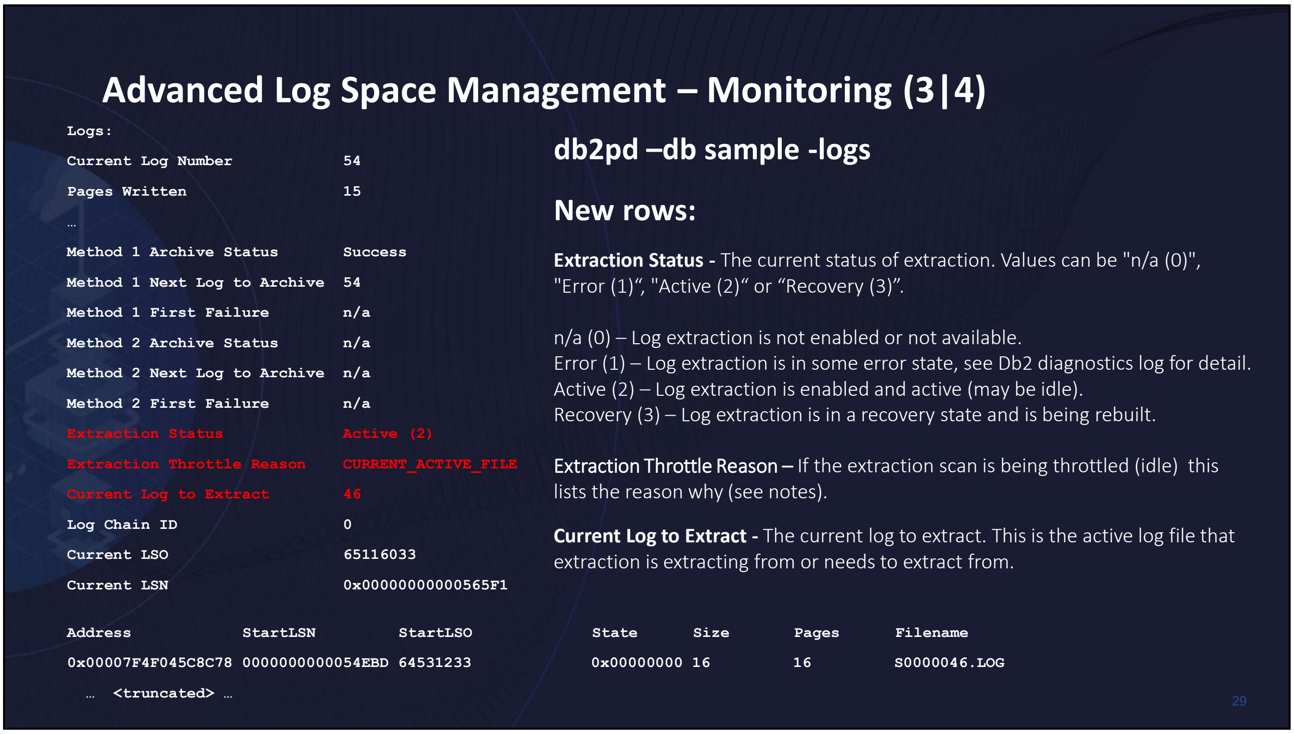
Advanced Log Space Management – Monitoring (2|4)

New column for MON_GET_UNIT_OF_WORK

Column Name	Data Type	Description
LOG_EXTRACTION_DISK_SPACE_USED	BIGINT	Number of bytes used in extraction log files

New XML element for MON_GET_UNIT_OF_WORK_DETAILS

Column Name	Data Type	Description
LOG_EXTRACTION_DISK_SPACE_USED	xs:nonNegativeInteger	Number of bytes used in extraction log files



Advanced Log Space Management – Monitoring (3|4)

```
Logs:
Current Log Number            54
Pages Written                 15
…
Method 1 Archive Status       Success
Method 1 Next Log to Archive  54
Method 1 First Failure        n/a
Method 2 Archive Status       n/a
Method 2 Next Log to Archive  n/a
Method 2 First Failure        n/a
Extraction Status             Active (2)
Extraction Throttle Reason    CURRENT_ACTIVE_FILE
Current Log to Extract        46
Log Chain ID                  0
Current LSO                   65116033
Current LSN                   0x00000000000565F1

Address            StartLSN         StartLSO          State      Size       Pages      Filename
0x00007F4F045C8C78 0000000000054EBD 64531233          0x00000000 16         16         S0000046.LOG
  …  <truncated> …
```

db2pd –db sample -logs

New rows:

Extraction Status - The current status of extraction. Values can be "n/a (0)", "Error (1)", "Active (2)" or "Recovery (3)".

n/a (0) – Log extraction is not enabled or not available.
Error (1) – Log extraction is in some error state, see Db2 diagnostics log for detail.
Active (2) – Log extraction is enabled and active (may be idle).
Recovery (3) – Log extraction is in a recovery state and is being rebuilt.

Extraction Throttle Reason – If the extraction scan is being throttled (idle) this lists the reason why (see notes).

Current Log to Extract - The current log to extract. This is the active log file that extraction is extracting from or needs to extract from.

Extraction Throttle Reasons are detailed at https://www.ibm.com/docs/en/db2/11.5?topic=reference-l#r_ext_status.

Throttling can occur for any of the following reasons:

Throttling not enabled

- Throttling not available, for example because ALSM has been disabled.
- Throttle reason: n/a

Disk full

- During runtime and recovery undo, throttle extraction if the extraction log files were to interfere with the upper limit of the configured number of log files for the database. This rule ensures that the disk space that would be needed for extraction files is not taken away from the configured log space for the database. Running out of disk space in the active log path during regular logging activities could result in an unexpected database shutdown, therefore this condition overrules any other throttle reasons.This rule accounts for the **`logprimary`** and **`logsecond`** database configuration parameters, and ensures that extraction log files never breach this space. There are special cases when Db2 may end up with more primary log files than specified by **`logprimary`** (for example, with extraction running), or with less primary log files (for example, when the database is in the process of being activated and log files are being allocated asynchronously). Additionally, the **`logsecond`** parameter can change dynamically. The rule accounts for all such cases.

  When log spooling is enabled on an HADR standby database, extraction is throttled to protect disk space for log spooling. If the database **`hadr_spool_limit`** configuration parameter is set to a fixed value, this rule will ensure that extraction does not take away any disk space needed for log spooling. If **`hadr_spool_limit`** is set to AUTOMATIC and disk space is limited, this rule forces extraction to share disk space between itself and log spooling. If **`hadr_spool_limit`** is set to -1, then this rule will not throttle extraction.
- Throttle reason: DISK_FULL

Distance from active log file

- Throttle extraction if the log file being considered for extraction is the current active log file for writing. Extraction only works on closed log files.
- Throttle reason: CURRENT_ACTIVE_FILE

Log archiving

- Throttle extraction if log archiving has not been enabled for the database, or if the log file being considered for extraction has not yet been archived. Active log files waiting to be archived will always stay in the active log path, and extraction would only duplicate disk space without any added benefit.
- Throttle reason: LOG_ARCHIVING

Log space usage

- Throttle extraction when consumed active log space is below a calculated threshold. This rule is designed to save system resources and ensures that extraction will only start when the database is close to running out of the configured log space.
- Throttle reason: DB_LOG_SPACE_USED

Extraction ratio

- Throttle extraction if the sum of the extracted data exceeds the calculated configured log space percentage limit. The purpose of this rule is to prevent extraction for very large transactions, thus potentially duplicating what is found in the active log files.
- Throttle reason: EXTRACTION_RATIO

New extraction zone

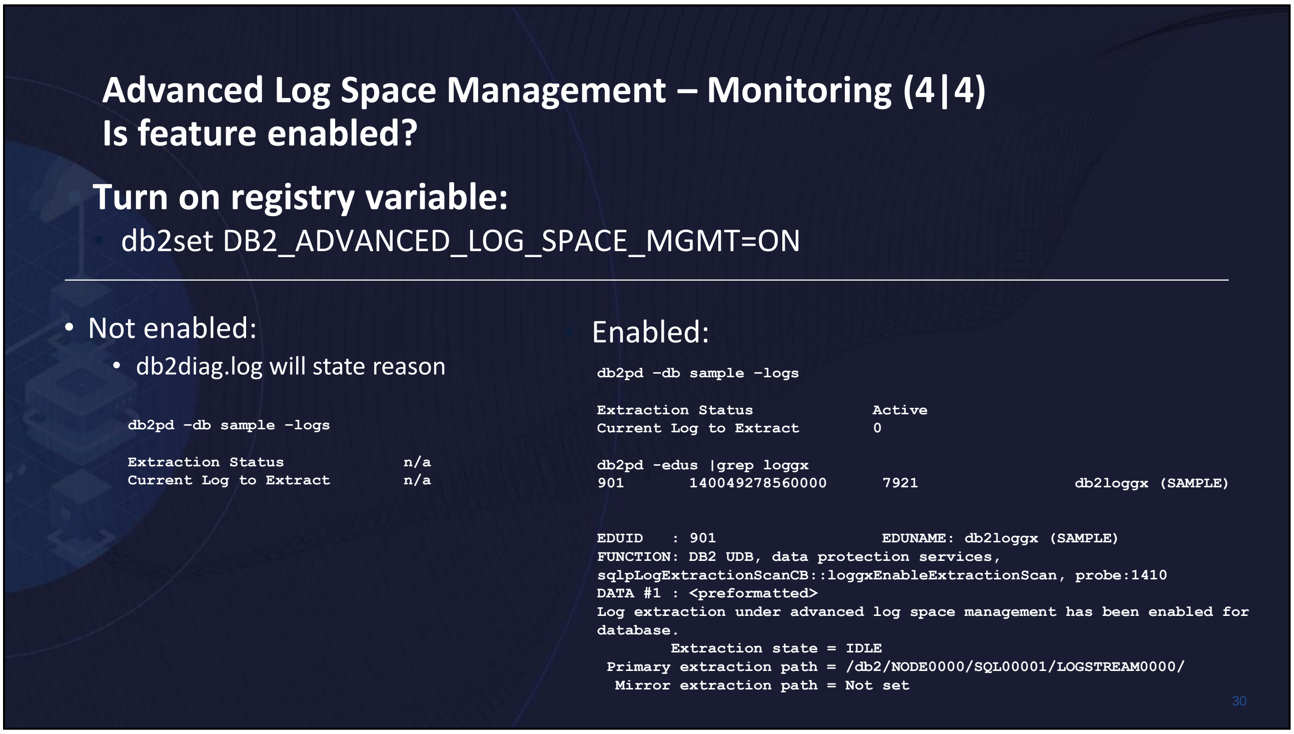
- Do not extract data if a new extraction zone has been detected and any extraction log files before this zone are no longer needed. An extraction zone is the range of log records from start to end that the extraction scan will need to process.
- Throttle reason: NEW_EXTRACTION_ZONE

Buffer pool flush needed

- Throttle extraction if the currently processed log record has not yet been flushed to the disk. Log records that have not been flushed from the buffer pool will always be needed for recovery purposes, and thus would always need to be extracted. This can be controlled by the database configuration parameters **PAGE_AGE_TRGT_MCR** and **PAGE_AGE_TRGT_GCR**
- Throttle reason: SLOW_BP_FLUSH

Previous extraction error

- Throttle extraction if the currently processed log file needs to be skipped. Upon encountering certain types of errors, the extraction scanner may decide to skip the currently processed log file and restart the scan in the next log file.
- Throttle reason: SCAN_ERROR



Advanced Log Space Management – Monitoring (4|4)
Is feature enabled?

Turn on registry variable:

- db2set DB2_ADVANCED_LOG_SPACE_MGMT=ON

- Not enabled:
  - db2diag.log will state reason

```
db2pd -db sample -logs

Extraction Status               n/a
Current Log to Extract          n/a
```

Enabled:

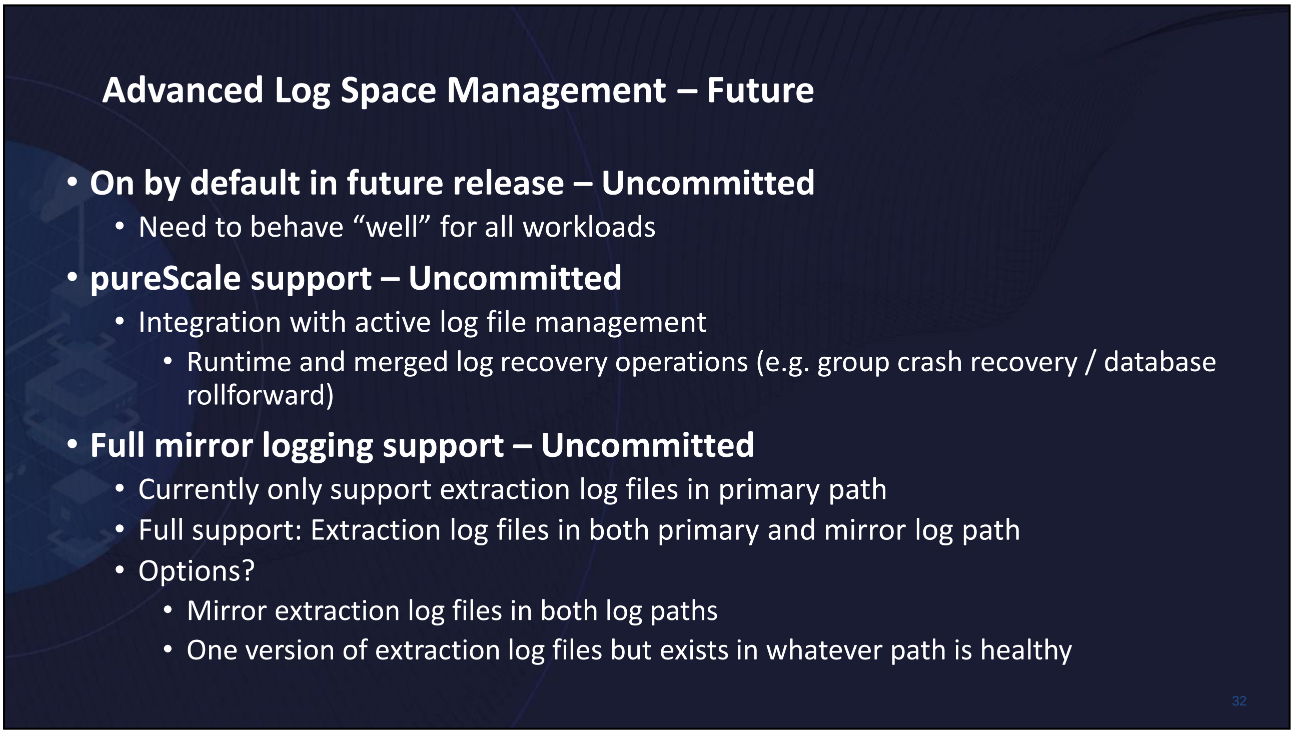
```
db2pd -db sample -logs

Extraction Status               Active
Current Log to Extract          0

db2pd -edus |grep loggx
901       140049278560000      7921                db2loggx (SAMPLE)


EDUID   : 901                  EDUNAME: db2loggx (SAMPLE)
FUNCTION: DB2 UDB, data protection services,
sqlpLogExtractionScanCB::loggxEnableExtractionScan, probe:1410
DATA #1 : <preformatted>
Log extraction under advanced log space management has been enabled for
database.
        Extraction state = IDLE
 Primary extraction path = /db2/NODE0000/SQL00001/LOGSTREAM0000/
  Mirror extraction path = Not set
```

30

Turn on the registry variable DB2_ADVANCED_LOG_SPACE_MGMT and activate database. The db2diag.log will display a message whether log extraction is enabled or not. db2pd –logs will also show state of extraction. All extraction is done by the new db2loggx EDU.

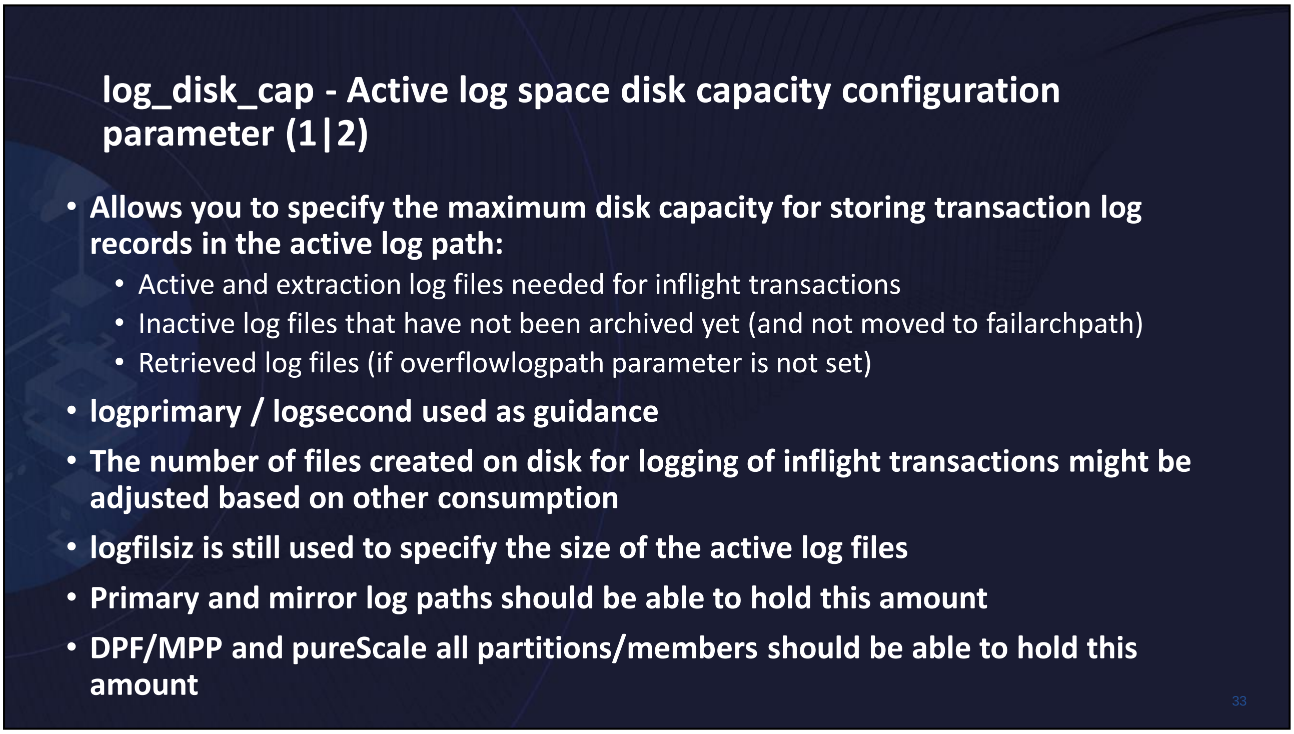
Agenda

- Advanced Log Space Management
  - Overview
  - Monitoring
  - Future



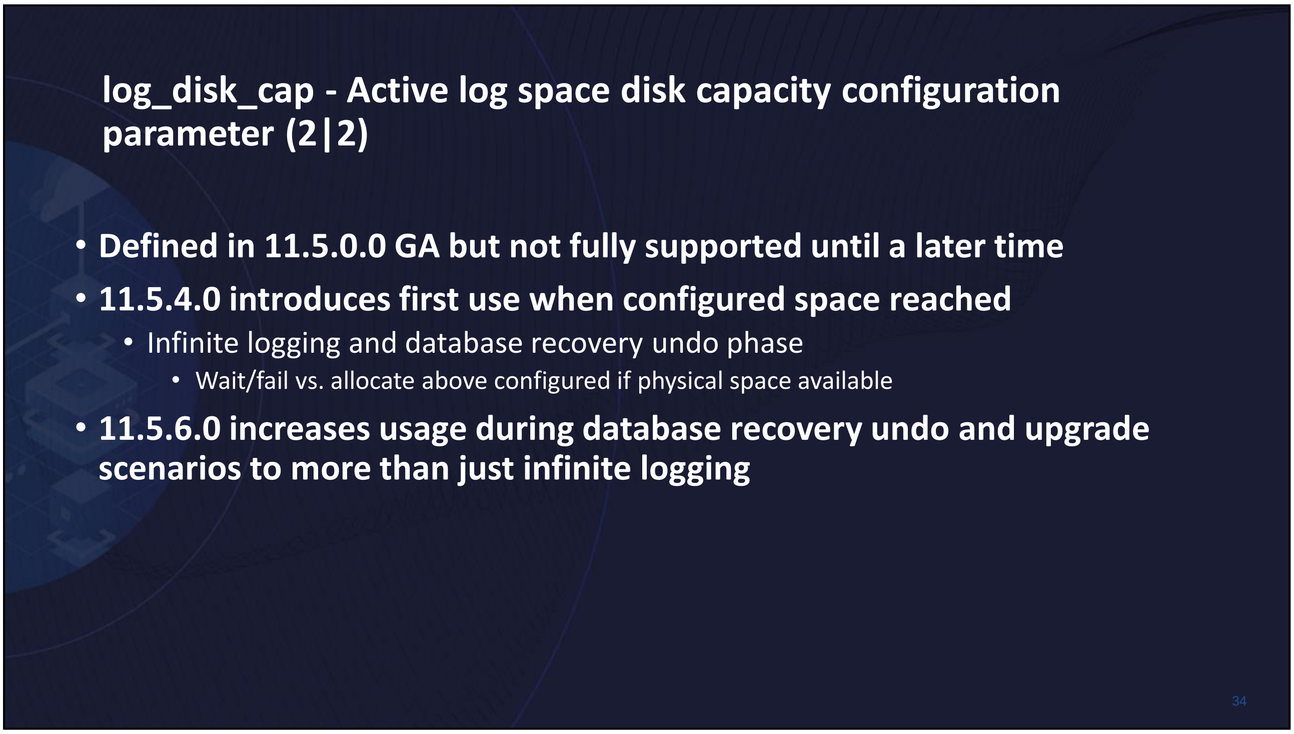
Advanced Log Space Management – Future

- **On by default in future release – Uncommitted**
  - Need to behave “well” for all workloads
- **pureScale support – Uncommitted**
  - Integration with active log file management
    - Runtime and merged log recovery operations (e.g. group crash recovery / database rollforward)
- **Full mirror logging support – Uncommitted**
  - Currently only support extraction log files in primary path
  - Full support: Extraction log files in both primary and mirror log path
  - Options?
    - Mirror extraction log files in both log paths
    - One version of extraction log files but exists in whatever path is healthy



log_disk_cap - Active log space disk capacity configuration parameter (1|2)

- **Allows you to specify the maximum disk capacity for storing transaction log records in the active log path:**
  - Active and extraction log files needed for inflight transactions
  - Inactive log files that have not been archived yet (and not moved to failarchpath)
  - Retrieved log files (if overflowlogpath parameter is not set)
- **logprimary / logsecond used as guidance**
- **The number of files created on disk for logging of inflight transactions might be adjusted based on other consumption**
- **logfilsiz is still used to specify the size of the active log files**
- **Primary and mirror log paths should be able to hold this amount**
- **DPF/MPP and pureScale all partitions/members should be able to hold this amount**



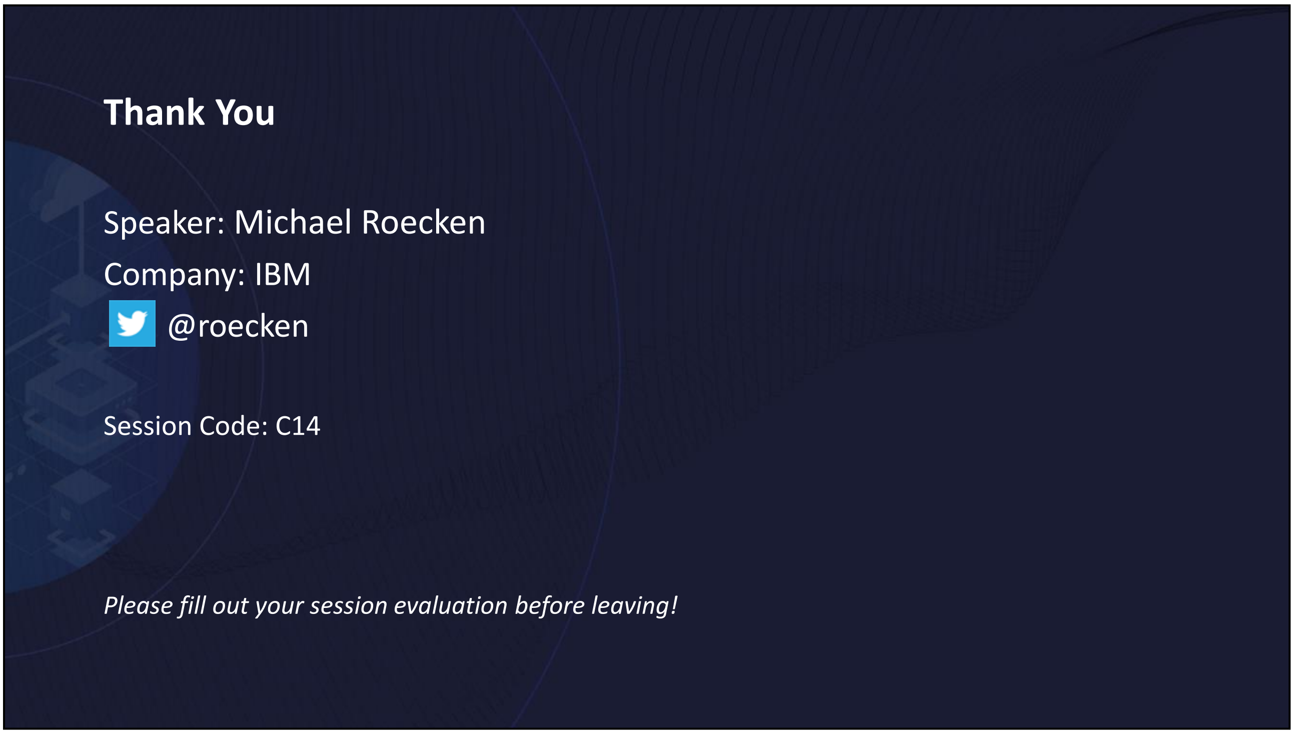
log_disk_cap - Active log space disk capacity configuration parameter (2|2)

- **Defined in 11.5.0.0 GA but not fully supported until a later time**
- **11.5.4.0 introduces first use when configured space reached**
  - Infinite logging and database recovery undo phase
    - Wait/fail vs. allocate above configured if physical space available
- **11.5.6.0 increases usage during database recovery undo and upgrade scenarios to more than just infinite logging**

Resources

- IBM Documentation
  - https://www.ibm.com/docs/en/db2/11.5?topic=logging-advanced-log-space-management
- What's New 11.5.6.0
  - https://www.ibm.com/docs/en/db2/11.5?topic=new-1156
- What's New 11.5.7.0
  - https://www.ibm.com/docs/en/db2/11.5?topic=new-1157

Questions ???



Thank You

Speaker: Michael Roecken

Company: IBM

@roecken

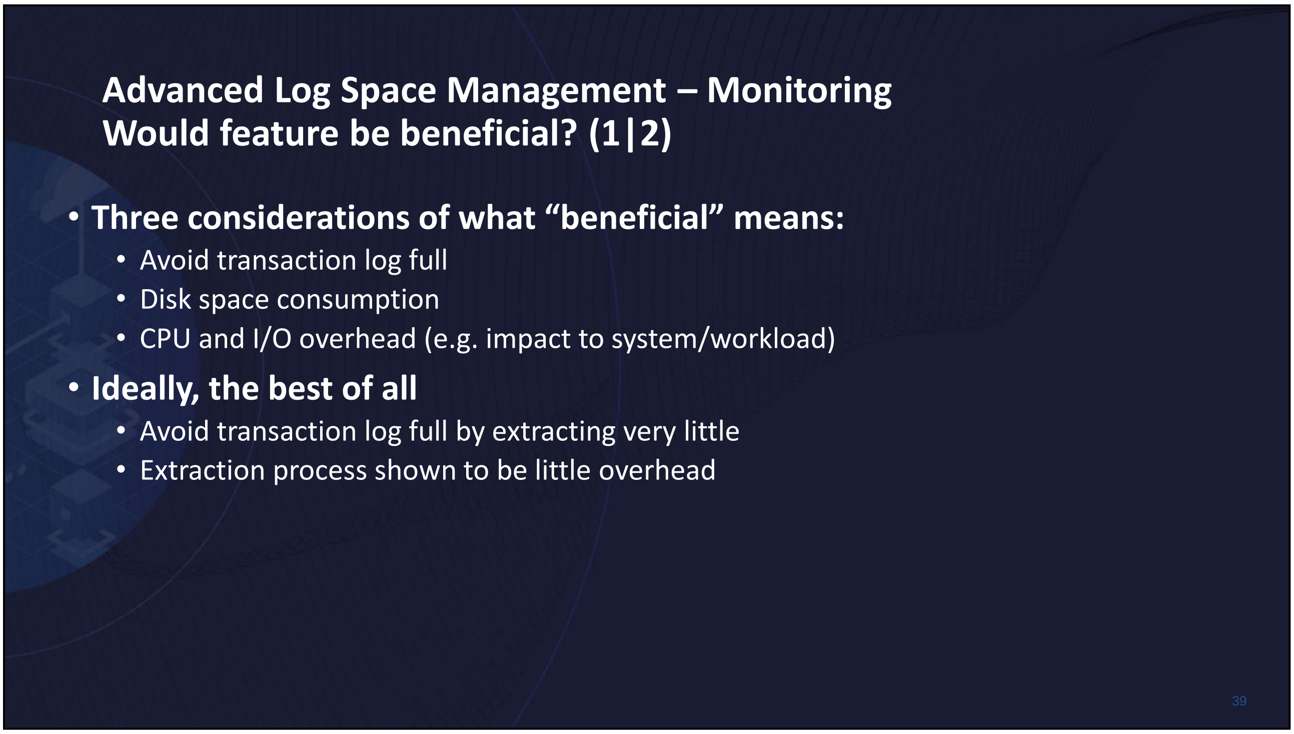
Session Code: C14

Please fill out your session evaluation before leaving!

Michael Roecken is a senior software developer with Db2 for Linux, UNIX, and Windows platforms at the IBM Toronto Lab. Michael has worked since 2000 designing, implementing and supporting various features and capabilities in the areas of: backup/restore, crash/rollforward recovery, high availability/disaster recovery, and logging/transaction management.

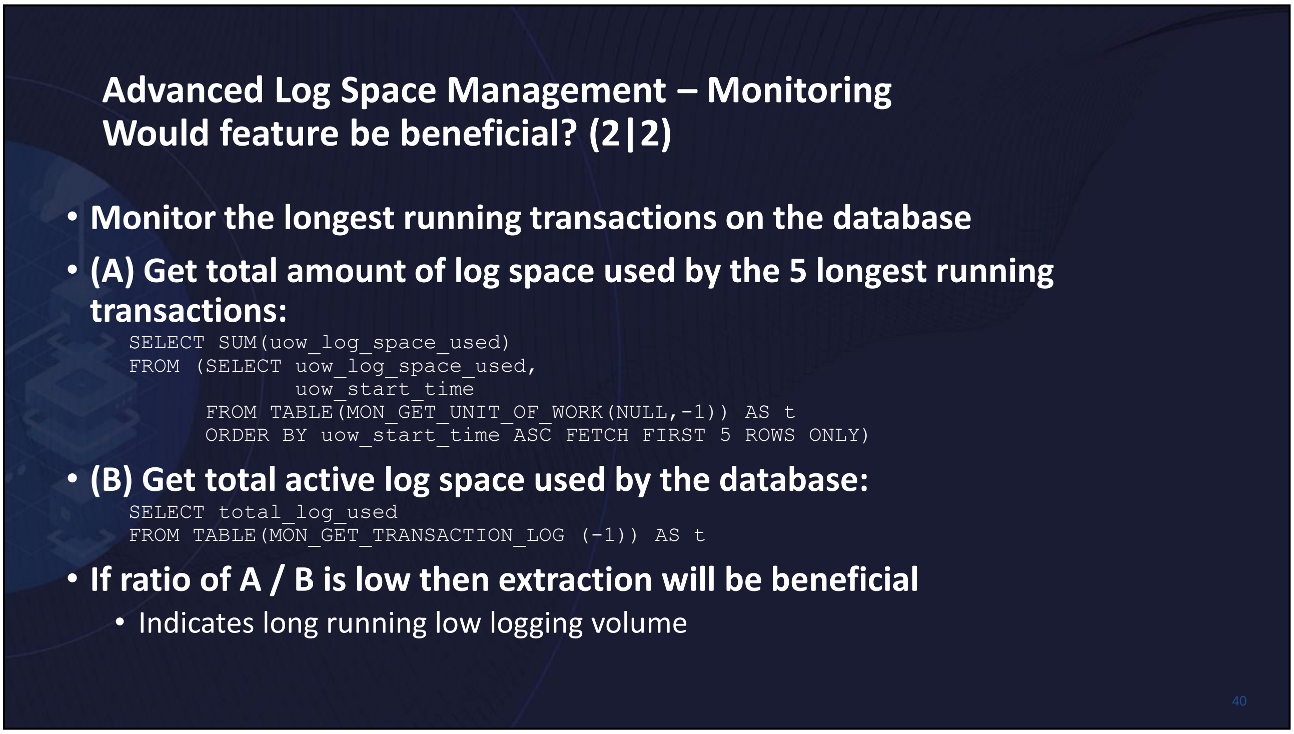
Please connect with Michael on Twitter at @roecken and LinkedIn.

Backup Slides



Advanced Log Space Management – Monitoring Would feature be beneficial? (1|2)

- **Three considerations of what “beneficial” means:**
  - Avoid transaction log full
  - Disk space consumption
  - CPU and I/O overhead (e.g. impact to system/workload)
- **Ideally, the best of all**
  - Avoid transaction log full by extracting very little
  - Extraction process shown to be little overhead



Advanced Log Space Management – Monitoring Would feature be beneficial? (2|2)

- **Monitor the longest running transactions on the database**
- **(A) Get total amount of log space used by the 5 longest running transactions:**

```
SELECT SUM(uow_log_space_used)
FROM (SELECT uow_log_space_used,
             uow_start_time
      FROM TABLE(MON_GET_UNIT_OF_WORK(NULL,-1)) AS t
      ORDER BY uow_start_time ASC FETCH FIRST 5 ROWS ONLY)
```

- **(B) Get total active log space used by the database:**

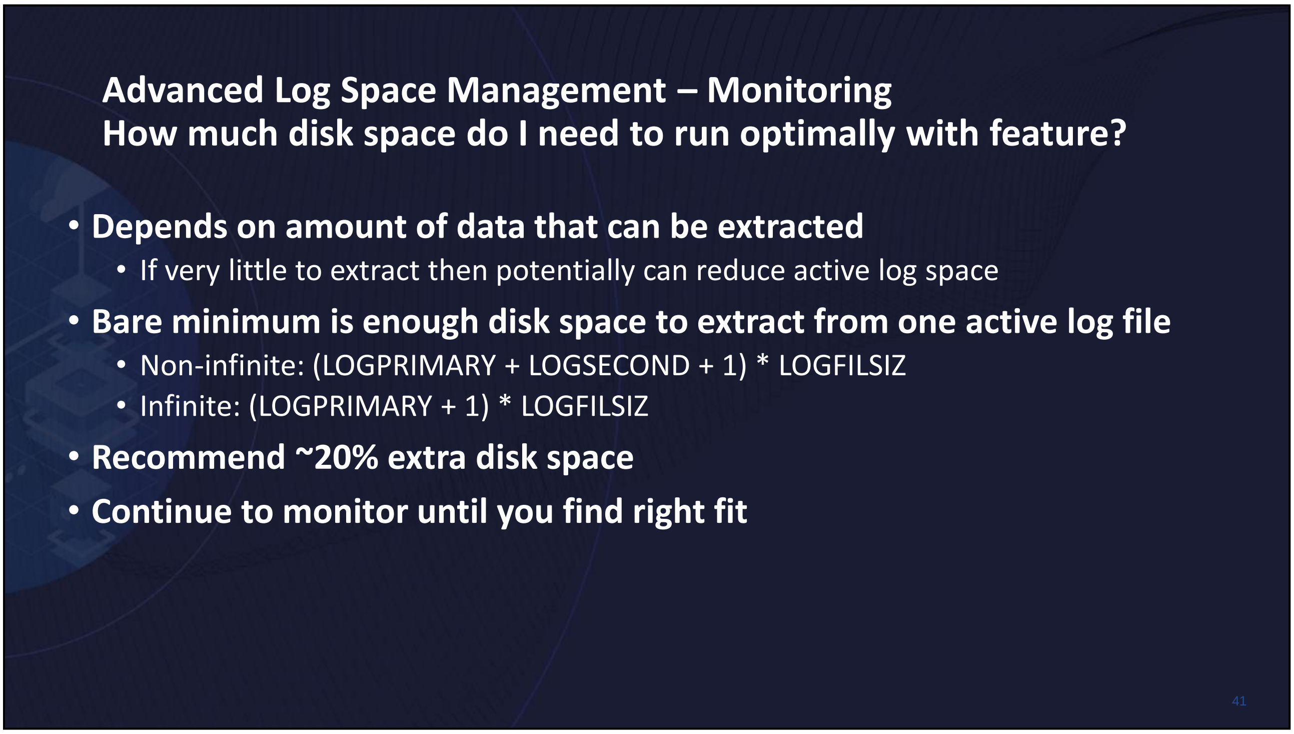
```
SELECT total_log_used
FROM TABLE(MON_GET_TRANSACTION_LOG (-1)) AS t
```

- **If ratio of A / B is low then extraction will be beneficial**
  - Indicates long running low logging volume

40

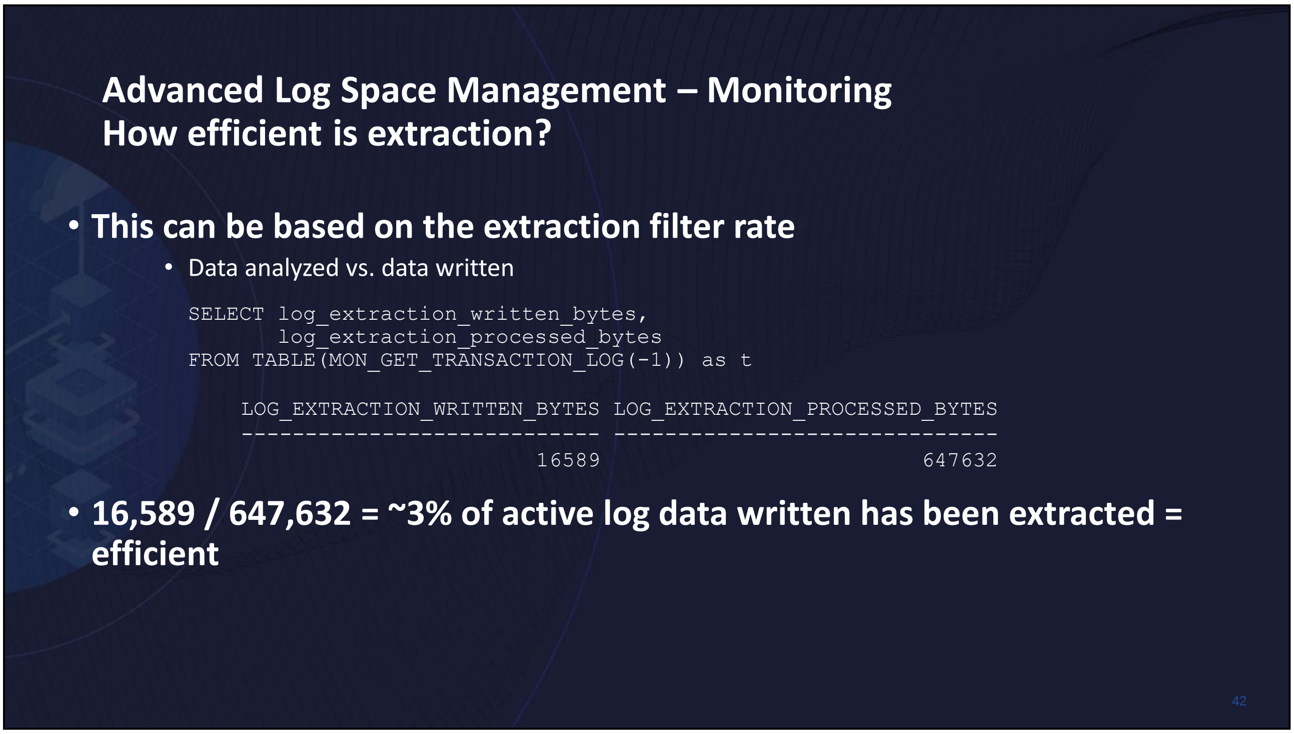
> "If ratio of A / B is low then extraction will be beneficial"

"low" here is relative. The lower the number the better the disk space saving. So need to decide what amount of disk space you are willing to save in order to avoid transaction log full.



Advanced Log Space Management – Monitoring
How much disk space do I need to run optimally with feature?

- **Depends on amount of data that can be extracted**
  - If very little to extract then potentially can reduce active log space
- **Bare minimum is enough disk space to extract from one active log file**
  - Non-infinite: (LOGPRIMARY + LOGSECOND + 1) * LOGFILSIZ
  - Infinite: (LOGPRIMARY + 1) * LOGFILSIZ
- **Recommend ~20% extra disk space**
- **Continue to monitor until you find right fit**

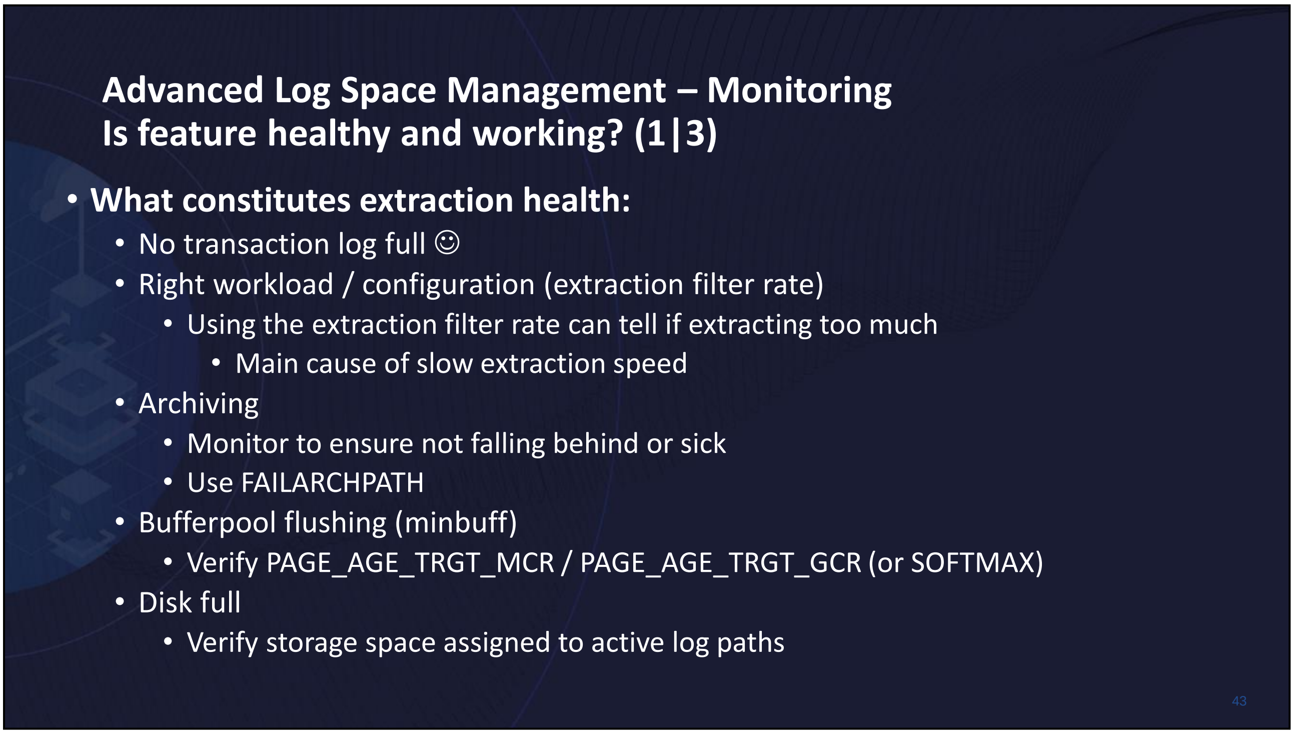


Advanced Log Space Management – Monitoring
How efficient is extraction?

- **This can be based on the extraction filter rate**
  - Data analyzed vs. data written

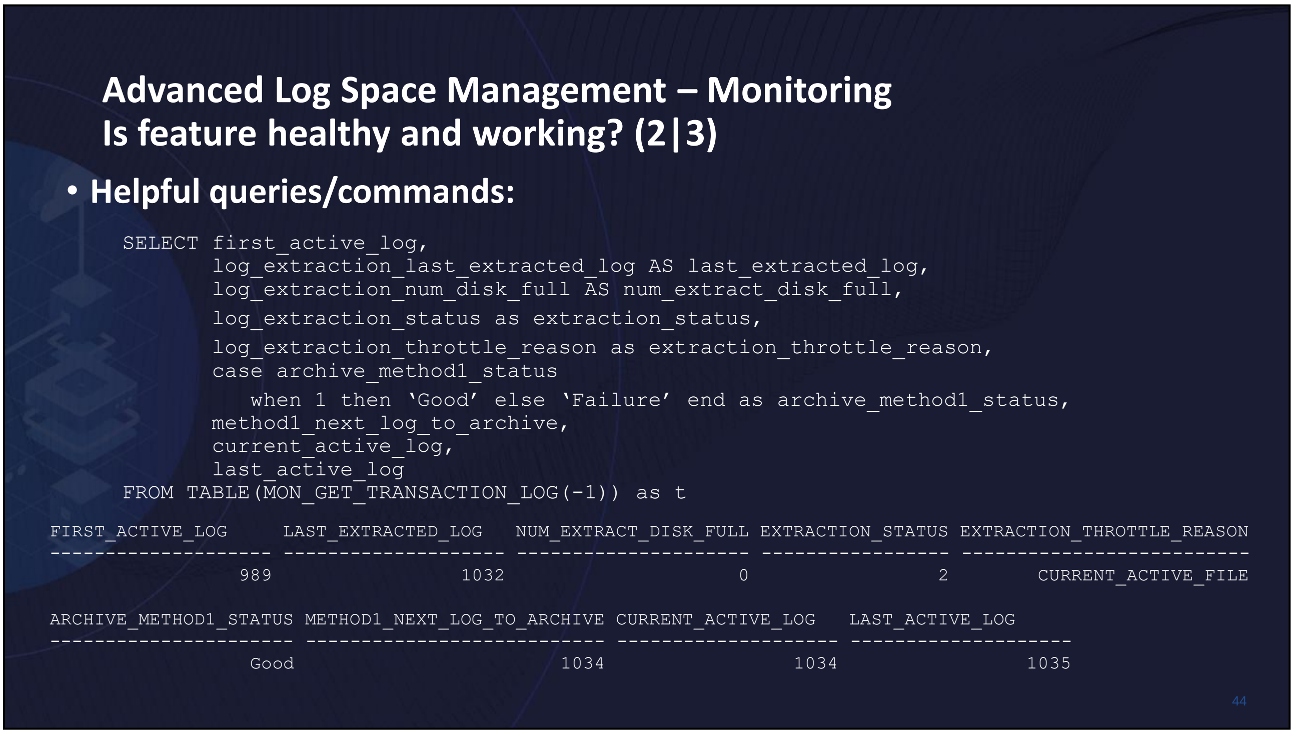
```
SELECT log_extraction_written_bytes,
       log_extraction_processed_bytes
FROM TABLE(MON_GET_TRANSACTION_LOG(-1)) as t

    LOG_EXTRACTION_WRITTEN_BYTES LOG_EXTRACTION_PROCESSED_BYTES
    ---------------------------- ------------------------------
                           16589                         647632
```

- **16,589 / 647,632 = ~3% of active log data written has been extracted = efficient**



Advanced Log Space Management – Monitoring
Is feature healthy and working? (1|3)

- **What constitutes extraction health:**
  - No transaction log full ☺
  - Right workload / configuration (extraction filter rate)
    - Using the extraction filter rate can tell if extracting too much
      - Main cause of slow extraction speed
  - Archiving
    - Monitor to ensure not falling behind or sick
    - Use FAILARCHPATH
  - Bufferpool flushing (minbuff)
    - Verify PAGE_AGE_TRGT_MCR / PAGE_AGE_TRGT_GCR (or SOFTMAX)
  - Disk full
    - Verify storage space assigned to active log paths



Advanced Log Space Management – Monitoring
Is feature healthy and working? (2|3)

- **Helpful queries/commands:**

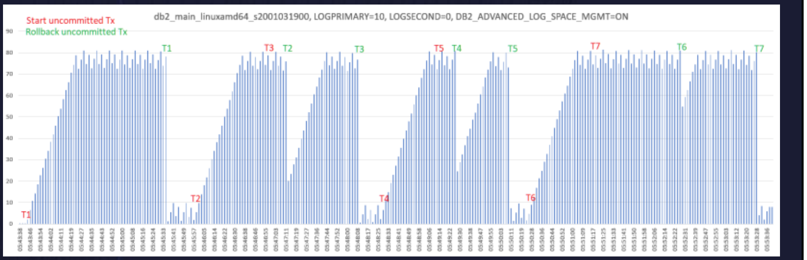
```
SELECT first_active_log,
       log_extraction_last_extracted_log AS last_extracted_log,
       log_extraction_num_disk_full AS num_extract_disk_full,
       log_extraction_status as extraction_status,
       log_extraction_throttle_reason as extraction_throttle_reason,
       case archive_method1_status
          when 1 then ‘Good’ else ‘Failure’ end as archive_method1_status,
       method1_next_log_to_archive,
       current_active_log,
       last_active_log
FROM TABLE(MON_GET_TRANSACTION_LOG(-1)) as t
```

```
FIRST_ACTIVE_LOG     LAST_EXTRACTED_LOG   NUM_EXTRACT_DISK_FULL EXTRACTION_STATUS EXTRACTION_THROTTLE_REASON
-------------------- -------------------- --------------------- ----------------- --------------------------
                 989                 1032                     0                 2        CURRENT_ACTIVE_FILE

ARCHIVE_METHOD1_STATUS METHOD1_NEXT_LOG_TO_ARCHIVE CURRENT_ACTIVE_LOG   LAST_ACTIVE_LOG
---------------------- --------------------------- -------------------- --------------------
                  Good                        1034                 1034                 1035
```

Advanced Log Space Management – Monitoring
Is feature healthy and working? (3|3)

- **db2pd -db sample -logs**

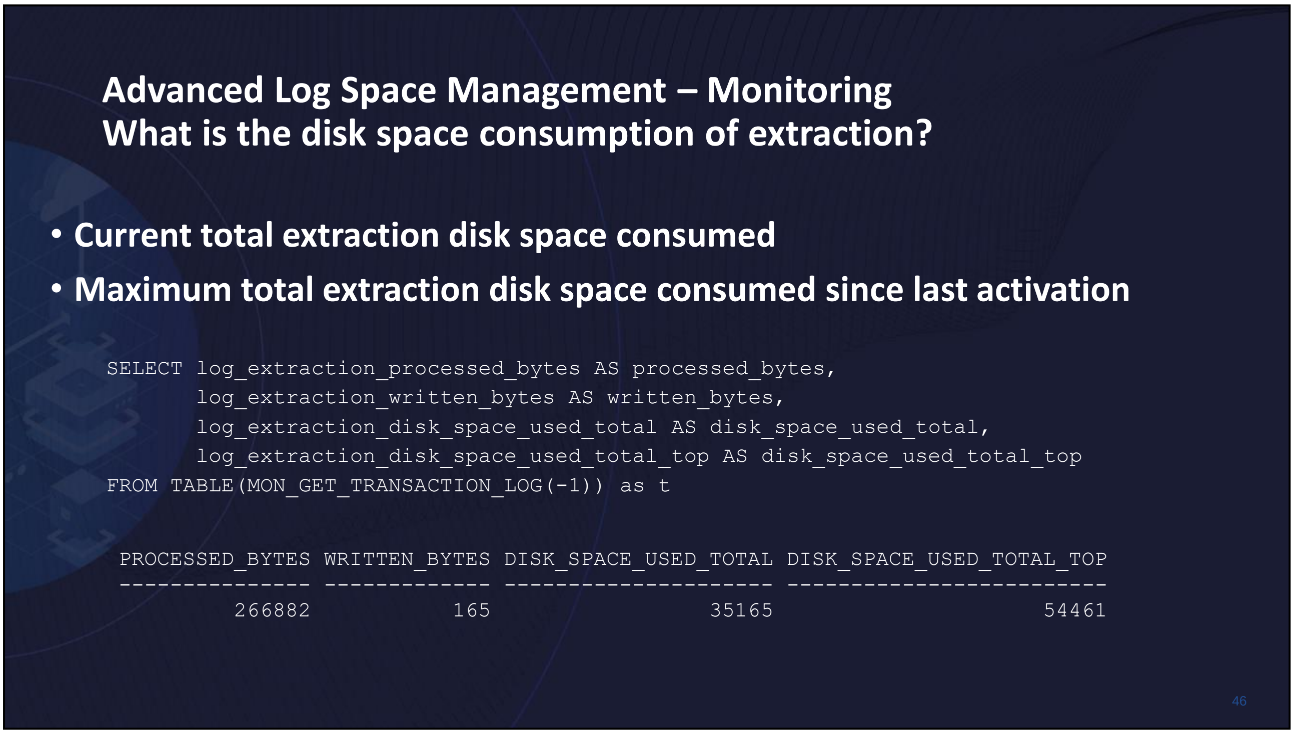
```
Logs:
Current Log Number            1034
…
Method 1 Archive Status       Success
Method 1 Next Log to Archive  1034
…
Extraction Status             Active
Extraction Throttle Reason    NONE
Current Log to Extract        1033
…
```



```
StartLSN         StartLSO         State      Size       Pages      Filename
00000000000729AB 78259201         0x00000000 4          4          S0001026.LOG
…
0000000000000000 78405937         0x00000000 4          4          S0001035.LOG
```

No log data will be extracted from an active log file that has not been archived yet. This would duplicate disk space. Ensure methx_status is 1 (healthy), not 0 (error).

No log data will be extracted from an active log file where minbufflsn exists. This is due to recovery algorithm that needs to replay all log records >= minbufflsn. So no benefit of extracting such data as it would duplicate disk space.



Advanced Log Space Management – Monitoring
What is the disk space consumption of extraction?

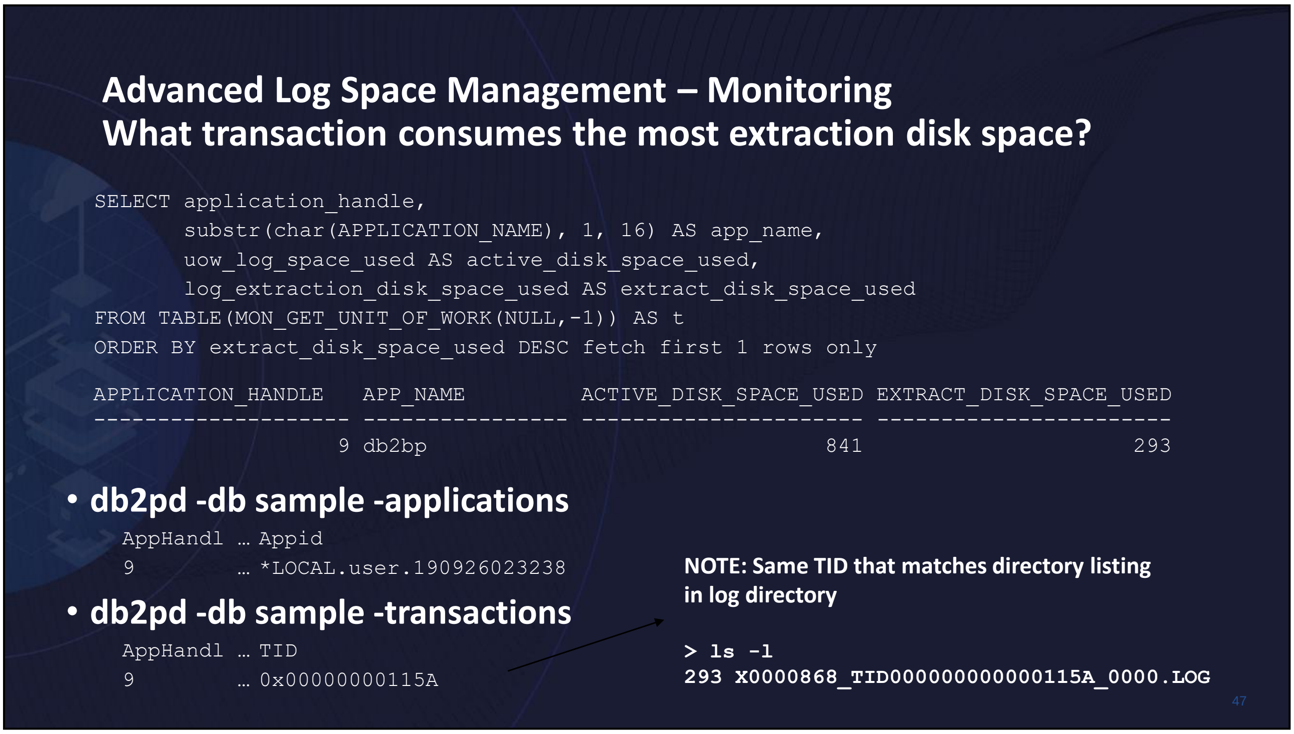
- Current total extraction disk space consumed
- Maximum total extraction disk space consumed since last activation

```
SELECT log_extraction_processed_bytes AS processed_bytes,
       log_extraction_written_bytes AS written_bytes,
       log_extraction_disk_space_used_total AS disk_space_used_total,
       log_extraction_disk_space_used_total_top AS disk_space_used_total_top
FROM TABLE(MON_GET_TRANSACTION_LOG(-1)) as t
```

```
 PROCESSED_BYTES WRITTEN_BYTES DISK_SPACE_USED_TOTAL DISK_SPACE_USED_TOTAL_TOP
 --------------- ------------- --------------------- -------------------------
          266882           165                 35165                     54461
```

46

This query tells you that since the last activation extraction processed 266,882 bytes of log data from the active log files. From that amount, 165 bytes of log data was written to extraction TID files. The current total amount of disk space consumed by extraction files, including log data and meta data, is 35,165 bytes. Since the last activation, extraction has taken up 54,461 bytes.



Advanced Log Space Management – Monitoring
What transaction consumes the most extraction disk space?

```
SELECT application_handle,
       substr(char(APPLICATION_NAME), 1, 16) AS app_name,
       uow_log_space_used AS active_disk_space_used,
       log_extraction_disk_space_used AS extract_disk_space_used
FROM TABLE(MON_GET_UNIT_OF_WORK(NULL,-1)) AS t
ORDER BY extract_disk_space_used DESC fetch first 1 rows only

APPLICATION_HANDLE   APP_NAME         ACTIVE_DISK_SPACE_USED EXTRACT_DISK_SPACE_USED
-------------------- ---------------- ---------------------- -----------------------
                   9 db2bp                               841                     293
```

- **db2pd -db sample -applications**

```
AppHandl … Appid
9        … *LOCAL.user.190926023238
```

- **db2pd -db sample -transactions**

```
AppHandl … TID
9        … 0x00000000115A
```

NOTE: Same TID that matches directory listing in log directory

```
> ls -l
293 X0000868_TID000000000000115A_0000.LOG
```

47

To find the transaction that is consuming the most extraction log space allows one to understand if this is a known expectation or not, maybe possibly a rogue transaction.

You can map a transaction ID (TID) from a directory listing or you can use a combination of commands to track down which application/transaction is consuming the most amount of extraction space.

The above example shows that this particular transaction has written 841 bytes of log data to the active files, but only 293 bytes have been extracted so far.

- **What's New in Version 11.5.6.0**
- What's New in Version 11.5.7.0

What’s New in Version 11.5.6.0 (1|5)

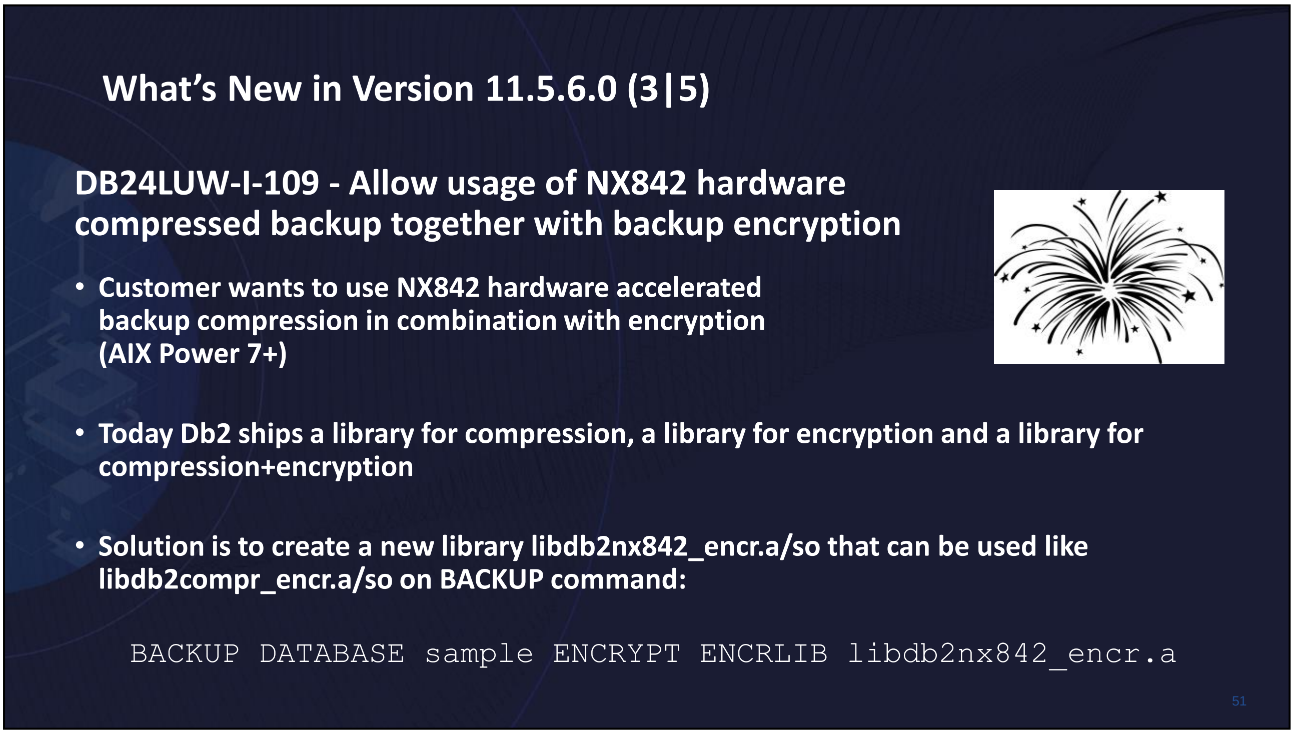
DB24LUW-I-1038 - db2adutl performance

- **db2adult delete takes long**
  - Deletes objects one by one using TSM delete API
  - High load on TSM server interferes with new workload
- **Re-wrote to batch as many objects (the TSM server allows) into one TSM delete API call**
- **Internal tests showed 3x improvement**

What’s New in Version 11.5.6.0 (2|5)

DB24LUW-I-941 - Tool to format log file

- **Problem:**
  - Way to tell: which tables are being changed, by what operations, the transaction id, the time of commit, and from whom. The current db2fmtlog tool does not format the log records
  - Help identify a good point-in-time to use for ROLLFORWARD in DPF
- **Goal:**
  - Provide all the internal/developer options of the db2fmtlog tool to customers such that the customer and internal versions of the tool have the same functionality
- **Highlights:**
  - Previously, a customer needed to enable the service password to access additional options, this is not needed any more
  - The customer version will not have the "-userdata" option due to data security concerns

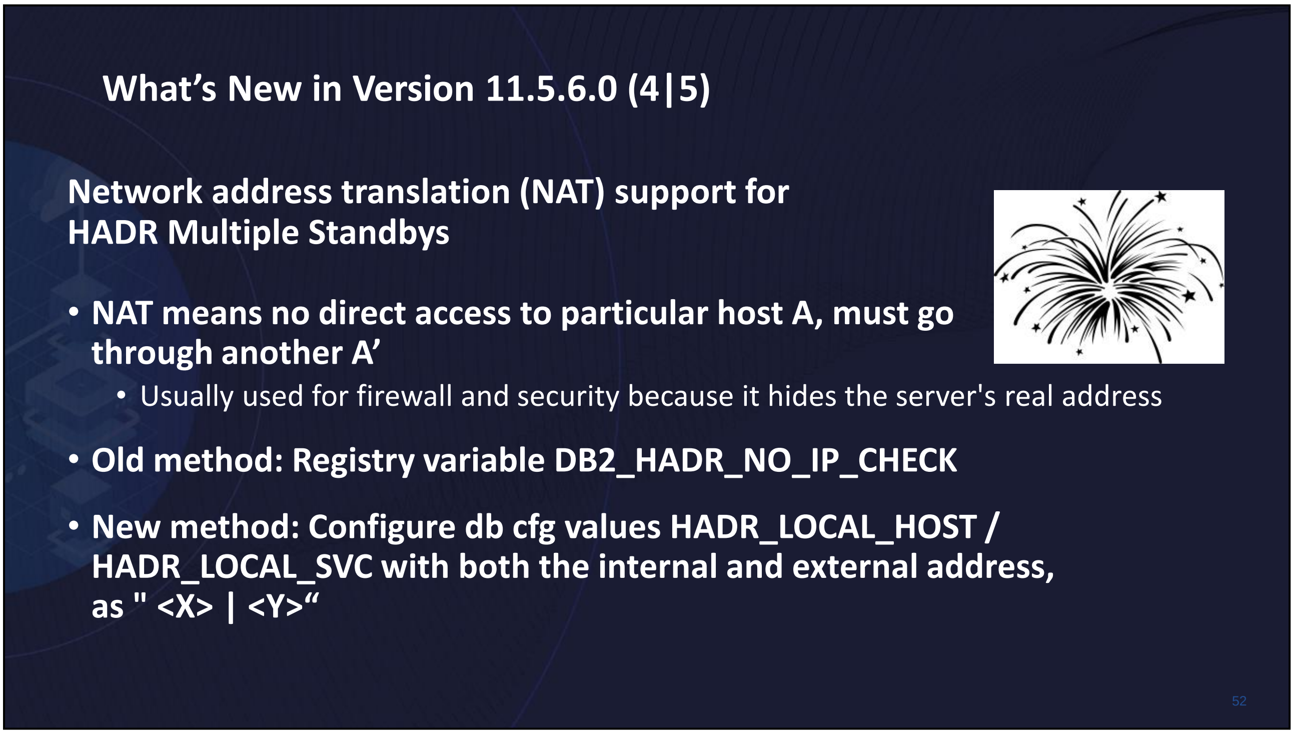


What’s New in Version 11.5.6.0 (3|5)

DB24LUW-I-109 - Allow usage of NX842 hardware compressed backup together with backup encryption

- Customer wants to use NX842 hardware accelerated backup compression in combination with encryption (AIX Power 7+)
- Today Db2 ships a library for compression, a library for encryption and a library for compression+encryption
- Solution is to create a new library libdb2nx842_encr.a/so that can be used like libdb2compr_encr.a/so on BACKUP command:

```
BACKUP DATABASE sample ENCRYPT ENCRLIB libdb2nx842_encr.a
```

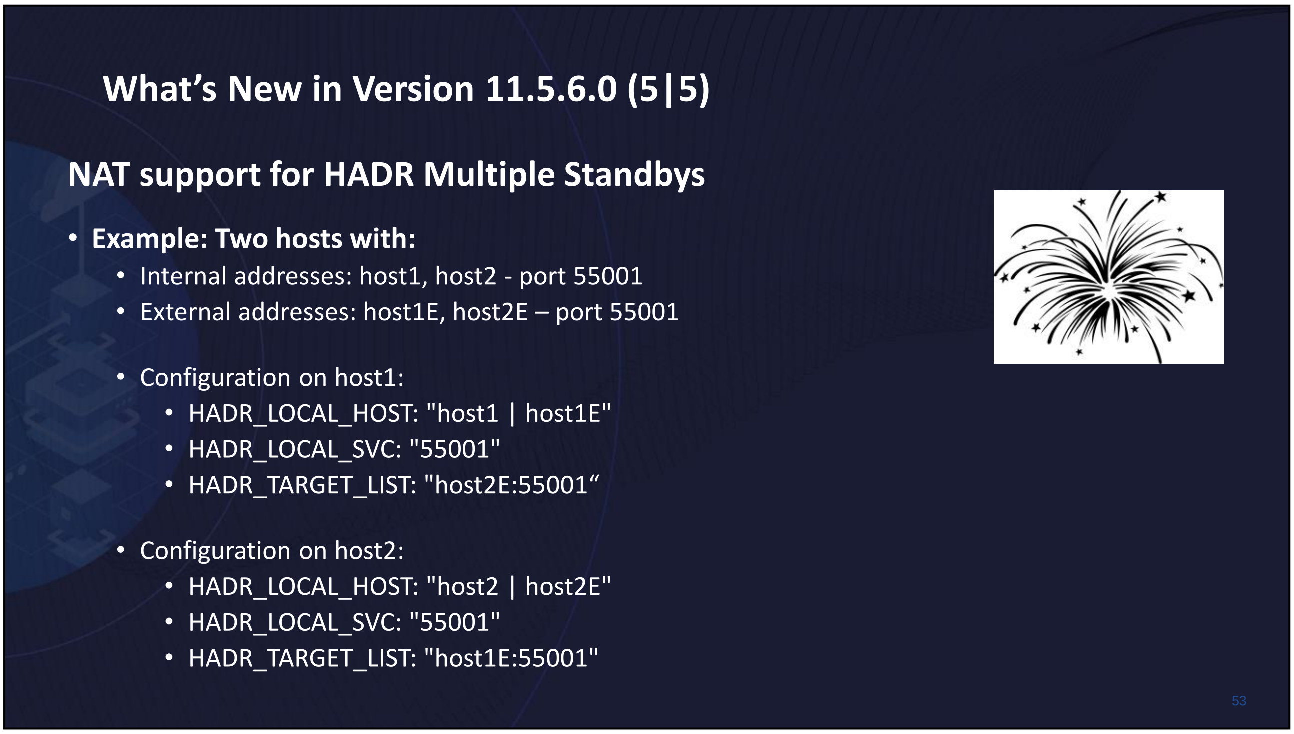


What’s New in Version 11.5.6.0 (4|5)

Network address translation (NAT) support for HADR Multiple Standbys

- **NAT means no direct access to particular host A, must go through another A’**
  - Usually used for firewall and security because it hides the server's real address
- **Old method: Registry variable DB2_HADR_NO_IP_CHECK**
- **New method: Configure db cfg values HADR_LOCAL_HOST / HADR_LOCAL_SVC with both the internal and external address, as " <X> | <Y>“**

52



What’s New in Version 11.5.6.0 (5|5)

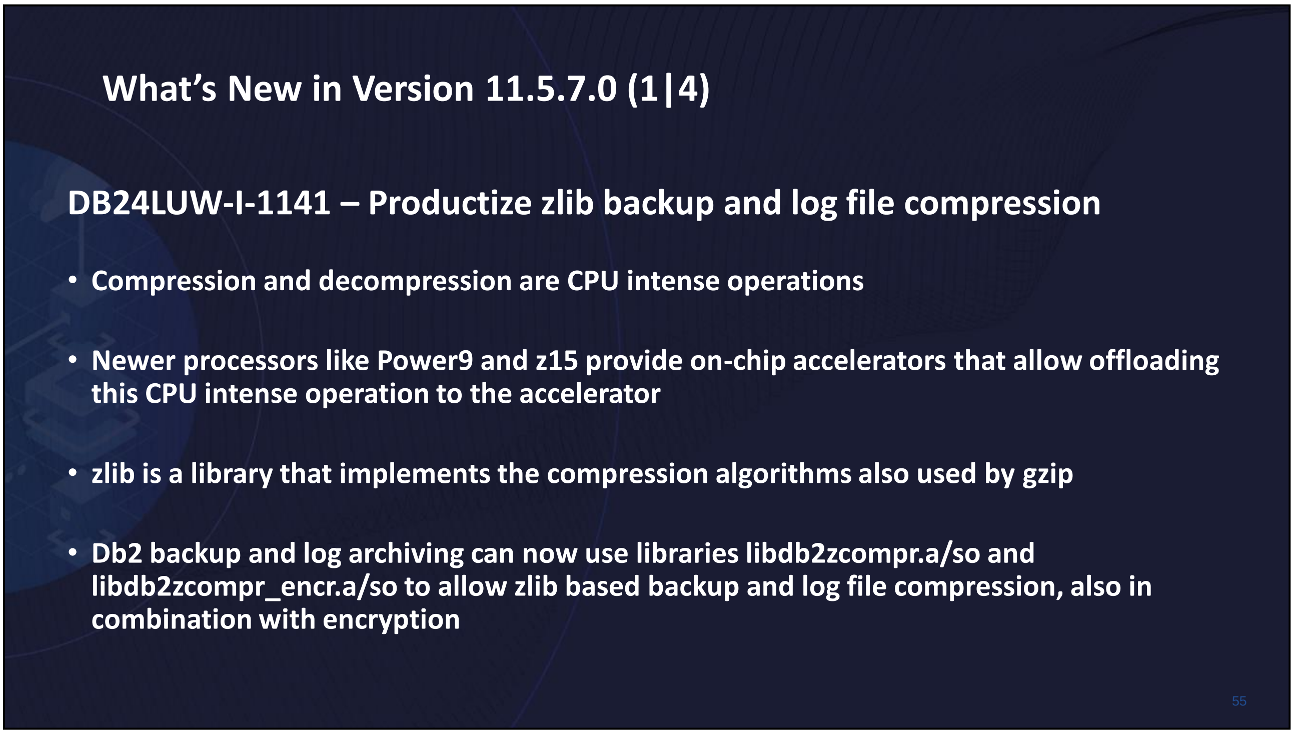
NAT support for HADR Multiple Standbys

- **Example: Two hosts with:**
  - Internal addresses: host1, host2 - port 55001
  - External addresses: host1E, host2E – port 55001

  - Configuration on host1:
    - HADR_LOCAL_HOST: "host1 | host1E"
    - HADR_LOCAL_SVC: "55001"
    - HADR_TARGET_LIST: "host2E:55001“

  - Configuration on host2:
    - HADR_LOCAL_HOST: "host2 | host2E"
    - HADR_LOCAL_SVC: "55001"
    - HADR_TARGET_LIST: "host1E:55001"

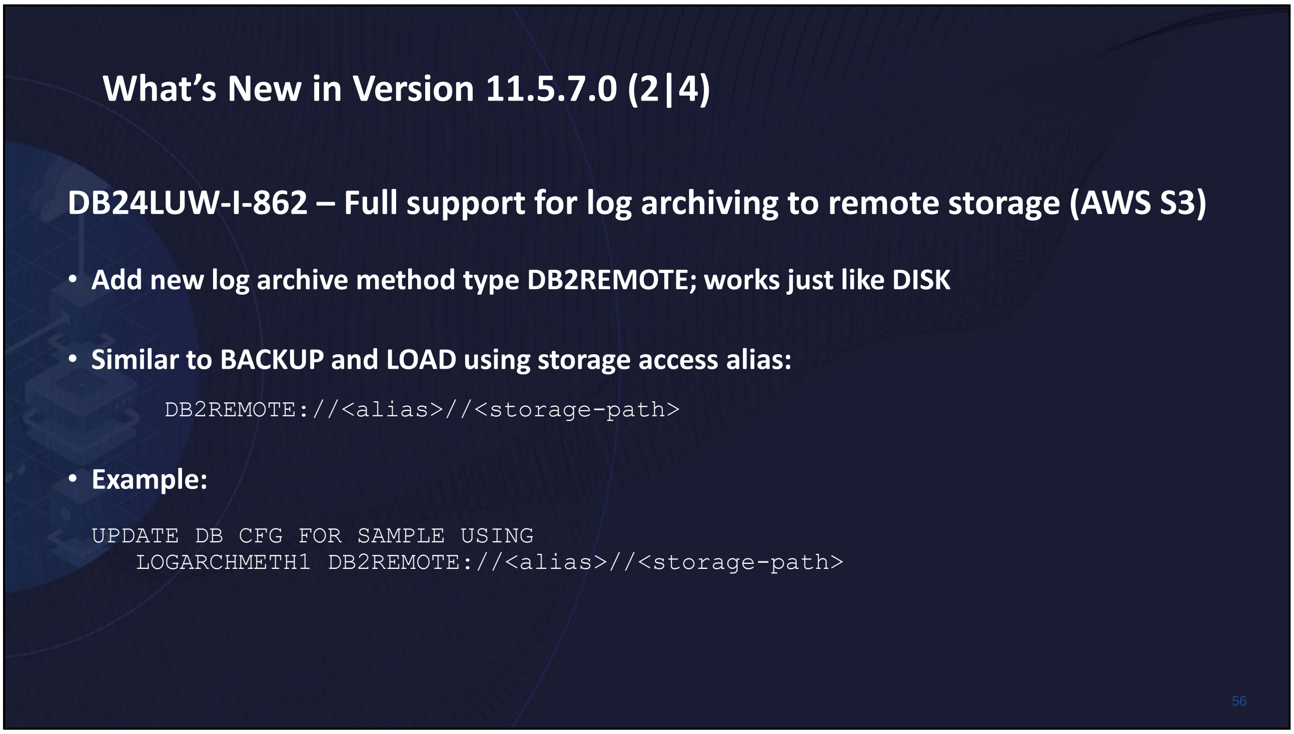
- What’s New in Version 11.5.6.0
- **What’s New in Version 11.5.7.0**



What’s New in Version 11.5.7.0 (1|4)

DB24LUW-I-1141 – Productize zlib backup and log file compression

- Compression and decompression are CPU intense operations
- Newer processors like Power9 and z15 provide on-chip accelerators that allow offloading this CPU intense operation to the accelerator
- zlib is a library that implements the compression algorithms also used by gzip
- Db2 backup and log archiving can now use libraries libdb2zcompr.a/so and libdb2zcompr_encr.a/so to allow zlib based backup and log file compression, also in combination with encryption



What’s New in Version 11.5.7.0 (2|4)

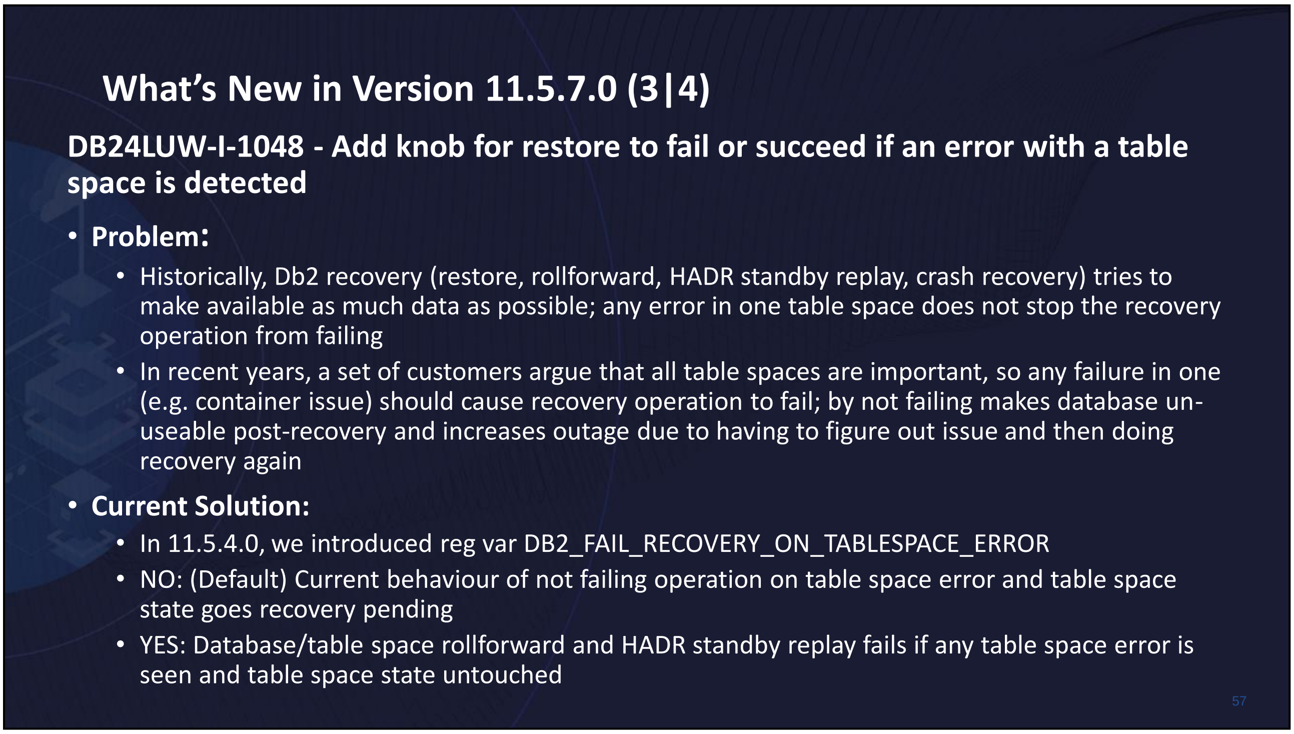
DB24LUW-I-862 – Full support for log archiving to remote storage (AWS S3)

- **Add new log archive method type DB2REMOTE; works just like DISK**
- **Similar to BACKUP and LOAD using storage access alias:**

```
DB2REMOTE://<alias>//<storage-path>
```

- **Example:**

```
UPDATE DB CFG FOR SAMPLE USING
   LOGARCHMETH1 DB2REMOTE://<alias>//<storage-path>
```



What’s New in Version 11.5.7.0 (3|4)

DB24LUW-I-1048 - Add knob for restore to fail or succeed if an error with a table space is detected

- **Problem:**
  - Historically, Db2 recovery (restore, rollforward, HADR standby replay, crash recovery) tries to make available as much data as possible; any error in one table space does not stop the recovery operation from failing
  - In recent years, a set of customers argue that all table spaces are important, so any failure in one (e.g. container issue) should cause recovery operation to fail; by not failing makes database un-useable post-recovery and increases outage due to having to figure out issue and then doing recovery again
- **Current Solution:**
  - In 11.5.4.0, we introduced reg var DB2_FAIL_RECOVERY_ON_TABLESPACE_ERROR
  - NO: (Default) Current behaviour of not failing operation on table space error and table space state goes recovery pending
  - YES: Database/table space rollforward and HADR standby replay fails if any table space error is seen and table space state untouched

What’s New in Version 11.5.7.0 (4|4)

DB24LUW-I-1048 - Add knob for restore to fail or succeed if an error with a table space is detected (cont’d)

- **In 11.5.7.0:**
  - Add restore to DB2_FAIL_RECOVERY_ON_TABLESPACE_ERROR
  - If set to YES, if restore has an issue with a table space container the restore will fail
    - Will consider any error when restoring the table space definition