Db2 13 DBA topics

ZPARM changes
REBIND simplification
Column names > 30 bytes
Insert performance for PBGs
Online conversion from PBG to PBR
Application deadlock and timeout control
Profile enhancements including local threads
Profiles: Monitoring connections for security
Profiles: new PRDID values
Online DELETE of active log data set
Index page split monitoring
Ownership flexibility
Concurrent ALTER DATA CAPTURE
Statement level dependency

Removed ZPARMs

FL 100

- HONOR_KEEPDICTIONARY
- IX_TB_PART_CONV_EXCLUDE
- DSVCI
- IMMEDWRI
- SUBQ_MIDX
- DDF_COMPATIBILITY
- PLANMGMTSCOPE
- REALSTORAGE_MANAGEMENT
- EXTRAREQ
- EXTRASRV
- MAXARCH
- MAXTYPE1
- OPT1ROWBLOCKSORT
- PARA_EFF
- RESYNC
- TRACSTR

These ZPARMs are removed in Db2 13 because they are deprecated, obsolete, or rarely utilized.
The behaviors follow the default settings.

ZPARM default change – PAGESET_PAGENUM

FL 100

Purpose of ZPARM: specifies default page numbering scheme used for PBR table spaces

- Db2 12 default: ABSOLUTE
- Db2 13 default: RELATIVE

Reason: going forward, relative page numbering (RPN) is best practice for PBR table spaces

- RPN advantages versus absolute page numbering:
  - Much greater data capacity
  - Maximum number of partitions not affected by choice of page size or DSSIZE
  - DSSIZE can be specified at partition level (and DSSIZE increase is immediate change)
- Also: when PBG table space is online-converted to PBR via Db2 13 enhancement, RPN is used for the new PBR table space
- And conversion from absolute to relative page numbering requires online REORG of entire table space, and that gets more expensive as table space gets larger – best to start out with RPN for a PBR table space

ZPARM default change – STATIME_MAIN

FL 100

STATIME_MAIN is one of two ZPARMs that determine the intervals at which Db2 statistics trace data will be generated (the other is STATIME)

Many of the statistics most important for diagnosing Db2 performance issues are generated at the interval specified via STATIME_MAIN

– Most notably, IFCIDs 0001 and 0002

The Db2 12 default STATIME_MAIN interval of 60 seconds is in some cases not granular enough to allow precise identification of a performance issue

With Db2 13, the default value for STATIME_MAIN is changed from 60 to 10 seconds

– The resulting finer-grained statistics interval should enhance diagnosis of performance issues

– CPU cost of default statistics trace classes remains negligible with this change

This is one of two Db2 13 changes that enhance diagnostic value of statistics trace information…

Aggregated accounting data in statistics trace class 1

FL 100

When IFCID 369 is active, it generates useful aggregated accounting data that can be included in a Db2 monitor-generated statistics report (below is an example of IFCID 369-generated data)

```
CONNTYPE   CL1 ELAPSED       CL1 CPU    CL1 SE CPU   CL2 ELAPSED       CL2 CPU    CL2 SE CPU      CL3 SUSP   CL2 NOT ACC  QUANTITY
--------  ------------  ------------  ------------  ------------  ------------  ------------  ------------  ------------  --------
BATCH     2:09:03.8976   1:54.670005      2.077926   3:48.608261   1:20.627524      2.077912   1:55.361532     32.722852   2673.00
CICS       19 21:44:57  5:53:38.4157      0.000000  20:47:58.981  3:12:33.4938      0.000000  15:47:46.693  1:47:38.7947   9189.8K
DDF       2 06:52:05.5  47:14.403228  1:27:43.9224  10:43:04.666  43:08.696542  1:23:20.6799  5:53:42.4927  4:06:13.4765  13196.1K
IMS                N/P           N/P           N/P           N/P           N/P           N/P           N/P           N/P      0.00
RRSAF     3 08:13:01.7     41.407479   1:29.233962   4:14.494267     28.994667   1:29.226115     33.980175   3:11.519425  24991.00
UTILITY      28.089146      1.678974      0.944658     21.912913      1.539285      0.944658     18.391770      1.981859     40.00
```

Db2 12: IFCID 369 was associated with statistics trace class 9 – not often started by users

Db2 13: IFCID 369 is added into statistics trace class 1 – one of the default statistics classes

– *IFCID 369 can still be activated via statistics trace class 9 in Db2 13 environment, for backward compatibility*

With this change, helpful aggregated accounting data will be more readily available than before

Updated ZPARM defaults

FL 100

- DDF - From NO to AUTO
- FTB_NON_UNIQUE_INDEX - From NO to YES
- MAXSORT_IN_MEMORY – From 1000 (KB) to 2000 (KB)
- SRTPOOL - From 10000 (KB) to 20000 (KB)
- EDM_SKELETON_POOL - From 51200 to 81920
- EDMDBDC - From 23400 to 40960
- MAXCONQN - From OFF to ON
- MAXCONQW – From OFF to ON
- NUMLKTS - From 2000 to 5000
- NUMLKUS - From 10000 to 20000
- OUTBUFF - From 4000 (KB) to 102400 (KB)

The default values of these ZPARMs are updated in Db2 13 to reflect best practices or more typical usage.

New max value for DSMAX

FL 100

– DSMAX – specifies the maximum number data sets that can be open at one time

– The new highest possible value is changed from 200,000 to 400,000

– Technical details of this change are covered in the scalability topic of this workshop

REBIND simplification … enhancement

FL 100

Enhancements:

- Improved REBIND performance for APREUSE(ERROR/WARN)
  - Skips reading of referenced object catalog statistics
- Reduced storage usage during BIND/REBIND for packages with many tables
- Reduced number of REBIND errors & warnings
  - When using APREUSE(ERROR/WARN) during new version migration
  - When an access path is not available in new version but was available in prior version
  - When query rewrite was used
  - (also available in Db2 12 via PH36728/UI75603)

Column names over 30 bytes

FL 100

Db2 12 limits column names to 30 bytes

- Primarily due to the length of the SQLNAME field of the SQLDA structure for host language programs

Db2 13 enhancement – allows you to define a column with a name longer than 30 bytes, up to **128 bytes**

- A new subsystem parameter is introduced, **TABLE_COL_NAME_EXPANSION**, to externalize the new subsystem parameter SPRMTCNE
- Db2 13 default: OFF
- Recommendation: set TABLE_COL_NAME_EXPANSION to the same value on all members of a data sharing group
- Note: Although you will be able to define a column with a name up to 128 bytes, column names with a length greater than 30 bytes may be truncated on a character boundary to at most 30 bytes when they are returned in an SQLDA. APIs that do not use the SQLDA to obtain a column name may return complete column names.

Improving insert success rate for PBG table spaces (1|3)

The challenge: insert failures for partition-by-growth table space with multiple partitions

- Assume a PBG table space with 7 partitions in a Db2 12 environment
- Based on clustering key value for to-be-inserted row, target page is in partition 4 – what happens next?
  - Db2 requests conditional IX lock on partition 4 (*conditional*: if not acquired right away, move on)
  - If IX lock cannot be acquired on partition 4, Db2 searches backwards or forwards through other partitions (backward/forward direction determined randomly for a given insert)
  - For each "next in line" partition, Db2 may either fail to get IX partition lock, or find that partition is full
  - After all partitions checked without success, insert will fail with -904 (resource unavailable), and reason code 00C90090 (partition lock failure) or 00C9009C (partition full)
  - Even worse, in case of 00C9009C (partition full): user might subsequently check the table space and find that there is plenty of space to accommodate new rows

Db2 13: smarter space search for insert into PBG (2|3)

FL 100

Main reason for insert failures described on preceding slide: Db2 12 tried only once to get conditional lock on a partition prior to performing insert

- Incompatible lock that blocked conditional IX partition lock request often has short duration – possible that conditional lock request *would have been successful if retried*

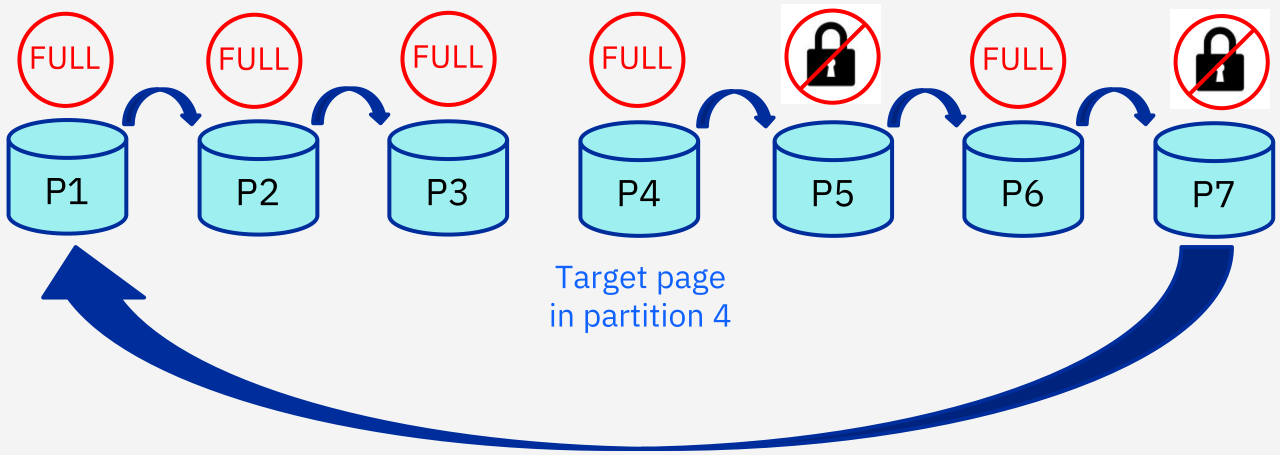
Db2 13 enhancement – new processing path for row insert:

- Db2 keeps track, in memory, of partitions for which initial conditional lock request failed
- Db2 also uses an in-memory partition bit map to track full partitions
- If all partitions are checked without success for an insert, Db2 will try again for the conditional lock on up to 5 of the partitions for which the lock could not initially be acquired
  - If those conditional lock retries are unsuccessful, Db2 will request an unconditional lock and will wait for IRLM timeout period to acquire the lock – if unsuccessful, then issue -904 with code 00C90090
- If Db2 sees that all partitions are full, a new partition will be added for insert of the row

A little more on smarter space search for PBG insert (3|3)

FL 100

The big picture:



Failed to get conditional lock on parts: P5, P7

Full partitions:
P1, P2, P3. P4, P6

Action: retry conditional lock requests for partitions P5, P7

Note: in data sharing environment, tracking of full partitions done at member level

– Could be variances due to workload – different members may have slightly different partition-full information

– This is consequence of objective: strike the right balance between optimal space search, optimal performance

Online conversion from PBG to PBR (1|5)

FL 500

- Partition-by-growth (PBG) is considered the default type of universal table space (UTS) and most table spaces converted to UTS are PBG
- PBG works well for small and medium sized tables
- For larger tables, partition-by-range (PBR) has several advantages over PBG:
  - Greater insert throughput
  - Enhanced query performance
  - Easier to maintain clustering within partitions
  - Ability to have partitioned indexes
  - Maximizes utility independence and parallelism

Online conversion from PBG to PBR (2|5)

FL 500

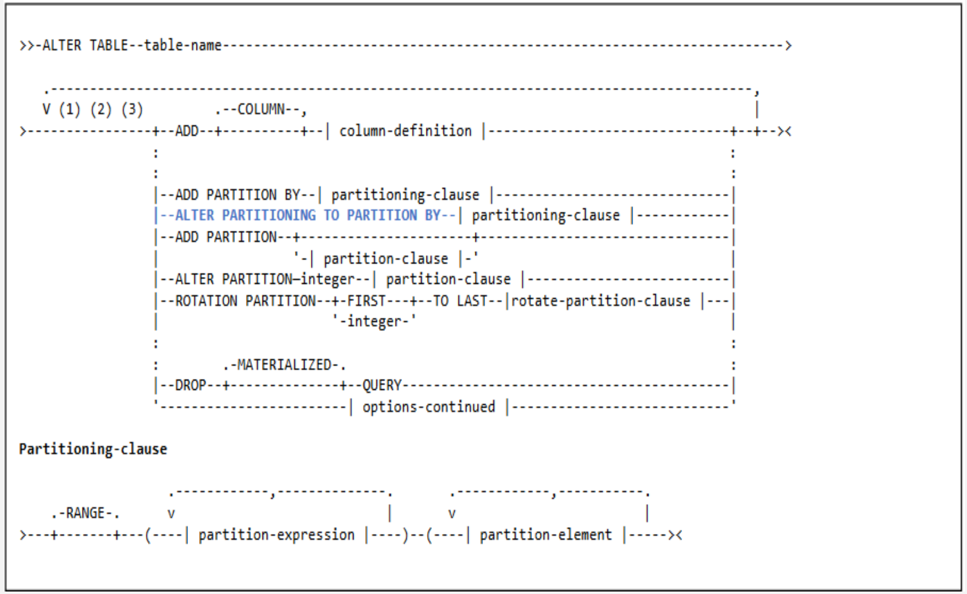
- Db2 13: introduced the capability to convert a table's partitioning scheme from partition-by-growth (PBG) table space to partition-by-range (PBR) table space online
- The partitioning scheme is altered directly to partition-by-range with relative page numbering
- The new PBR table space does not need to have the same number of partitions as the prior PBG table space. It can have more, the same or fewer partitions
- The existing indexes on the table are handled as a part of the conversion process
  - Db2 does not change any aspects or attributes of those indexes
- Users may create partitioned indexes on the table as desired after the conversion has been completed

Online conversion from PBG to PBR (3|5)

FL 500

– The ALTER TABLE statement has been enhanced with a new ALTER PARTITIONING clause

```
ALTER TABLE E8054.TB01
ALTER PARTITIONING TO PARTITION BY
  RANGE (COLINT, COLCHAR)
(PARTITION 1 ENDING AT ( 5, ‘CCC’),
 PARTITION 2 ENDING AT (10, ‘MMM’),
 PARTITION 3 ENDING AT (MAXVALUE,
  MAXVALUE))
```



```
>>-ALTER TABLE--table-name------------------------------------------------------------------->

   .-------------------------------------------------------------------------------------------,
   V (1) (2) (3)          .--COLUMN--,                                                          |
>-----------------+--ADD--+----------+--| column-definition |-----------------------------+--+--><
                  :                                                                       :
                  :                                                                       :
                  |--ADD PARTITION BY--| partitioning-clause |----------------------------|
                  |--ALTER PARTITIONING TO PARTITION BY--| partitioning-clause |----------|
                  |--ADD PARTITION--+--------------------+-------------------------------|
                  |                 '-| partition-clause |-'                              |
                  |--ALTER PARTITION—integer--| partition-clause |------------------------|
                  |--ROTATION PARTITION--+-FIRST---+--TO LAST--|rotate-partition-clause |---|
                  |                      '-integer-'                                      |
                  :                                                                       :
                  :        .-MATERIALIZED-.                                               :
                  |--DROP--+--------------+--QUERY----------------------------------------|
                  '-----------------------| options-continued |---------------------------'

Partitioning-clause

                      .-----------,------------.     .-----------,----------.
    .-RANGE-.         v                        |     v                      |
>---+-------+---(----| partition-expression |----)--(----| partition-element |----><
```

Online conversion from PBG to PBR (4|5)

FL 500

- Determine a suitable partitioning scheme to use for the table, including the columns that will define the partitions and the limit key values for each partition, and evaluate the following considerations:
  - The number of partitions that will be created
    - Note that if the table to be converted is defined with DATA CAPTURE CHANGES, then the number of partitions in the new PBR table space cannot be less than the number of partitions in the old PBG table space
  - The data set size for each partition
    - Initial DSSIZE for the new PBR table space is inherited from the old PBG table space
    - Either ensure that each partition of new PBR table space can fit within that DSSIZE, or if necessary, alter the PBG table space to have a larger DSSIZE (if fix for APAR PH51359 is applied, DSSIZE change and ALTER PARTITIONING change can be put into effect via one online REORG; otherwise, those two changes will require two REORGs of the table space)
- You can use the RUNSTATS utility to collect useful statistics for planning the range-partitioning scheme

Online conversion from PBG to PBR (5|5)

FL 500

- REORG TABLESPACE SHRLEVEL REFERENCE or CHANGE must be run to materialize the ALTER TABLE ALTER PARTITIONING TO PARTITION BY RANGE pending definition change
- The entire table space needs to be reorganized to convert it from PBG to PBR after the pending definition change has been issued
- The high limit key for the last partition requires MAXVALUE for ascending key columns or MINVALUE for descending key columns
- Consider creating partitioned indexes on the table to support parallel processing advantages
- The materializing REORG invalidates dependent packages
- Table space cannot be recovered to a point in time prior to the materializing REORG
  - You can still run the UNLOAD utility on the old image copies of the table space or partitions created prior to REORG materialization for data mining or recovery purposes, and the LOAD utility can be used to reload the data into a different tablespace

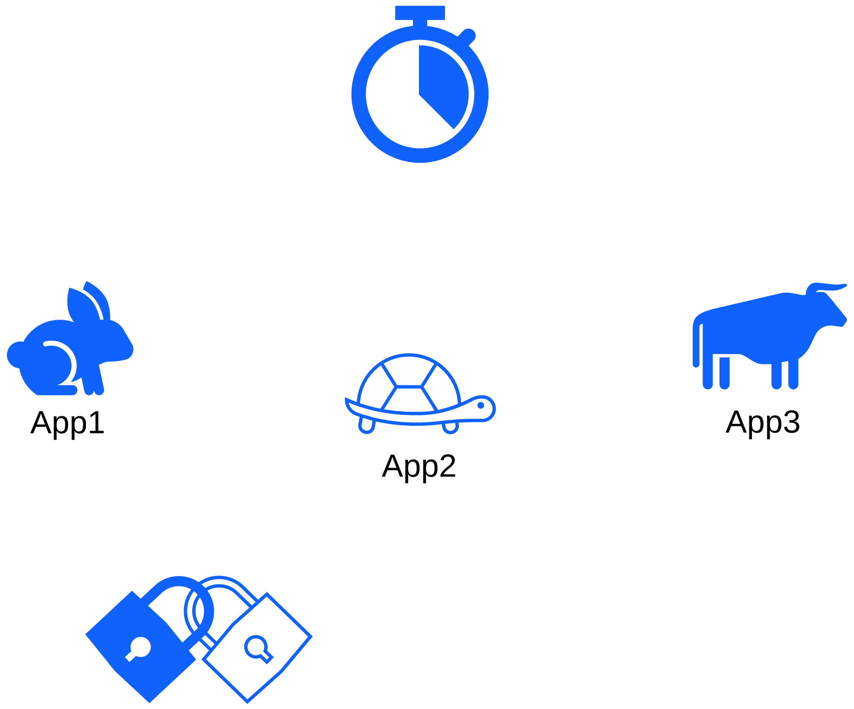
Application deadlock and timeout control

Current behavior

– Single subsystem parameter IRLMRWT

- Seconds before resource timeout detected
- Challenges:
  – No granularity by application process
  – Constrains multi-tenancy
  – Short response time application requirements constrained by single value across Db2 member or subsystem

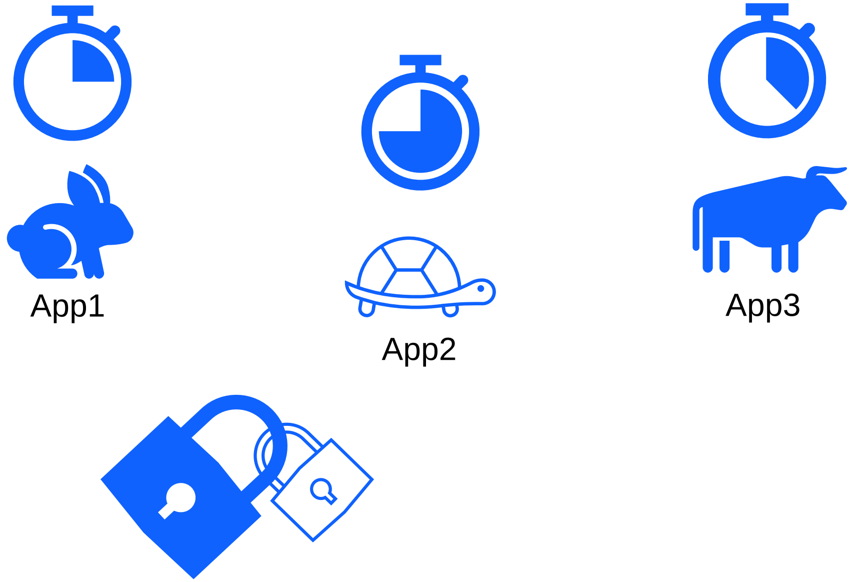
– Deadlock resolution applies equally to all processes in the Db2 member or subsystem



Application deadlock and timeout control

New behavior

- New special register — FL 500
  - CURRENT LOCK TIMEOUT
    - Set at application or SQL statement level
- New global variable — FL 501
  - DEADLOCK_RESOLUTION_PRIORITY
    - Weighting factor in resolving deadlocks with other threads
- Either or both can be set with system profile monitoring



Application deadlock and timeout control - details

CURRENT LOCK TIMEOUT

FL 500

- INTEGER, range -1 to 32767
  - -1 – no timeouts; wait until lock released or deadlock detected
  - 0 – application does not wait for a lock
  - 1-32767 – seconds to wait for a lock
    - Limited by subsystem parameter SPREG_LOCK_TIMEOUT_MAX
  - DSNT376I modified to include special register settings, if applicable

DEADLOCK_RESOLUTION_PRIORITY

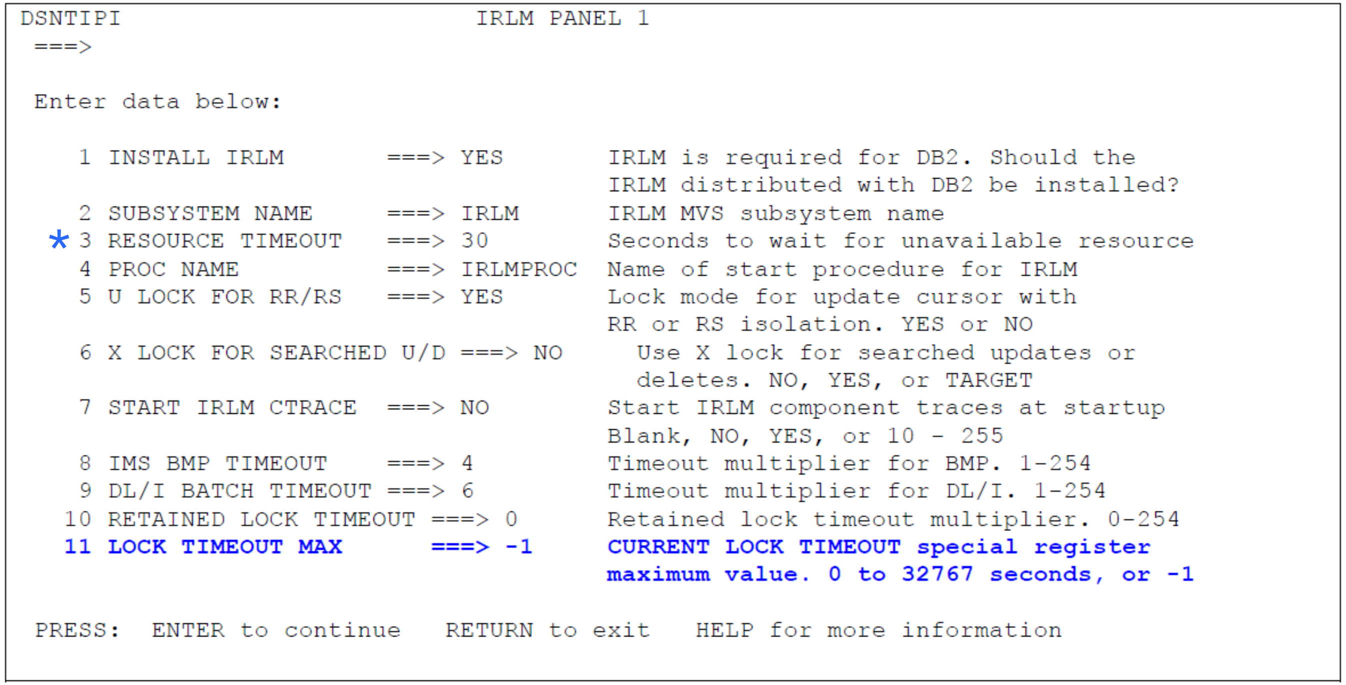
FL 501

- SMALLINT, range 0-255
- Process with highest priority wins

SPREG_LOCK_TIMEOUT_MAX

FL 100

* IRLMRWT now online changeable



```
DSNTIPI                        IRLM PANEL 1
 ===>

 Enter data below:

    1 INSTALL IRLM       ===> YES        IRLM is required for DB2. Should the
                                         IRLM distributed with DB2 be installed?
    2 SUBSYSTEM NAME     ===> IRLM       IRLM MVS subsystem name
  * 3 RESOURCE TIMEOUT   ===> 30         Seconds to wait for unavailable resource
    4 PROC NAME          ===> IRLMPROC   Name of start procedure for IRLM
    5 U LOCK FOR RR/RS   ===> YES        Lock mode for update cursor with
                                         RR or RS isolation. YES or NO
    6 X LOCK FOR SEARCHED U/D ===> NO      Use X lock for searched updates or
                                           deletes. NO, YES, or TARGET
    7 START IRLM CTRACE  ===> NO         Start IRLM component traces at startup
                                         Blank, NO, YES, or 10 - 255
    8 IMS BMP TIMEOUT    ===> 4          Timeout multiplier for BMP. 1-254
    9 DL/I BATCH TIMEOUT ===> 6          Timeout multiplier for DL/I. 1-254
   10 RETAINED LOCK TIMEOUT ===> 0       Retained lock timeout multiplier. 0-254
   11 LOCK TIMEOUT MAX      ===> -1      CURRENT LOCK TIMEOUT special register
                                         maximum value. 0 to 32767 seconds, or -1

 PRESS:  ENTER to continue   RETURN to exit   HELP for more information
```

LOCK TIMEOUT MAX

-1: any supported value can be specified in SET CURRENT LOCK TIMEOUT statement (default)

0-32767: maximum value that can be specified in a SET CURRENT LOCK TIMEOUT statement

Profile table support for local threads

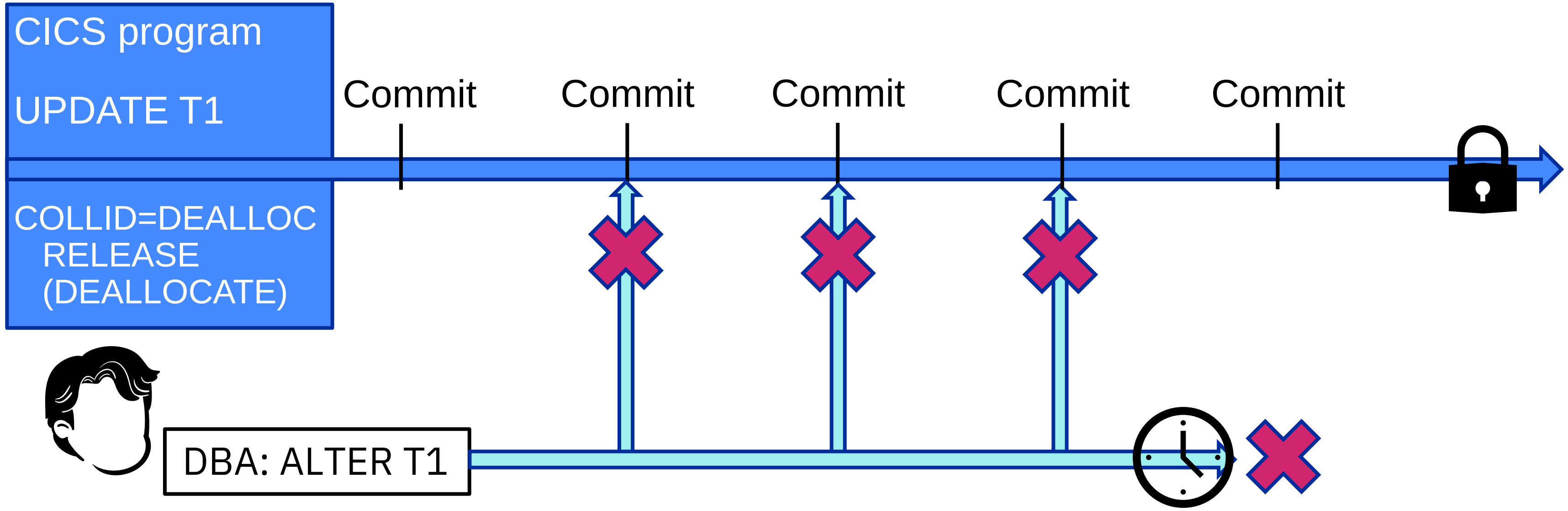
Current behavior

- Profile tables can be used by DBA to set special registers and global variables
  - Distributed threads only
- Local applications cannot easily change:
  - Special registers
  - Global variables
- Application developer required to make changes to local applications

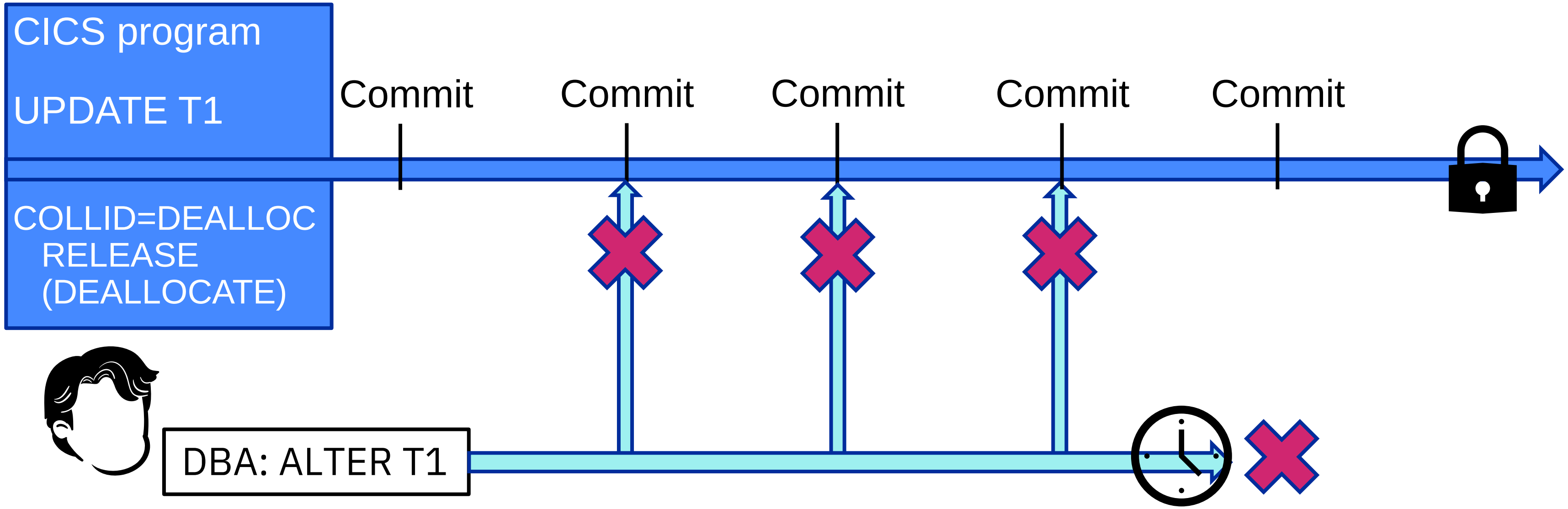
New behavior

- Profile tables enhanced
  - Local thread support in some situations
  - New special register:
    - CURRENT LOCK TIMEOUT — FL 500
  - New built-in global variable:
    - SYSIBMADM.DEADLOCK_RESOLUTION_PRIORITY — FL 501
  - New keyword:
    - RELEASE_PACKAGE — FL 500

Challenge scenario – release deallocate



Solution scenario (1|3) – release deallocate



DBA: INSERT INTO SYSIBM.DSN_PROFILE TABLE … ;
INSERT INTO SYSIBM.DSN_PROFILE_ATTRIBUTES … ;

Solution scenario (2|3) – release deallocate

INSERT INTO SYSIBM.DSN_PROFILE_TABLE…

PROFILEID	COLLID	PROFILE_ENABLE
99	DEALLOC	Y

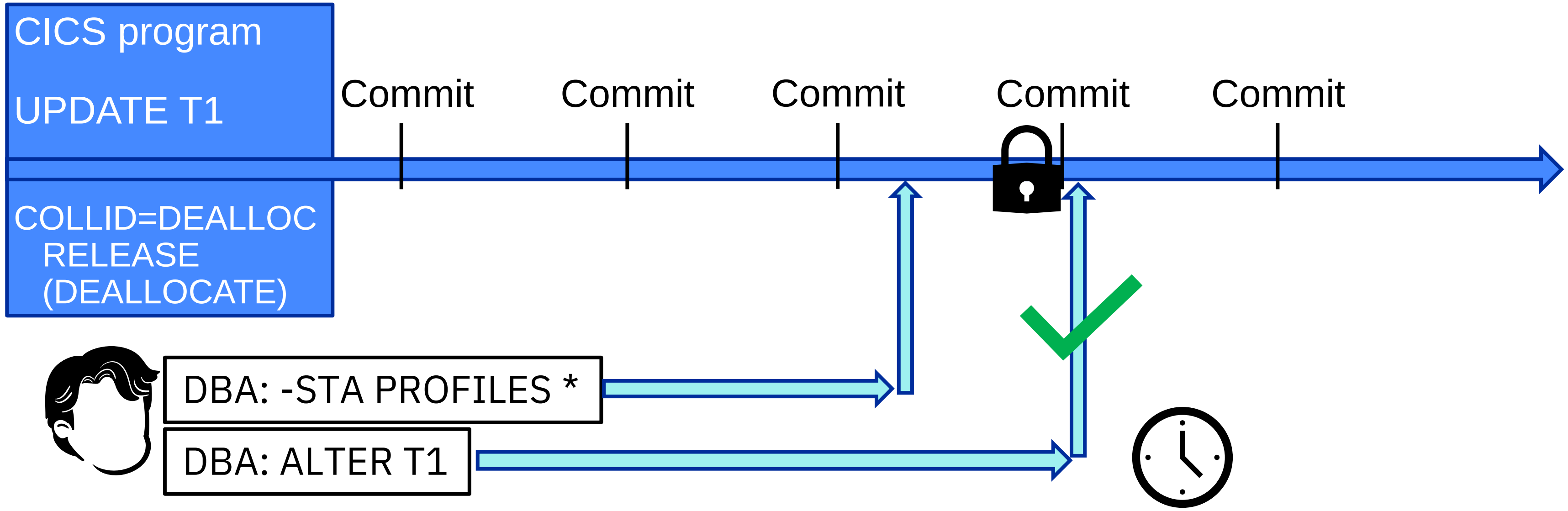
INSERT INTO SYSIBM.DSN_PROFILE_ATTRIBUTES…

PROFILEID	KEYWORDS	ATTRIBUTE1	ATTRIBUTE2
99	RELEASE_PACKAGE	COMMIT	2

ATTRIBUTE1 COMMIT overrides the RELEASE setting for the collection

ATTRIBUTE2 determines scope: NULL – distributed only; 1 – local only; 2 – local and distributed

Solution scenario (3|3) – release deallocate



* as described on prior page

Profile support – SYSIBM.DSN_PROFILE_ATTRIBUTES table

KEYWORDS	ATTRIBUTE1		ATTRIBUTE2
GLOBAL_VARIABLE	SET SYSIBMADM.DEADLOCK_RESOLUTION_PRIORITY = *value*	FL 501	NULL | 1 | 2
RELEASE_PACKAGE	COMMIT	FL 500	NULL | 1 | 2
SPECIAL_REGISTER	SET CURRENT LOCK TIMEOUT = *value*	FL 500	NULL | 1 | 2

ATTRIBUTE2 determines scope: NULL – distributed only; 1 – local only; 2 – local and distributed

– Deadlock resolution and current lock timeout:

- Remote threads: profiles are evaluated and SET statements are processed only when ***first*** package is loaded and when first non-SET SQL statement is executed
- Local threads: profiles are evaluated and SET statements are processed when ***each*** package is loaded

– RELEASE_PACKAGE

- Profiles evaluated when ***each*** package is loaded

Profile support for local threads

Db2 DDF address space must be loaded to use system profile monitoring, even if only monitoring local threads

- Subsystem parameter DDF must be set to AUTO or COMMAND
  - For local thread support, *ssnm*DIST must be started, but –STA DDF not required

Profiles for local threads specified in SYSIBM.DSN.PROFILE_TABLE columns

- Option 1: AUTHID, ROLE, or both
- Option 2: COLLID, PKGNAME, or both
- Option 3: ***One*** of CLIENT_APPLNAME, CLIENT_USERID, or CLIENT_WORKSTNNAME

- Column choice options are mutually exclusive; each profile can only have values for one of the three options

Profiles: Monitoring connections for security (1|3)

FL 500

Current behavior

- Distributed thread security behavior determined by single subsystem parameter (DSNZPARM) TCPALVER
  - YES – new connection accepted with user ID only
    - CLIENT – alternative to YES
  - <u>NO</u> – user ID and password required, or PassTicket or Kerberos
    - SERVER – alternative to NO
  - SERVER_ENCRYPT – user ID and password required, or Kerberos tickets, plus one of:
    - User ID and password AES-encrypted
    - Connection on AT-TLS port, e.g. SECPORT

Address several security use cases

- There may be different requirements for different Db2 access types
- Case 1: JDBC clients require multi-factor authentication (MFA) or client certificates
- Case 2: REST access from z/OS Connect only needs the z/OS Connect userid and password
- Case 3: Db2 for z/OS access as defined with TCPALVER
- Case 4: any other access needs a client certificate

Monitoring connections for security (2|3)

FL 500

- APAR PH48764 introduces the capability to discover and enforce the use of approved authentication and encryption methods by Db2 clients using profiles
- New actions added to the KEYWORDS column of the DSN_PROFILE_ATTRIBUTES table
  - MONITOR *product-type* CONNECTIONS FOR SECURITY, where *product-type* can be REST, JDBC, CLI, DB2CONNECT, DSN or *
- Specify the attributes of the profile in the ATTRIBUTE*n* columns
  - ATTRIBUTE1 specifies the action and console message (warning or exception)
  - ATTRIBUTE2 specifies the desired authentication mechanism (basic, MFA, client certificate, etc.)
  - ATTRIBUTE3 specifies whether the connection must be secured with an AT-TLS policy
- The new keyword values can only be specified for profiles using the default location filtering criteria ('*', '::0', or '0.0.0.0‘ in LOCATION column of the DSN_PROFILE_TABLE)

Use case example (3|3)

FL 500

DSN_PROFILE_TABLE

PROFILEID	LOCATION	ROLE	AUTHID	PRDID	COLLID	PKGNAME
101	::0	null	null	null	null	null

DSN_PROFILE_ATTRIBUTES

PROFILEID	KEYWORDS	ATTRIBUTE1	ATTRIBUTE2	ATTRIBUTE3
101	MONITOR JDBC CONNECTIONS FOR SECURITY	EXCEPTION_ DIAGLEVEL3	6	null
101	MONITOR REST CONNECTIONS FOR SECURITY	EXCEPTION_ DIAGLEVEL2	1	null
101	MONITOR DSN CONNECTIONS FOR SECURITY	EXCEPTION	null	null
101	MONITOR * CONNECTIONS FOR SECURITY	EXCEPTION	4	null

- EXCEPTION or WARNING imply DIAGLEVEL1, which does not include profile information
- DIAGLEVEL1 or DIAGLEVEL2 produces at most 1 message per 5 minutes
- DIAGLEVEL3 issued for each occurrence

Monitoring connections for security: reference

FL 500

- 'product-type' in
  - REST
  - JDBC
  - CLI
  - DB2CONNECT
  - DSN
  - * (applications other than specified above)

- ATTRIBUTE1 - message and action
  - EXCEPTION | EXCEPTION_DIAGLEVEL1 – fail request
  - EXCEPTION_DIAGLEVEL2 – fail request
  - EXCEPTION_DIAGLEVEL3 – fail request
    - » Issue DSNT776I for every exception
  - WARNING or WARNING_DIAGLEVEL1 – allow request
  - WARNING_DIAGLEVEL2 – allow request
  - WARNING_DIAGLEVEL3 – allow request
    - » Issue DSNT775I for every warning

Monitoring connections for security: reference

FL 500

- ATTRIBUTE2 – authentication mechanism
  - NULL – honor value in TCPALVER
    - ATTRIBUTE3 ignored; no warnings or exceptions
  - 1 – use basic auth: user ID and password or passphrase for authentication
  - 2 – use basic auth with MFA
  - 4 – use client certificate for authorization
    - Connection must be secured with AT-TLS policy
    - ATTRIBUTE3 ignored
  - 5 – use basic auth or client certificate: use 1 or 4
  - 6 – use basic auth with MFA or client certificate: use 2 or 4

- ATTRIBUTE3 – whether AT-TLS required
  - NULL use following default AT-TLS policy behavior for mechanism on ATTRIBUTE2
    - Basic auth: connection does not require AT-TLS
    - Basic auth with MFA: connection does not require AT-TLS
    - Client certificate: connection must be secured with AT-TLS
  - 1 – connection must be secured with AT-TLS policy

Another example: reference

FL 500

DSN_PROFILE_TABLE

PROFILEID	LOCATION	ROLE	AUTHID	PRDID	COLLID	PKGNAME
1	::0	null	null	null	null	null

DSN_PROFILE_ATTRIBUTES

PROFILEID	KEYWORDS	ATTRIBUTE1	ATTRIBUTE2	ATTRIBUTE3
1	MONITOR REST CONNECTIONS FOR SECURITY	EXCEPTION	5	null
1	MONITOR JDBC CONNECTIONS FOR SECURITY	WARNING	1	1

- Native REST applications that connect to Db2 by using client certificates with an AT-TLS policy defined are allowed. REST connections using user ID and password are also allowed regardless of AT-TLS policy. Other forms of authentication are rejected with a DSNT776I exception message.
- JDBC applications connecting to Db2 by using basic user ID and password with an AT-TLS policy defined are allowed. Other forms of authentication are also allowed with a DSNT771I warning message.
- Connections from product types other than REST or JDBC use the TCPALVER subsystem parameter to determine what form of security is required.

Product identifier (PRDID) values for specific DRDA levels

FL 500

- PRDID format is *pppvvrrm*, where ppp = product code, vv = version, rr = release, m = modification level [0-9, A-Z])
- With APAR PH48184 applied, Db2 now returns PRDID values that accurately reflect the DRDA level of the Db2 server, which correspond to specific function levels
- Before this APAR and in Db2 12, the PRDID values indicate only a range of function levels, e.g. in Db2 12:
  - DSN12015 for V12R1M500 or higher
  - DSN12010 for V12R1M100

PRDID	Function Level
DSN13010	V13R1M100
DSN13011	V13R1M500
DSN13012	V13R1M501
…	…
DSN1301A	V13R1M509
…	…
DSN1301Z	V13R1M534

Online delete active log data set from BSDS

FL 500

Current behavior

- Db2 outage required to delete active log data set
  - DSNJU003 stand-alone utility

Reasons to delete active log data set

- All 93 copies defined and need to increase size beyond 4 GB
- Encrypted active log data sets
  - Remove unencrypted data sets
  - Rotate encryption key and remove data sets with old encryption keys
- Moving active log data sets to new storage system

New behavior

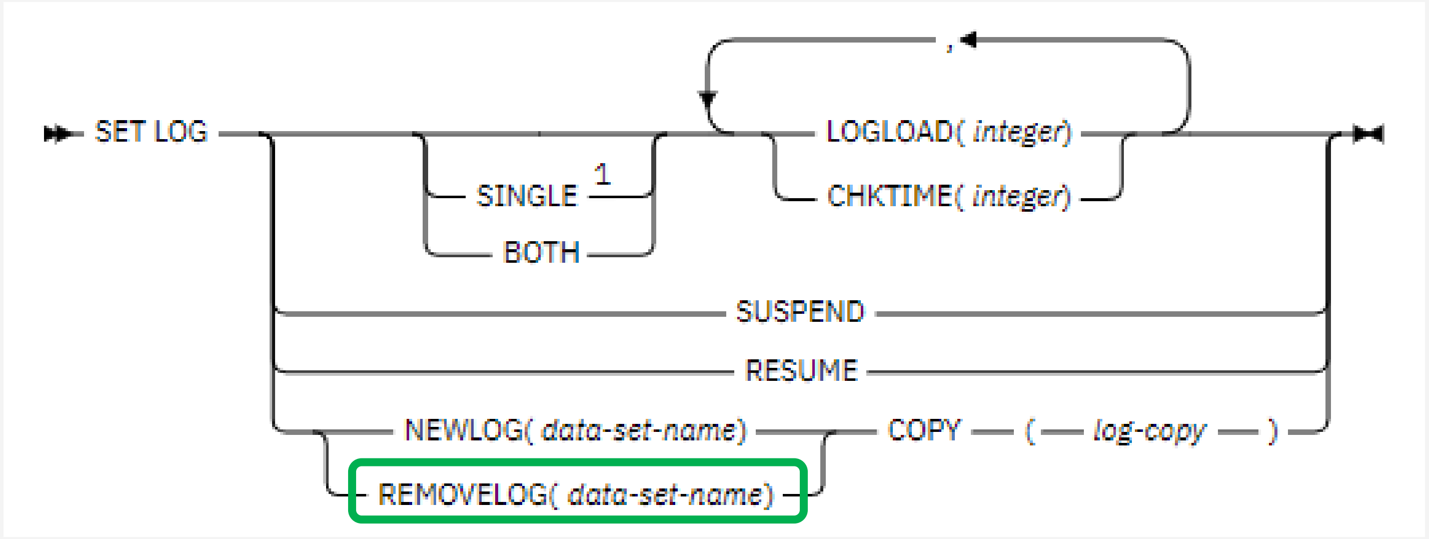
- -SET LOG REMOVELOG command:
  - If log not in use, deleted from BSDS
  - If log in use, marked REMOVAL PENDING
  - If error, failed command, for example:
    - Log is current active log
    - Log status is NOTREUSABLE
- Physical deletion of data set controlled by user
- GDPS Active/Active zero data loss (ZDL) not supported

-SET LOG command syntax

FL 500

REMOVELOG option added to –SET LOG command

- COPY is required
  - Double check for intended data set



Examples:

-DBP1 SET LOG REMOVELOG(DB2P.DBP1.LOGCOPY1.DS03) COPY(1)

-DBP1 SET LOG REMOVELOG(DB2P.DBP1.LOGCOPY2.DS03) COPY(2)

Online delete active log data set: details

FL 500

When log marked for REMOVAL PENDING

- Log cannot be unmarked
- Log cannot be used for new log read or write
  - Db2 will use another active log copy, if it exists
    - Else read from archive log
- Monitor status; reissue –SET LOG REMOVELOG
- After Db2 restart
  - If non-data sharing, log deleted from BSDS
  - If data sharing, REMOVAL PENDING persists
- Log can be deleted using DSNJU003

Requirements

- To remove active log data set
  - Specified COPY must exist in BSDS
  - Must be marked REUSABLE
  - Must not be current active log
  - Must not be next current active log
  - There must be at least 3 REUSABLE log data sets for the COPY specified in the command
  - FL 500 or higher; not supported for FL 100*
  - Data sharing: no peer Db2s have it allocated

Online delete active log data set: monitoring

-SET LOG REMOVELOG new messages

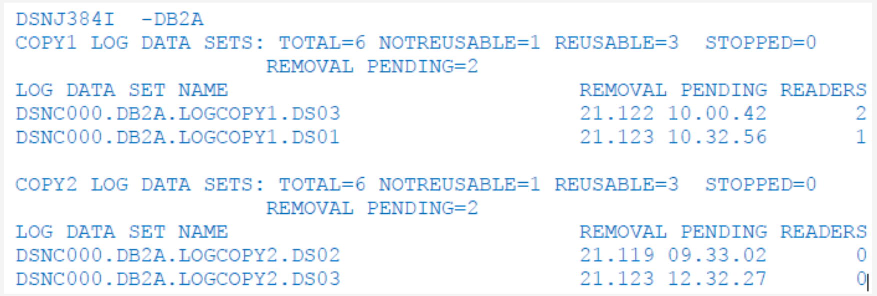
FL 500

- DSNJ391I … REMOVELOG OPERATION FAILED…
  - Includes data set name, reason
- DSNJ392I … *data-set* REMOVED FROM THE ACTIVE LOG INVENTORY
- DSNJ393I … *data-set* MARKED AS REMOVAL PENDING IN THE ACTIVE LOG INVENTORY
- DSNJ394I … INVALID KEYWORD COMBINATION…

-DISPLAY LOG DETAIL

FL 500 FL 100*

- Includes new message DSNJ384I



```
DSNJ384I  -DB2A
COPY1 LOG DATA SETS: TOTAL=6 NOTREUSABLE=1 REUSABLE=3  STOPPED=0
                    REMOVAL PENDING=2
LOG DATA SET NAME                            REMOVAL PENDING READERS
DSNC000.DB2A.LOGCOPY1.DS03                   21.122 10.00.42       2
DSNC000.DB2A.LOGCOPY1.DS01                   21.123 10.32.56       1

COPY2 LOG DATA SETS: TOTAL=6 NOTREUSABLE=1 REUSABLE=3  STOPPED=0
                    REMOVAL PENDING=2
LOG DATA SET NAME                            REMOVAL PENDING READERS
DSNC000.DB2A.LOGCOPY2.DS02                   21.119 09.33.02       0
DSNC000.DB2A.LOGCOPY2.DS03                   21.123 12.32.27       0
```

Other messages modified also

DSNJU004 adds support for REMOVAL PENDING

Monitoring index page split activity (1|2)

FL 500

Index page splits, which happen when Db2 has to insert an entry into an index page that is full, can have a significant negative impact on performance of high-volume INSERT processes

- Negative performance impact can be especially pronounced in Db2 data sharing environment, when index in question is GBP-dependent

Db2 12 problem: information that could help in monitoring and mitigating index page split activity often *not available* or *incomplete*

- *Often not available,* because index page split activity only captured in IFCID 359
  - IFCID 359 activated via performance trace class 4, which is not on by default (relatively high-overhead trace record – generated for <u>every</u> index page split)
- *Incomplete:* even if IFCID 359 active, record lacks important diagnostic information such as unit of recovery (UR) ID and data sharing member number

Monitoring index page split activity (2|2)

FL 500

Db2 13 solution:

- With function level 500 activated, new trace IFCID 396 generated when statistics trace class 3 is active
  - Stats class 3 is low-overhead trace, active by default
  - IFCID 396 is low-cost: only generated when elapsed time of index page split action unusually high (> 1 sec)
  - More-complete information, including *UR ID* and *data sharing member number* associated with IX page split
- Also, when catalog level is V13R1M501 (do-able when function level 500 activated), 3 new columns related to index page split activity added to SYSINDEXSPACESTATS and SYSIBM.SYSIXSPACESTATS_H tables:
  - REORGTOTALSPLITS – total number of index page splits since last reorganization or rebuild of index
  - REORGSPLITTIME – total/aggregated elapsed time of index page splits since last IX REORG or rebuild
  - REORGEXCSPLITS – total number of abnormal index splits (elapsed time > 1 second) since last reorganization or rebuild of index

Bottom line: more-complete, targeted information will aid in mitigating index page split issues

Flexibility for package ownership

FL 500

Previous behavior

– DBA can not directly specify role or ID ownership of plans, packages, and SQL PL routines when using both a role and an authorization ID

– Instead, DBA has indirect control of role or ID ownership:

  • To make role owner, run with role defined with AS OBJECT OWNER

  • To make ID owner run with role defined without AS OBJECT OWNER, or run without role

– This limitation impedes customer ability to adopt role-based security to meet their compliance requirements

New behavior

– New capability added to specify:

  • Role or ID ownership for plans or packages (including packages of SQL PL routines)

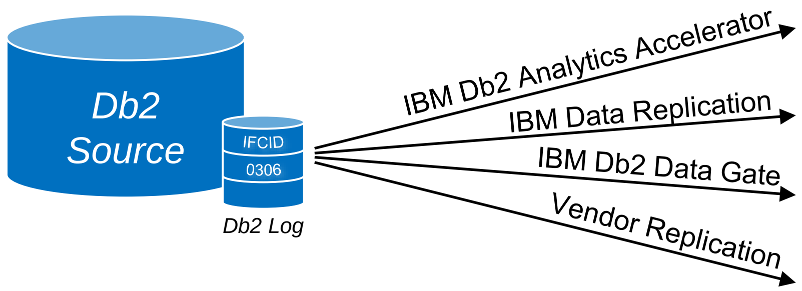
– How ownership can be specified as role or ID with Db2 13:

  • PACKAGE OWNER… **AS ROLE | USER** (for SQL PL routines)

  • OWNER… **OWNERTYPE (ROLE | USER)** (for BIND/REBIND PLAN/PACKAGE)

Concurrent ALTER TABLE ... DATA CAPTURE CHANGES

FL 500



- *Data Replication solutions retrieve changed data from the Db2 log via IFCID 0306*
- *Data Replication requires replication source tables to be altered with DATA CAPTURE CHANGES for 2 reasons:*
  - *Permission to replicate the table*
  - *Let Db2 log the full before image (instead of partial before image) plus partial after image (always) in case of a DML UPDATE operation*

Previous behavior:

– As part of the DATA CAPTURE alteration

- Db2 quiesces (but does not invalidate) static packages dependent on the altered table
- Db2 quiesces and invalidates cached and potentially stabilized dynamic statements dependent on the altered table

New behavior:

– As part of the DATA CAPTURE alteration

- Db2 does *not* quiesce static packages dependent on the altered table
- Db2 does *not* quiesce or invalidate cached and potentially stabilized dynamic statements dependent on the altered table

Concurrent ALTER TABLE .. DATA CAPTURE CHANGES

FL 500

Advantages of the improved concurrency:

– DATA CAPTURE alteration no longer waits for dependent DML statements to commit and the alteration can be executed successfully even when there is concurrently running static or dynamic DML against the table

– Given customers' 24x7 requirements the improved concurrency enables DATA CAPTURE alterations in parallel with the regular workload

- ALTER TABLE DATA CAPTURE no longer fails with time out due to continuous concurrent DML activity

– And, DATA CAPTURE alterations no longer cause the invalidation of cached or stabilized dynamic statements. This eliminates additional overhead (reprepare) and avoids potential performance regression due to unintended access path changes during reprepare.

Note

– The DML concurrency does not change other existing serialization mechanisms or locks that are obtained by ALTER TABLE .. DATA CAPTURE processing, such as DBD locks, catalog row locks, etc.

Statement-level dependency infrastructure

FL 500

FL 502

Previous behavior

- Db2 tracks application dependencies at package level
- An operation on any object requiring invalidation results in the entire package marked as invalid; even when only a subset of SQL statements in that package needs to be invalidated
- This is broad and limits Db2 flexibility to enhance and improve invalidation processing

Db2 13 and statement-level dependency

- When function level 500 activated, catalog can be taken to V13R1M501 level – one of the new tables is SYSPACKSTMTDEP
- Function level 502 introduces DEPLEVEL option for BIND and REBIND PACKAGE
  - DEPLEVEL(STATEMENT): Db2 puts *statement-level* dependency information in SYSPACKSTMTDEP
  - PACKAGE_DEPENDENCY_LEVEL in ZPARM provides default value (STATEMENT or <u>PACKAGE</u>) for DEPLEVEL
- With FL 502 activated, consider binding/rebinding packages with DEPLEVEL(STATEMENT) – will position you to benefit from future enhancement that will reduce impact of package invalidation