IDUG

2026

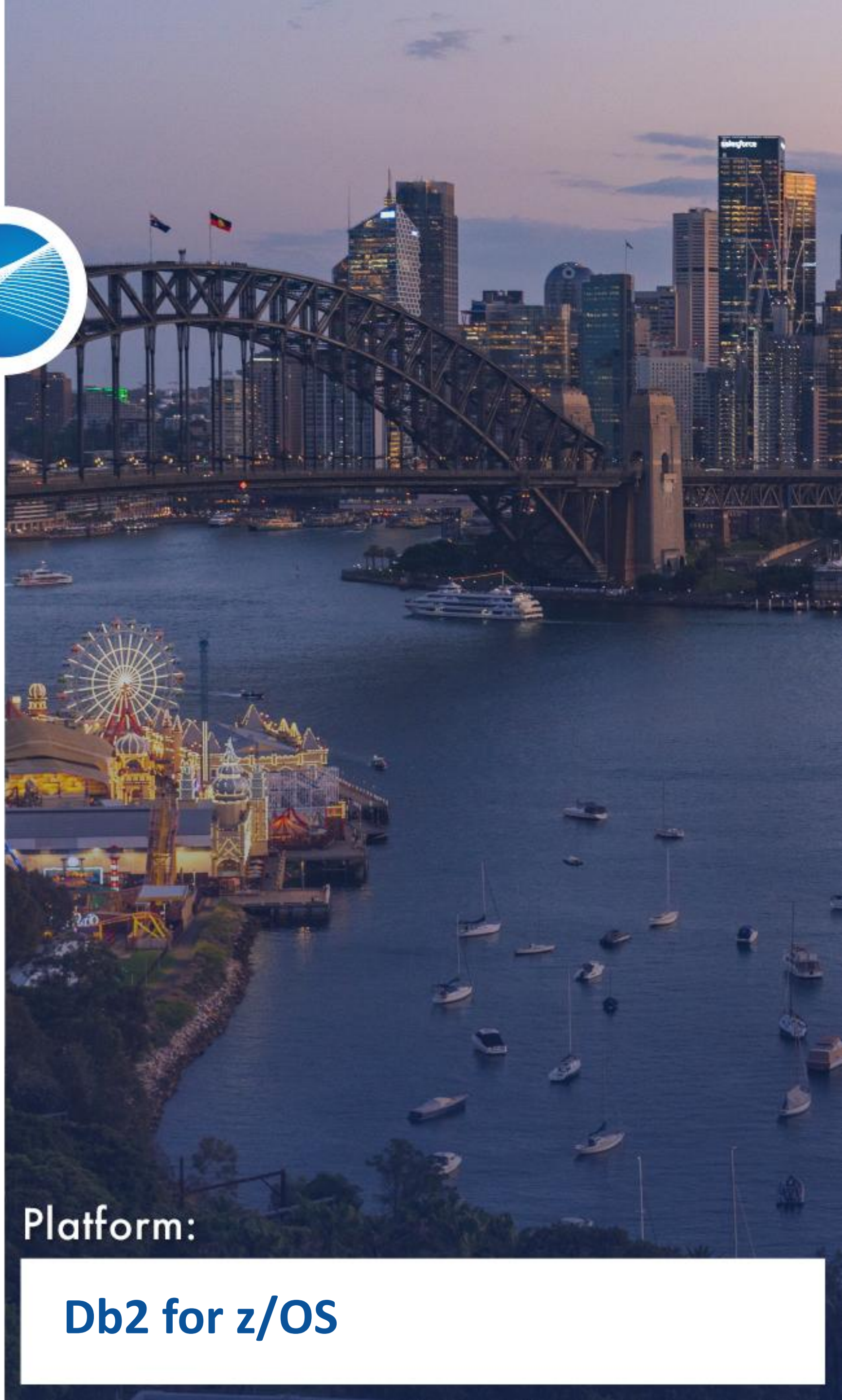
Sydney | March 16 - 18

AU Db2 TECH CONFERENCE

40 Years of Battle Scars Managing Db2 for z/OS

Steen Rasmussen, *Broadcom*

Session Code: C08



Platform:

Db2 for z/OS

ABSTRACT

- *Having been a DBA from 1985 to 1995 followed by working in the field with Db2 customers around the world for 30 years, I have faced quite a few wildfires and wars.*
- *Join me and experience some of the most interesting ones and learn how partner tooling can augment native Db2 features – hopefully you will bring home some ideas and draw some benefits.*

AGENDA

- Intelligent/Surgical Recovery
- Optimizing ERP Application Upgrade
- Eliminate resource waste : Unconditional REORGs, Data Cloning
- Recovering “mistakenly” dropped objects, removed grant’s, freed packages
- Security Cleanup – various topics

Steen's IDUG Participation (including virtual ones)

- IDUG EMEA 1995-2025 (minus three) : 29
- IDUG NA 2000-2025 (minus 1) : 25
- APJ : 12 + 1 (this year)
- India : 3
- LATAM : 1
- Total 71

AGENDA

- Intelligent/Surgical Recovery
- Optimizing ERP Application Upgrade
- Eliminate resource waste : Unconditional REORGs, Data Cloning
- Recovering “mistakenly” dropped objects, removed grant’s, freed packages
- Security Cleanup – various topics

#IDUGDb2

Intelligent / Surgical Recovery

Copyright © 2026 Broadcom. All Rights Reserved. The term "Broadcom" refers to Broadcom, Inc. and/or its subsidiaries.

Intelligent/Surgical Recovery (1|2)

- Batch application being tested in production to estimate execution time – "forgot to comment-out one DELETE stmt" – Db2 Explain insufficient.

  - A few thousands rows mistakenly deleted from one table.
  - Unfortunately, a parent table with many FK-tables.
  - Also "logical data" relationships with IMS DB's.
  - Incident discovered hours later
  - System wide recovery would be a disaster
  - Decision to use LOG-tool to UNDO the DELETES including FK-tables - minutes to "recover" as opposed to many hours.
  - ***BUT ……….***

Intelligent/Surgical Recovery (2|2)

- This can be very powerful

- **Exercise with great caution**
  - *Need to get acceptance from business application owners*
  - *What if applications make decisions based on the row-existence ?*
  - *Be aware of TRIGGERS – you might need to drop these prior to "re-inserting deleted data" – if DELETE/INSERT triggers*
  - *Using the log tool – remember to include TRIGGERED SQL (special log-records)*

#IDUGDb2

Optimizing ERP Application Upgrade

Copyright © 2026 Broadcom. All Rights Reserved. The term "Broadcom" refers to Broadcom, Inc. and/or its subsidiaries.

Optimizing ERP Application Upgrade (1|3)

- An ERP schema change "philosophy"

```
CREATE TABLE PX1.TB01          (OLD VERSION DDL)
(C1 CHAR(5)
 C2 INTEGER
 C3 DATE);
```

```
CREATE TABLE PX1.TB01          (NEW VERSION DDL)
(C1 CHAR(7)
 C2 INTEGER
 C3 DATE;
```

- A pretty simple schema change which can be done using ALTER and REORG
- **BUT ……… this is the "philosophy"**

Optimizing ERP Application Upgrade (2|3)

- An “interesting and ***scary***” implementation

```
CREATE TABLE PX1.TB01_NEW;

INSERT INTO PX1.TB01_NEW (col-names)
SELECT C1,C2,C3 FROM PX1.TB01:

ALTER TABLE PX1.TB01 TO PX1.TB01_OLD;
ALTER TABLE PX1.TB01_NEW TO PX1.TB01;

.........SCARY !!!!!!!!
```

- It works great – but this ERP application has 30,000 tables and 40,000 indexes *(SELECT-INSERT not the most efficient tool)*
- Why was this approach picked ?
  - It’s universal for all RDBMS’s (Db2 z/OS, LUW, Oracle, SQL-server ………..

Optimizing ERP Application Upgrade (3|3)

- What did we do to optimize this "interesting and scary" approach ?
  - Pick your favorite DDL generation tool.
  - You have the current ERP application in a "sandbox".
  - Generate DDL for the entire ERP application.
    - Create "OLD ERP" environment (we used DEFINE NO).
  - You receive the DDL/DML for the new ERP release.
  - Execute the script on the "sandbox" instance.
  - Execute DDL Compare between NEW release and the "DEFINE NO" environment
    - Result: online/deferred schema change DDL, REORG etc.
    - Carry this script to subsequent environments to avoid the CREATE-INSERT-ALTER-RENAME scenario
    - Huge elapsed-time savings.
    - (please be aware of index design differences – schema synchronization tool has option to KEEP un-matched).

#IDUGDb2

Intelligent and Cost Effective Reorg

Copyright © 2026 Broadcom. All Rights Reserved. The term "Broadcom" refers to Broadcom, Inc. and/or its subsidiaries.

Intelligent and Cost Effective Reorg (1|5)

- Homegrown REXX used to “automate” reorg processes
- Maintenance became a burden during Db2 upgrades and requirements impossible to implement
- And ……. the author retired
- Implemented new reorg approach to avoid “issues”
  - All SMALL tables every week
  - All MEDIUM tables every month
  - All LARGE tables every quarter

NOT Best Practices !!

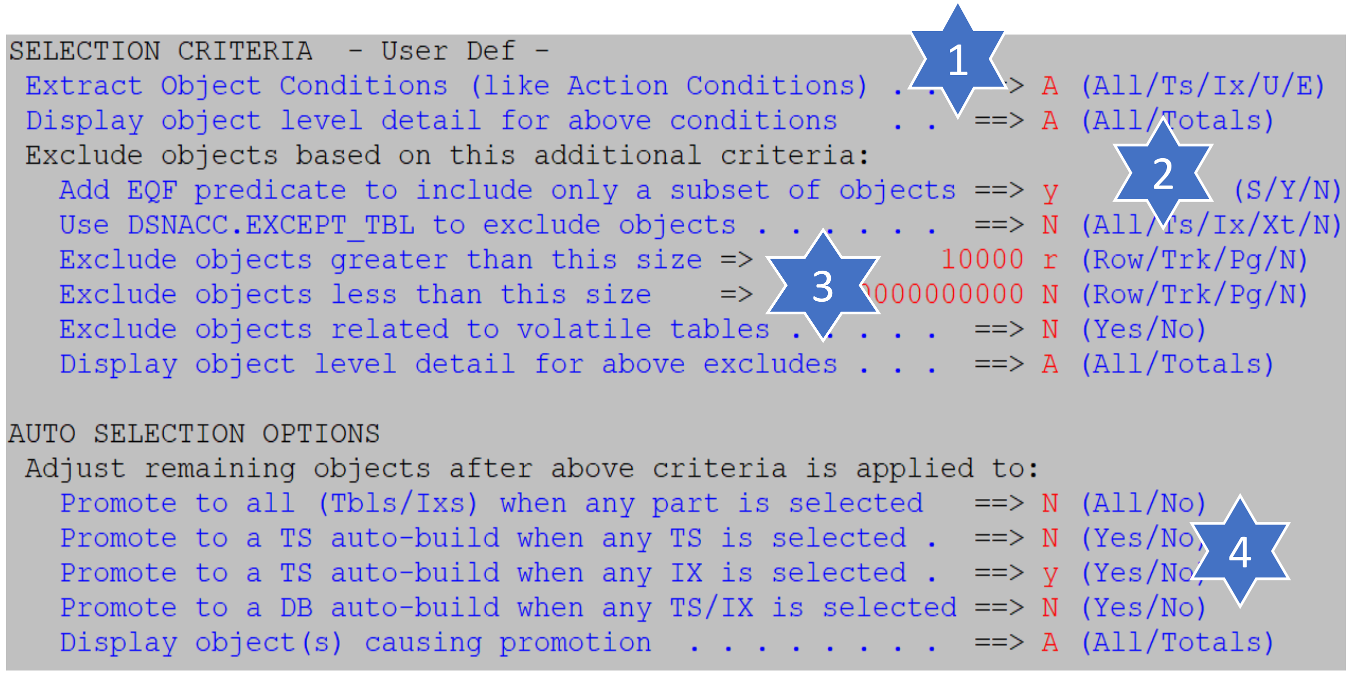
- …………
- Database Analyzer implemented and nice improvements achieved *(other vendor solutions do exist)*

Intelligent and Cost Effective Reorg (2|5)

- Implement a solution that requires minimal maintenance
  - No JCL coding
  - No to little need monitoring new/dropped objects
  - Easy adoption of new features in Db2
  - Intelligent & flexible "skeletons" to handle site specific requirements and potential object characteristics
  - Integration with job-scheduling system

Intelligent and Cost Effective Reorg (3|5)

- Some of the most desired requirements/needs observed:
  1. Conditional thresholds *(to be covered on next slide)*
  2. Use EQF to potentially exclude objects (DBA exception table)
  3. Exclude small objects since reorg might not benefit anyway
  4. When index (or partition) needs reorg – promote to tablespace (part)
  5. All partitions for a tablespace needing reorg to be in one LISTDEF

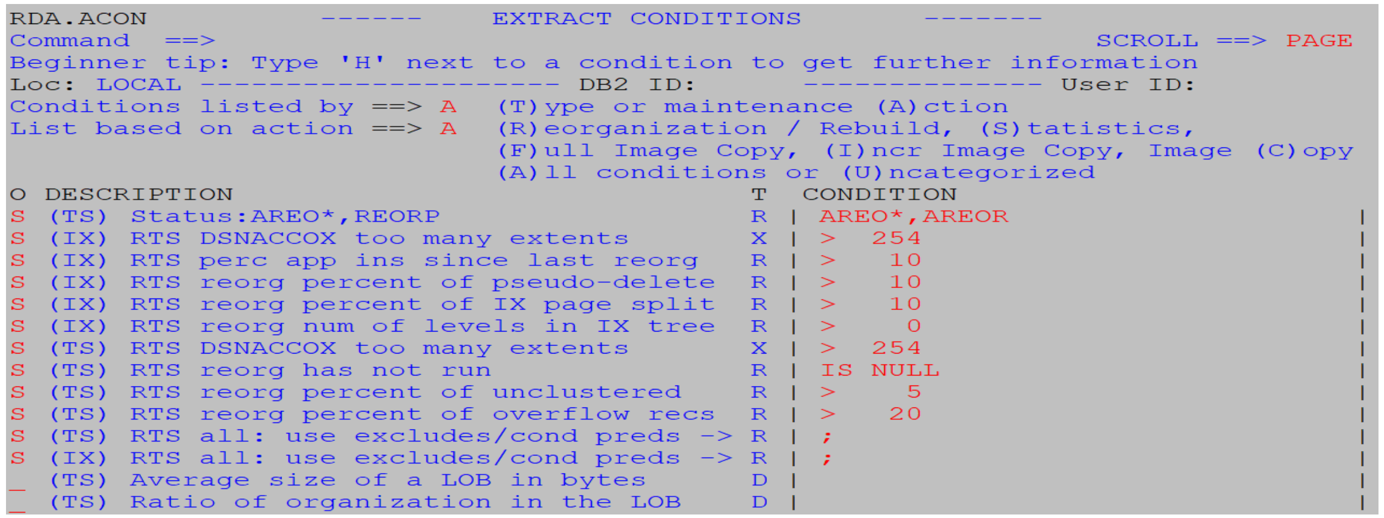


```
SELECTION CRITERIA  - User Def -
 Extract Object Conditions (like Action Conditions) . .    > A (All/Ts/Ix/U/E)
 Display object level detail for above conditions   . .  ==> A (All/Totals)
 Exclude objects based on this additional criteria:
   Add EQF predicate to include only a subset of objects ==> y         (S/Y/N)
   Use DSNACC.EXCEPT_TBL to exclude objects . .   . . .  ==> N (All/ s/Ix/Xt/N)
   Exclude objects greater than this size =>         10000 r (Row/Trk/Pg/N)
   Exclude objects less than this size    =>    000000000 N (Row/Trk/Pg/N)
   Exclude objects related to volatile tables .   . . .  ==> N (Yes/No)
   Display object level detail for above excludes . . .  ==> A (All/Totals)

AUTO SELECTION OPTIONS
 Adjust remaining objects after above criteria is applied to:
   Promote to all (Tbls/Ixs) when any part is selected   ==> N (All/No)
   Promote to a TS auto-build when any TS is selected .  ==> N (Yes/No
   Promote to a TS auto-build when any IX is selected .  ==> y (Yes/No
   Promote to a DB auto-build when any TS/IX is selected ==> N (Yes/No)
   Display object(s) causing promotion  . . . . . . . .  ==> A (All/Totals)
```

Intelligent and Cost Effective Reorg (4|5)

- Most used conditions (RTS based):
  - Advisory reorg status
  - Extents, appended inserts, pseudo-deleted index entries
  - Page splits and index-levels are crucial
  - Unclustered rows and relocated rows



```
RDA.ACON              ------     EXTRACT CONDITIONS         -------
Command   ==>                                                   SCROLL ==> PAGE
Beginner tip: Type 'H' next to a condition to get further information
Loc: LOCAL --------------------- DB2 ID:         -------------- User ID:
Conditions listed by ==> A  (T)ype or maintenance (A)ction
List based on action ==> A  (R)eorganization / Rebuild, (S)tatistics,
                            (F)ull Image Copy, (I)ncr Image Copy, Image (C)opy
                            (A)ll conditions or (U)ncategorized
O DESCRIPTION                         T  CONDITION
S (TS) Status:AREO*,REORP             R | AREO*,AREOR                         |
S (IX) RTS DSNACCOX too many extents  X | >  254                              |
S (IX) RTS perc app ins since last reorg R | >   10                            |
S (IX) RTS reorg percent of pseudo-delete R | >   10                          |
S (IX) RTS reorg percent of IX page split R | >   10                          |
S (IX) RTS reorg num of levels in IX tree R | >    0                          |
S (TS) RTS DSNACCOX too many extents  X | >  254                              |
S (TS) RTS reorg has not run          R | IS NULL                             |
S (TS) RTS reorg percent of unclustered R | >    5                            |
S (TS) RTS reorg percent of overflow recs R | >   20                          |
S (TS) RTS all: use excludes/cond preds -> R | ;                             |
S (IX) RTS all: use excludes/cond preds -> R | ;                             |
_ (TS) Average size of a LOB in bytes D |                                     |
_ (TS) Ratio of organization in the LOB D |                                   |
```

Intelligent and Cost Effective Reorg (5|5)

- **Conclusion:**

- The processes were implemented into job scheduler
- Review of outcome after a few weeks:
  - +90% of reorgs eliminated
  - Executed in a fraction of the time
  - No need to maintain/create JCL anymore
  - Automatically include new objects
  - Some objects got reorg'ed more frequently
  - Application performance improved due to this
  - Keep track of REORG REASON:
    - Potentially altering tablespace/index attributes can postpone reorgs

#IDUGDb2

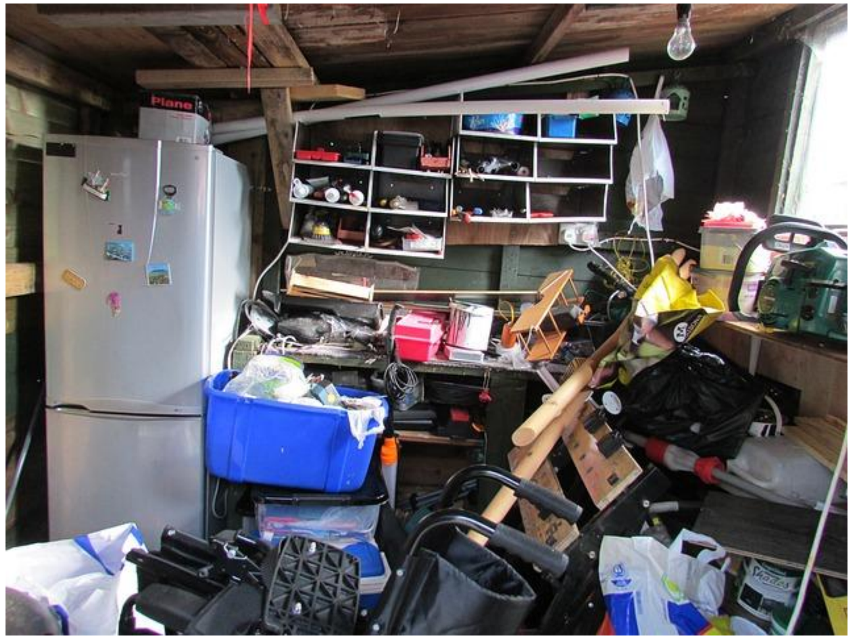
Recovering Dropped Database

Copyright © 2026 Broadcom. All Rights Reserved. The term "Broadcom" refers to Broadcom, Inc. and/or its subsidiaries.

Recovering Dropped Objects (1|4)

- A business application group requested a database to be dropped.
  - Multiple tablespaces and indexes
  - Dozens of DBRM's / Packages
- Five weeks later they had the need to do forensic analysis of the "lost data"
- A backup/unload of data & DDL etc. wasn't performed.

- *Extend archive logs retention period*
- *Extend imagecopy datasets retention (thank GOD they were still there)*
- *Log Tool "Dropped Object Recovery" feature*



Recovering Dropped Objects (2|4)

- Very easy to panic and lose control
- Important to stay calm – think about what really happened behind the scenes !!
- When an object is dropped – *(no need to recover catalog):*
  - SYSCOPY entries are DELETE's – recorded in the Db2 log (we can find the imagecopy dataset names)
  - Object details are just a number of DELETE's (SYSTABLESPACE, SYSCOLUMNS, SYSTABLES, SYSSYSINDEXES, ...)
  - Authorizations are DELETE's from xxxxxxAUTH tables
  - Package details are basically UPDATE statements in the Db2 log
  - When the objects are re-created : New OBID's might ne used – find the dropped OBID's in the log to map new/old objects
  - All data changes from the last imagecopy to the DROP-RBA are in the Db2 log
- All the ingredients are present to get rid of the nightmare

Recovering Dropped Objects (3|4)

- You can do this manually – OR – use your log tool of choice

1. Identify the time-range for the DROP
2. Generate the DDL CREATE statements from the log and execute
3. Identify old/new Internal Identifiers
4. Find the last usable imagecopy dataset and DSN1COPY using OBIDXLAT (or a recovery utility like Fast Recover)
5. Apply the log-records from the imagecopy-RBA to the DROP-RBA
6. RUNSTATS or restore from the log
7. Generate REBIND statements from the log
8. Potentially GRANTs
9. Maybe it's a good PIT to take an imagecopy

Recovering Dropped Objects (4|4)

You can do this manually – OR – use your log tool of choice

- A lot of tasks to complete manually and room for errors.
- Pick your LOG tool of choice and avoid the hours/days it might take.
- This customer surrendered after two days and called in the troops.

✓ Fill out the time range

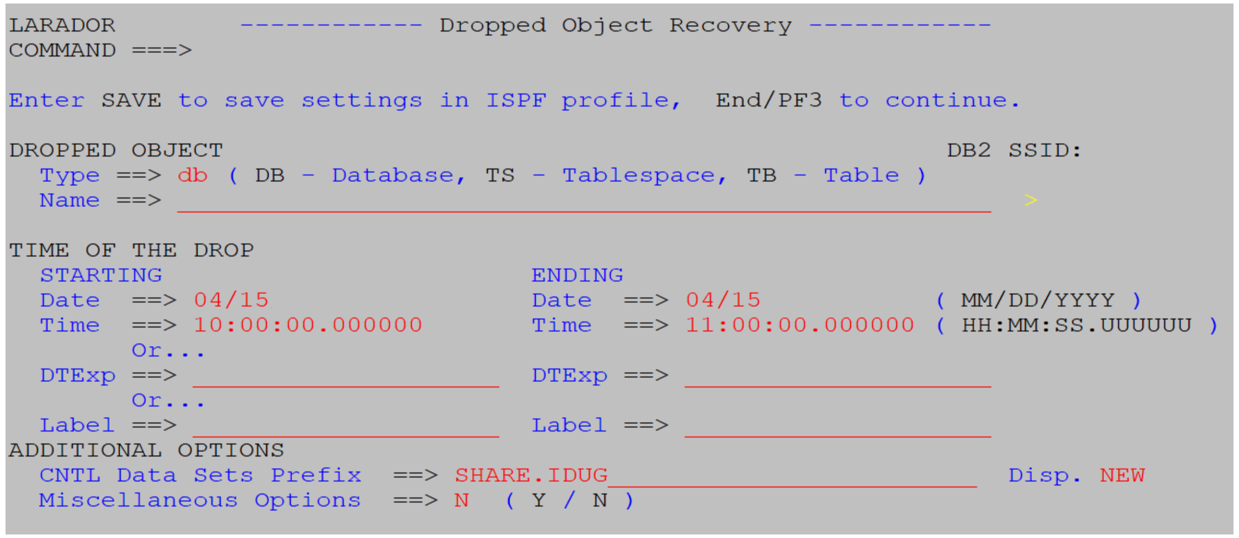
✓ Specify the object type and name

✓ A few seconds/minutes to scan the log

✓ One job with 9/10 steps created

✓ In 15 minutes everything was restored

Be prepared: Easy to PRACTICE + TEST



```
LARADOR        ------------ Dropped Object Recovery ------------
COMMAND ===>

Enter SAVE to save settings in ISPF profile,  End/PF3 to continue.

DROPPED OBJECT                                                   DB2 SSID:
  Type ==> db ( DB - Database, TS - Tablespace, TB - Table )
  Name ==> _____________________________________________________  >

TIME OF THE DROP
  STARTING                         ENDING
  Date   ==> 04/15                 Date   ==> 04/15              ( MM/DD/YYYY )
  Time   ==> 10:00:00.000000       Time   ==> 11:00:00.000000    ( HH:MM:SS.UUUUUU )
        Or...
  DTExp  ==> ____________________  DTExp  ==> ____________________
        Or...
  Label  ==> ____________________  Label  ==> ____________________
ADDITIONAL OPTIONS
  CNTL Data Sets Prefix    ==> SHARE.IDUG__________________________  Disp. NEW
  Miscellaneous Options    ==> N  ( Y / N )
```

#IDUGDb2

Packages Missing

Copyright © 2026 Broadcom. All Rights Reserved. The term "Broadcom" refers to Broadcom, Inc. and/or its subsidiaries.

Packages Missing (1|2)

- Packages failed to execute
- Not only a few ………..
- Db2 log scanned and ~8000 packages were FREE'd
- Only the DBA's had authority – nobody claimed to be guilty
  - Log scan indicated who did it
  - Maybe one reason why many customers execute audit reports for DDL, Commands, BINDs, GRANT's and REVOKEs
- This customer spent the weekend restoring the Db2 catalog to an alternate subsystem
  - Then the BINDs were generated and transported to the failing system
- BUT . . . . . Again it's time to step back and think what happened

Packages Missing (2|2)

- What happens during a FREE package ?
- A number of catalog tables are impacted to reflect the command
  - SYSPACKAGE, SYSPACKDEP, SYSPACKSTMT, ………….
  - Basically a lot of DELETEs
  - They all go into the Db2 log – and which tool has helped a lot ?
- The customer called to get advice to be prepared for future identical incidents
- The Db2 archive logs were still present from the incident a few days ago
- We used a log tool to generate UNDO statements for the FREE command
  - The scan of the log for the approximate time range executed for a couple of minutes
  - All ~8000 BIND statements were generated
  - This approach could have caused a stress free weekend

#IDUGDb2

Security Clean-Up

Copyright © 2026 Broadcom. All Rights Reserved. The term "Broadcom" refers to Broadcom, Inc. and/or its subsidiaries.

Security Cleanup (1|4)

- Topic gaining increased interest
  - Internal Db2 security – external is less an issue
  - Remove the evidence/ownership from retired user-id's
- Easier nowadays compared to the past
  - Personal story from 1988'ish – a DBA quit – he was SYSADM
  - Late Friday I decided to REVOKE
    - It did run for a while …………… longer than expected
    - Five minutes later Operation called – nothing was executing
    - Views were missing
    - Plans (packages were not invented) were invalidated
    - No LOG TOOL to assist in finding all the "missing pieces" – no vendor revoke-cascade-repair
    - I faced a "nice" drop-cascade : NOT a nice Friday evening !!

Security Cleanup (2|4)

- One frequent used method :
  - Make “retired user-id” INSTALL SYSADM to avoid cascade

- Relatively new native Db2 features can come to the rescue:
  - TRANSFER OWNERSHIP
  - REVOKE_DEP_PRIVILEGES

Security Cleanup (3|4)

- TRANSFER OWNERSHIP in action:
  - STEEN01 created DB=KDB01

TRANSFER OWNERSHIP OF DATABASE KDB01 TO USER SACHIN revoke privileges ;

```
 DB2 Object ===> DB                          Option   ===> L
  Data Base ===> KDB%                      > Creator ===> *
  Qualifier ===> *                         > N/A      ===> *
Loc: LOCAL ---------- SSID:      ---------- STEEN01 -
CMD       DATABASE CREATOR  STOGROUP CREATEDB   DBID BPOOL
________ KDB01    SACHIN   SYSDEFLT STEEN01   15042 BP0
******************************* BOTTOM OF DATA ***********
```

```
UPDATE "SYSIBM".SYSDATABASE
   SET CREATOR = 'SACHIN'
   WHERE "NAME" = 'KDB01'     ;

 UPDATE "SYSIBM".SYSDBAUTH
   SET GRANTOR = 'SACHIN'
     , GRANTEE = 'SACHIN'
     , TIMESTAMP = X'C5D4D8D1E4C7F9E4F2E3C9C8'
     , DATEGRANTED = '230602'
     , TIMEGRANTED = '08252643'
     , GRANTEDTS = '2023-06-02-08.25.26.435675'
   WHERE GRANTOR = 'STEEN01'
     AND GRANTEE = 'STEEN01'
     AND "NAME" = 'KDB01'
     AND TIMESTAMP = X'C5D4D8C3F3E7F3E5E8D2F5D6'
     AND DATEGRANTED = '230601'
     AND TIMEGRANTED = '15544316'
     AND GRANTEETYPE = ' '
     AND AUTHHOWGOT = ' '
     AND CREATETABAUTH = 'G' ……….. 
```

- It works very well
- Using a log tool to generate REDO DML illustrates PURE UPDATES
- SYSVIEWS handled the same way – but SYSTABLES and SYSTABAUTH have an INSERT and DELETE

Security Cleanup (4|4)

- Db2 REVOKE_DEP_PRIVILEGES subsystem parameter
- The REVOKE_DEP_PRIVILEGES subsystem parameter controls whether revoking a privilege from a user is to cause dependent privileges to be revoked. If dependent privileges are to be revoked, revoking a privilege from a user also revokes the privilege from anyone that the user has granted that privilege to.
  - **NO** – REVOKE statements do not include dependent privileges. An error occurs if a REVOKE statement contains the INCLUDING DEPENDENT PRIVILEGES clause.
  - **YES** – REVOKE statements include dependent privileges, except when ACCESSCTRL, DATAACCESS, and system DBADM authorities are revoked. An error occurs if a REVOKE statement contains the NOT INCLUDING DEPENDENT PRIVILEGES clause, except when ACCESSCTRL, DATAACCESS, and system DBADM authorities are revoked.
  - **SQLSTMT** – Allows revoking of dependent privileges to be controlled at the SQL level, as specified in REVOKE statements. Db2 recognizes the dependent privileges clause (INCLUDING DEPENDENT PRIVILEGES or NOT INCLUDING DEPENDENT PRIVILEGES) of the REVOKE statement
- Note: This is a security-related parameter. If it is set to NO, privileges that were granted by a user are retained even if that user loses the authority that allowed the user to perform the grant

#IDUGDb2

Data Cloning / Refresh
(are you wasting resources)

Copyright © 2026 Broadcom. All Rights Reserved. The term "Broadcom" refers to Broadcom, Inc. and/or its subsidiaries.

Data Cloning / Refresh (1|4)

Common methods used to accomplish this task:

- Besides from data sub-setting and masking (different topic):
    1. DSN1COPY to do IC-VSAM or VSAM-VSAM. Challenges when +1 2GB VSAM pagesets (the good old segmented tablespaces), unequal number of PBG's, extent processing etc.
    2. Unload-Load is resource intensive – advantage is schema differences
    3. Recover from production IC to target – with or without log-apply
- All three alternatives might have consistency issues
    - IC probably done as SHRLEVEL CHANGE
    - What about RI constraints
    - Depending on the use case, "fuzzy" data might be "good enough"

Data Cloning / Refresh (2|4)

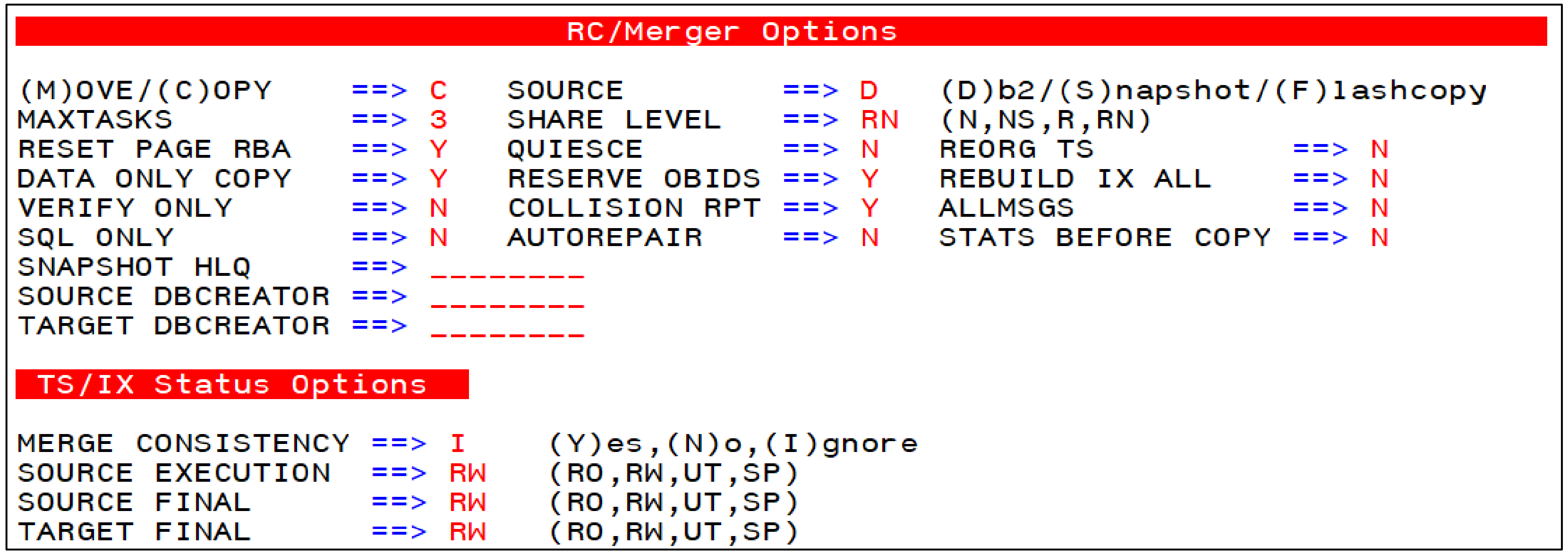
- Benchmark for traditional approach using unload/load:
  - Object characteristics:
  - ~550M rows , average row length 98 byte , 14M 4K pages
  - 26 x 2GB pagesets and one index with 3 x 2GB pagesets
  - 572 cpu-sec and ~60 minute elapsed (on a busy system)
- Benchmark for RC/Merger implementation:
  - 67 cpu-sec and ~ 6 minute elapsed (on a busy system)
  - RC/Merger COPY-method used (all source pages are read and written)
  - If MOVE used, the savings are much more (VSAM RENAME and potentially OBID translation)
- Let's have a look at the considerations needed

Data Cloning / Refresh (3|4)

- Using FlashCopy technology proves even better performance:
  - COPY FLASHCOPY YES CONSISTENT YES (for both *tablespace* and *indexes*)
  - All IN-FLIGHTS backed out → VSAM RENAME (*plus XLAT*)
  - If “internal IDs” can be honored savings can be even better.
- Using RC/Merger provided 98% CPU and savings and elapsed time ~5 minutes at a large pharmaceutical company
  - One table of 4bn rows, 100 partitions and 3 NPIs
  - Requirement of cloning into 8 environments at least weekly could not be honored
  - Unload/Load took 3.5 hours and consumed hours of CPU for just one environment
  - Now all 8 environments can be refreshed daily providing the business more accurate data
  - RC/Merger can preserve a lot of resources while cloning data
  - Due to the low CPU consumption – more data refresh can be completed

Data Cloning / Refresh (4|4)

- Some important pieces to consider when cloning and not using unload-LOAD
  - REPAIR when schema versioning
  - If source NOT consistent do INDEX REBUILD to avoid 00C90101
  - Attempt to RESERVE OBIDs
  - RESET PAGE RBA



```
                              RC/Merger Options

(M)OVE/(C)OPY     ==> C   SOURCE         ==> D   (D)b2/(S)napshot/(F)lashcopy
MAXTASKS          ==> 3   SHARE LEVEL    ==> RN  (N,NS,R,RN)
RESET PAGE RBA    ==> Y   QUIESCE        ==> N   REORG TS          ==> N
DATA ONLY COPY    ==> Y   RESERVE OBIDS  ==> Y   REBUILD IX ALL    ==> N
VERIFY ONLY       ==> N   COLLISION RPT  ==> Y   ALLMSGS           ==> N
SQL ONLY          ==> N   AUTOREPAIR     ==> N   STATS BEFORE COPY ==> N
SNAPSHOT HLQ      ==> ________
SOURCE DBCREATOR  ==> ________
TARGET DBCREATOR  ==> ________

 TS/IX Status Options

MERGE CONSISTENCY ==> I    (Y)es,(N)o,(I)gnore
SOURCE EXECUTION  ==> RW   (RO,RW,UT,SP)
SOURCE FINAL      ==> RW   (RO,RW,UT,SP)
TARGET FINAL      ==> RW   (RO,RW,UT,SP)
```

#IDUGDb2

Conclusion

Copyright © 2026 Broadcom. All Rights Reserved. The term "Broadcom" refers to Broadcom, Inc. and/or its subsidiaries.

Conclusion / Wrap-up

Don't let panic take control

Think outside the box

Take a step back and think about what happened

You need to really understand your tool box

……. Stay tuned on features and capabilities

#IDUGDb2

THANK YOU !!

Copyright © 2026 Broadcom. All Rights Reserved. The term "Broadcom" refers to Broadcom, Inc. and/or its subsidiaries.

IDUG

2026

Sydney | March 16 - 18

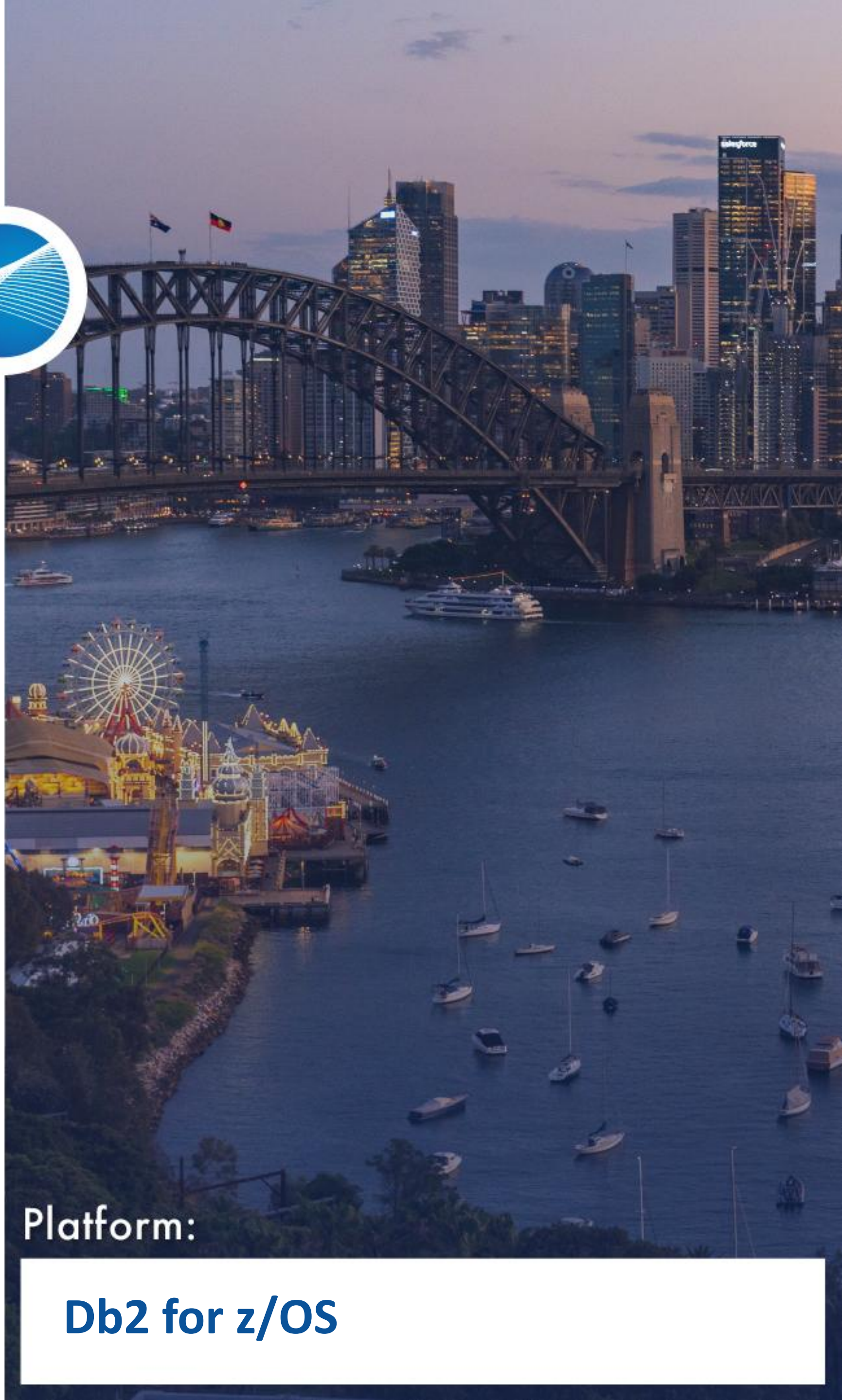
AU Db2 TECH CONFERENCE

40 Years of Battle Scars Managing Db2 for z/OS

Steen Rasmussen, *Broadcom*

Contact: steen.Rasmussen@broadcom.com / db2steen@yahoo.com

Session Code: 7



Platform:

Db2 for z/OS

IDUG

2026 Australia **Db2** Tech Conference