

---

# Db2 vNext Migration

## Are you Ready Player One?\*

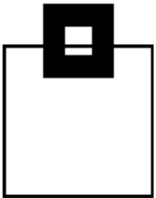
### (It's Free and Easy)

Roy Boxwell, SOFTWARE ENGINEERING GmbH

\* [https://en.wikipedia.org/wiki/Ready\\_Player\\_One](https://en.wikipedia.org/wiki/Ready_Player_One)



# Agenda



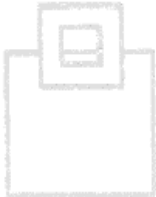
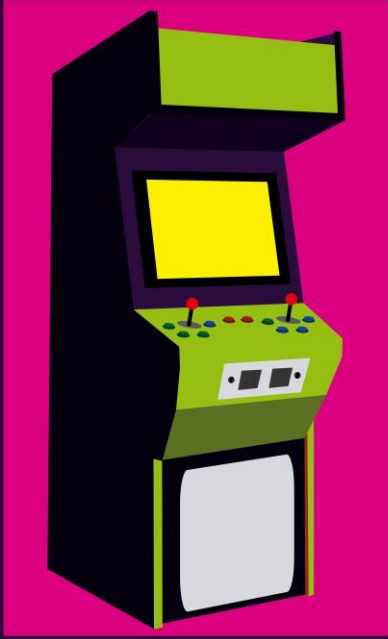
PRESS  
START!



PRESS  
START!



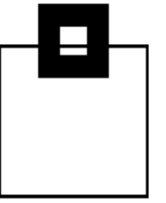
PRESS  
START!



# Agenda

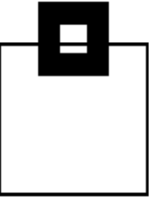
---

- Migration? Doesn't IBM do that for me?
- In the beginning was the Haakon...
- Details, details, details...
- How do I find and fix them all?
- Questions and Answers



# Agenda

---



- Migration? Doesn't IBM do that for me?

- In the beginning was the Haakon...

- Details, details, details...

- How do I find and fix them all?

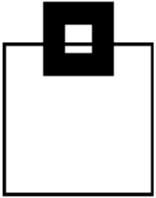
- Questions and Answers



# Migration? Doesn't IBM do that for me?

---

**Spoiler alert:**

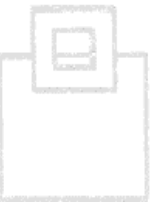
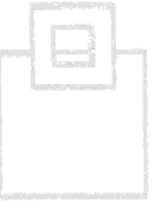
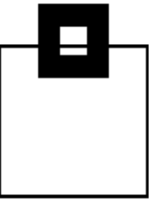


# Migration? Doesn't IBM do that for me?

---

**Spoiler alert:**

**No.**



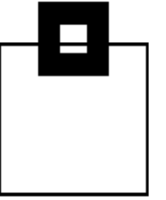
# Migration? Doesn't IBM do that for me?

---

**Spoiler alert:**

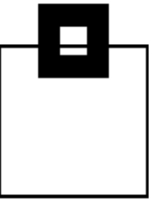
**No.**

**Well, actually, “bits” of it... But first, let's back up a bit and see why we are discussing this at all today!**



# Agenda

---



- Migration? Doesn't IBM do that for me?
- **In the beginning was the Haakon...**
- Details, details, details...
- How do I find and fix them all?
- Questions and Answers



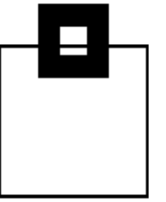
# In the beginning was the Haakon...

---

Haakon Roberts announced at the IDUG 2024 EMEA in Valencia a list of “things” that will **\*stop\*** a Db2 migration to vNext:

- Simple tablespace
- Segmented tablespace
- Classic partitioned tablespace
- Basic (6 byte) format
- BRF
- Hash Access
- Synonym
- VTAM

4



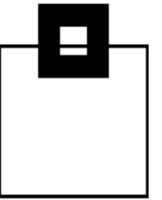
# In the beginning was the Haakon...

---

He then expanded upon it at the next IDUG EMEA 2025 in Düsseldorf by stating that “in fact we will stop you even getting to Db2 13 FL511 if any of these items exist”:

- Simple tablespace
- Segmented tablespace
- Classic partitioned tablespace
- Basic (6 byte) format
- BRF
- Hash Access
- Synonym
- VTAM

4



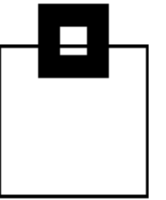
# In the beginning was the Haakon...

---

**This caused a bit of a stir in the audience... Since this time a lot of people and companies have asked me “When is IBM supplying us a migration CATMAINT to resolve all of the issues?”**

- Simple tablespace
- Segmented tablespace
- Classic partitioned tablespace
- Basic (6 byte) format
- BRF
- Hash Access
- Synonym
- VTAM

4



# In the beginning was the Haakon...

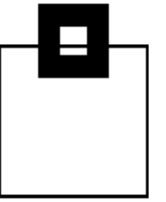
---

Naturally, I cannot answer this, but I think the phrase:

**“Don’t hold your breath.”**

**springs to mind!\***

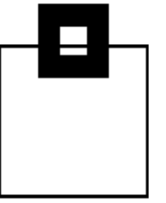
**\*I could be wrong and next month a CATMAINT is rolled out...**



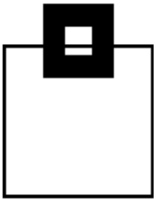
# Agenda

---

- Migration? Doesn't IBM do that for me?
- In the beginning was the Haakon...
- Details, details, details... **DEPRECATED**
- How do I find and fix them all?
- Questions and Answers

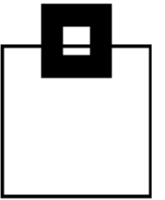


# Details, details, details...



# Details, details, details...

---



## Deprecated function

SIMPLE TS

Non-UTS base tablespace

6-byte RBA and LRSN format for the BSDS

BRF

## Recommended alternative

Migrate to PBG, SEGM, PBR

Migrate to PBG or PBR

Starting in DB2 11, convert the BSDS to use the extended 10-byte RBA and LRSN formats. The BSDS conversion must be completed before migration to DB2 12.

Migrate to RRF

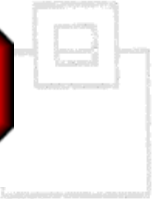
## Support removed

Db2 12 FL504

Db2 12 FL504

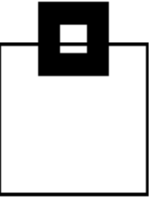
Db2 12

Db2 12



# Details, details, details...

---



## Deprecated function

HASH Tables  
SYNONYMS

## Recommended alternative

Drop hash  
Migrate to ALIAS

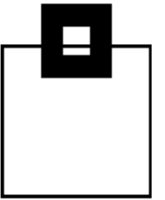
## Support removed

Db2 12 FL504  
Db2 12 FL504  
for create



# Details, details, details...

---



## Deprecated function

HASH Tables  
SYNONYMS

VTAM/SNA

## Recommended alternative

Drop hash  
Migrate to ALIAS

TCP/IP

## Support removed

Db2 12 FL504  
Db2 12 FL504  
for create  
Db2 vNext



# Details, details, details...

---

## Deprecated function

HASH Tables  
SYNONYMS

VTAM/SNA

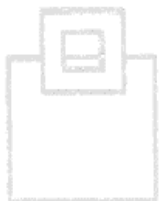
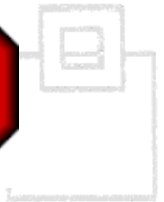
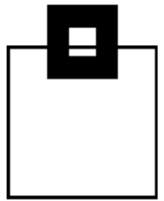
## Recommended alternative

Drop hash  
Migrate to ALIAS

TCP/IP

## Support removed

Db2 12 FL504  
Db2 12 FL504  
for create  
Db2 vNext

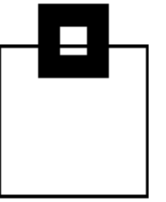


**So now, armed with a few SQLs, you can trawl through the Db2 Catalog and get a list of all the bad guys.**

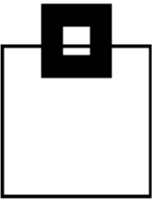
# Agenda

---

- Migration? Doesn't IBM do that for me?
- In the beginning was the Haakon...
- Details, details, details... **DEPRECATED**
- How do I find and fix them all?
- Questions and Answers

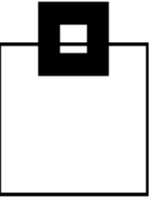


# How do I find and fix them all?



# How do I find and fix them all?

---



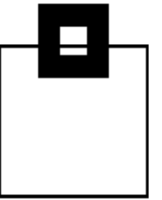
## Segmented or simple tablespaces with one or more tables:

```
SELECT TS.DBNAME
       , TS.NAME
FROM SYSIBM.SYSTABLESPACE TS
     , SYSIBM.SYSDATABASE DB
WHERE TS.DBNAME      = DB.NAME
      AND DB.TYPE    = ' '
      AND NOT DB.NAME = 'DSNDB01'
      AND NOT DB.NAME = 'DSNDB06'
      AND TS.PARTITIONS = 0
      AND TS.TYPE      = ' '
      AND TS.NTABLES   >= 1
ORDER BY 1 , 2
FOR FETCH ONLY
WITH UR ;
```



# How do I find and fix them all?

---



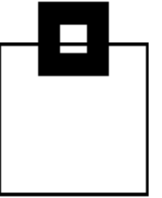
## Classic index-based partitioning tables:

```
SELECT TS.DBNAME, TS.NAME
       , STRIP(TP.IXCREATOR) CONCAT '.' CONCAT STRIP(TP.IXNAME)
FROM SYSIBM.SYSTABLESPACE TS
     , SYSIBM.SYSTABLEPART TP
     , SYSIBM.SYSDATABASE DB
WHERE TS.DBNAME = DB.NAME
      AND DB.TYPE = ' '
      AND TS.SEGSIZE = 0
      AND TS.PARTITIONS > 0
      AND TS.TYPE IN (' ' , 'L')
      AND TS.DBNAME = TP.DBNAME
      AND TS.NAME = TP.TSNAME
      AND NOT TP.IXCREATOR = ''
      AND TP.PARTITION = 1
      AND NOT DB.NAME = 'DSNDB01'
      AND NOT DB.NAME = 'DSNDB06'
ORDER BY 1 , 2
FOR FETCH ONLY
WITH UR ;
```



# How do I find and fix them all?

---



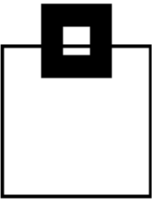
## Classic table-based partitioning tables:

```
SELECT TS.DBNAME
       , TS.NAME
FROM SYSIBM.SYSTABLESPACE TS
     , SYSIBM.SYSTABLEPART TP
     , SYSIBM.SYSDATABASE DB
WHERE TS.DBNAME = DB.NAME
      AND DB.TYPE = ' '
      AND TS.SEGSIZE = 0
      AND TS.PARTITIONS > 0
      AND TS.TYPE IN (' ' , 'L')
      AND TS.DBNAME = TP.DBNAME
      AND TS.NAME = TP.TSNAME
      AND TP.IXCREATOR = ' '
      AND TP.PARTITION = 1
      AND NOT DB.NAME = 'DSNDB01'
      AND NOT DB.NAME = 'DSNDB06'
ORDER BY 1 , 2
FOR FETCH ONLY
WITH UR ;
```



# How do I find and fix them all?

---



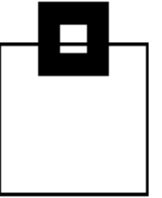
## Six byte RBA table partitions:

```
SELECT TP.DBNAME
       , TP.TSNAME
       , TP.PARTITION
       , TP.RBA_FORMAT
FROM SYSIBM.SYSTABLEPART TP
     ,SYSIBM.SYSDATABASE DB
WHERE TP.DBNAME      = DB.NAME
      AND DB.TYPE    = ' '
      AND NOT DB.NAME = 'DSNDB01'
      AND NOT DB.NAME = 'DSNDB06'
      AND (TP.RBA_FORMAT = ' '
           OR TP.RBA_FORMAT = 'B')
      AND NOT TP.TYPE = 'O'
ORDER BY 1 , 2 , 3
FOR FETCH ONLY
WITH UR ;
```



**Partition is 0 for TS objects.**

# How do I find and fix them all?



## Six byte RBA index partitions:

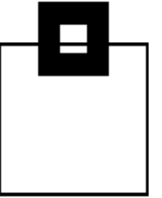
```
SELECT STRIP(IP.IXCREATOR) CONCAT '.' CONCAT STRIP(IP.IXNAME)
      , IP.PARTITION
      , IP.RBA_FORMAT
FROM SYSIBM.SYSINDEXPART IP
     ,SYSIBM.SYSINDEXES IX
WHERE IX.CREATOR = IP.IXCREATOR
      AND IX.NAME = IP.IXNAME
      AND NOT IX.DBID IN (1 , 6)
      AND (IP.RBA_FORMAT = ' '
           OR IP.RBA_FORMAT = 'B')
ORDER BY 1 , 2 , 3
FOR FETCH ONLY
WITH UR ;
```



**Partition is 0 for IX objects.**

# How do I find and fix them all?

---



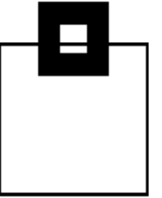
## BRF table partitions:

```
SELECT TS.DBNAME
       , TS.NAME
       , TP.PARTITION
FROM SYSIBM.SYSTABLESPACE TS
     ,SYSIBM.SYSTABLEPART TP
WHERE NOT TS.DBNAME = 'DSNDB01'
      AND NOT TS.DBNAME = 'DSNDB06'
      AND      TS.DBNAME = TP.DBNAME
      AND      TS.NAME     = TP.TSNAME
      AND NOT TS.TYPE     = 'O'
      AND NOT TS.TYPE     = 'P'
      AND      TP.FORMAT  = ' '
ORDER BY 1 , 2 , 3
FOR FETCH ONLY
WITH UR ;
```



# How do I find and fix them all?

---

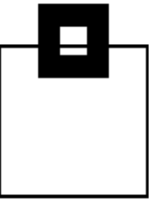


## Hash-organized tablespaces:

```
SELECT TS.DBNAME
       , TS.NAME
       , STRIP(TB.CREATOR) CONCAT '.' CONCAT STRIP(TB.NAME)
FROM SYSIBM.SYSTABLESPACE TS
     , SYSIBM.SYSDATABASE DB
     , SYSIBM.SYSTABLES TB
WHERE TS.DBNAME = DB.NAME
      AND DB.TYPE = ' '
      AND TS.ORGANIZATIONTYPE = 'H'
      AND TS.DBNAME = TB.DBNAME
      AND TS.NAME = TB.TSNAME
      AND NOT DB.NAME = 'DSNDB01'
      AND NOT DB.NAME = 'DSNDB06'
ORDER BY 1 , 2
FOR FETCH ONLY
WITH UR ;
```



# How do I find and fix them all?



## Synonyms:

```
SELECT STRIP(SY.CREATOR)   CONCAT '.' CONCAT STRIP(SY.NAME)  
      , STRIP(SY.TBCREATOR) CONCAT '.' CONCAT STRIP(SY.TBNAME)  
FROM SYSIBM.SYSSYNONYMS SY  
ORDER BY 1  
FOR FETCH ONLY  
WITH UR ;
```



# How do I find and fix them all?

---

## VTAM/SNA:

Check your Communication Database (CDB) tables. Any rows in these tables is not good:

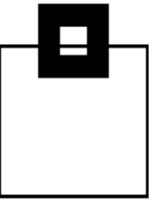
**SYSIBM.LULIST**

**SYSIBM.LUMODES**

**SYSIBM.LUNAMES with a non-blank LUNAME**

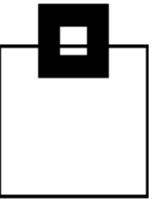
**SYSIBM.MODESELECT**

Check your DDF parameters, on every member, to make sure that the IPNAME is **\*not\*** set to “-NONE”.



# How do I find and fix them all?

---



## VTAM/SNA:

But why go to all the bother? This is not a trivial migration! IBM state:

### Why TCP/IP?

Switching to TCP/IP isn't just mandatory—it brings real advantages:

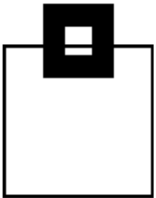
- **Enhanced security:** Supports modern encryption and authentication methods like AT-TLS and JSON Web Tokens (JWT).
- **Improved performance:** Reduces CPU consumption and supports 64-bit communication buffers.
- **Cost efficiency:** Improves efficiency by leveraging zIIP processors.

### Details here:

<https://www.ibm.com/docs/en/db2-for-zos/13.0.0?topic=systems-converting-tcpip-communication-from-vtam>

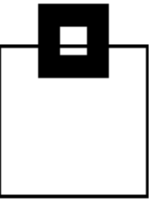


# How do I find and fix them all?



# How do I find and fix them all?

---



**Segmented or simple tablespaces with single tables:**

**Fix:** `ALTER TABLESPACE db.ts MAXPARTITIONS 1 ;`

**For multi-table tablespaces you must be on Db2 12 FL508 or higher:**

**Fix:** `ALTER TABLESPACE db.ts  
MOVE TABLE xxx.yyy  
TO TABLESPACE db.TSnew... ;`

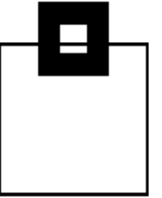
**Note:** You must first create the new tablespace(s) with correct BP, CCSID and LOGGED parameters. Then after all of the ALTER(s) you execute a REORG db.ts to actually resolve all of the pending DDL. Next you will need a RUNSTATS db.TSnew... SHRLEVEL REFERENCE and a REBIND of any packages that were invalidated by the MOVE TABLE.

**The Db2 Dir/Catalog will be done by IBM, so we are saved that at least!**



# How do I find and fix them all?

---



## Classic index-based partitioning tables:

**Two-stage fix:** `ALTER INDEX aaa.bbb NOT CLUSTER ;`  
`ALTER INDEX aaa.bbb CLUSTER ;`



## Now you have table-based so now just the flip to UTS PBR:

`ALTER TABLESPACE ddd.eee SEGSIZE 64 ;`

## Classic table-based partitioning tables or zero DSSIZE:

**Fix:** `ALTER TABLESPACE ddd.eee SEGSIZE 64 ;`

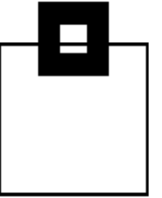


**Note:** After these ALTERs you will require REORGs to action the pending DDL changes. Not required if DEFINE NO.



# How do I find and fix them all?

---



## Six byte RBA index or table partitions:

**Fix:** REORG

For DEFINE NO objects only a DROP and a CREATE will “fix” the problem  
– However, watch out for any dependencies as even DEFINE NO can be referred to by SQL, of course!



## BRF table partitions:

**Fix:** REORG

**Note:** This REORG will not fix the BRF problem if a table exists with a VALID or EDIT procedure (VALPROC or EDPROC).



For DEFINE NO objects make sure the REORG uses “ROWFORMAT RRF”



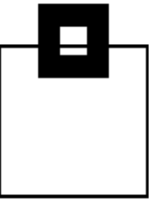
# How do I find and fix them all?

---

## Hash-organized tablespaces:

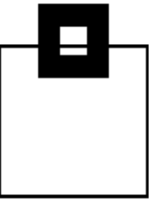
**Fix:** `ALTER TABLE xxxxxxxx.yyyyyyyy DROP ORGANIZATION ;`

**Then possibly create a new index for normal access as the Hash Index is automatically dropped by using this command.**



# How do I find and fix them all?

**SYNONYMS:**



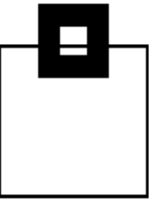
# How do I find and fix them all?

---

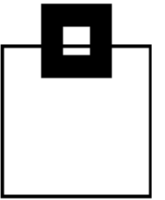
## SYNONYMS (page 1 of 6):

These are “tricky” and you must take a multi-modal approach. Start with dependency checks on the SYNONYM to actually see if it might cause problems by the DROP:

```
SELECT BCOLNAME
       , BOWNER
       , DSCHEMA
       , DNAME
       , DCOLNAME
       , CASE DTYPE
         WHEN 'B' THEN 'Basic Trigger'
         WHEN 'C' THEN 'Generated Column'
         WHEN 'F' THEN 'Function'
         WHEN 'I' THEN 'Index'
         WHEN 'M' THEN 'Materialized Query table'
         WHEN 'O' THEN 'Procedure'
```



# How do I find and fix them all?



## SYNONYMS (page 2 of 6):

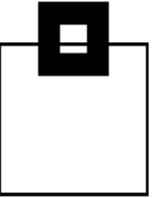
```
        WHEN 'V' THEN 'View'                '
        WHEN 'X' THEN 'Row Permission'      '
        WHEN 'Y' THEN 'Column Mask'        '
        WHEN '1' THEN 'Advanced Trigger'    '
        ELSE          'Unknown'            '
        END
    , DOWNER
FROM SYSIBM.SYSDEPENDENCIES
WHERE BSHEMA = 'Synonym Schema'
    AND BNAME = 'Synonym Name'
    AND BTYPE = 'S'
ORDER BY 1 , 2 , 3 , 5 , 6 , 7
FOR FETCH ONLY
WITH UR ;
```



**Note:** If you have old, DB2 10 or earlier, base objects: Views/MQTs, SQL Scalar functions, procedures or triggers. Then you must also check the SYSVIEWDEP, SYSPACKDEP and SYSSEQUENCEDEP tables as well.

# How do I find and fix them all?

---



## SYNONYMS (page 3 of 6):

The output shows you any extra work you might have to do! Like **DROPPing** any dependent **FUNCTIONs**, **VIEWs** or **MQTs** etc. Then you run another SQL to check out any package dependencies:

```
SELECT DCOLLID
       , DNAME
       , HEX(DCONTOKEN)
       , CASE DTYPE
           WHEN 'F' THEN 'Compiled SQL scalar function      '
           WHEN 'N' THEN 'Native SQL routine package       '
           WHEN 'O' THEN 'Original copy of a package       '
           WHEN 'P' THEN 'Previous copy of a package       '
           WHEN 'R' THEN 'Reserved for IBM use            '
           WHEN 'T' THEN 'Basic Trigger                '
           WHEN ' ' THEN 'Not a Trigger or native SQL package'
```



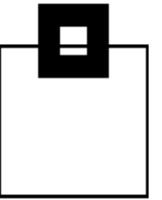
# How do I find and fix them all?

---

## SYNONYMS (page 4 of 6):

```
        WHEN '1' THEN 'Advanced Trigger'
        ELSE          'Unknown'
        END
    , DOWNER
FROM SYSIBM.SYSPACKDEP
WHERE BQUALIFIER = 'Synonym Schema'
    AND BNAME     = 'Synonym Name'
    AND BTYPE     = 'S'
ORDER BY 1 , 2
FOR FETCH ONLY
WITH UR ;
```

**This output shows you a list of packages that will require at least a REBIND after you have got rid of the SYNONYM.**



# How do I find and fix them all?

---

**SYNONYMS (page 5 of 6):**

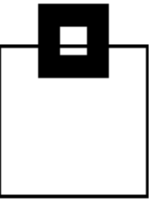
**Then, finally, comes the fix:**

**Generate a set of SPUFI DDL statements:**

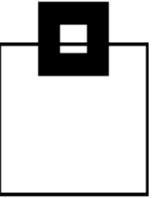
```
SET CURRENT SQLID = 'Synonym schema' ;
DROP SYNONYM 'Synonym name' ;
COMMIT ;
CREATE ALIAS 'Synonym schema'.'Synonym name'
    FOR 'Table creator'.'Table name' ;
COMMIT ;
```

**GRANTS do not have to be checked for SYNONYMS as GRANTS are recorded in the catalog against the underlying TABLES which have not been dropped.**

**Once this is done then recreate all of the dependent objects that had to be dropped and all of their GRANTS and any of their dependencies as well.**



# How do I find and fix them all?



**SYNONYMS (page 6 of 6):**

**So what is the difference between a SYNONYM and an ALIAS anyway?**

Characteristic	Synonyms (deprecated)	Aliases
Can be created in application compatibility V12R1M504 and higher?	No	Yes
Requires authorization to create?	No	Yes
Can be defined on objects not at the current server?	No	Yes
Can be defined on the name of an object that does not yet exist?	No	Yes, but it must exist when used
Is dropped when referenced objects are dropped?	Yes	No
Uses a qualified object name for the object?	No, one-part name	Yes
Can be referenced or used by users other than the object owner?	No	Yes

# How do I find and fix them all?

---

## VTAM/SNA:

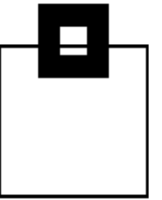
**Fix:** There is no quick fix for this at all. If you are using VTAM/SNA for inter Db2 communication you **\*must\*** move to TCP/IP as soon as you can!

CDB Entries should simply be deleted apart from the last empty LUNAME entry.

A BSDS update must be run using DSNJU003 to set the IPNAME value to anything but **-NONE**:

```
DDF IPNAME=myhost
```

This completely switches VTAM/SNA, **for this member**, off.



# How do I find and fix them all?

---

**That's a lot of work, isn't it?**

**Wouldn't it be much better, faster and easier if you had a small piece of free software that does it nearly \*all\* for you?**

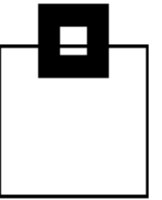
**That scanned and listed out all the vNext problems you are going to hit?**

**That, in a paid version, generates all the ALTERs that you need. Including REORG, RUNSTATS, REBIND and DROP statements? Leaving just CDB, VTAM and SYNONYMS to be investigated by us humans!**

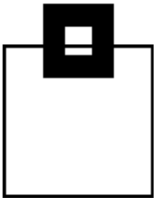
**It already exists today and you can download it and run it within 20 minutes.**

**MigrationReadiness HealthCheck for Db2 z/OS**

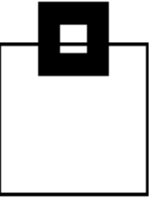
**<https://www.segus.com/products/productlist/migrationreadiness-healthcheck/>**



# How do I find and fix them all?



# How do I find and fix them all?



## When you run it the output looks like (1 of 5):

Db2 MigrationReadiness HealthCheck V2.0 for DD10 V13R1M508 started at 2026-02-05-17.33.04

CATALOG AND DIRECTORY REORG NOT AVAILABLE

Lines with \*\*\* are deprecated features

Lines with **MMM** are migration blockers

Lines with XXX are definition errors

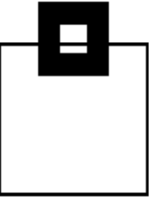
```
Number of DATABASES           :      311
# of empty DATABASES         :      137
# of implicit DATABASES      :      221
# of empty implicit DATABASES:      132

Number of TABLESPACES       :     3625
of which HASH organized      :         0
of which PARTITIONED CLASSIC :         0
# Partitions                  :         0
of which SEGMENTED           :      26 MMM
of which SIMPLE               :         3 MMM
of which LOB                  :        121
of which UTS PBG              :     3453
# Partitions                  :     3457
of which UTS PBR (Absolute)  :         0
# Partitions                  :         0
of which UTS PBR (Relative)  :         5
# Partitions                  :        21
of which XML                  :         17

Number of tablespaces as LARGE :         0
Number of empty tablespaces   :        101
Number of multi-table TSs     :         11 MMM
# of tables within these      :         47
Number of incomplete TS      :        111 XXX
Number of INSERT ALG 0 TS     :     3607
Number of INSERT ALG 1 TS     :         18
```



# How do I find and fix them all?



## When you run it the output looks like (2 of 5):

```
Number of INSERT ALG 2 TS      :      0

Number of tables                :    7098
  of which ACCELERATOR ONLY    :      0
  of which ALIASes             :   3397
  of which ARCHIVES            :      0
  of which AUXs                :    111
  of which CLONES              :      0
  of which GTTs                :    140
  of which HISTORYs           :      2
  of which MQTs                :      1
  of which TABLES             :   3419
  of which VIEWS               :     11
  of which XMLs                :     17
Number of tables with Audit    :      1
Number of tables with Data Cap :      0
Number of tables incomplete    :     2 XXX
Number of tables with control  :      1

Number of RLF DSNRLMT__ tables :      0
  of which columns deprecated  :      0
Number of RLF DSNRLST__ tables :      1
  of which columns deprecated  :      0

Number of PLAN_TABLES         :     60
  of which deprecated          :    44 ***

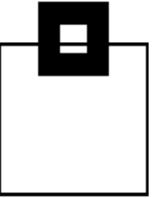
Number of SYNONYMS            :      0

Number of UNICODE V11 Columns :      0

Number of PROCEDURES          :    143
  of which SQL EXTERNAL        :      0
```



# How do I find and fix them all?



## When you run it the output looks like (3 of 5):

```
of which EXTERNAL      :      111
of which NATIVE SQL    :       32

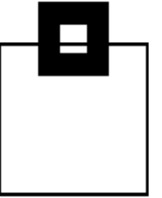
Number of FUNCTIONS    :       93
of which EXTERNAL TABLE :      39
of which EXTERNAL SCALAR :      42
of which SOURCED AGGREGATE :      0
of which SOURCED SCALAR :      0
of which SQL TABLE    :      0
of which SQL SCALAR    :      10
of which SYSTEM-GENERATED :      2

Number of Indexes      :     4665
of which HASH          :      0
of which type 2        :     4664
  # of partitioned IXs :      0
  # Partitions          :      0
of which DPSI          :      1
  # Partitions          :      3
of which PI            :      0
  # Partitions          :      0
Number of indexes COPY YES :      18
Number of indexes COMPRESS YES :      0

Number of table partitions :     3645
of which DEFINE NO      :     1626
of which six byte RBA <11 NFM:      0
of which six byte RBA Basic :      0
of which ten byte RBA   :     2019
Number of TP in BRF      :      17 MMM
Number of TP with COMPRESS Y :      97
Number of TP with COMPRESS F :      0
Number of TP with COMPRESS H :      0
```



# How do I find and fix them all?



## When you run it the output looks like (4 of 5):

```
Number of TP with TRACKMOD YES :      3642

Number of index partitions      :      4667
  of which DEFINE NO           :      2100
  of which six byte RBA <11 NFM:         0
  of which six byte RBA Basic   :         0
  of which ten byte RBA         :      2567

Number of STOGROUPS             :         5
Number of non-SMS VOLUMES       :         0

Number of PLANS                  :         49
  of which DBRMs direct         :         0
  # of SQL statements           :         0

Number of PACKAGES (total)      :      3619
  of which VALID = A            :         10
  of which VALID = H            :         0
  of which VALID = N            :         1
  of which VALID = Y            :      3607
  of which VALID = S            :         1
  of which OPERATIVE = N        :         0
  of which OPERATIVE = Y        :      3619
  of which OPERATIVE = R        :         0

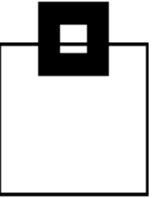
Old RELBOUND executed packages :         0

Number of PACKAGES (distinct)   :         565

Number of Original PACKAGES     :      2206
Number of Previous PACKAGES     :      2206
Number of Phased-out PACKAGES   :      2438
Total number of PACKCOPY        :      6850
```



# How do I find and fix them all?



## When you run it the output looks like (5 of 5):

```
of which VALID = A      :      630
of which VALID = H      :          0
of which VALID = N      :      259
of which VALID = Y      :     5255
of which VALID = S      :      706
of which OPERATIVE = N  :          0
of which OPERATIVE = Y  :     6850
of which OPERATIVE = R  :          0

Number of SQL statements :    166744

LULIST entries found    :          0
LUMODES entries found   :          0
LUNAMES entries found :      6 MMM
MODESELECT entries found :          0

ZPARAM CHECK_FASTREPLICATION set to REQUIRED is at correct value REQUIRED and Ok.
ZPARAM CMTSTAT set to INACTIVE is at correct value INACTIVE and Ok.
ZPARAM DISALLOW_SEL_INTO_UNION set to YES is at correct value YES and Ok.
ZPARAM MATERIALIZE_NODET_SQLTUDF set to NO is not at correct value YES. ***
ZPARAM PREVENT_NEW_IXCTRL_PART set to YES is at correct value YES and Ok.

DDF command prefix -DD10 the IPNAME is still set to "-NONE". MMM

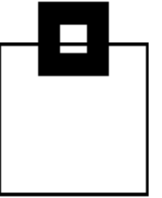
Db2 MigrationReadiness HealthCheck V2.0 for DD10 V13R1M508 ended at 2026-02-05-17.33.05

Migration to vNext is not possible

Db2 MigrationReadiness HealthCheck ended with RC:      4
```



# How do I find and fix them all?



## It also outputs a detailed list of everything found:

```
Segmented DB: ROYXTEST TS: DBRMTS01
Classic partitioned DB: DOGTEST1 TS: DOGS02
Synonym: BOXWEL2.ELVISB
for SYSIBM.SYSTABLES
Has the following SYSIBM.SYSPACKDEF:
DCOLLID   : MDB2VNEX_TEST
DNAME     : MORE0001
DCONTOKEN : 1B1E7E6F0E8FEFE0
DTYPE     : Not a Trigger or native SQL package
```

This package has been deprecated

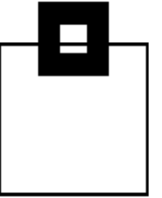
*Author message:*

Package no longer supported. Use at your own risk.



# How do I find and fix them all?

---

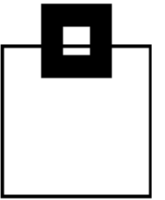


**As well as outputting a list of everything found that is a blocker:**

```
Simple DB: DSNDB01 TS: SCT02
Segmented DB: DSNDB01 TS: SYSUTILX
Segmented DB: DSNDB06 TS: SYSALTER
Segmented DB: DSNDB06 TS: SYSCONTX
Segmented DB: DSNDB06 TS: SYSDDF
Segmented DB: DSNDB06 TS: SYSEBCDC
Simple DB: DSNDB06 TS: SYSGPAUT
Segmented DB: DSNDB06 TS: SYSGRTNS
Segmented DB: DSNDB06 TS: SYSHIST
Segmented DB: DSNDB06 TS: SYSJAVA
Segmented DB: DSNDB06 TS: SYSROLES
Segmented DB: DSNDB06 TS: SYSSEQ
Segmented DB: DSNDB06 TS: SYSSEQ2
Segmented DB: DSNDB06 TS: SYSSTATS
Segmented DB: DSNDB06 TS: SYSTARG
Segmented DB: DSNDB06 TS: SYSTSASC
Segmented DB: DSNDB06 TS: SYSTSUNI
Segmented DB: DSNDB06 TS: SYSTSXTM
Segmented DB: DSNDB06 TS: SYSTSXTS
Simple DB: DSNDB06 TS: SYSUSER
Segmented DB: DSNDB06 TS: SYSXML
Segmented DB: DSNDB07 TS: DSN32K00
Segmented DB: DSNDB07 TS: DSN32K01
.
.
.
```



# How do I find and fix them all?



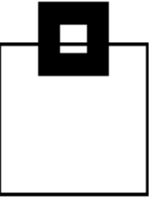
If the optional paid version is licensed, you get a list of ALTERs, DROPs REORGs, RUNSTATS and REBINDs:

```
ALTER TABLESPACE ROYXTEST.DBRMST01 MAXPARTITIONS 1 ;
COMMIT ;
ALTER INDEX DOGTEST1.INDEX_GREATER_THAN_EIGHT_ON_DOGTAB2
  NOT CLUSTER ;
ALTER INDEX DOGTEST1.INDEX_GREATER_THAN_EIGHT_ON_DOGTAB2
  CLUSTER ;
COMMIT ;
ALTER TABLESPACE DOGTEST1.DOGS02 SEGSIZE 64 ;
COMMIT ;
SET CURRENT SQLID = 'BOXWEL2' ;
DROP SYNONYM ELVISB ;
COMMIT ;
CREATE ALIAS BOXWEL2.ELVISB
FOR SYSIBM.SYSTABLES ;
COMMIT ;
ALTER TABLE DB2V11.DB2V11TB1
  ALTER COLUMN X06
    SET DATA TYPE VARCHAR( 25)
;
COMMIT ;
```



# How do I find and fix them all?

---



## Deprecated work files?

As you have just seen workfiles are included but their usage is a bit “unclear” ...

- If your workfile tablespace is segmented and non-UTS and with zero as a SECQTY it will be used for CGTTs, Large Sorts, Materializing Views etc. which can span more than one tablespace
- If your workfile tablespace is a PBG with or without a SECQTY it will be used for DGTTs, Scrollable cursors and SQL MERGE operations where the data cannot span more than one tablespace

You normally need both!  
Then a few ZPARMs rear their ugly heads...



# How do I find and fix them all?

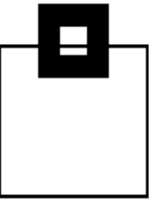
---

**WFDBSEP** default is NO. If set to YES Db2 will, for DGTG usage etc., allocate to PBG or Segmented with non zero secqty. If non-workfile usage is required it will attempt to allocate to segmented non-UTS with zero SECQTY. If either of these selections fails SQL code -904 is returned.

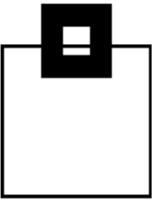
If set to NO it will still attempt the “preferred space” but if none are available it will fail over to another type of workfile tablespace.

**MAXTEMPS** default is zero. You can put in here a number of MB or GB that is the limit an agent can allocate. This is quite handy for stopping run-away cartesian join style transactions.

**WFSTGUSE\_AGENT\_THRESHOLD** default is zero. It can send an alert when nn% of all workfiles are in use by a single agent. You could set this to, say, 30 and monitor the xxxxMSTR to see who is hogging the workfile space and take corrective actions.



# Selection order: **PBG** and WFDBSEP



- 32K **PBG**
- 4K **PBG**
- 32K segm  
– SECQTY>0
- 4K segm  
– SECQTY>0
- 32K segm  
– SECQTY=0
- 4K segm  
– SECQTY=0

- 32K **PBG**
- 4K **PBG**
- 32K segm  
– SECQTY>0
- 4K segm  
– SECQTY>0
- 32K segm  
– SECQTY=0
- 4K segm  
– SECQTY=0
- 32K segm  
– SECQTY>0
- 4K segm  
– SECQTY>0
- 32K **PBG**
- 4K **PBG**

**PBGs** are selected independent of their SECQTY

**SORT** is spannable, **TEMP** is not (DGTTs etc.)

Thanks to Peter Hartmann for the use of this graphic

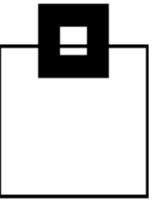


# How do I find and fix them all?

---

**In Db2 13 FL508 came a small DDL change where you can now set the “type” of a workfile tablespace much better than ever before.**

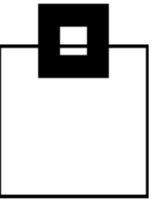
**In the DDL for CREATE TABLESPACE you simply specify FOR SORT or FOR DGTG and you are done!**



# Agenda

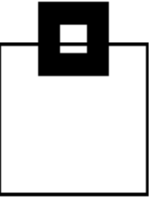
---

- Migration? Doesn't IBM do that for me?
- In the beginning was the Haakon...
- Details, details, details... **DEPRECATED**
- How do I find and fix them all?
- Questions and Answers



# Results

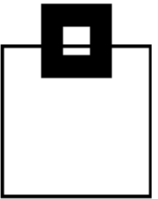
---



**So far we have had only positive feedback from the Pocket Tool:**

- **It does what it says and found way too much!**
- **Discovered things we didn't know we had...**
- **At least we can plan the pre-migration now...**
- **What about a regex to generate "good" new Tablespace names?**
- **Can it fix VTAM/SNA usage?**





## Yoo hoo!



ZERTIFIKAT | CERTIFICATE | CERTIFICAT | CERTIFICADO | CERTIFIKAT | 51314 | 证书 | 인증서



## CERTIFICATE



**Management system as per ISO/IEC 27001:2022**

In accordance with TÜV AUSTRIA procedures, it is hereby certified that

**SOFTWARE ENGINEERING**

**SOFTWARE ENGINEERING GmbH**  
**Vagedesstraße 19**  
**D-40479 Düsseldorf**

applies a management system in line with the above standard for the following scope

**development, sales and support of software for optimizing Db2 for z/OS including all associated tasks**

Valid Statement of Applicability (SoA): V1.0, 20.11.2025  
 Certificate Registration No. 10201250015745

Valid until: 2029-01-26  
 Initial certification: 2026-01-27

*E. Richter*  
 Certification Body  
 at TÜV AUSTRIA GMBH

Vienna, 2026-01-27

This certification was conducted in accordance with TÜV AUSTRIA auditing and certification procedures and is subject to regular surveillance audits.  
 TÜV AUSTRIA GMBH · Deutschstraße 10 · A-1230 Wien · www.tuv.at



Online Verification  
[www.tuv.at/onlineverification](https://www.tuv.at/onlineverification)

Verification of all data on this certificate is subject to the approval of TÜV AUSTRIA [10]

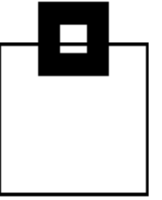
026699-25-01





# Questions & Answers

---



# Questions & Answers

