

# DB2 13 FOR Z/OS FL 504

## OBJECT LEVEL UTILITY HISTORY

### WHAT IS IT AND HOW DO I USE IT ?

Patric Becker

Rocket Software

# Thank you !

All credits for this presentation go to:

Laura Kunioka-Weiss, Principal Software Engineer

Db2 for z/OS Utilities Development, Backup and Recovery !

# Agenda

- Overview
- Understanding History Information
- Using Utility History
- Closing

# Agenda

- Overview
- Understanding History Information
- Using Utility History
- Closing

# The Backdrop

---

Utilities are essential tasks run every day to create backups, reorganize data, load data, unload data, and gather statistics. It's critical for these processes to run successfully, efficiently and in a specific timeframe to avoid application impact.

However, information about utility executions, both near real-time and historical, is difficult to gather and analyze.

# Who Will Benefit From Utility History ?

---

Do you need  
to manage  
and tune your  
utility  
executions ?

- Quickly check if any utilities failed in the past 24 hours and take corrective actions ?
- Assess the execution of a specific utility job ?
- Compare the performance of certain utilities over time ?
  - COPY, RUNSTATS, LOAD, REORG
- Look for
  - beneficial trends and apply the same strategy to other areas ?
  - worsening trends and take preventative actions ?
- Analyze utility executions for balancing workloads
  - Move objects from one job to another
  - Move job executions from one window to another
- When was the last REORG on table space x ?
- What is the trending size of table space x ?
- How can I determine objects affected by an active or stopped utility ?

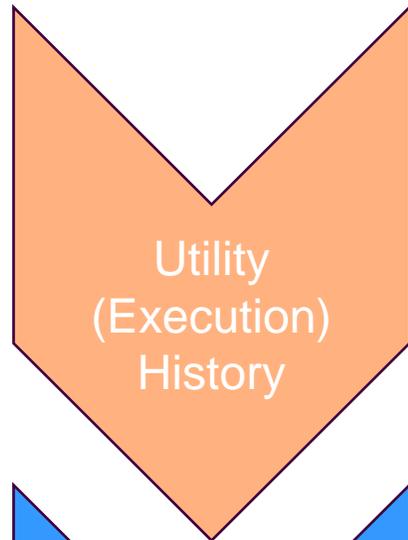
# Solution Design Points

---

- ❑ Collect and save useful, essential execution information common across IBM Db2 utilities in new Db2 catalog tables
  - ❑ Two-tiered table approach correlated with an event ID
    - ❑ Utility (Execution) History in **SYSIBM.SYSUTILITIES**
    - ❑ Utility Object-level History in **SYSIBM.SYSOBJEVENTS**
  - ❑ Information available in near real-time while the utility is running
  - ❑ Information easily accessible with SQL queries
- ❑ Prioritize successful utility execution over history collection
  - ❑ May not be suitable for audit purposes **⚠ CAUTION**

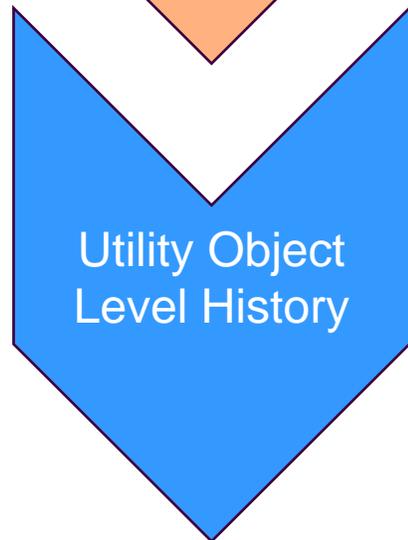
# Staged Implementation

---



Db2 13 for z/OS – Function Level 501, Catalog Level 501

- ❑ GA May 2022



Db2 13 for z/OS – Function Level 504, Catalog Level 504

- ❑ Available October 2023
- ❑ APARs PH55476, PH55914, PH55915, PH55916, PH54919 (FL 504)

# History Activation and EVENTID

---

- ❑ Activate utility history with UTILITY\_HISTORY subsystem parameter
  - ❑ **UTILITY** (FL 501 and above)
    - ❑ Execution information
  - ❑ **OBJECT**(FL 504 and above)
    - ❑ Execution information and Object-level information
- ❑ EVENTID column in **SYSUTILITIES**, **SYSOBJEVENTS**, SYSCOPY catalog tables
- ❑ Unique utility event (execution) identifier
- ❑ Reported for each execution in job output with new message  
DSNU3031I UTILITY HISTORY COLLECTION IS ACTIVE.  
LEVEL: *UTILITY-or-OBJECT*, EVENTID: *event-id*
- ❑ REPORT RECOVERY shows EVENTID for SYSCOPY rows and system-level backups

# SYSIBM.SYSUTILITIES Catalog Table

Db2 13 FL501

❑ One row for each utility execution

❑ **EVENTID** unique utility event ID #

❑ Also added to SYSCOPY

❑ **NAME**

❑ Utility executed

❑ **INSERTEDBY** = ,DB2'

❑ **JOBNAME**

❑ **UTILID**

❑ **USERID**

❑ **STARTTS**

❑ **ENDTS**

❑ **ELAPSEDTIME**

❑ **CPUTIME**

❑ **ZIIPTIME**

❑ **RETURNCODE**

❑ **CONDITION**

❑ Active, stopped, ended, terminated, forced

❑ **RESTART**

❑ **NUMOBJECTS**

❑ **LISTNAME**

❑ **STARTLOGPOINT**

❑ **GROUP\_MEMBER**

❑ **SORTNAME**

❑ **SORTCPUTIME**

❑ **SORTZIIPTIME**

❑ **PHASEnNAME**

❑ **PHASEnET**

❑ **PHASEnCPUT**

❑ **PHASEnZIIP**

❑ **PHASEnDATA**

Where n is 1 - 14.  
Reserved for future  
use, currently set  
to NULL.

❑ Table Space **DSNDB06.SYSTSUTL**

# SYSIBM.SYSOBJEVENTS Catalog Table

Db2 13 FL504

- ❑ One row for each partition or non-partitioned object for Db2 Utility executions

- ❑ **EVENTID** utility event ID #
  - ❑ Correlates with SYSUTILITIES.EVENTID and SYSCOPY.EVENTID
- ❑ **DBID**
- ❑ **PSID**
- ❑ **PARTITION**
  - ❑ Partition number or zero for non-partitioned objects
- ❑ **INSTANCE**
- ❑ **DBNAME**
- ❑ **SPACENAME**
- ❑ **OBJTYPE**
  - ❑ ,T' or ,I'
- ❑ **EVENTTS**
  - ❑ Begin processing **TIMESTAMP**
- ❑ **COUNT**
  - ❑ Number of rows, records, keys, pages or LOBs processed
- ❑ **ELAPSEDTIME**
  - ❑ **COPY** or **RECOVER** time for creation or restore of sequential image copy
- ❑ **INSERTEDBY = ,DB2'**
  - ❑ ,DB2' for IBM Db2 Utilities

- ❑ Table Space **DSNDB06.SYSTSOEV**

# Utility Support

## Execution and Object-Level History

- CHECK DATA
- CHECK INDEX
- CHECK LOB
- COPY
- COPYTOCOPY
- LOAD
- MERGECOPY
- MODFIY RECOVERY
- MODIFY STATISTICS
- QUIESCE
- REBUILD INDEX
- RECOVER
- REORG
- REPAIR
- RUNSTATS
- UNLOAD

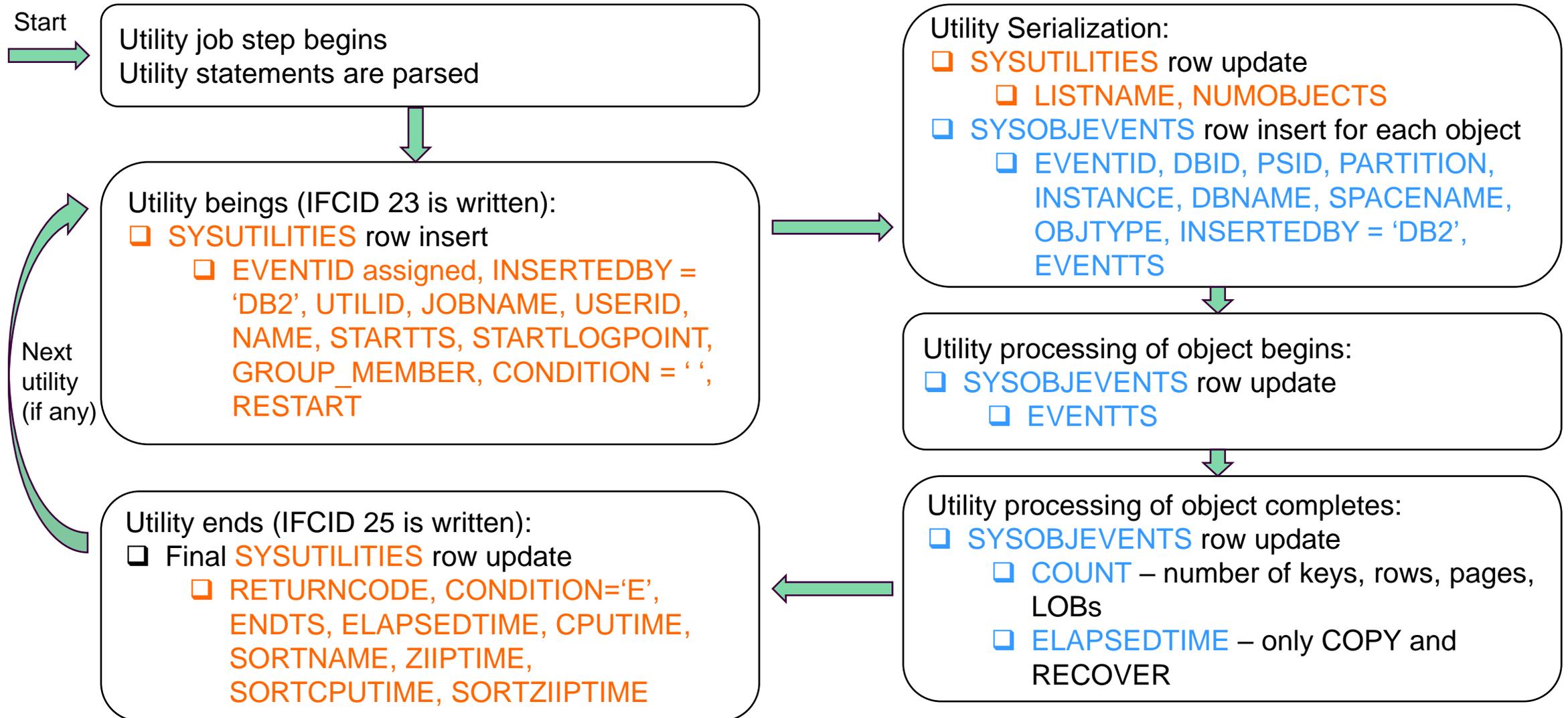
## Execution History

- BACKUP SYSTEM
- CATMAINT
- REPORT RECOVERY
- REPORT TABLESPACESET
- STOSPACE

## Not supported

- DIAGNOSE
- EXEC SQL
- LISTDEF
- OPTIONS
- RESTORE SYSTEM
- TEMPLATE
- Stand-alones: DSN1COPY,  
DSNJU003, etc.

# Utility History - Flow



# Agenda

- Overview
- **Understanding History Information**
- Using Utility History
- Closing

# Utility Execution History **SYSUTILITIES** row

How many **SYSUTILITIES** rows are inserted for a utility statement ?  
It depends...

Explicit object names or no object names → **One row**

LISTDEF list - Number of rows vary based on utility and LISTDEF options

- ❑ COPY, RECOVER, COPYTOCOPY, QUIESCE → **One row**
- ❑ REORG TABLESPACE, RUNSTATS TABLESPACE, UNLOAD, REORG INDEX, REBUILD INDEX on a list of table spaces, MODIFY RECOVERY, MODIFY STATISTICS, MERGECOPY, REPORT RECOVERY:
  - ❑ Requested at the table space or index space level → **One row** for each space
- ❑ REORG TABLESPACE, RUNSTATS INDEX, UNLOAD, REBUILD INDEX on a list of index spaces, CHECK INDEX:
  - ❑ Requested on one or more partitions → **One row** for each related group of indexes or partitions
- ❑ Related means partitions in the same table space or indexes over the same table

# Understanding **SYSUTILITIES** Columns

---

## ❑ **EVENTID**

- ❑ Unique utility event (execution) identifier, BIGINT data type
- ❑ Set by new Db2 sequence SYSIBM.DSNSEQ\_EVENTID
- ❑ Starts at 1000. Max value is 9,223,372,036,854,775,807.
  - ❑ May not increase sequentially according to order of utility executions
    - ❑ Each Db2 member in a data sharing group has a sequence cache
    - ❑ Sequence numbers in cache are not preserved when Db2 is restarted
- 💡 ❑ For chronological order of utility execution use STARTTS (start timestamp) column value to order SYSUTILITIES rows in queries

## ❑ **NAME**

- ❑ Utility name
- 💡 ❑ Like utility functions are grouped together
  - ❑ REORG TABLESPACE and REORG INDEX grouped under REORG
  - ❑ RECOVER TABLESPACE and RECOVER INDEX grouped under RECOVER

# Understanding **SYSUTILITIES** Columns (cont'd)

---

- ❑ **JOBNAME, UTILID, USERID** columns
  - ❑ Job information
- ❑ **STARTTS, ENDTS** columns
  - ❑ Local timestamps
- ❑ **LISTNAME** column
  - ❑ LISTDEF list name, otherwise NULL
- ❑ **RESTART** column
  - ❑ Utility restarted? 'N' or 'Y'
- ❑ **STARTLOGPOINT** column
  - ❑ RBA or LRSN when utility started
- ❑ **GROUPMEMBER** column
  - ❑ Data sharing –member name where utility executed
  - ❑ Non-data sharing -NULL
- ❑ **INSERTEDBY** column
  - ❑ 'DB2' for IBM Db2 utilities

# Understanding **SYSUTILITIES** Columns (cont'd)

## ❑ **NUMOBJECTS** column

- ❑ Number of objects the utility will process
- ❑ Objects accessed for read or write
- 💡 ❑ Matches number of **SYSOBJEVENTS** rows
- ❑ Partitioned objects – each partition is counted as one object
- ❑ Non partitioned objects – counted as one object

## ❑ **RETURNCODE** and **CONDITION** columns

RETURNCODE	CONDITION	Explanation
NULL	Blank	Utility is active or stopped. Issue –DIS UTIL(utilid) command -> DSNU105I for active or DSNU100I for stopped.
0 or 4	E	Utility ended successfully. For RETURNCODE=4, one or more warning messages issued.
8	E	Utility ended with one or more errors
8	T	Active utility accepted –TERM UTIL command and terminated itself
NULL	T	-TERM UTIL command terminated the stopped utility
NULL	F	-START DB SP ACCESS(FORCE) command terminated the stopped utility

# Understanding **SYSUTILITIES** Columns (cont'd)

---

History performance statistics are consistent with IFCID 25 utility end trace record.

## ❑ **ELAPSEDTIME, CPUTIME, ZIIPTIME** columns

- ❑ In microseconds ( $\mu$ s), divide by

- ❑ 1 mil. for seconds

- ❑ 60 mil. for minutes

## ❑ **CPUTIME** is general processor CPU time, does not include zIIP time

## ❑ **ZIIPTIME** when accounting class 1 trace is active, does not include SORT zIIPtime



- ❑ Column values reflect total for all objects processed by the utility

## ❑ **SORTNAME** column

- ❑ When applicable DB2SORT or DFSORT

## ❑ **SORTCPUTIME, SORTZIIPTIME** columns

- ❑ When applicable and provided to Db2 by sort program

- ❑ In  $\mu$ s

# Example: **SYSUTILITIES** Row

EVENTID	NAME				INSERTEDBY
22974	COPY				DB2

JOBNAME	UTILID	USERID
TS5817I	IMAGE	TS5817

STARTTS	ENDTS	ELAPSEDTIME	CPUTIME
2024-04-29-09.11.42.018390	2024-04-29-09.11.43.836198	1817808	166244

ZIIPTIME	RETURNCODE	CONDITION	RESTART	NUMOBJECTS	LISTNAME
0	0	E	N	17	DSN8LDEF

STARTLOGPOINT	GROUP_MEMBER	SORTNAME	SORTCPUTIME
00DF04C48EEA438C7800	I9A2	?	?

# Utility Object-Level History **SYSOBJEVENTS** Rows

**SYSOBJEVENTS** rows are inserted during initialization for target objects (partitions or space):

- Utility-in-progress state - UTUT, UTRW or UTRO
- Read or write access for main processing
- Set in a restrictive state during processing, i.e., RECP, RBDP, CHKP, etc.

**DSNU3033I SYSIBM.SYSOBJEVENTS ROWS INSERTED FOR OBJECT-LEVEL HISTORY**

## + Includes:

- Related LOBs, XML, and their indexes – REORG TABLESPACE AUX(YES), LOAD
- Dependent table (spaces), parent table (spaces), and their indexes – CHECK DATA, LOAD
- Exception tables and their indexes – CHECK DATA
- Mapping index – REORG TABLESPACE SHRLEVEL CHANGE

## - Excludes:

- Objects with errors or undefined DEFINE NO objects (except for utilities that will define them)
- Catalog and directory objects when not a target of the utility
  - DSNDB06.SYSTSCPY, DSNDB01.SYSLGRNX and others, e.g., MODIFY RECOVERY on a user table space

# Utility Object-Level History **SYSOBJEVENTS** Rows

---

Special cases for **SYSOBJEVENTS** rows:

PBG partition grown during utility execution

- ❑ LOAD or REORG inserts a **SYSOBJEVENTS** row for the partition

Online REORG table space type conversion

- ❑ Initialization (UTILINIT phase)
- ❑ **SYSOBJEVENTS** rows inserted for original table space type
- ❑ Termination (UTILTERM phase)
- ❑ **SYSOBJEVENTS** rows for original table space type are deleted
- ❑ New rows inserted for new table space type

# Understanding SYSOBJEVENTS Columns

---

## ❑ EVENTID column

- ❑ Utility event and execution identifier

## ❑ DBID, PSID, DBNAME, SPACENAME, OBJTYPE, INSTANCE columns

- ❑ Identifies object

## ❑ PARTITION column

- ❑ Physical partition number for partitioned objects, always non-zero

- ❑ Zero for non-partitioned objects



- ❑ Consistent with how real-time statistics (RTS) information is kept

## ❑ EVENTTS column

- ❑ Current local timestamp when processing of object begins

## ❑ COUNT column

- ❑ Number of rows, records, keys, pages, or LOBs

- ❑ Updated when processing of object completes, end of phase, or end of utility

## ❑ ELAPSEDTIME column (in $\mu$ s)

- ❑ COPY and RECOVER elapsed time for creation or restoration of sequential image copy



- ❑ Can be used for recovery time estimation of image copy restoration

- ❑ NULL for all other utilities

## ❑ INSERTEDBY column

- ❑ 'DB2' for IBM Db2 utilities

# Example: SYSOBJEVENTS Row

	EVENTID	DBID	PSID	PARTITION	INSTANCE	DBNAME
1_	22974	374	2	1	1	DSN8D13A
2_	22974	374	4	1	1	DSN8D13A
3_	22974	374	4	2	1	DSN8D13A
4_	22974	374	4	3	1	DSN8D13A
5_	22974	374	4	4	1	DSN8D13A
6_	22974	374	4	5	1	DSN8D13A
7_	22974	374	6	1	1	DSN8D13A
8_	22974	374	8	1	1	DSN8D13A
9_	22974	374	10	1	1	DSN8D13A
10_	22974	374	12	1	1	DSN8D13A
11_	22974	374	14	1	1	DSN8D13A
12_	22974	374	18	1	1	DSN8D13A

	SPACENAME	OBJTYPE	EVENTTS	COUNT	ELAPSEDTIME
1_	DSN8S13D	T	2024-04-29-09.11.42.749277	4	12097
2_	DSN8S13E	T	2024-04-29-09.11.42.835064	36	5700
3_	DSN8S13E	T	2024-04-29-09.11.42.848378	3	11267
4_	DSN8S13E	T	2024-04-29-09.11.42.863097	4	7027
5_	DSN8S13E	T	2024-04-29-09.11.42.873498	3	4607
6_	DSN8S13E	T	2024-04-29-09.11.42.895843	3	5762
7_	DSN8S13F	T	2024-04-29-09.11.42.966144	3	5190
8_	DSN8S13G	T	2024-04-29-09.11.43.064486	5	11343
9_	DSN8S13H	T	2024-04-29-09.11.43.186483	3	5341
10_	DSN8S13I	T	2024-04-29-09.11.43.260540	3	3766
11_	DSN8S13J	T	2024-04-29-09.11.43.329069	3	3771
12_	DSN8S13M	T	2024-04-29-09.11.43.397785	4	5119

# Example: **SYSUTILITIES** and **SYSOBJEVENTS** Row from IBM High Performance Unload

```
DB2 Admin          DD1A Interpretation of an Event in SYSUTILITIES          04:20
Command ==>

Details for event ID . . : 1137265

Utility . . . . . : UNLOAD
Inserted by . . . . . : HPU
Job name . . . . . : TS5941T
Utility ID . . . . . : DB2UNLOAD
Userid . . . . . : TS5941
Start timestamp . . . . . : 2024-06-11-04.17.33.491084
End timestamp . . . . . : 2024-06-11-04.18.09.303065
CPU time (usecs) . . . . . : 131758 ( .131)
Elapsed time (usecs) . . . . . : 35811981 ( 35.811)
ZIIP time (usecs) . . . . . : ? ( ?)
Return code . . . . . : 4
Status . . . . . : E - Ended
Restart . . . . . : N
Number of objects . . . . . : 1
LISTDEF name . . . . . : ?
Start logpoint (hex) . . . . . : 00000000A7E0E9E6B64
Group member name . . . . . : ?
Sort name . . . . . : ?
Sort CPU time (usecs) . . . . . : ? ( ?)
Sort ZIIP time (usecs) . . . . . : ? ( ?)
```

```
DB2 Admin          DD1A Interpretation of an Event in SYSOBJEVENTS          04:21
Command ==>

Details for event ID . . : 1137265

Utility . . . . . : UNLOAD
Object type . . . . . : T - Tablespace
Database name . . . . . : JTHDCH DB ID . . . . . : 4046
Space name . . . . . : JTHSCHK PS ID . . . . . : 13
Partition . . . . . : 1
Instance . . . . . : 1
Count . . . . . : 0
Elapsed time (usecs) . . . . . : 20758 ( .020)
Event timestamp . . . . . : 2024-06-11-04.18.09.262095
```

# Examples: **SYSUTILITIES** Rows from IBM Db2 Recovery Expert for z/OS

```

DB2 Admin DD1A Interpretation of an Event in SYSUTILITIES 04:23
Command ==>

Details for event ID . . : 1134181
Utility . . . . . : DSN1COPY
Inserted by . . . . . : ARY
Job name DB2 Admin DD1A Interpretation of an Event in SYSUTILITIES 04:24
Utility Command ==>
Userid
Start time
End time
CPU time
Utility
Elapsed time
ZIIP time
Return code
Status
Restart
Number of
LISTDEF
Start log
Group member
Sort name
Sort CPU
Sort ZIIP

Details for event ID . . : 1134347
Utility
Elapsed time
ZIIP time
Return code
Status
Restart
Number of
LISTDEF
Start log
Group member
Sort name
Sort CPU
Sort ZIIP

DB2 Admin DD1A Interpretation of an Event in SYSUTILITIES 04:26
Command ==>

Details for event ID . . : 1127461
Utility . . . . . : Fast Apply
Inserted by . . . . . : ARY
Job name . . . . . : STD2140P
Utility ID . . . . . : TS6501.STD2140P
Userid . . . . . : TS6501
Start timestamp . . . . . : 2024-06-04-02.24.55.510000
End timestamp . . . . . : 2024-06-04-02.24.55.720000
CPU time (usecs) . . . . . : 20000 ( .020)
Elapsed time (usecs) . . . . . : 210000 ( .210)
ZIIP time (usecs) . . . . . : 0 ( .000)
Return code . . . . . : 0
Status . . . . . : E - Ended
Restart . . . . . : N
Number of objects . . . . . : 14
LISTDEF name . . . . . : ?
Start logpoint (hex) . . . . . : 00000000000000000000
Group member name . . . . . : ?
Sort name . . . . . : ?
Sort CPU time (usecs) . . . . . : ? ( ?)
Sort ZIIP time (usecs) . . . . . : ? ( ?)

```

**SYSOBJEVENTS**  
also available

# Example: **SYSUTILITIES** and **SYSOBJEVENTS** Row from IBM Db2 Cloning Tool

```
DB2 Admin          DD1A Interpretation of an Event in SYSUTILITIES          04:25
Command ==>

Details for event ID . . : 1018510

Utility . . . . . : IBM Db2 Cloning Tool 3.2 - Source job
Inserted by . . . . . : CKZ32
Job name . . . . . : RR000SRC
Utility ID . . . . . : J0487990
Userid . . . . . : TS6396
Start timestamp . . . . . : 2024-04-15-07.15.24.862228
End timestamp . . . . . : 2024-04-15-07.15.27.338568
CPU time (usecs) . . . . . : 156691          (          .156)
Elapsed time (usecs) . . . . . : 2476340      (          2.476)
ZIIP time (usecs) . . . . . : ?              (          ?)
Return code . . . . . : 0
Status . . . . . : E - Ended
Restart . . . . . : N
Number of objects . . . . . : 0
LISTDEF name . . . . . : RR
Start logpoint (hex) . . . . . : ?
Group member name . . . . . : DD1A
Sort name . . . . . : ?
Sort CPU time (usecs) . . . . . : ?          (          ?)
Sort ZIIP time (usecs) . . . . . : ?          (          ?)
```

# Agenda

- Overview
- Understanding History Information
- **Using Utility History**
- Closing

# Using Utility History

---

## Execute your utilities workload

- Activate utility history collection
- History information automatically collected and accumulates over time
- Analyze utility history information
  - Rerun failed utilities
  - Calibrate utility jobs and execution

## Run utilities on utility history catalog objects as needed

- COPY, REORG, RUNSTATS, etc.
- Don't include other objects because utility history will not be collected

## Regularly assess size of **SYSUTILITIES** and **SYSOBJEVENTS** tables

- Delete information no longer needed
- Delete oldest by timestamp
- Optionally, keep information for your most important utilities
- SYSOBJEVENTS** rows are not too useful without corresponding **SYSUTILITIES** rows

# Utility Object-Level History

---

- ❑ Function may be turned on and off multiple times
  
- ❑ Information is not collected
  - ❑ For utilities executed on Db2 utility history catalog objects and other special cases
  - ❑ Resource-unavailable condition during the update of the utility history catalog objects
  - 💡 ❑ Does not affect successful completion of the utility !
  
- ❑ Information can be inserted or deleted by users or tools
  
- ❑ No automated cleanup process by Db2 utilities
  - ❑ Db2 for z/OS Tools can help with this

# Considerations for Queries

---

## SQL SELECT

- 💡  Filter, order and join utility history information in **SYSUTILITIES**, **SYSOBJEVENTS** and **SYSCOPY** correlating on **EVENTID** column to check, analyze, compare and manage utility executions
- 💡  Use ISOLATION(UR) to avoid contention
  - 'WITH UR'
  - Create user defined indexes over **SYSUTILITIES**, **SYSOBJEVENTS** and **SYSCOPY** if needed for better performance

## Negative SQLCODEs returned for SELECT on utility history catalog tables

- Function/catalog level is too low
- Utility history table spaces may be in a restricted Db2 state

# Sample Queries

---

Utilities with errors in the last 24 hours:

```
SELECT EVENTID, NAME, JOBNAME, UTILID, STARTTS, RETURNCODE, CONDITION
FROM SYSIBM.SYSUTILITIES
WHERE STARTTS >= CURRENT TIMESTAMP - 1 DAYS
AND RETURNCODE >= 8
WITH UR;
```

All active or stopped utilities:

```
SELECT EVENTID, NAME, JOBNAME, UTILID, STARTTS, RESTART, RETURNCODE, CONDITION
FROM SYSIBM.SYSUTILITIES
WHERE RETURNCODE IS NULL
AND CONDITION=' '
WITH UR;
```



Issue `-DIS UTIL(utilid)` command to check if stopped or active.

# Sample Queries

---

How long did a utility/job take to execute ?

```
SELECT EVENTID, NAME, JOBNAME, UTILID, STARTTS, ELAPSEDTIME, RETURNCODE, CONDITION
FROM SYSIBM.SYSUTILITIES
WHERE NAME      = 'utility'
      AND JOBNAME = 'jobname'
      AND STARTTS >= CURRENT TIMESTAMP - 7 DAYS
ORDER BY ELAPSEDTIME DESC
WITH UR;
```

Trend utility timings – detect deviations from ‘the norm’:

```
SELECT NAME, JOBNAME, STARTTS, ELAPSEDTIME, CPUTIME, NUMOBJECTS, LISTNAME
FROM SYSIBM.SYSUTILITIES
WHERE NAME = 'utility' AND JOBNAME = 'jobname'
      AND STARTTS >= STARTTS - 6 MONTHS
ORDER BY STARTTS
WITH UR;
```

# Sample Queries (cont'd)

Total count of executions by utility name in the past week:

```
SELECT NAME AS UTILITY_TYPE, COUNT(*) AS UTILITY_COUNT
  FROM SYSIBM.SYSUTILITIES
 WHERE STARTTS >= CURRENT TIMESTAMP - 7 DAYS
  GROUP BY NAME
  WITH UR;
```



UTILITY_TYPE	UTILITY_COUNT
COPY	5
LOAD	32
MODIFY RECOVERY	128
REORG	40
REPORT RECOVERY	13
RUNSTATS	2

# Sample Queries (cont'd)

Most recent utility executed on partition 3 of table space DSN8D13A.DSN8S13E:

```
SELECT O.EVENTID, O.DBNAME, O.SPACENAME, O.PARTITION,  
       U.NAME, U.ENDTS, U.RETURNCODE, U.CONDITION  
FROM SYSIBM.SYSOBJEVENTS O INNER JOIN SYSIBM.SYSUTILITIES U  
     ON O.EVENTID=U.EVENTID  
WHERE DBNAME='DSN8D13A'  
     AND SPACENAME='DSN8S13E'  
     AND PARTITION=3  
ORDER BY EVENTTS DESC  
FETCH FIRST 1 ROWS ONLY  
WITH UR;
```

EVENTID	DBNAME	SPACENAME	PARTITION
22976	DSN8D13A	DSN8S13E	3

NAME	ENDTS	RETURNCODE	CONDITION
REORG	2024-04-29-10.01.17.412180	0	E



# Utility History Use Cases

- ❑ Tooling should assist in the analysis
  - ❑ Here we see the options for querying the catalog from my favorite Admin Tool

```
Option ==>
Enter option and optional criteria:
1 - Display utility history rows
    Utility . . . . . (RUNSTATS, REORG, ? to lookup)
    Event Id . . . . . (DB2 or other product name)
    Inserted by . . . . .
    Userid . . . . .
    Jobname . . . . .
    Utility ID . . . . .
    Return code >= . . . . .
    Status . . . . . (A-Active, E-Ended, T-Term., F-Forced)
    Start time within . . . . . (e.g. n DAYS or n MONTHS)
    Age >= . . . . . (e.g. n DAYS or n MONTHS)
    CPU time >= . . . . . (in msec)
    Elapsed time >= . . . . . (in msec)
2 - Delete Utility history rows
    Status . . . . . (E-Ended, T-Terminated, F-Forced)
    Age >= . . . . . (e.g. n DAYS or n MONTHS)
```



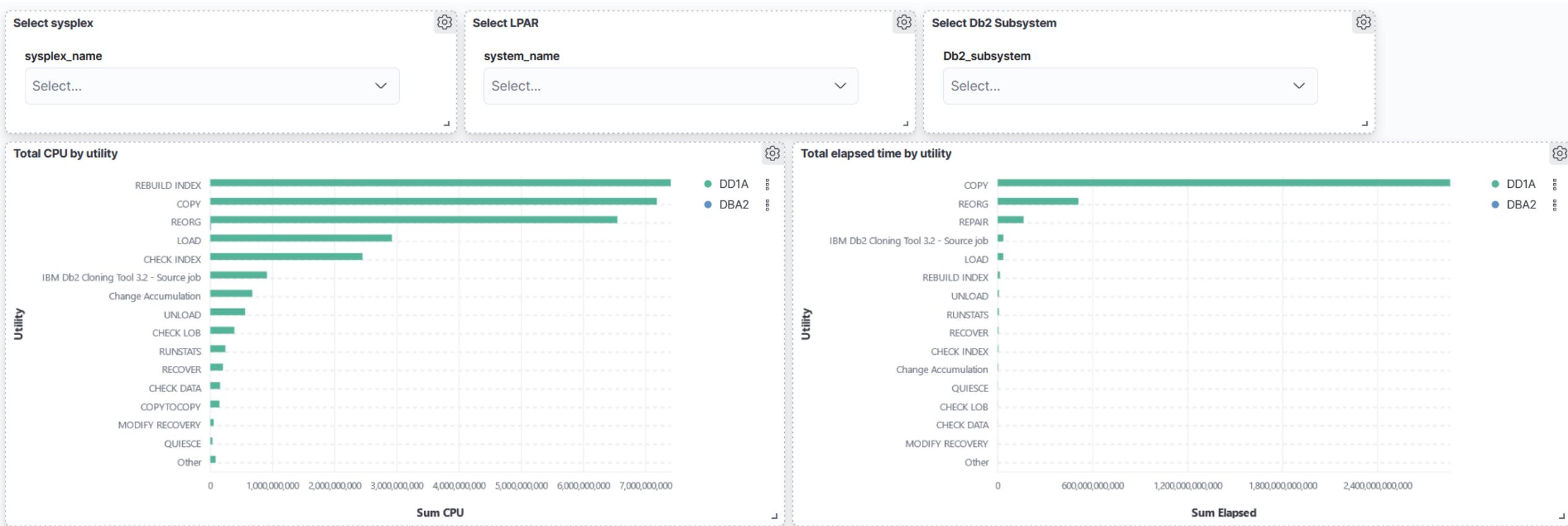
# Utility History Use Cases

- ❑ Tooling can provide more complex query capabilities
  - ❑ Highlighting the most resource intensive utilities
  - ❑ Showing deviation from normal execution times etc

```
Option ==> 

CPU time >= . . . . . (in msec)           More: -
Elapsed time >= : : . (in msec)
2 - Delete Utility history rows
   Status . . . . . (E-Ended, T-Terminated, F-Forced)
   Age >= . . . . . (e.g. n DAYS or n MONTHS)
Predefined queries:
3 - Show top 25 completed utilities with longest CPU time
4 - Show top 25 completed utilities with longest elapsed time
5 - Show all utilities group by utility
6 - Show top 25 active utilities with longest elapsed time
7 - Show top 25 utilities with return code >= 8
   Start time within . (e.g. n DAYS or n MONTHS)
Create own query:
8 - Use the SELECT prototype on SYSUTILITIES
Display exceptions:
9 - Show utilities that differs from the average
   Start time within . 7 (e.g. n DAYS or n MONTHS)
   CPU time >= . . . . 100 (in msec, default 1000)
   Std deviations . . 3 (1-9, default 3)
```

# Utility History Use Cases – Sample Dashboard



# Utility History Use Cases – Sample Dashboard

Most CPU consuming utility 1 ⚙️

Event id	Utility name	Utility id	CPU time
963,981	LOAD	LOAD1	17,346,337
915,299	COPY	TS5745.CIMSC2P	15,253,659
915,300	COPY	TS5745.CIMSC2R	15,188,828
915,298	COPY	TS5745.CIMSC2M	15,151,101
915,301	COPY	TS5745.CIMSC2Q	13,764,159
921,683	COPY	TS5745.CIMSC2M	13,436,233
921,684	COPY	TS5745.CIMSC2P	12,673,413
921,689	COPY	TS5745.CIMSC2T	12,667,908
921,685	COPY	TS5745.CIMSC2Q	12,392,854
878,114	Change Accumulation	TS3556.RDR0620P	12,290,000

Event id	Db2 subsystem	Utility name	Start	End	Utility id	User id	CPU time	Elapsed time	zIIP time	Sort CPU time	Sort zIIP time
963,981	DD1A	LOAD	2024-03-07-05...	2024-03-07-05...	LOAD1	TS6025	17,346,337	137,697,322	7,836,591	359,188	0

# Utility History Use Cases

---

- Parameter Efficiency
  - What difference did changing parameters make to the execution ?
  - What types of objects benefitted most ?
  
- Problem determination - estimate time to completion
  - Example – Called out overnight due to long running utility
    - Do you cancel ?
    - Look at previous execution times and trends to make better decision
  
- Find unsuccessful utility executions
  - Trend failure patterns
  - Do utilities commonly fail at the same time ?
  
- Partition //ism
  - Are partitions handled in the most efficient way ?
  
- Change Management history

# Usage Notes

---

- ❑ Rows not deleted when objects dropped
  - ❑ User responsible for deletion – keep track of sizes
- ❑ Resource unavailable stops collection but does not affect utility execution
- ❑ Near '*real-time*' updates
- ❑ Utility Restart
  - ❑ Inserts, or updates SYSOBJEVENTS as needed
  - ❑ Note: COUNT represents processing (original & restart)
  - ❑ ELAPSED TIME for COPY/RECOVER represents restart only
- ❑ Utility failures
  - ❑ Utility Abends do not update SYSOBJEVENTS
  - ❑ TERM UTIL does not update SYSOBJEVENTS
  - ❑ START ACCESS(FORCE) does not update SYSOBJEVENTS
- ❑ BACKUP/RECOVERY of SYSOBJEVENTS/SYSUTILITIES
  - ❑ Can be backed up and recovered together
  - ❑ Cannot be combined with other tablespace(s)
  - ❑ Restriction to avoid contention if history collection is active

# Agenda

- Overview
- Understanding History Information
- Using Utility History
- Closing

# Future Possibilities

---

What other essential utility information do you need?

AHA! Ideas and requests for:

- SHRLEVEL
- Utility control statement
- Utility job output
- ZSORT
- DSNTIAUL support
- Ability to activate or request object-level history in job step

# Publication Links

---



IBM Community Blog Db2 13 for z/OS: Capturing utility and object-level history

➤ <https://community.ibm.com/community/user/datamanagement/blogs/kate-wheat1/2023/11/21/objecthistory>

Db2 13 Monitoring utility history

➤ <https://www.ibm.com/docs/en/db2-for-zos/13?topic=utilities-monitoring-utility-history>

Db2 13 SYSIBM.SYSUTILITIES catalog table

➤ <https://www.ibm.com/docs/en/db2-for-zos/13?topic=tables-sysutilities>

Db2 13 SYSIBM.SYSOBJEVENTS catalog table

➤ <https://www.ibm.com/docs/en/db2-for-zos/13?topic=tables-sysobjevents>

Db2 13 Overview of what's new in Db2

➤ [https://www.ibm.com/docs/en/db2-for-zos/13?topic=13-overview-whats-new-in-db2#db2z\\_13\\_wnewoverview\\_\\_sect-utilities](https://www.ibm.com/docs/en/db2-for-zos/13?topic=13-overview-whats-new-in-db2#db2z_13_wnewoverview__sect-utilities)

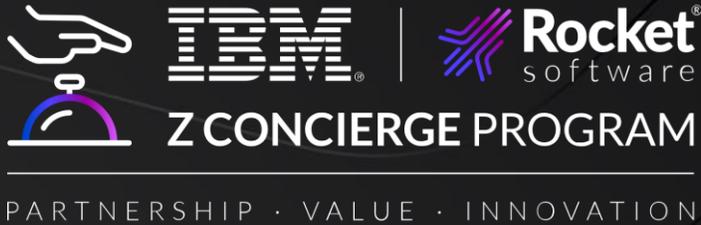
Db2 13 Function level 504 (APAR PH54919 –October 2023)

➤ <https://www.ibm.com/docs/en/db2-for-zos/13?topic=levels-function-level-504-apar-ph54919-october-2023>

IBM Redbook Db2 13 for z/OS and More

➤ <https://www.redbooks.ibm.com/abstracts/sg248527.html>

For more information, email us:  
zconciierge@rocketsoftware.com



# Z Conciierge Program

*Deploy and learn faster, and achieve ROI sooner, with our experts*

No-charge IBM Db2 Tools deployment assistance offers, take advantage today!

Access to our IBM Db2 Tools experts

Help you get started with your Db2 Tools deployment

Establishment of best practices

Dedicated point of contact for support /questions

# Thank you.

rocketsoftware.com

zconciierge@rocketsoftware.com

