

LUWs hidden treasures - administrative views and routines

Koen De Backer
ABIS Training & Consulting

Session Code: D7

Tuesday, 15 October 2013, 15:15 - 16:15 | Platform: LUW



Agenda

Introducing administrative objects

General purpose administrative objects

Administrative objects and monitoring

Administrative objects for administrative tasks

Usage scenarios - easier monitoring, swifter problem detection?



LUW administrative objects (1)

- DB2 LUW Administrative 'objects' provide an easy-to-use [application programming] interface to DB2 administrative tasks and [performance] data, using familiar SQL statements
- Administrative objects include:
 - table/scalar functions
 - views - probably created on some of the table functions mentioned above
 - routines and procedures



LUW administrative objects (2)

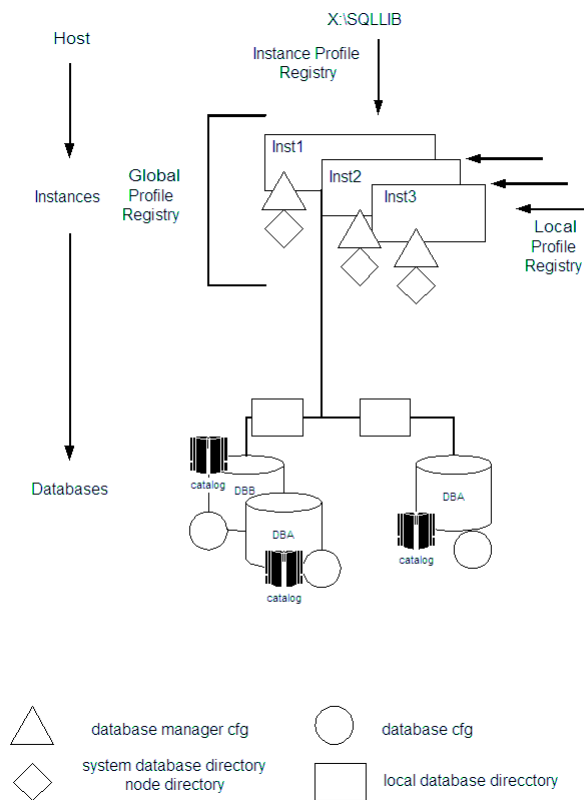
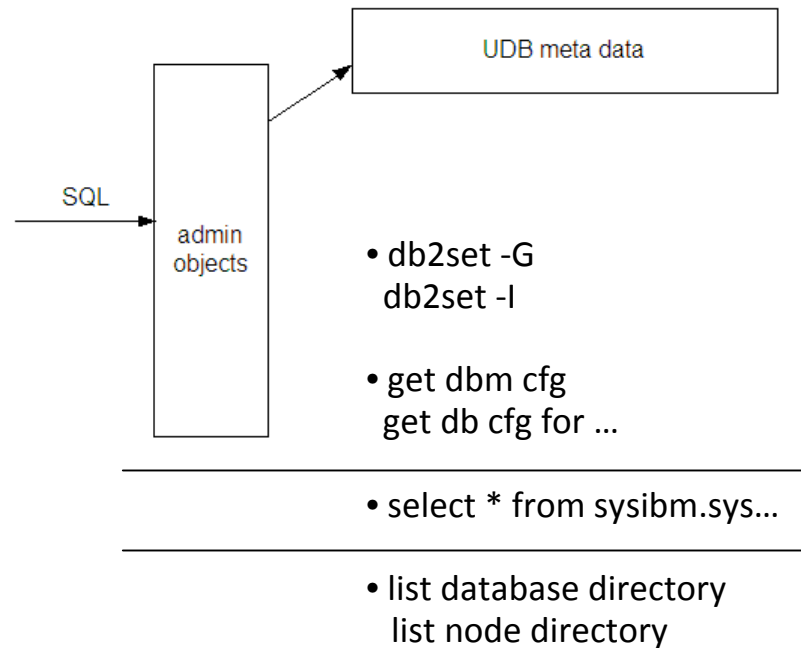
The aim?

- Provide a simple, well-known, easy to use interface to administrative and monitoring data by means of SQL + compute efficiency ratios
- Get rid of context specific, tool specific APIs for access to this data
- Provide additional (specialised?) functionality
[perhaps not (yet) externalised through GUI interface]
- Make our lives easier - and that of the tool vendors!

Oracle: v\$_*, dbms_*

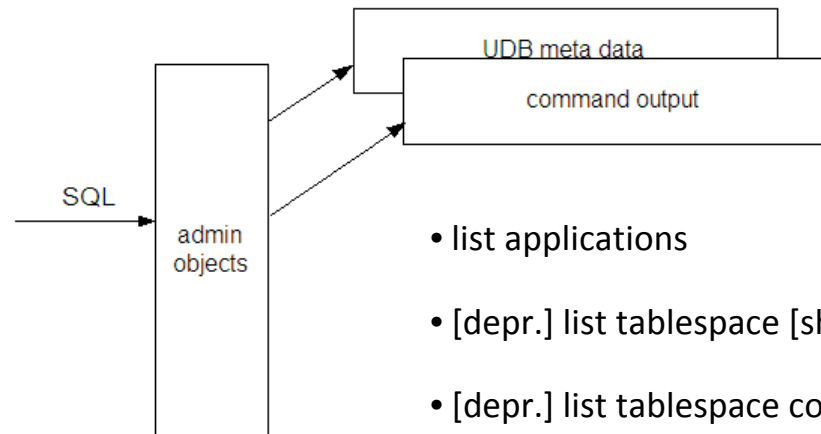


LUW administrative objects - sources (1)





LUW administrative objects - sources (2)

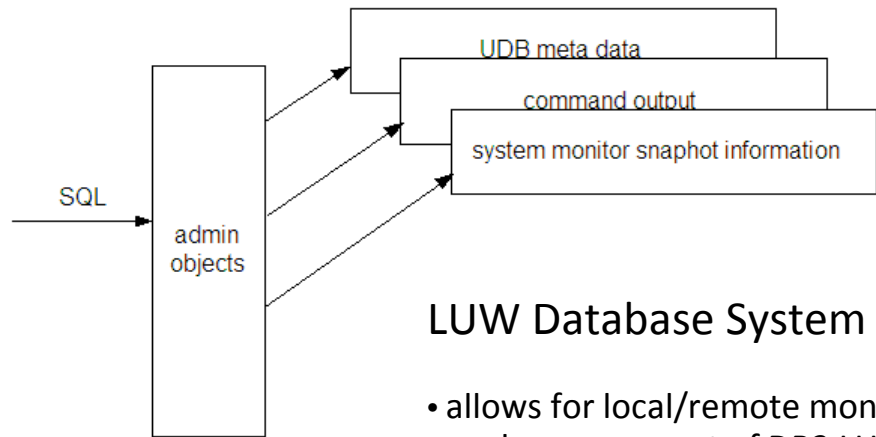


- list applications
- [depr.] list tablespace [show detail]
- [depr.] list tablespace containers
- list utilities
- ...

confusion!
some deprecated, some not!



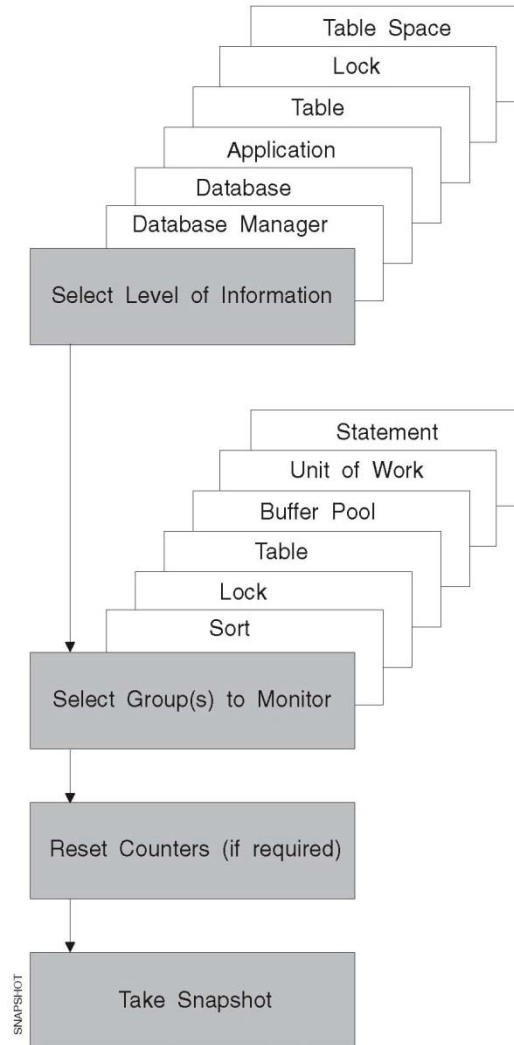
LUW administrative objects - sources (3)



deprecated!

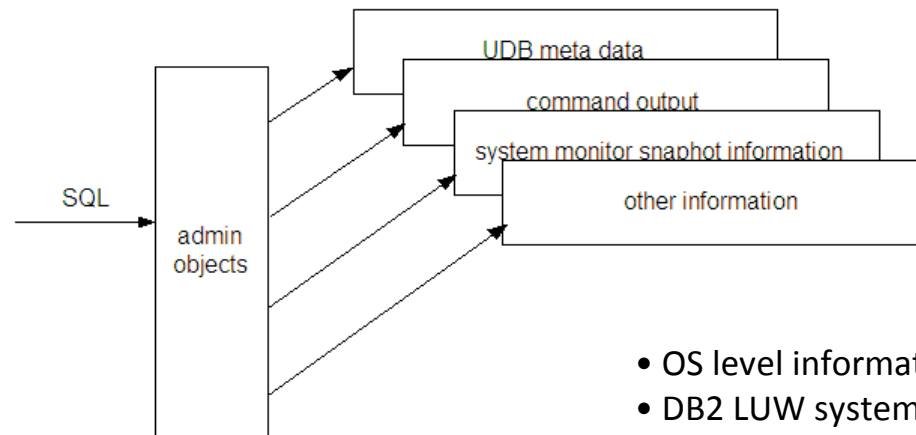
LUW Database System Monitor

- allows for local/remote monitoring/tuning and management of DB2 LUW databases and applications
- get monitor switches
- update monitor switches using <name> on | off
- get snapshot for dbm | db | all | ..





LUW administrative objects - sources (4)



- OS level information
- DB2 LUW system/environment/licencing information
- ...



LUW administrative objects - schemas

- **sysproc** schema
 - contains all procedures, scalar/table functions required for accessing relevant data using SQL like constructs [and others]
- **sysibmadm** schema
 - contains all views required for accessing relevant data using SQL like constructs
 - views often build on the above mentioned sysproc table functions

```
SELECT count(*)  
from sysibm.sysroutines  
where routineschema = 'SYSPROC'  
  
1  
-----  
445
```

```
SELECT count(*)  
from sysibm.systables  
where creator = 'SYSIBMADM'  
  
1  
-----  
79
```



LUW administrative objects - sample structure

```
get dbm cfg
```

```
select * from  
TABLE(SYSPROC.DBM_GET_CFG()) AS CONFIG
```

```
select *  
from sysibmadm.dbmcfg
```

```
select * from sysibm.sysfunctions  
where name = 'DBM_GET_CFG'  
select * from sysibm.sysroutines  
where routinename = 'DBM_GET_CFG'
```

```
select text from sysibm.sysviews  
where name = 'DBMCFG'
```



LUW administrative objects - remarks (1)

Considerations!

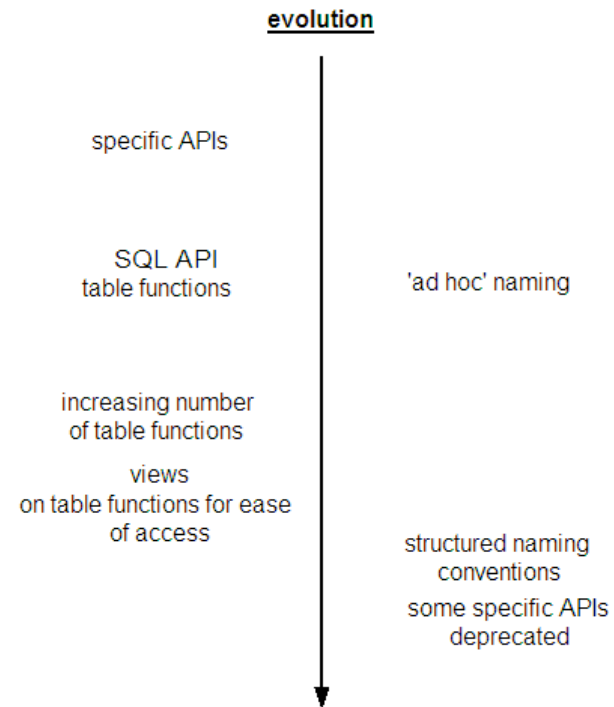
- not all table functions have a corresponding admin view defined
use of table functions might still be required in specific situations
- not all admin views are [exclusively] based on table functions:
 - some interpret and/or extend other admin views
 - some are simply based on eg. sysibm.sys... catalog tables
 - some combine both catalog table info and info obtained from table function sources
 - some include calculations
- introduction of these object has signaled 'the end' of some well-known, frequently used tools/commands!

snapshots
list tablespaces
memtrack



LUW administrative objects - remarks (2)

- work in progress...
- structuring, restructuring ...





LUW administrative objects - remarks (3)

- interpretation of results - what is what?
[meaning, significance]

```
select datatype, count(*) as 'Nnbr'
from sysibmadm.dbmcfg group by datatype
```

<u>DATATYPE</u>	<u>Nnbr</u>
BIGINT	17
INTEGER	23
REAL	2
SMALLINT	1
VARCHAR(1023)	5
VARCHAR(128)	5
VARCHAR(14)	2

```
select name
from sysibmadm.dbmcfg
where datatype = 'INTEGER'
```

```
agent_stack_sz, agentpri,
conn_elapse, diaglevel,
mon_heap_sz, notifylevel, numdb, ...
```

- what data gets generated/registered? And when?



Required privileges

- objects in sysproc schema
 - routines require *execute*
[SYSCAT.ROUTINEAUTH]
- objects in sysibmadm schema
 - views require *select*
[SYSCAT.TABAUT]



Administrative views/table functions - types

environment views/table functions - *ENV_**

- eg. (v) **ENV_CF_SYS_RESOURCES**, ENV_FEATURE_INFO, **ENV_INST_INFO**, **ENV_PROD_INFO**, ENV_SYS_INFO, **ENV_SYS_RESOURCES**
- eg. (p) GET_DB2_SYSTEM_RESOURCES, GET_DB2_EDU_SYSTEM_RESOURCES, ENV_GET_REG_VARIABLES
- describe OS system characteristics and available resources, DB2 software characteristics, installed options, and licencing information, etc.

Deprecated: get_db_config(p), get_dbm_config(f)

Bold: example provided - v=view; p=procedure; f=function



Admin views/table functions - types (ex)

```
select substr(name, 1, 20), substr(value, 1, 20),
substr(unit,1, 8) from sysibmadm.env_sys_resources
```

1	2	3

OS_NAME	WIN32_NT	-
HOST_NAME	WS34EB24J	-
OS_VERSION	5.1	-
OS_RELEASE	Service Pack 3	-
MACHINE_IDENTIFICATI	x86 Family 6, model	-
CPU_TOTAL	1	-
CPU_ONLINE	1	-
CPU_CONFIGURED	1	-
CPU_SPEED	2394	MHz
CPU_HMT_DEGREE	1	-
CPU_CORES_PER_SOCKET	1	-
MEMORY_TOTAL	1024	MB
MEMORY_FREE	379	MB
MEMORY_SWAP_TOTAL	414	MB
MEMORY_SWAP_FREE	406	MB
VIRTUAL_MEM_TOTAL	1438	MB
VIRTUAL_MEM_FREE	785	MB
CPU_USAGE_TOTAL	1	PERCENT

18 record(s) selected.

```
db2 describe table
sysibmadm.env_prod_info
```

Column name

```
-----
INSTALLED_PROD
INSTALLED_PROD_FULLNAME
LICENSE_INSTALLED
PROD_RELEASE
LICENSE_TYPE
```

```
db2 describe table
sysibmadm.env_inst_info
```

Column name

```
-----
INST_NAME
IS_INST_PARTITIONABLE
NUM_DBPARTITIONS
INST_PTR_SIZE
RELEASE_NUM
SERVICE_LEVEL
BLD_LEVEL
PTF
FIXPACK_NUM
NUM_MEMBERS
```




Admin views/table functions - types

configuration & setup views/table functions

- DBMCFG (v), DBCFG (v), REG_VARIABLES (v)
- list:
 -) **database**manager and **database** configuration information
 -) **registry** profile settings

No description/help/unit indication
provided - a pity!

```
get dbm cfg
```

```
get db cfg for ...
```

```
db2set -I xxx -all
```

v=view; p=procedure; f=function



Admin views/table functions - types

table/index related views/table functions

- **ADMINTABINFO** (v) and ADMIN_GET_TABINFO (f) -- temp table version exists as well!
- ADMIN_EST_INLINE_LENGTH (f)
- **ADMIN_GET_TAB_COMPRESS_INFO** (f) - changed interpretation
- **ADMIN_GET_TAB_DICTIONARY_INFO** (f) - changed interpretation

- ADMIN_GET_INDEX_COMPRESS_INFO (f)
- ADMIN_GET_INDEX_INFO (f)

Deprecated: admintabcompressinfo (v)

Bold: example provided - v=view; p=procedure; f=function



Admin views/table functions - types (ex1)

```
db2 describe table sysibmadm.admintabinfo
```

```
Column name [edited]
```

```
-----  
TABSCHEMA, TABNAME, TABTYPE
```

```
...
```

```
DATA_PARTITION_ID
```

```
AVAILABLE
```

```
DATA_OBJECT_L_SIZE, DATA_OBJECT_P_SIZE
```

```
INDEX_OBJECT_L_SIZE, INDEX_OBJECT_P_SIZE
```

```
LONG_OBJECT_L_SIZE, LONG_OBJECT_P_SIZE
```

```
LOB_OBJECT_L_SIZE, LOB_OBJECT_P_SIZE
```

```
XML_OBJECT_L_SIZE, XML_OBJECT_P_SIZE
```

```
INDEX_TYPE
```

```
REORG_PENDING, INPLACE_REORG_STATUS, LOAD_STATUS
```

```
READ_ACCESS_ONLY
```

```
NO_LOAD_RESTART
```

```
NUM_REORG_REC_ALTERS
```

```
INDEXES_REQUIRE_REBUILD
```

```
...
```

```
DICTIONARY_SIZE
```

```
BLOCKS_PENDING_CLEANUP
```

```
...
```

```
RECLAIMABLE_SPACE
```

```
XML_DICTIONARY_SIZE
```

```
AMT_STATUS
```

```
...
```

Alters causing reorgs

Log/Phy sizes base component, KB

Log/Phy sizes other components, KB

External action status information

Regular table: runstats free space



Admin views/table functions - types (ex2)

```
create table Mytable -- compress yes adaptive
insert data into Mytable
reorg table mytables resetdictionary
```

```
SELECT ROWCOMPMODE, BUILDER, BUILD_TIMESTAMP, SIZE, HISTORICAL_DICTIONARY, ROWS_SAMPLED,
       PCTPAGESSAVED, AVGCOMPRESSEDROWSIZE
FROM TABLE(SYSPROC.ADMIN_GET_TAB_DICTIONARY_INFO( 'DUSER', 'MYTABLES' ));
```

```
SELECT ROWCOMPMODE, PCTPAGESSAVED_CURRENT, AVGROWSIZE_CURRENT, PCTPAGESSAVED_STATIC,
       AVGROWSIZE_STATIC, PCTPAGESSAVED_ADAPTIVE, AVGROWSIZE_ADAPTIVE
FROM TABLE(SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO('DUSER', 'MYTABLES'));
```

ROWCOMPMODE	A
BUILDER	REORG
BUILD_TIMESTAMP	<TS>
SIZE	36096
HISTORICAL_DICTIONARY	N
ROWS_SAMPLED	5181
PCTPAGESSAVED	67
AVGCOMPRESSEDROWSIZE	15

ROWCOMPMODE	A
PCTPAGESSAVED_CURRENT	67
AVGROWSIZE_CURRENT	15
PCTPAGESSAVED_STATIC	67
AVGROWSIZE_STATIC	15
PCTPAGESSAVED_ADAPTIVE	67
AVGROWSIZE_ADAPTIVE	15

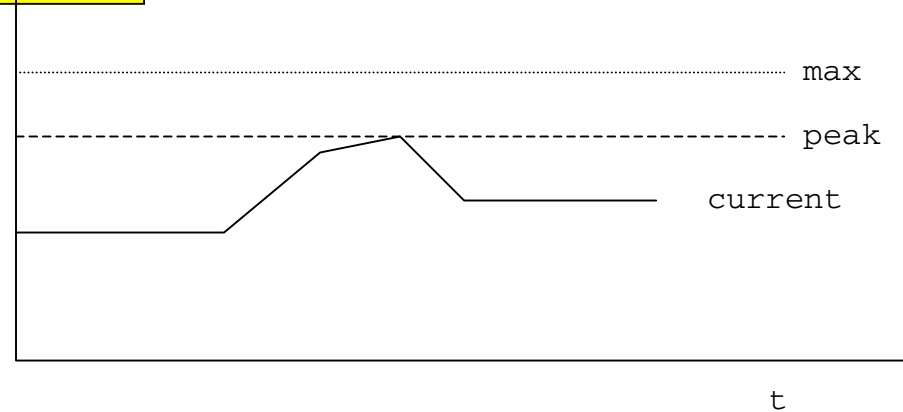


Admin views/table functions - types

memory related views/table functions

- ADMIN_GET_MEM_USAGE (f)

Column name	
MEMBER	
MAX_MEMBER_MEM	
CURRENT_MEMBER_MEM	Memory
PEAK_MEMBER_MEM	(bytes)





Admin views/table functions - types

storage, container, tablespace related views/table functions

- ADMIN_GET_STORAGE_PATHS (f)

```
Column name [edited]
-----
STORAGE_GROUP_NAME
STORAGE_GROUP_ID
...
DB_STORAGE_PATH - Automatic storage path
...
DB_STORAGE_PATH_STATE - Storage path state (IN_USE, NOT_IN_USE, PENDING)
DB_STORAGE_PATH_ID - Storage path identifier
FS_ID - Unique file system identification number
FS_TOTAL_SIZE - Total size of a file system
FS_USED_SIZE - Amount of space used on a file system
STO_PATH_FREE_SIZE - Automatic Storage path free space
```



Admin views/table functions - types

monitoring related views/table functions

- mon*, evmon*
- obtain 'runtime statistics' describing system state, health, performance
- views - a starting point of any analysis:

basic system highlights, including calculation of standard [efficiency] ratios
stable: strategically important

Deprecated: health*, most snap*



Admin views/table functions - types

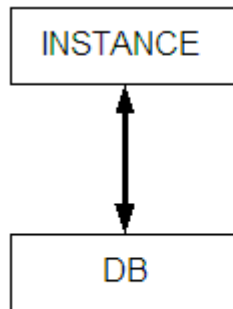
monitoring related views/table functions

- table functions - more detailed analysis (tuning and troubleshooting)
 - stable: strategically important
 - DO NOT use table functions with version number in name; always select columns, never use '*'
 - table functions WITHOUT and WITH '_DETAILS' suffix (=XML output)
 - metrics always reset at database (de)activation - no stored data!
 - pureScale aware



Admin views/table functions - types

monitoring related views/table functions



- MON_GET_INSTANCE (F)
- MON_GET_DATABASE (F)
- MON_GET_DATABASE_DETAILS (F)
- MON_GET_TRANSACTION_LOG (F)
- MON_GET_UTILITY (F)
- MON_TRANSACTION_LOG_UTILIZATION (V)
- MON_DB_SUMMARY (V)

```
db2 describe table
sysibmadm.mon_db_summary

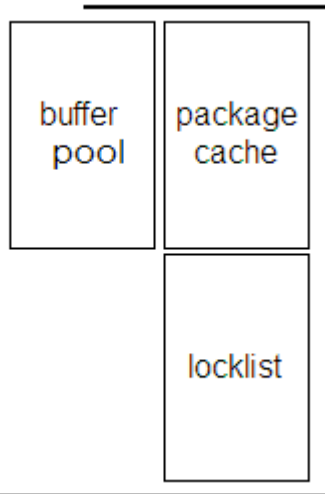
Column name [edited]
-----
TOTAL_APP_COMMITS
TOTAL_APP_ROLLBACKS
...
APP_RQSTS_COMPLETED_TOTAL
AVG_RQST_CPU_TIME
ROUTINE_TIME_RQST_PERCENT
RQST_WAIT_TIME_PERCENT
ACT_WAIT_TIME_PERCENT
IO_WAIT_TIME_PERCENT
LOCK_WAIT_TIME_PERCENT
AGENT_WAIT_TIME_PERCENT
NETWORK_WAIT_TIME_PERCENT
SECTION_PROC_TIME_PERCENT
SECTION_SORT_PROC_TIME_PERCENT
COMPILE_PROC_TIME_PERCENT
TRANSACTION_END_PROC_TIME_PERCENT
UTILS_PROC_TIME_PERCENT
...
AVG_LOCK_WAITS_PER_ACT
AVG_LOCK_TIMEOUTS_PER_ACT
AVG_DEADLOCKS_PER_ACT
...
ROWS_READ_PER_ROWS_RETURNED
TOTAL_BP_HIT_RATIO_PERCENT
...
```

Deprecated: log_utilization (v)



Admin views/table functions - types

monitoring related views/table functions



MON_GET_PKG_CACHE_STMT (F)
 MON_GET_APPL_LOCKWAIT (F)
 MON_GET_BUFFERPOOL (F)

MON_LOCKWAITS (V)
 MON_PKG_CACHE_SUMMARY (V)
 MON_BP_UTILISATION (V)

```
db2 describe table
mon_pkg_cache_summary
```

Column name [edited]

```
-----
SECTION_TYPE
EXECUTABLE_ID
NUM_COORD_EXEC
NUM_COORD_EXEC_WITH_METRICS
TOTAL_STMT_EXEC_TIME
AVG_STMT_EXEC_TIME
TOTAL_CPU_TIME
AVG_CPU_TIME
TOTAL_LOCK_WAIT_TIME
AVG_LOCK_WAIT_TIME
TOTAL_IO_WAIT_TIME
AVG_IO_WAIT_TIME
PREP_TIME
ROWS_READ_PER_ROWS_RETURNED
AVG_ACT_WAIT_TIME
...
stmt_text
```

```
db2 describe table
mon_bp_utilisation
```

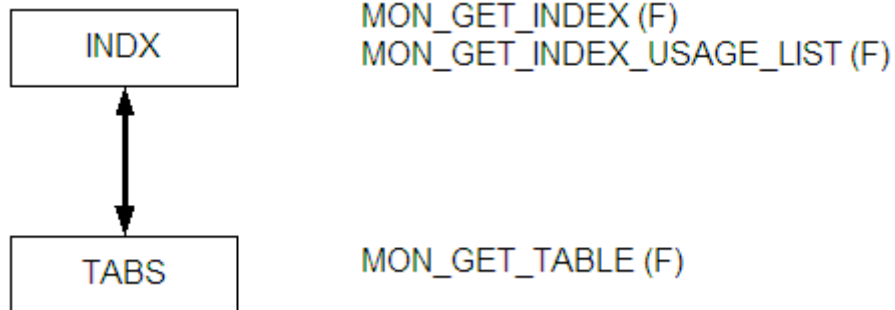
Column name [edited - many more cols]

```
-----
BP_NAME
DATA_PHYSICAL_READS      - DATA_HIT_RATIO_PERCENT
INDEX_PHYSICAL_READS     - INDEX_HIT_RATIO_PERCENT
```



Admin views/table functions - types

monitoring related views/table functions

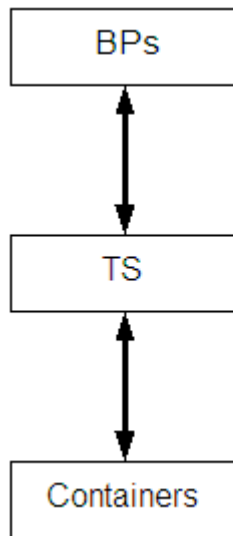


Deprecated: _v95 views, _v97 views



Admin views/table functions - types

monitoring related views/table functions



MON_BP_UTILIZATION (V)
 MON_GET_BUFFERPOOL (F)

MON_TBSP_UTILIZATION (V)
 MON_GET_TABLESPACE (F)

MON_GET_CONTAINER (F)



```
db2 describe table
mon_tbasp_utilization

Column name [edited]
-----
TBSP_NAME
...
TBSP_TYPE
TBSP_CONTENT_TYPE
TBSP_STATE
TBSP_PAGE_SIZE
TBSP_EXTENT_SIZE
TBSP_PREFETCH_SIZE
TBSP_USING_AUTO_STORAGE
TBSP_AUTO_RESIZE_ENABLED
TBSP_TOTAL_SIZE_KB
TBSP_USABLE_SIZE_KB
TBSP_UTILIZATION_PERCENT
TBSP_PAGE_TOP
DATA_PHYSICAL_READS
DATA_HIT_RATIO_PERCENT
...
```

Deprecated: bp_hitratios (v), bp_read_io(v), bp_write_io(v), container_utilization(v)



Admin views/table functions - types (ex)

```
SELECT varchar(tbsp_name, 30),
       tbsp_type, pool_data_p_reads, direct_reads, direct_writes, pool_data_writes
FROM TABLE(SYSPROC.mon_get_tablespace('',-1))
```

		BPPR	DR	DW	PDW
SYSCATSPACE	DMS	945	7	288	1428
TEMPSPACE1	SMS	0	0	0	0
USERSPACE1	DMS	2	0	0	0
SYSTOOLSPACE	DMS	9	300	0	39

	C_ID	CONTAINER_TYPE	STR	TP	UP	FSgb	USgb
SYSTOOLSPACE	0	FILE_EXTENT_TAG	0	8192	8188	446	139

```
SELECT varchar(tbsp_name, 30), container_id, container_type, stripe_set, TOTAL_PAGES,
       USABLE_PAGES, fs_total_size/1024/1024/1024, fs_used_size/1024/1024/1024
FROM TABLE(SYSPROC.mon_get_container('SYSTOOLSPACE',-1))
```



Admin views/table functions - types

monitoring related views/table functions

applications, connections, agents, ...

locks

MON_GET_CONNECTION (F), MON_GET_SERVICE_SUBCLASS (F)
MON_GET_UNIT_OF_WORK (F), MON_GET_WORKLOAD (F)
MON_GET_AGENT (F)

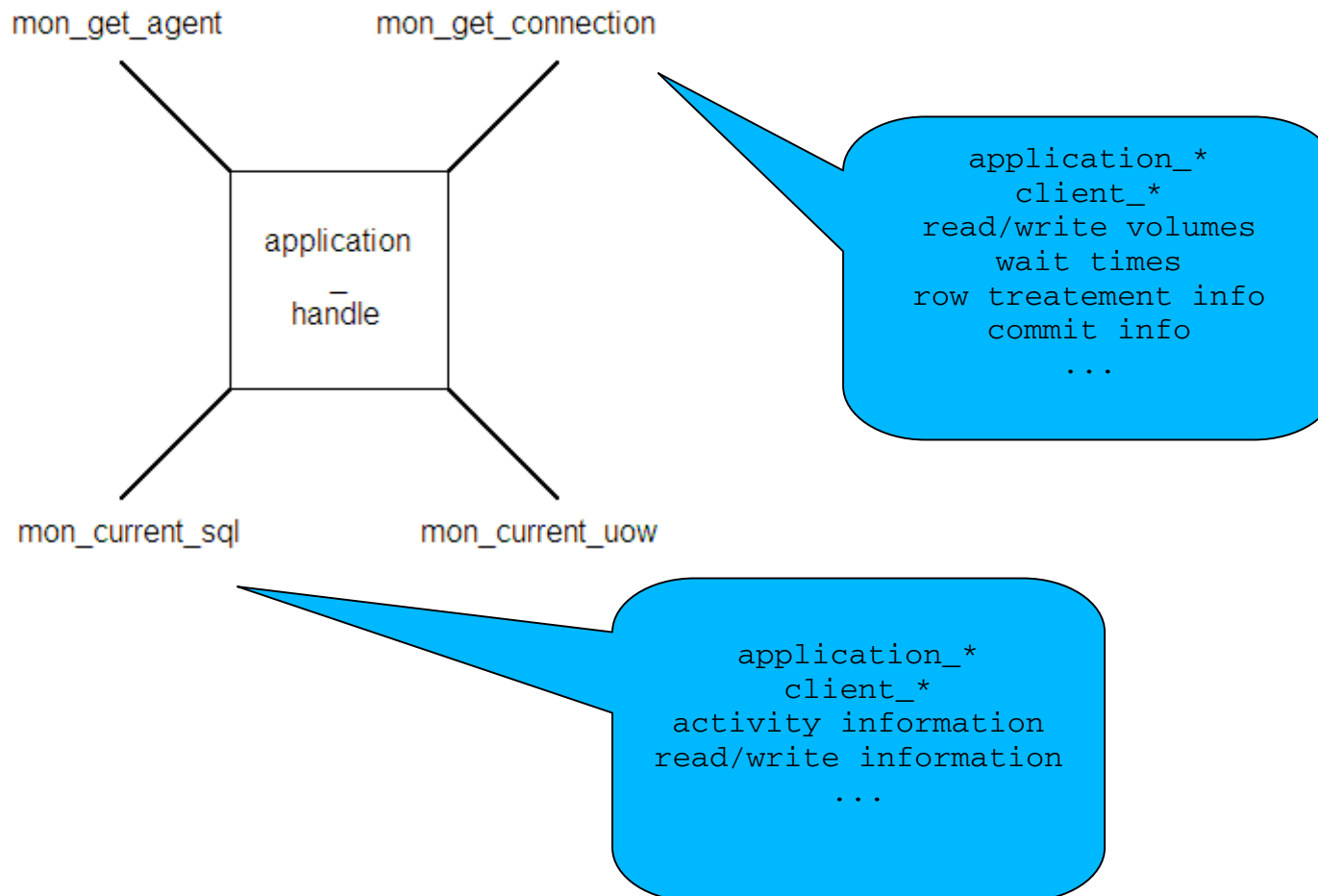
MON_CURRENT_SQL (V), MON_CONNECTION_SUMMARY (V)
MON_WORKLOAD_SUMMARY (V), MON_CURRENT_UOW(V)

MON_GET_EXTENDED_LATCH_WAIT (F)
MON_GET_LOCKS (F)

Deprecated: appl_performance (v), applications (v), long_running_sql(v)



Admin views/table functions - types





L UW administrative objects - admin tasks

Some routines/procedures allow the DBA to perform specific administrative tasks otherwise:

- requiring the use of external APIs - eg. utilities
- not possible with the scope of one, single statement

The sysproc schema contains a number of procedures allowing the DBA to issue administrative commands through SQL applications, without the need to use external APIs (CLP interpretation not required).



LUW admin routine - admin_cmd

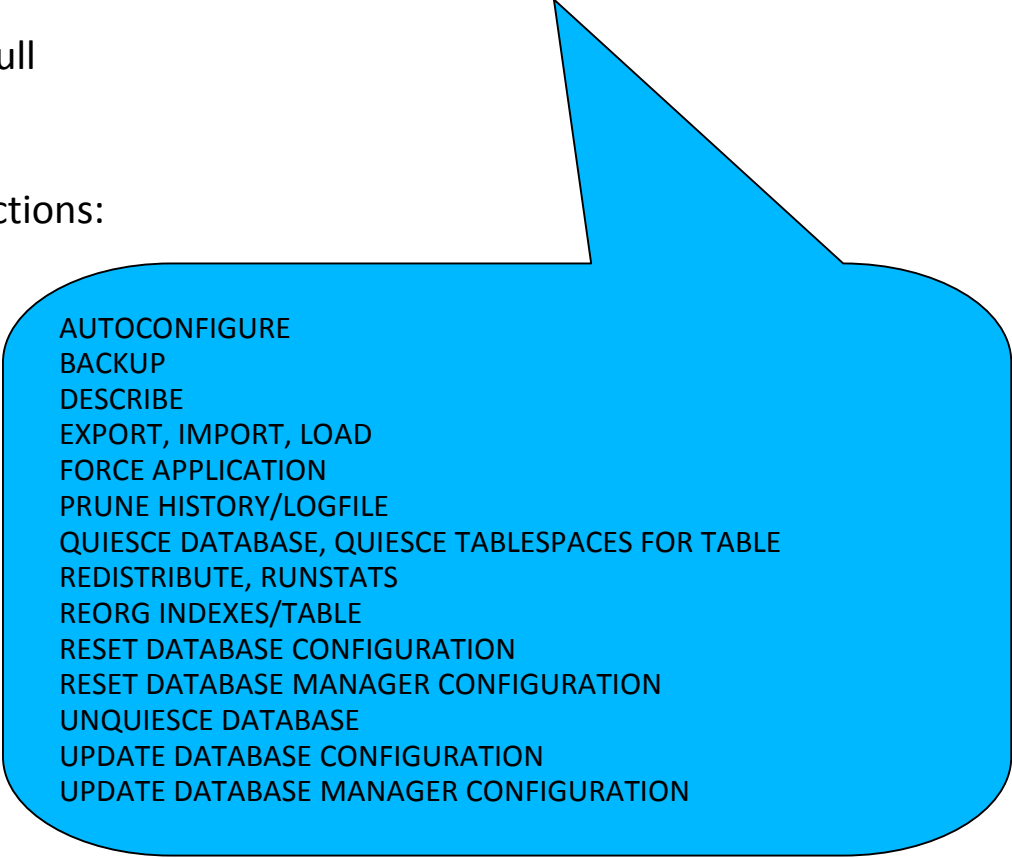
ADMIN_CMD: procedure invoked through CALL allowing the execution of administrative commands (including CLP commands).

- °) not all commands are supported/usefull
- °) some are supported with restrictions

Message handling for server generated actions:

ADMIN_GET_MSGS (f)

ADMIN_REMOVE_MSGS (p)



AUTOCONFIGURE
BACKUP
DESCRIBE
EXPORT, IMPORT, LOAD
FORCE APPLICATION
PRUNE HISTORY/LOGFILE
QUIESCE DATABASE, QUIESCE TABLESPACES FOR TABLE
REDISTRIBUTE, RUNSTATS
REORG INDEXES/TABLE
RESET DATABASE CONFIGURATION
RESET DATABASE MANAGER CONFIGURATION
UNQUIESCE DATABASE
UPDATE DATABASE CONFIGURATION
UPDATE DATABASE MANAGER CONFIGURATION



LUW admin routine - admin_cmd (ex1)

```
CALL SYSPROC.ADMIN_CMD
('EXPORT to c:\scratchad\mytables.ixf
OF ixf MESSAGES ON SERVER
select * from mytables');
```

Result set 1

ROWS_EXPORTED	MSG_RETRIEVAL	MSG_REMOVAL
<nbrOf>	sql1	sql2

Statement to fetch error messages:
SELECT SQLCODE, MSG
FROM TABLE(SYSPROC.ADMIN_GET_MSGS('9005_DUSER '))

Statement to clear error messages:
CALL SYSPROC.ADMIN_REMOVE_MSGS('9005_DUSER ')

```
SELECT SQLCODE, MSG
FROM TABLE(SYSPROC.ADMIN_GET_MSGS('9005_DUSER '))
```

```
SQL3104N The Export utility is beginning to export data to
file"c:\scratchad\mytables.ixf".
SQL3001C An I/O error (reason = "sqlopen -2029060079")
occurred while opening the output file.
SQL3105N The Export utility has finished exporting "0"
rows.
```

Missing a 'p':
scratchpad

```
CALL SYSPROC.ADMIN_REMOVE_MSGS('9005_DUSER ')
Return Status = 0
```



LUW admin routine - admin_cmd (ex2)

```
CALL SYSPROC.ADMIN_CMD  
('EXPORT to c:\scratchad\mytables.del  
OF del MESSAGES ON SERVER  
select * from mytables');  
  
Result set 1  
-----  
ROWS_EXPORTED    MSG_RETRIEVAL    MSG_REMOVAL  
          5181                sql1                sql2
```



LUW admin routine - admin_cmd (ex3)

```
db2 -tvf cmd.sql
CALL SYSPROC.ADMIN_CMD
  ('RUNSTATS ON TABLE MYTABLES ON KEY COLUMNS and INDEXES ALL')
  Return Status = 0

db2 -tvf cmd.sql
CALL SYSPROC.ADMIN_CMD
  ('RUNSTATS ON TABLE MYTABS ON KEY COLUMNS and INDEXES ALL')
SQL2306N  The table or index "DUSER.MYTABS" does not exist.
```

No consistent use of messaging interface!

```
C:\Scratchpad>db2 -tvf cmd.sql
CALL SYSPROC.ADMIN_CMD ('REORG TABLE MYTABLES RESETDICTIONARY')

  Return Status = 0
```



LUW admin routine - schema

schema related procedures

- ADMIN_COPY_SCHEMA
- ADMIN_DROP_SCHEMA

- ADMIN_MOVE_TABLE_UTIL,
ADMIN_MOVE_TABLE

```
(c/d) sourceschema --> from
(c) targetschema --> to
(c) copymode --> DDL, COPY, COPYNO
(d) dropmode --> NULL
(c) objectowner
(c) sourcetbsp
(c) targettbsp
(c/d) errortabschema --> systoolsspace
(c/d) errortab
```

Complex and complete!

Table can be created beforehand (easier?) or within the stored procedure itself!

V1 (tabschema, tabname, target_tabname, options(+), operation)

V2 (tabschema, tabname, data_tbsp, index_tbsp, lob_tbsp, organize_by_clause,
partkey_cols, data_part, coldef, option(+), operation)



LUW admin routine - schema (ex1)

```
CALL SYSPROC.ADMIN_COPY_SCHEMA  
( 'DUSER', 'EUSER', 'COPY', NULL, NULL, NULL, 'DUSER', 'CPERROR' )
```

Value of output parameters

Parameter Name : ERRORTABSCHEMA

Parameter Value : -

Parameter Name : ERRORTABNAME

Parameter Value : -

Return Status = 0

```
CALL SYSPROC.ADMIN_DROP_SCHEMA( 'EUSER', NULL, 'DUSER', 'DRPERROR' )
```

Value of output parameters

Parameter Name : ERRORTABSCHEMA

Parameter Value : DUSER

Parameter Name : ERRORTAB

Parameter Value : DRPERROR

Return Status = 0

Errors exist!



LUW admin routine - schema (ex2)

```
CALL SYSPROC.ADMIN_MOVE_TABLE  
( 'DUSER',  
  'MYTABLES',  
  'MYSPACE', '', '', '', '', '',  
  'MYNAME VARCHAR(128), MYCREATOR VARCHAR(128), MYTYPE VARCHAR(1), MYPK INTEGER',  
  '',  
  'MOVE' )
```

```
SQL2105N  The ADMIN_MOVE_TABLE procedure could not be completed because  
prerequisite for running the procedure was not satisfied. Reason code: "24"  
SQLSTATE=5UA0M
```

Validity of execution
environment checked

```
CALL SYSPROC.ADMIN_MOVE_TABLE  
( 'DUSER', 'MYTABLES', 'MYSPACE', 'MYSPACE', 'MYSPACE', '', '', '',  
  'MYNAME VARCHAR(128), MYCREATOR VARCHAR(128), MYTYPE VARCHAR(1), MYPK  
  INTEGER', '', 'CANCEL' )
```

Result set 1

KEY

VALUE

0 record(s) selected.

Return Status = 0

Cancel work in progress!



LUW admin routine - schema (ex2 ...)

```

CALL SYSPROC.ADMIN_MOVE_TABLE (
  'DUSER',
  'MYTABLES',
  'MYSPACE',
  'MYSPACE',
  'MYSPACE',
  '',
  '',
  '',
  'NAME VARCHAR(128), CREATOR VARCHAR(128), TYPE VARCHAR(1), MYPK
  INTEGER',
  '',
  'MOVE');
  
```

Result set 1

KEY	VALUE
AUTHID	DUSER
CLEANUP_END	2013-09-01-17.39.12.789000
CLEANUP_START	2013-09-01-17.39.12.664000
COPY_END	2013-09-01-17.39.10.758000
COPY_OPTS	ARRAY_INSERT, NON_CLUSTER
COPY_START	2013-09-01-17.39.10.024000
COPY_TOTAL_ROWS	5181
INDEXNAME	
INDEXSCHEMA	
INIT_END	2013-09-01-17.39.09.774000
INIT_START	2013-09-01-17.39.08.164000
AR_COLDEF	NAME VARCHAR(128), CREATOR VARCHAR(128),
<etc>	



LUW administrative routine - system

configuration related procedures

- GET_SYSTEM_INFO (p)
- GET_CONFIG (p)
- SET_CONFIG (p)



LUW administrative routine - utilities

utilities

- REORGCHECK_IX_STATS (p)
- REORGCHECK_TB_STATS (p)



Usage scenario 1 - quick-win analysis

- get acquainted with a limited set of crucial admin views/functions allowing for a quick access to and analysis of performance data 'just-in-case'
- allows for prompt evaluation of 'perceived' performance issues



Usage scenario 2 - enhanced secured acces

- use of admin views/functions allows for a more granual security setup
 - access to system/performance data can now be customised/tailored to specific needs
[using standard grants]
- views on views/table functions
- usage through procedures and functions allows for incorporation of business/function - specific access, and/or additional security specifics
[eg. detailed logging, auditing, ...]



Usage scenario 3 - management packs

- build a set of 'private' views, procedures and functions in a 'private' management schema
- install the schema and schema objects in all instances/databases to be managed
- creates an integrated, customisable and very flexible interface across systems for identifying and tackling system bottlenecks.



Usage scenario 4 - performance infrastructure

- set-up a centralised 'repository structure', to persistently store the 'volatile' data returned by the management functions
 - local
 - remote
- integrate with the 'management pack' discussed before
[provide an integrated approach to 'timing' and 'naming' across systems]
- perform time-based/system-based reporting and analysis
- careful when storing data: remember data significance!
Interpretation is obviously 'key'!

Koen De Backer

ABIS Training & Consulting

kdebacker@abis.be

Session: D7

LUWs hidden treasures -
administrative views and routines