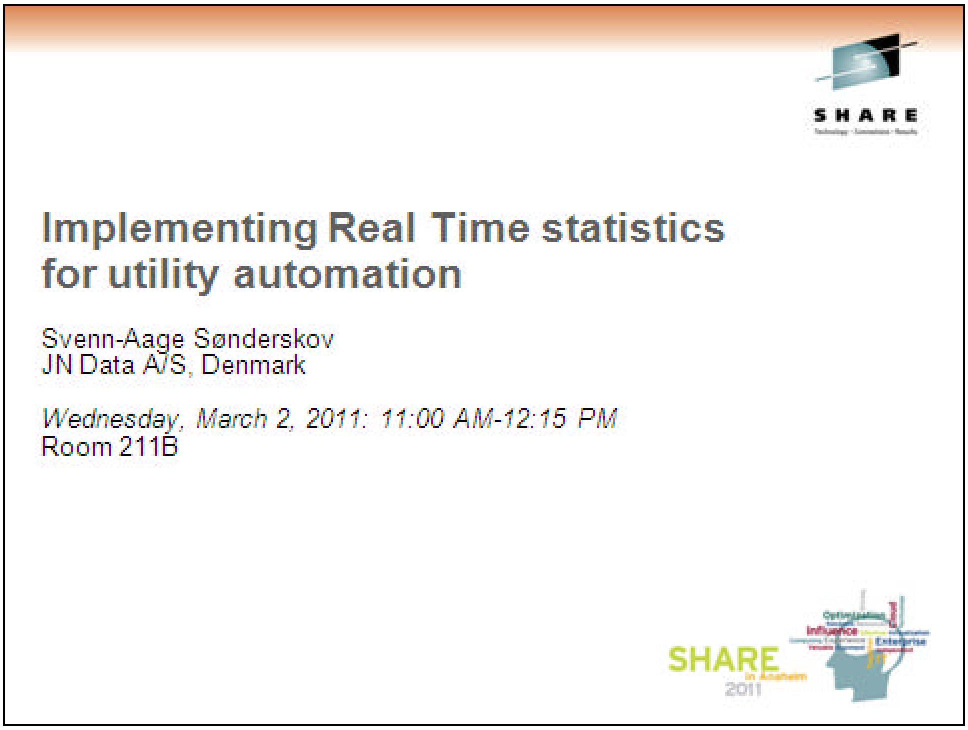
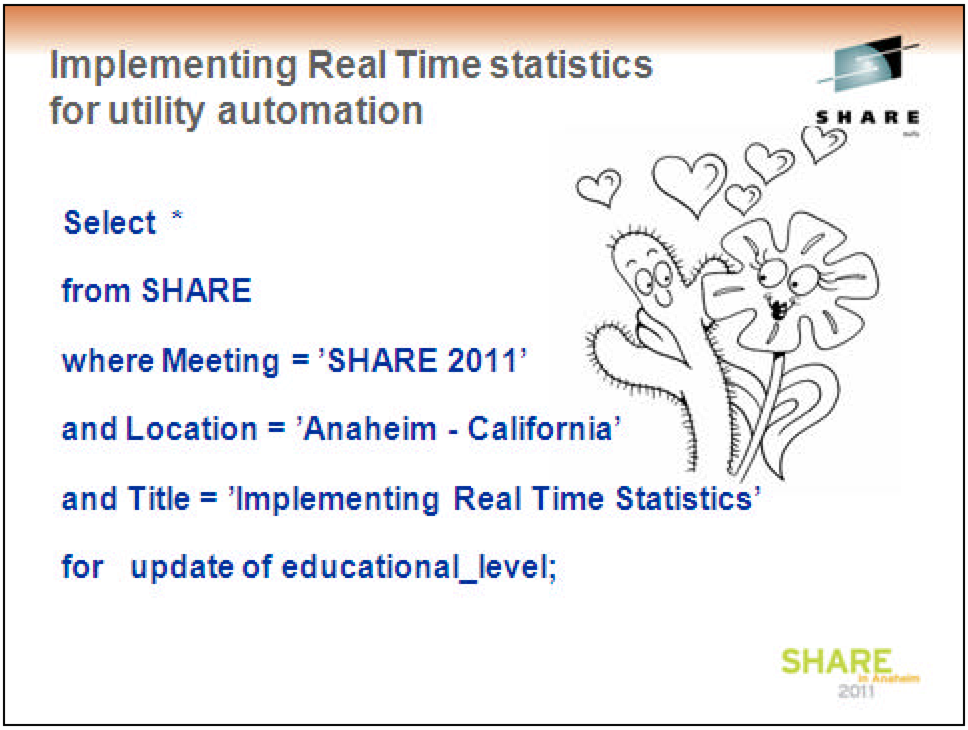


# Implementing Real Time statistics for utility automation

Svenn-Aage Sønderskov
JN Data A/S, Denmark

*Wednesday, March 2, 2011: 11:00 AM-12:15 PM*
Room 211B



## Implementing Real Time statistics for utility automation

SHARE

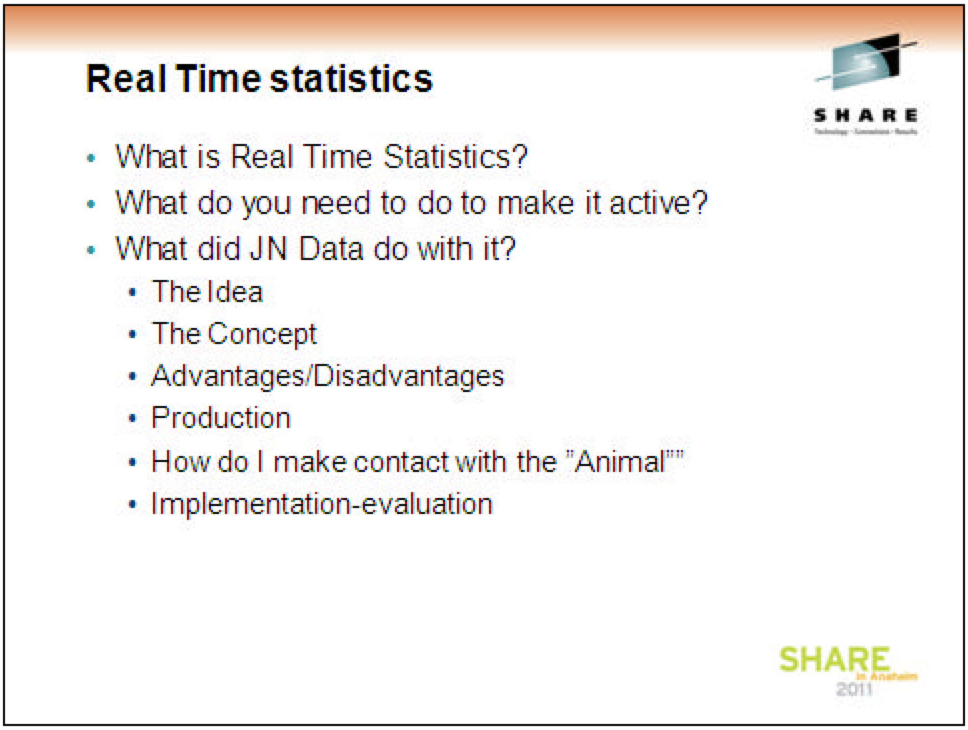
```
Select  *
from SHARE
where Meeting = 'SHARE 2011'
and Location = 'Anaheim - California'
and Title = 'Implementing  Real Time Statistics'
for   update of educational_level;
```

SHARE
in Anaheim
2011

It is amazing how precise SQL is in decribing things ;-) and a simple SQL-code can be the answer

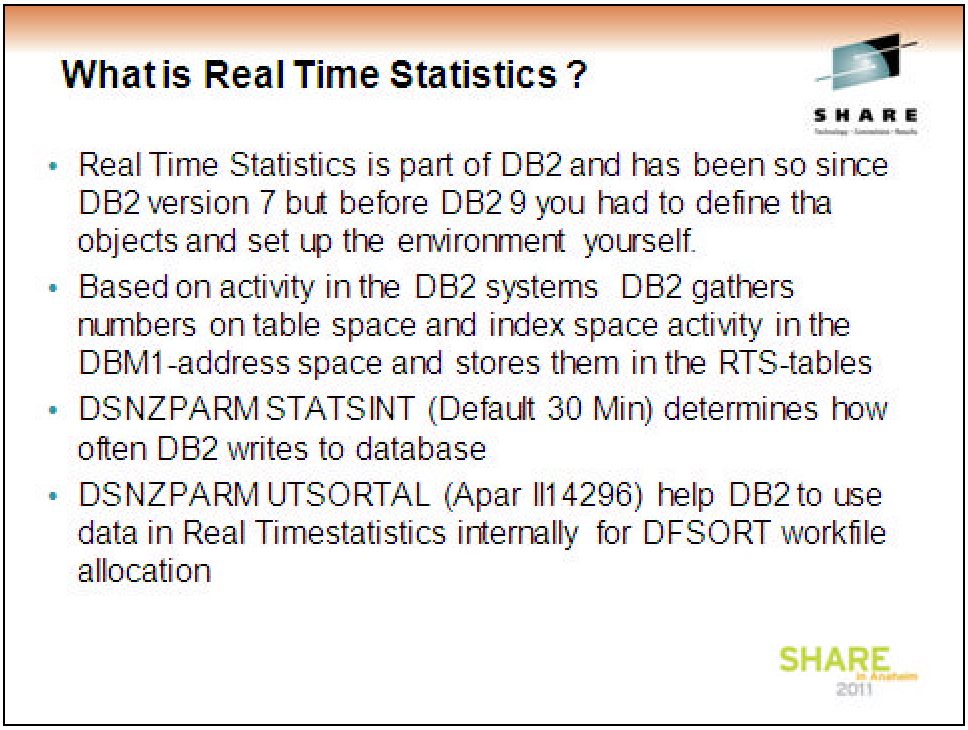
Imagine, that your children were taught SQL as the first foreign Language in school. A question from them to get money for a new computer could be answered by a -904 (ressource Unavailable) and whne putting the same question to their Mother she could simple answer -803 (Duplicate Row – We have hadf that question before) -)

So please join us in making SQL a spoken language ;-))



## Real Time statistics

- What is Real Time Statistics?
- What do you need to do to make it active?
- What did JN Data do with it?
  - The Idea
  - The Concept
  - Advantages/Disadvantages
  - Production
  - How do I make contact with the "Animal""
  - Implementation-evaluation

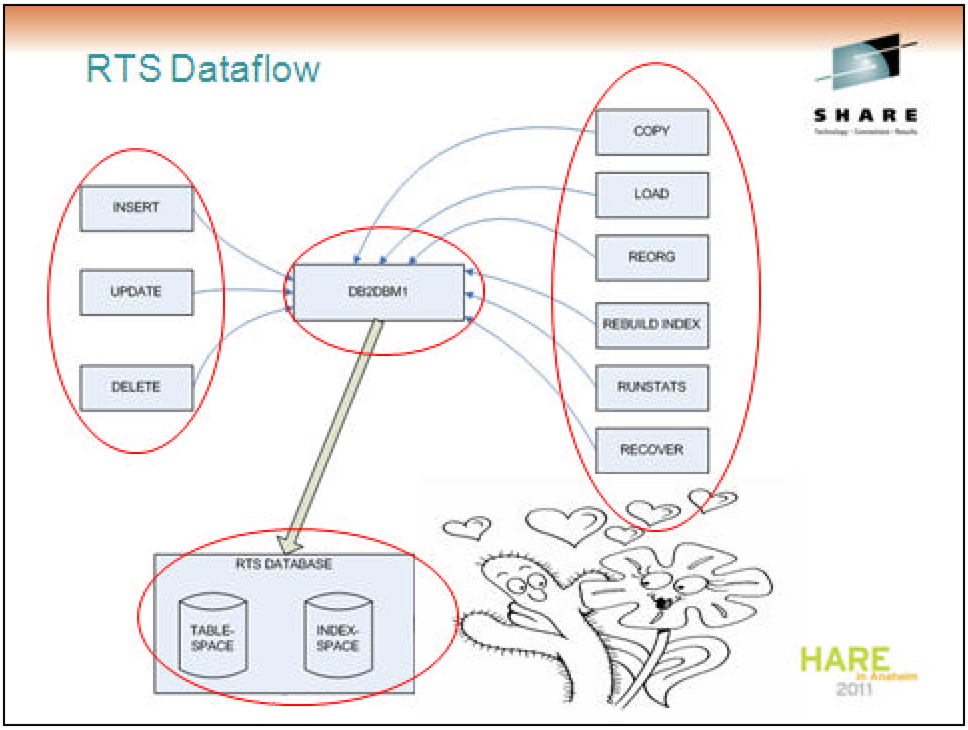


## What is Real Time Statistics ?

- Real Time Statistics is part of DB2 and has been so since DB2 version 7 but before DB2 9 you had to define tha objects and set up the environment yourself.
- Based on activity in the DB2 systems DB2 gathers numbers on table space and index space activity in the DBM1-address space and stores them in the RTS-tables
- DSNZPARM STATSINT (Default 30 Min) determines how often DB2 writes to database
- DSNZPARM UTSORTAL (Apar II14296) help DB2 to use data in Real Timestatistics internally for DFSORT workfile allocation

We have set the STATSINT to 1 minute, whiche means that the DBM1-addressspace outputs activity-records to the RTS tables every 1 minute.

Since the number of affected records are reasonably small we have seen no problems of having this value set at 1 minute

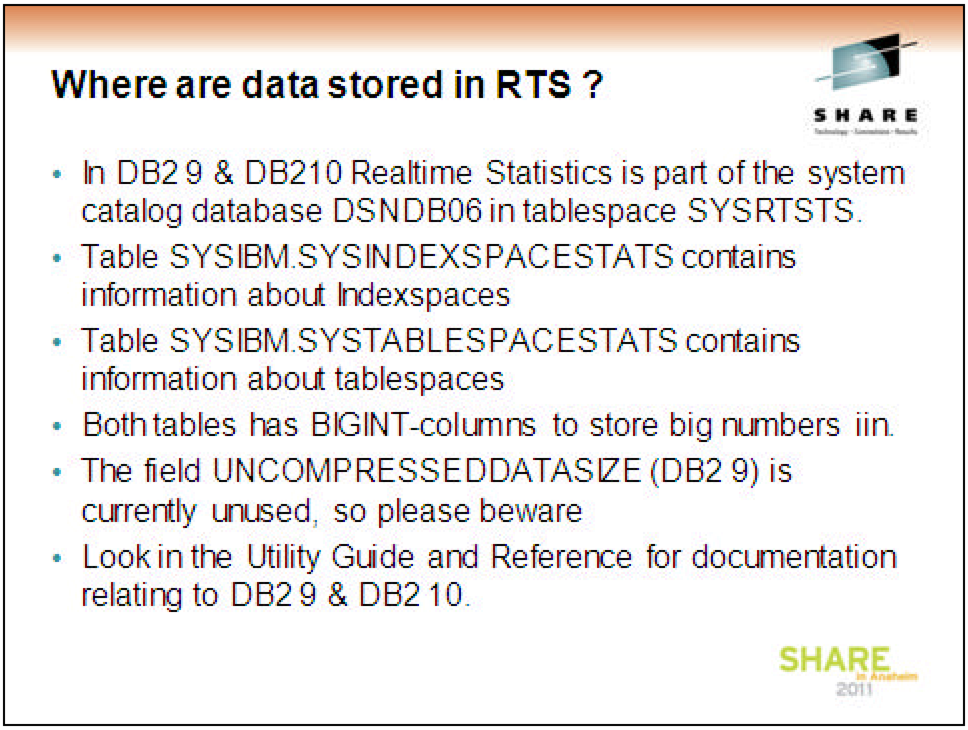


## RTS Dataflow

In order to get the full benefit of RTS you have to run a REORG-utility so that the DBM1-addressspace can register an initial Cardinality on Table Spaces and Index Spaces.

When Insert, Update and Delete's happens the DBM1-addrress space collect these data and updates the RTS-tables at the STATSINT Interval in DSNZPARM

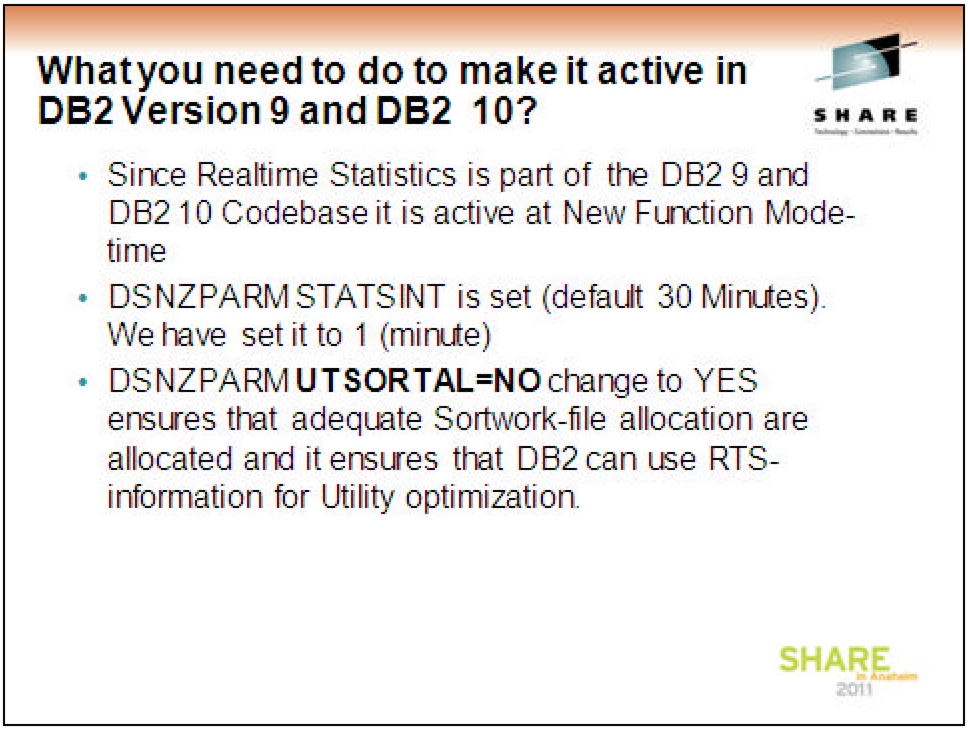
This gives you the operational basis for using RTS for utility automation.



## Where are data stored in RTS ?

- In DB2 9 & DB210 Realtime Statistics is part of the system catalog database DSNDB06 in tablespace SYSRTSTS.
- Table SYSIBM.SYSINDEXSPACESTATS contains information about Indexspaces
- Table SYSIBM.SYSTABLESPACESTATS contains information about tablespaces
- Both tables has BIGINT-columns to store big numbers iin.
- The field UNCOMPRESSEDDATASIZE (DB2 9) is currently unused, so please beware
- Look in the Utility Guide and Reference for documentation relating to DB2 9 & DB2 10.

SHARE in Anaheim 2011



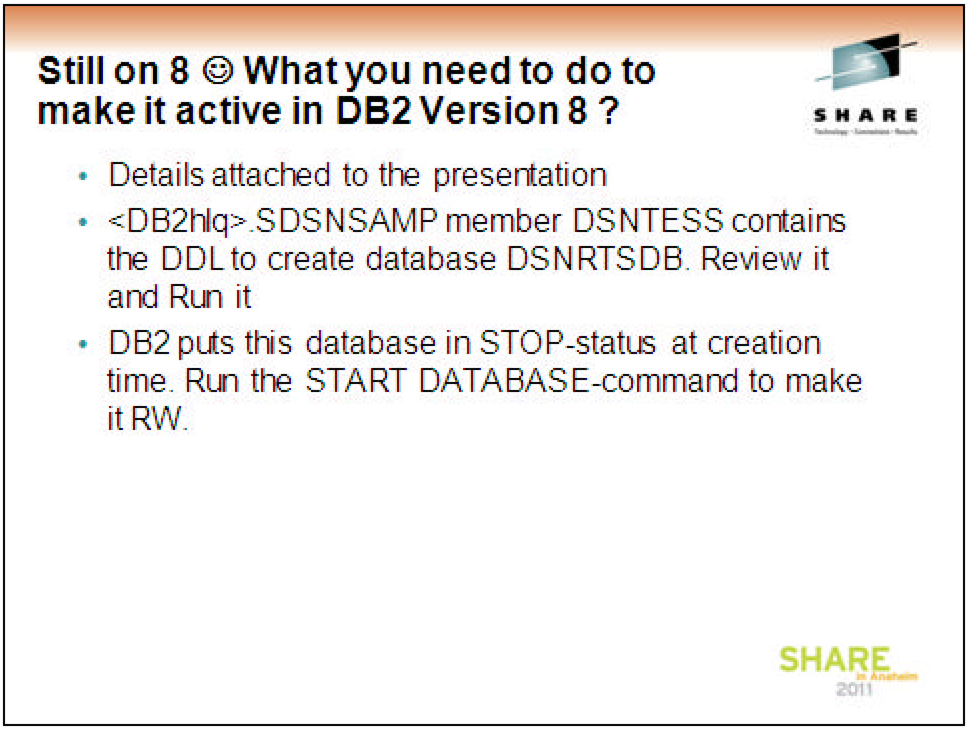
## What you need to do to make it active in DB2 Version 9 and DB2 10?

- Since Realtime Statistics is part of the DB2 9 and DB2 10 Codebase it is active at New Function Mode-time
- DSNZPARM STATSINT is set (default 30 Minutes). We have set it to 1 (minute)
- DSNZPARM **UTSORTAL=NO** change to YES ensures that adequate Sortwork-file allocation are allocated and it ensures that DB2 can use RTS-information for Utility optimization.

SHARE in Anaheim 2011

The Installation SDSNSAMP-library contains the basic RTS-definitions.

If you want to have shadow tablesets for viewing growth etc you have to build them yourself. You can yuse the table-ddl from tha basic ones and add a timestamp which you set when you load from the basic table

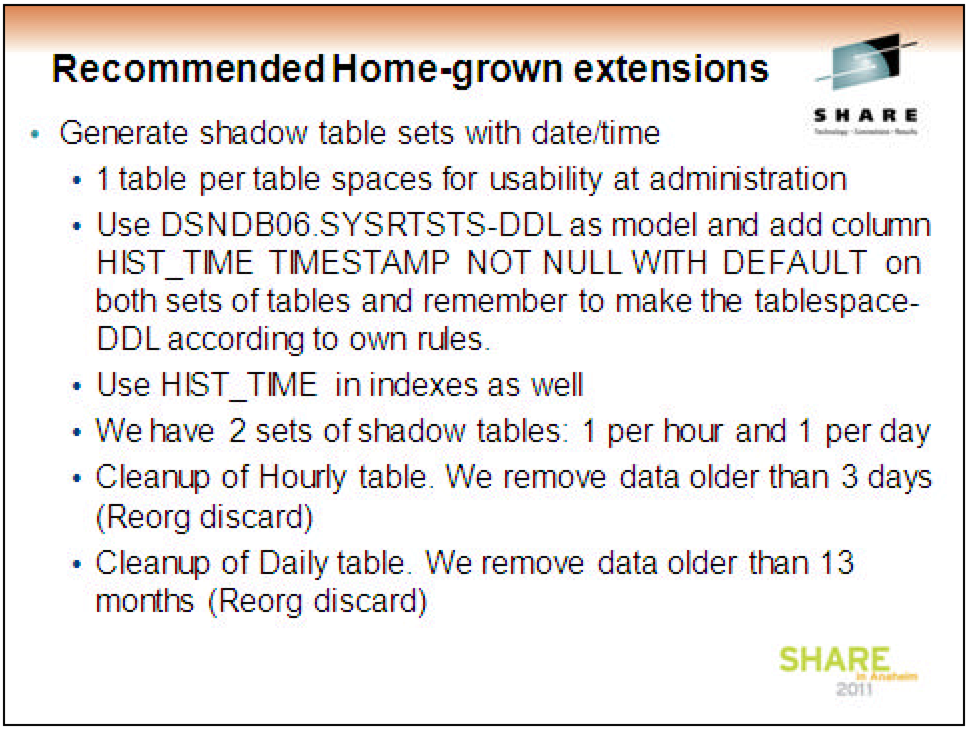


## Still on 8 ☺ What you need to do to make it active in DB2 Version 8 ?

- Details attached to the presentation
- <DB2hlq>.SDSNSAMP member DSNTESS contains the DDL to create database DSNRTSDB. Review it and Run it
- DB2 puts this database in STOP-status at creation time. Run the START DATABASE-command to make it RW.

The Installation SDSNSAMP-library contains the basic RTS-definitions.

If you want to have shadow tablesets for viewing growth etc you have to build them yourself. You can yuse the table-ddl from tha basic ones and add a timestamp which you set when you load from the basic table



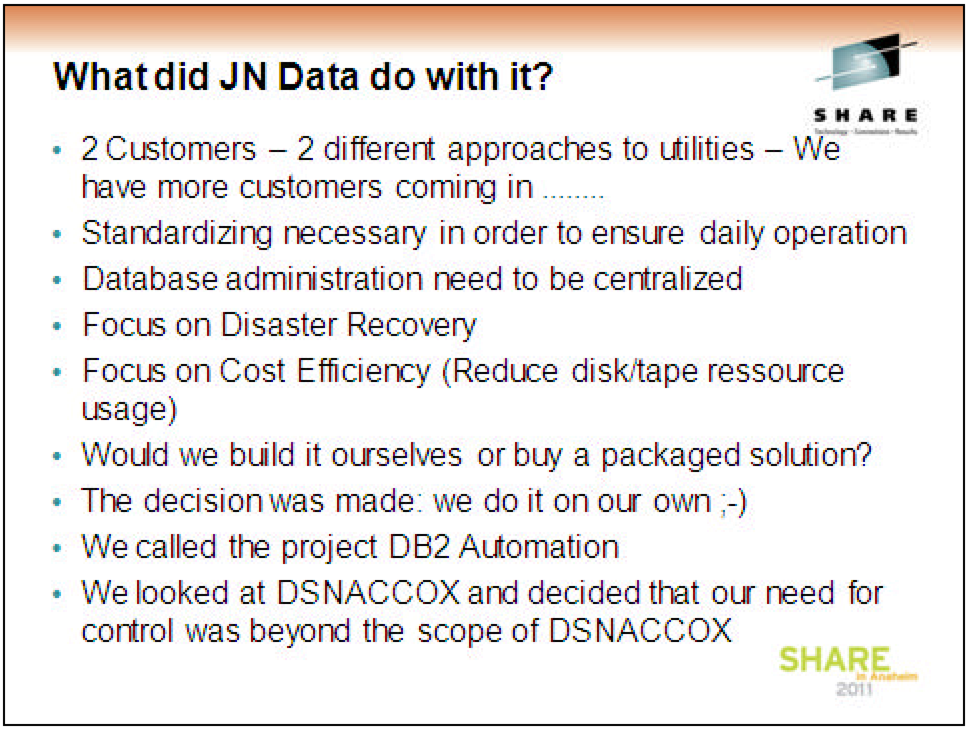
## Recommended Home-grown extensions

- Generate shadow table sets with date/time
  - 1 table per table spaces for usability at administration
  - Use DSNDB06.SYSRTSTS-DDL as model and add column HIST_TIME TIMESTAMP NOT NULL WITH DEFAULT on both sets of tables and remember to make the tablespace-DDL according to own rules.
  - Use HIST_TIME in indexes as well
  - We have 2 sets of shadow tables: 1 per hour and 1 per day
  - Cleanup of Hourly table. We remove data older than 3 days (Reorg discard)
  - Cleanup of Daily table. We remove data older than 13 months (Reorg discard)

SHARE in Anaheim 2011

The RTS-tables is updated in place, so if you want to follow the numbers to see when activity is happening you have to undload data on your own timeframe. We use 1 unload pr hour and load the data into a shadow-table where we have build a timestamp-column into the shadow table.

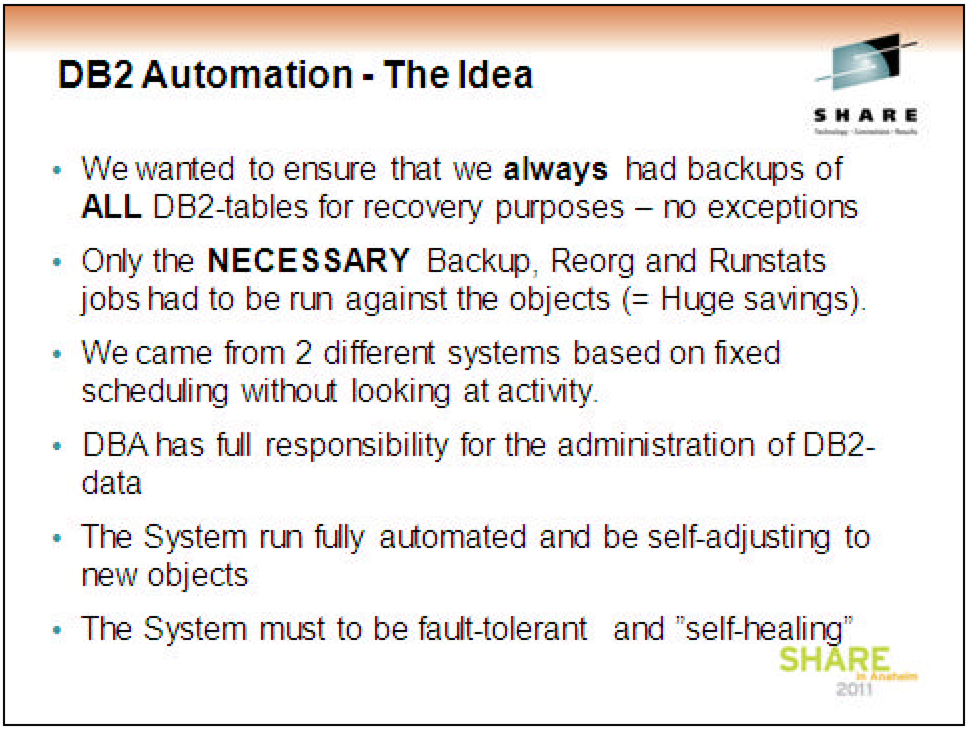
This means that we in the shadow table have one entry pr hour in tha detailed table and 1 entry pr day in the other shadow table.



## What did JN Data do with it?

- 2 Customers – 2 different approaches to utilities – We have more customers coming in ........
- Standardizing necessary in order to ensure daily operation
- Database administration need to be centralized
- Focus on Disaster Recovery
- Focus on Cost Efficiency (Reduce disk/tape ressource usage)
- Would we build it ourselves or buy a packaged solution?
- The decision was made: we do it on our own ;-)
- We called the project DB2 Automation
- We looked at DSNACCOX and decided that our need for control was beyond the scope of DSNACCOX

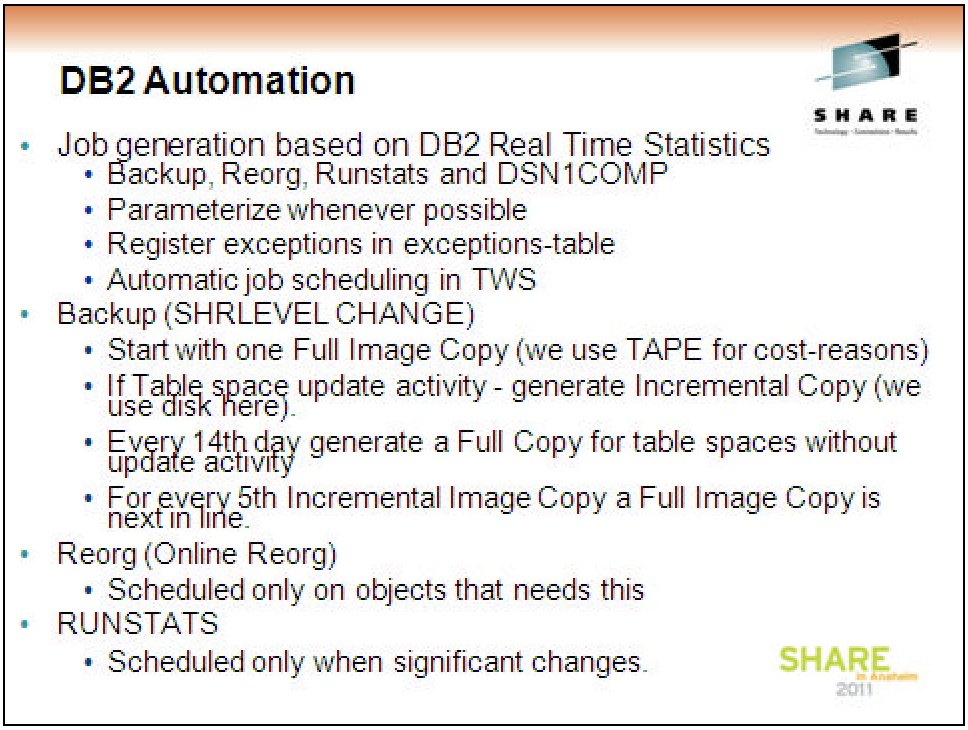
Having routines in place that ensures atandardized enviironments both ensures correct opration an ensures that in the Disaster Recovery situation you can rely on data being available for that purpose as well.



## DB2 Automation - The Idea

- We wanted to ensure that we **always** had backups of **ALL** DB2-tables for recovery purposes – no exceptions
- Only the **NECESSARY** Backup, Reorg and Runstats jobs had to be run against the objects (= Huge savings).
- We came from 2 different systems based on fixed scheduling without looking at activity.
- DBA has full responsibility for the administration of DB2-data
- The System run fully automated and be self-adjusting to new objects
- The System must to be fault-tolerant and "self-healing"

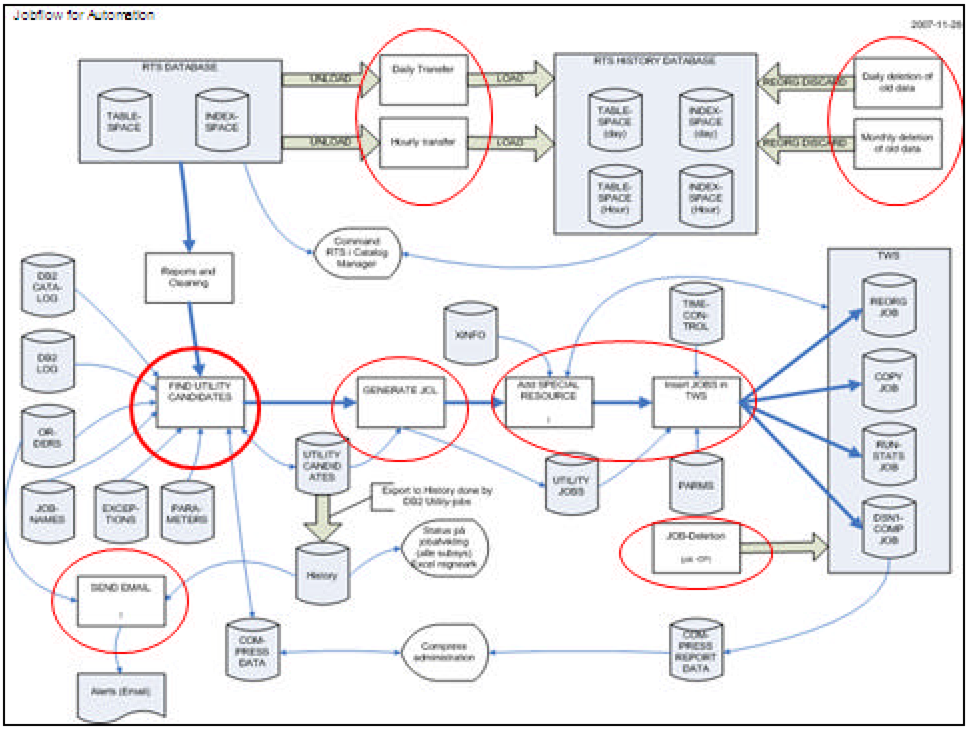
We wanted to make the utility-approach for both of our customers the same.



## DB2 Automation

- Job generation based on DB2 Real Time Statistics
  - Backup, Reorg, Runstats and DSN1COMP
  - Parameterize whenever possible
  - Register exceptions in exceptions-table
  - Automatic job scheduling in TWS
- Backup (SHRLEVEL CHANGE)
  - Start with one Full Image Copy (we use TAPE for cost-reasons)
  - If Table space update activity - generate Incremental Copy (we use disk here).
  - Every 14th day generate a Full Copy for table spaces without update activity
  - For every 5th Incremental Image Copy a Full Image Copy is next in line.
- Reorg (Online Reorg)
  - Scheduled only on objects that needs this
- RUNSTATS
  - Scheduled only when significant changes.

Trigger values atre by default the ones mentioned in DB2 Administration Guide: ’When to reorganize indexes and tablespaces’



The RTS Database is unloaded and loaded into 2 history table space sets

1. On an hourly basis – kept for 3 days (deletion done by REORG DISCARD)

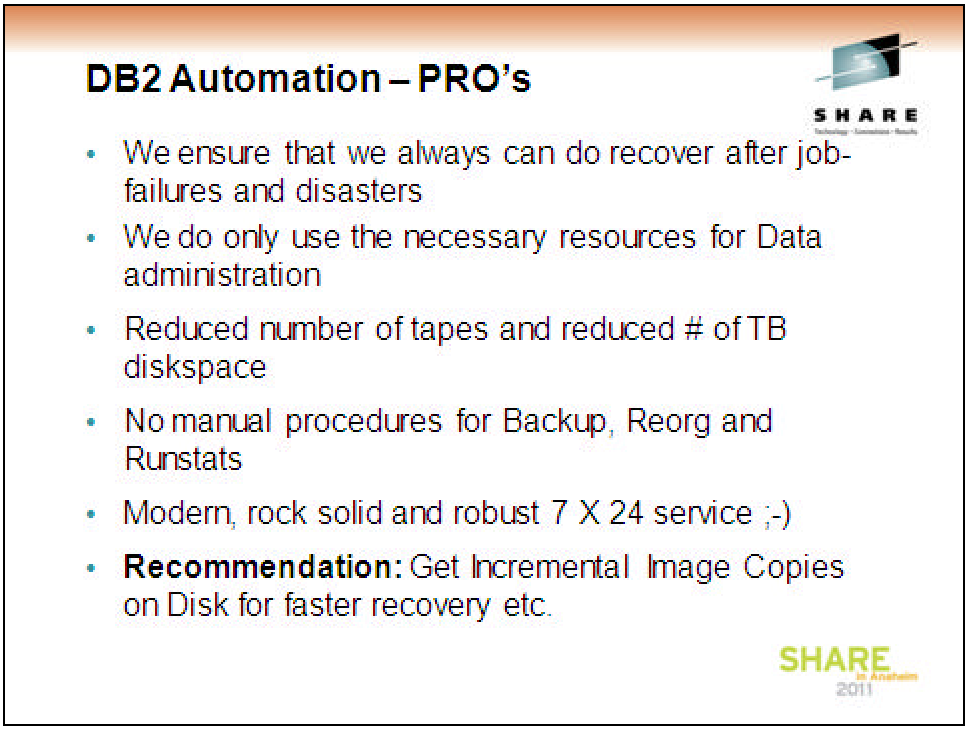
2. On a daily basis – Kept for 13 months (Delation done by REORG DISCARD)

The system looks in RTS to see if thresholds are met and put the objects in the utility candidate list for future JCL Generation.

The system generates JCL and registers the jobs and special resources in TWS.

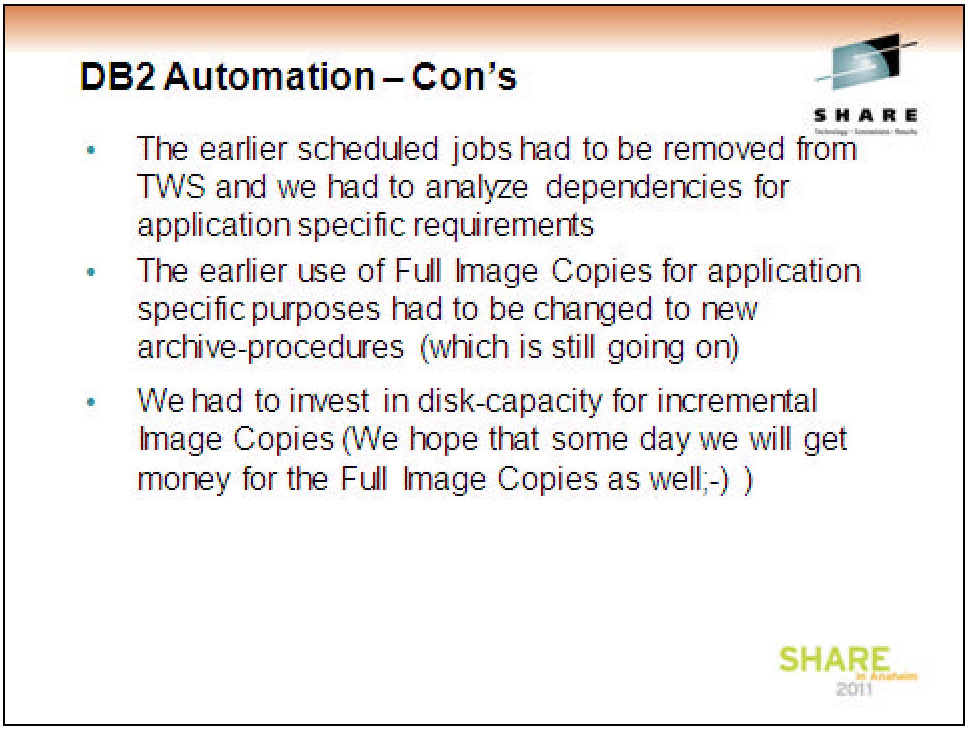
If the Jobs are not run becaus of timeframes etc an Email is sent to the DBA-Team, who decides on what has to be done.

The system cleans up the TWS-database for the already completed jobs and resources.



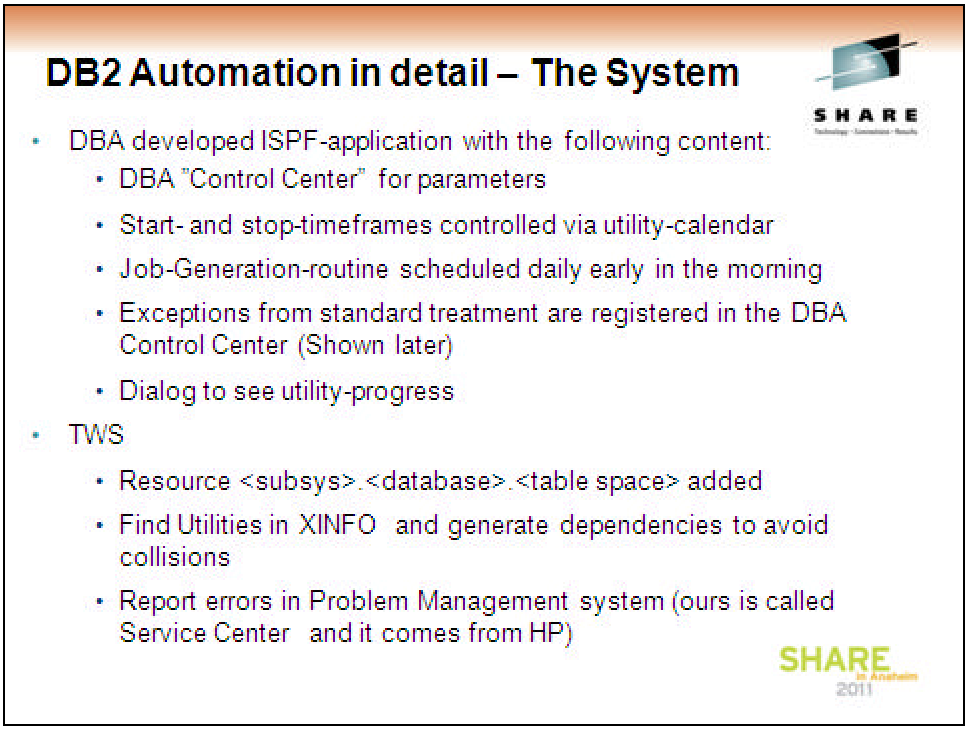
## DB2 Automation – PRO's

- We ensure that we always can do recover after job-failures and disasters
- We do only use the necessary resources for Data administration
- Reduced number of tapes and reduced # of TB diskspace
- No manual procedures for Backup, Reorg and Runstats
- Modern, rock solid and robust 7 X 24 service ;-)
- **Recommendation:** Get Incremental Image Copies on Disk for faster recovery etc.



## DB2 Automation – Con's

- The earlier scheduled jobs had to be removed from TWS and we had to analyze dependencies for application specific requirements
- The earlier use of Full Image Copies for application specific purposes had to be changed to new archive-procedures (which is still going on)
- We had to invest in disk-capacity for incremental Image Copies (We hope that some day we will get money for the Full Image Copies as well;-) )

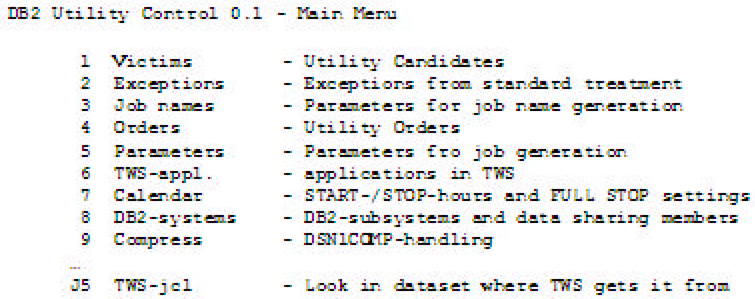


## DB2 Automation in detail – The System

- DBA developed ISPF-application with the following content:
  - DBA "Control Center" for parameters
  - Start- and stop-timeframes controlled via utility-calendar
  - Job-Generation-routine scheduled daily early in the morning
  - Exceptions from standard treatment are registered in the DBA Control Center (Shown later)
  - Dialog to see utility-progress
- TWS
  - Resource <subsys>.<database>.<table space> added
  - Find Utilities in XINFO and generate dependencies to avoid collisions
  - Report errors in Problem Management system (ours is called Service Center and it comes from HP)

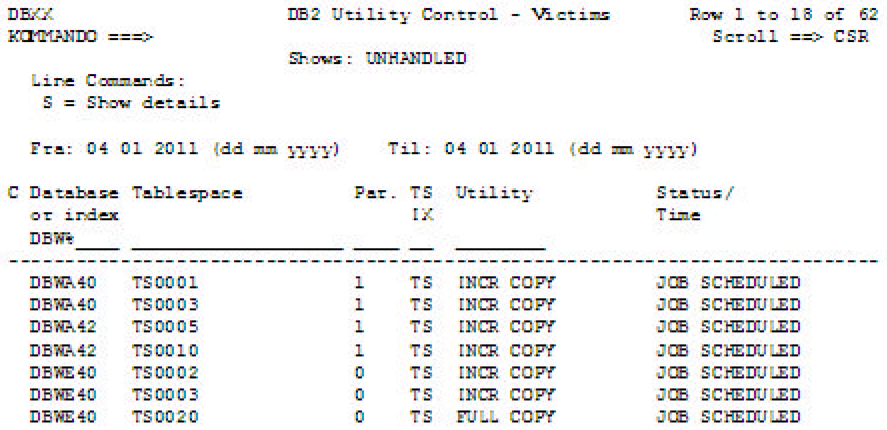
XINFO is a product from Horizont in Germany, which exttracts information from JCL, TWS, Sourcelibraries, DB2 and many other sources and put them in DB2 tables, that we can use to search for jobs that contains utilities.

# DB2 Utility Control Center Main Menu



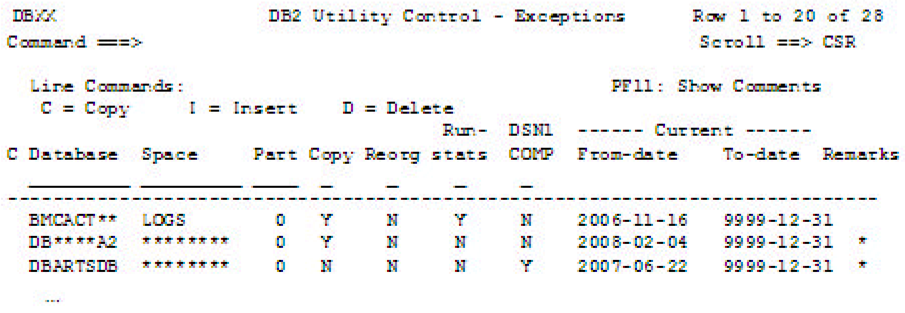
```
DB2 Utility Control 0.1 - Main Menu

       1  Victims         - Utility Candidates
       2  Exceptions      - Exceptions from standard treatment
       3  Job names       - Parameters for job name generation
       4  Orders          - Utility Orders
       5  Parameters      - Parameters fro job generation
       6  TWS-appl.       - applications in TWS
       7  Calendar        - START-/STOP-hours and FULL STOP settings
       8  DB2-systems     - DB2-subsystems and data sharing members
       9  Compress        - DSN1COMP-handling
      …
      J5  TWS-jcl         - Look in dataset where TWS gets it from
```

# Victims = Those who meet the thresholds



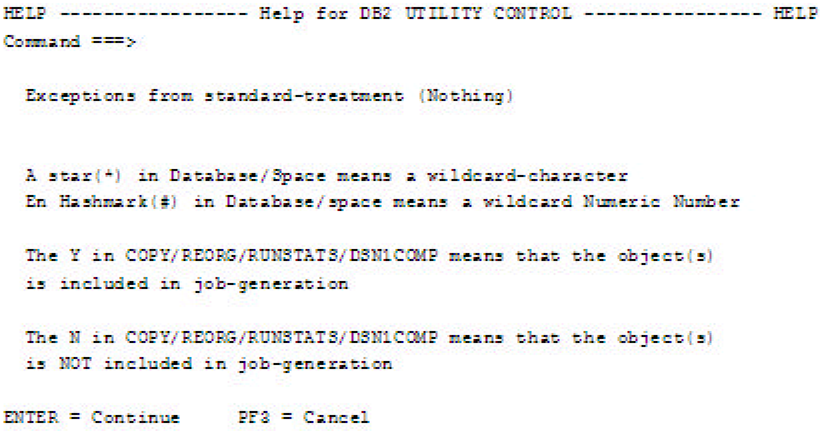
```
DBXX                     DB2 Utility Control - Victims      Row 1 to 18 of 62
KOMMANDO ===>                                                 Scroll ==> CSR
                         Shows: UNHANDLED
  Line Commands:
   S = Show details

  Fra: 04 01 2011 (dd mm yyyy)    Til: 04 01 2011 (dd mm yyyy)

C Database Tablespace          Par. TS  Utility            Status/
  or index                          IX                     Time
  DBW*____ ___________________ ____ __  ________
------------------------------------------------------------------------------
  DBWA40   TS0001              1    TS  INCR COPY          JOB SCHEDULED
  DBWA40   TS0003              1    TS  INCR COPY          JOB SCHEDULED
  DBWA42   TS0005              1    TS  INCR COPY          JOB SCHEDULED
  DBWA42   TS0010              1    TS  INCR COPY          JOB SCHEDULED
  DBWE40   TS0002              0    TS  INCR COPY          JOB SCHEDULED
  DBWE40   TS0003              0    TS  INCR COPY          JOB SCHEDULED
  DBWE40   TS0020              0    TS  FULL COPY          JOB SCHEDULED
```

# Exceptions – those who will not be handled



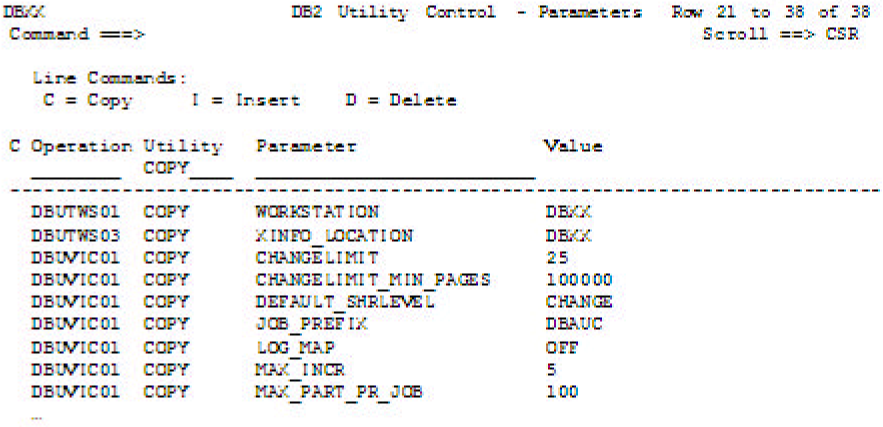
```
DBXX                   DB2 Utility Control - Exceptions      Row 1 to 20 of 28
Command ==>                                                  Scroll ==> CSR

  Line Commands:                                     PF11: Show Comments
   C = Copy     I = Insert    D = Delete
                                       Run- DSN1  ------ Current ------
C Database  Space     Part Copy Reorg stats COMP  From-date    To-date  Remarks

  _________ _________ ____  _     _     _     _
-----------------------------------------------------------------------------

  BMCACT**  LOGS        0   Y     N     Y     N   2006-11-16   9999-12-31
  DB****A2  ********    0   Y     N     N     N   2008-02-04   9999-12-31   *
  DBARTSDB  ********    0   N     N     N     Y   2007-06-22   9999-12-31   *
    ...
```

# help – panel to understand setting



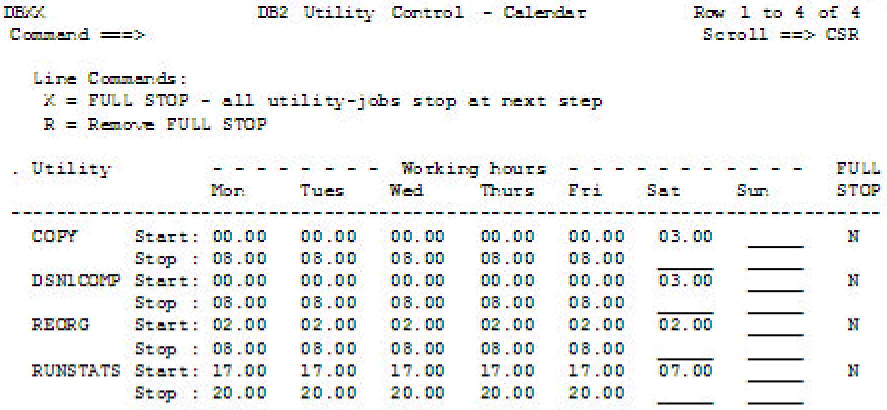
```
HELP ----------------- Help for DB2 UTILITY CONTROL ---------------- HELP
Command ===>

  Exceptions from standard-treatment (Nothing)


  A star(*) in Database/Space means a wildcard-character
  En Hashmark(#) in Database/space means a wildcard Numeric Number

  The Y in COPY/REORG/RUNSTATS/DSN1COMP means that the object(s)
  is included in job-generation

  The N in COPY/REORG/RUNSTATS/DSN1COMP means that the object(s)
  is NOT included in job-generation

ENTER = Continue     PF3 = Cancel
```

# Parameter-settings for COPY



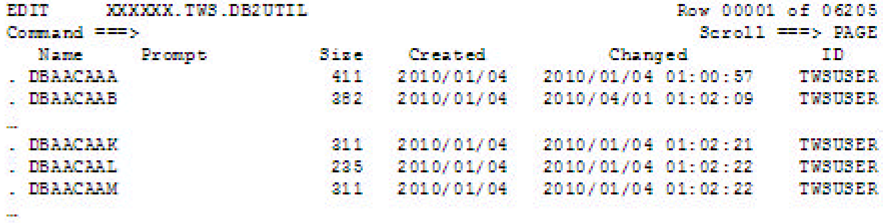
```
DBXX                         DB2  Utility  Control  - Parameters   Row 21 to 38 of 38
 Command ===>                                                          Scroll ==> CSR

   Line Commands:
    C = Copy       I = Insert     D = Delete

 C Operation Utility   Parameter                        Value
             COPY____
   _________ ________  ____________________________
-----------------------------------------------------------------------------------------
   DBUTWS01   COPY      WORKSTATION                      DBXX
   DBUTWS03   COPY      XINFO_LOCATION                   DBXX
   DBUVIC01   COPY      CHANGELIMIT                      25
   DBUVIC01   COPY      CHANGELIMIT_MIN_PAGES            100000
   DBUVIC01   COPY      DEFAULT_SHRLEVEL                 CHANGE
   DBUVIC01   COPY      JOB_PREFIX                       DBAUC
   DBUVIC01   COPY      LOG_MAP                          OFF
   DBUVIC01   COPY      MAX_INCR                         5
   DBUVIC01   COPY      MAX_PART_PR_JOB                  100
   …
```

# Calendar – When can we run jobs ?

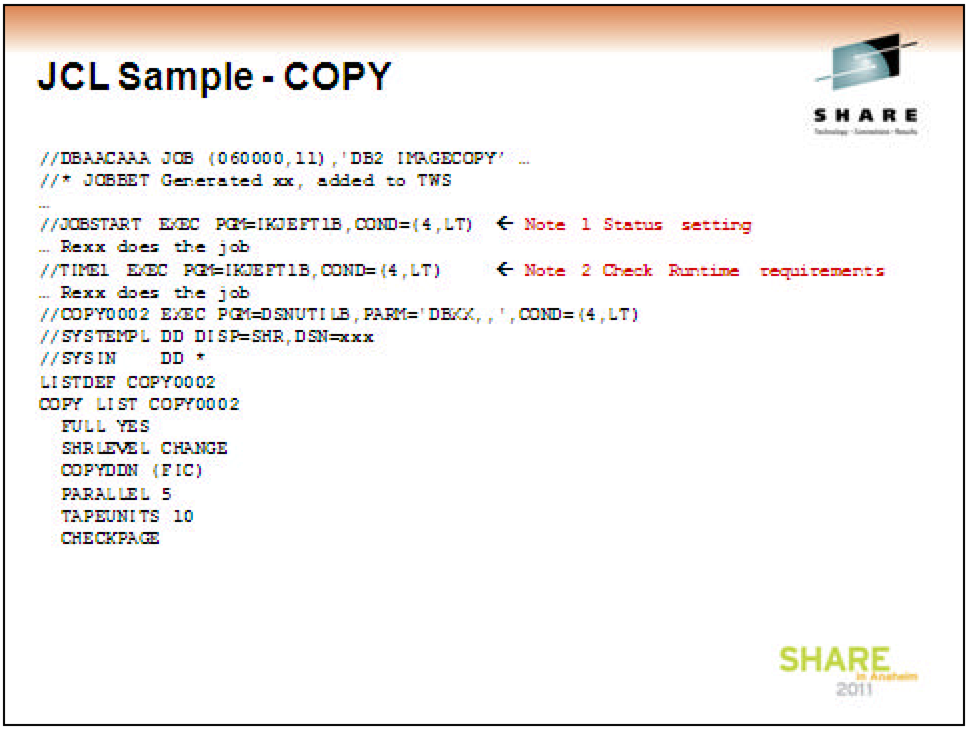


```
DBXX                     DB2 Utility Control - Calendar               Row 1 to 4 of 4
 Command ===>                                                          Scroll ===> CSR

   Line Commands:
    X = FULL STOP - all utility-jobs stop at next step
    R = Remove FULL STOP

 . Utility            - - - - - - - -  Working hours  - - - - - - - - - - -   FULL
                      Mon      Tues     Wed      Thurs    Fri      Sat      Sun      STOP
 -----------------------------------------------------------------------------------------
   COPY      Start: 00.00    00.00    00.00    00.00    00.00    03.00    _____     N
             Stop : 08.00    08.00    08.00    08.00    08.00    _____    _____
   DSN1COMP  Start: 00.00    00.00    00.00    00.00    00.00    03.00    _____     N
             Stop : 08.00    08.00    08.00    08.00    08.00    _____    _____
   REORG     Start: 02.00    02.00    02.00    02.00    02.00    02.00    _____     N
             Stop : 08.00    08.00    08.00    08.00    08.00    _____    _____
   RUNSTATS  Start: 17.00    17.00    17.00    17.00    17.00    07.00    _____     N
             Stop : 20.00    20.00    20.00    20.00    20.00    _____    _____
```

# JCL Generated by the system for TWS



```
EDIT      XXXXXX.TWS.DB2UTIL                              Row 00001 of 06205
Command ===>                                                Scroll ===> PAGE
   Name      Prompt            Size    Created           Changed           ID
. DBAACAAA                      411   2010/01/04   2010/01/04 01:00:57   TWSUSER
. DBAACAAB                      382   2010/01/04   2010/04/01 01:02:09   TWSUSER
…
. DBAACAAK                      311   2010/01/04   2010/01/04 01:02:21   TWSUSER
. DBAACAAL                      235   2010/01/04   2010/01/04 01:02:22   TWSUSER
. DBAACAAM                      311   2010/01/04   2010/01/04 01:02:22   TWSUSER
…
```



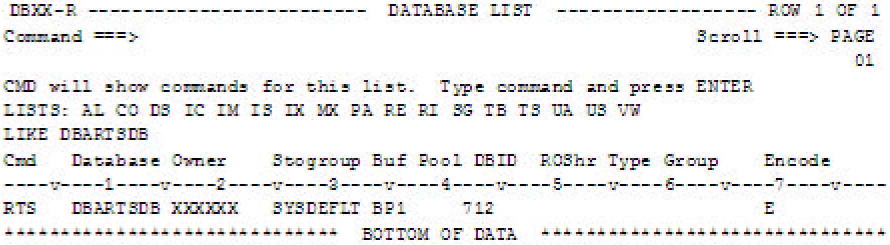
## JCL Sample - COPY

```
//DBAACAAA JOB (060000,11),'DB2 IMAGECOPY' …
//* JOBBET Generated xx, added to TWS
…
//JOBSTART  EXEC  PGM=IKJEFT1B,COND=(4,LT)  ← Note 1 Status  setting
… Rexx does the job
//TIME1  EXEC  PGM=IKJEFT1B,COND=(4,LT)     ← Note 2 Check Runtime  requirements
… Rexx does the job
//COPY0002 EXEC PGM=DSNUTILB,PARM='DBXX,,',COND=(4,LT)
//SYSTEMPL DD DISP=SHR,DSN=xxx
//SYSIN    DD *
LISTDEF COPY0002
COPY LIST COPY0002
  FULL YES
  SHRLEVEL CHANGE
  COPYDDN (FIC)
  PARALLEL 5
  TAPEUNITS 10
  CHECKPAGE
```

JOBSTART set status for Job Started in the utility-system

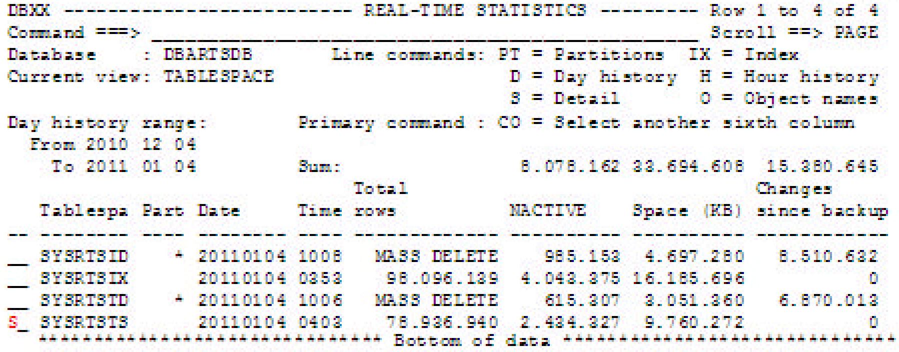
TIME1 checks for Full Stop and that threshholds are OK

# Viewing RTS-data (Home-grown) (from BMC Catalog Manager)



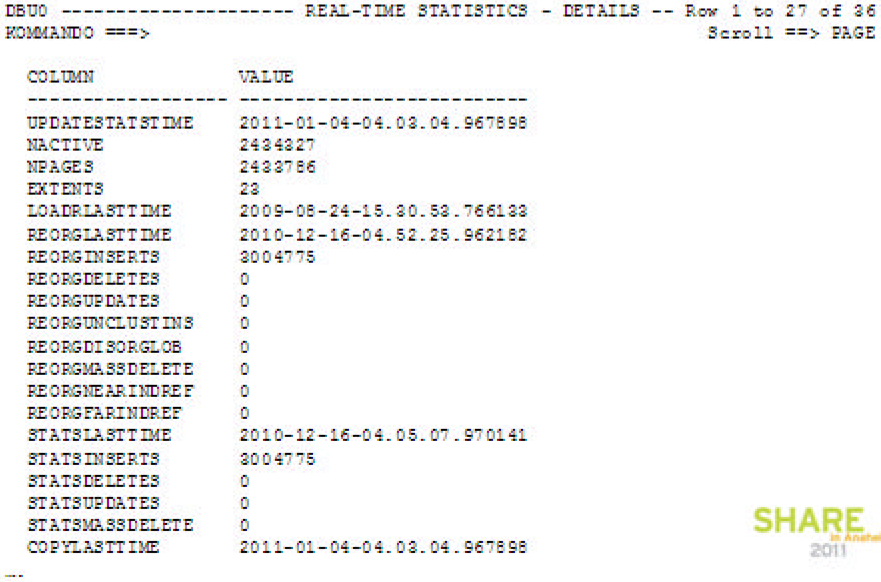
```
 DBXX-R ------------------------  DATABASE LIST  ----------------- ROW 1 OF 1
Command ===>                                                Scroll ===> PAGE
                                                                          01
CMD will show commands for this list.  Type command and press ENTER
LISTS: AL CO DS IC IM IS IX MX PA RE RI SG TB TS UA US VW
LIKE DBARTSDB
Cmd   Database Owner    Stogroup Buf Pool DBID  ROShr Type Group    Encode
----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v----
RTS   DBARTSDB XXXXXX   SYSDEFLT BP1     712                        E
*****************************  BOTTOM OF DATA  ******************************
```

# Viewing RTS Data



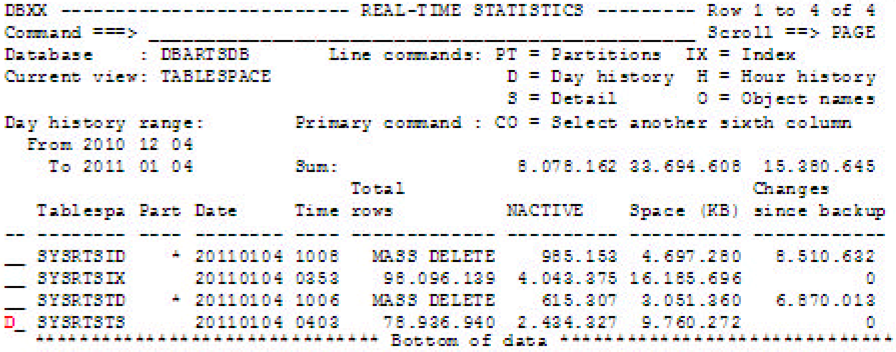
```
DBXX -------------------------- REAL-TIME STATISTICS --------- Row 1 to 4 of 4
Command ===> _________________________________________________ Scroll ==> PAGE
Database    : DBARTSDB       Line commands: PT = Partitions  IX = Index
Current view: TABLESPACE                     D = Day history  H = Hour history
                                             S = Detail       O = Object names
Day history range:        Primary command : CO = Select another sixth column
  From 2010 12 04
    To 2011 01 04         Sum:                8.078.162 33.694.608  15.380.645
                               Total                               Changes
   Tablespa Part Date     Time rows          NACTIVE    Space (KB) since backup
-- -------- ---- -------- ---- ------------- ---------- ---------- ------------
__ SYSRTSID    * 20110104 1008   MASS DELETE    985.153  4.697.280    8.510.632
__ SYSRTSIX      20110104 0353    98.096.139  4.043.375 16.185.696            0
__ SYSRTSTD    * 20110104 1006   MASS DELETE    615.307  3.051.360    6.870.013
S_ SYSRTSTS      20110104 0403    78.936.940  2.434.327  9.760.272            0
   ******************************* Bottom of data ******************************
```

# Viewing RTS Data – Table space Detail



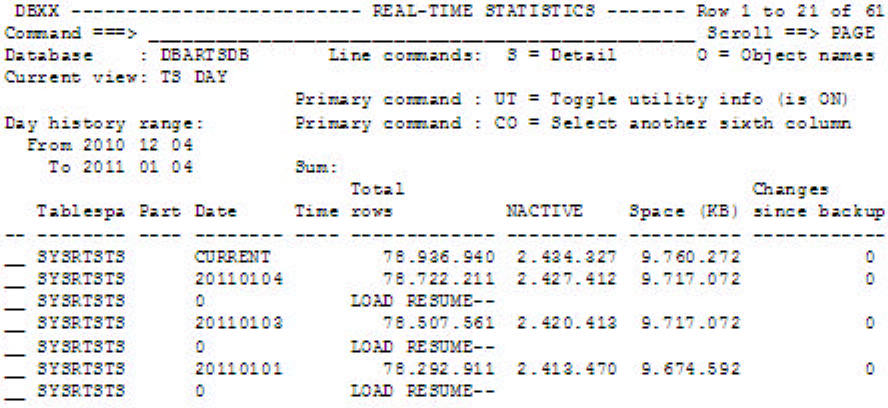
```
DBU0 --------------------- REAL-TIME STATISTICS - DETAILS -- Row 1 to 27 of 36
KOMMANDO ===>                                                   Scroll ==> PAGE

  COLUMN             VALUE
  ------------------ --------------------------
  UPDATESTATSTIME    2011-01-04-04.03.04.967898
  NACTIVE            2434327
  NPAGES             2433786
  EXTENTS            23
  LOADRLASTTIME      2009-08-24-15.30.53.766133
  REORGLASTTIME      2010-12-16-04.52.25.962182
  REORGINSERTS       3004775
  REORGDELETES       0
  REORGUPDATES       0
  REORGUNCLUSTINS    0
  REORGDISORGLOB     0
  REORGMASSDELETE    0
  REORGNEARINDREF    0
  REORGFARINDREF     0
  STATSLASTTIME      2010-12-16-04.05.07.970141
  STATSINSERTS       3004775
  STATSDELETES       0
  STATSUPDATES       0
  STATSMASSDELETE    0
  COPYLASTTIME       2011-01-04-04.03.04.967898
```

# Viewing RTS Data



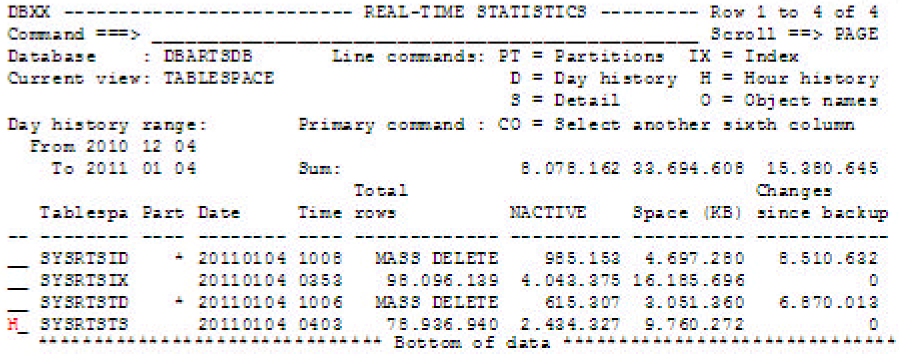
```
DBXX --------------------------- REAL-TIME STATISTICS --------- Row 1 to 4 of 4
Command ===> __________________________________________________ Scroll ==> PAGE
Database    : DBARTSDB        Line commands: PT = Partitions  IX = Index
Current view: TABLESPACE                      D = Day history  H = Hour history
                                              S = Detail       O = Object names
Day history range:         Primary command : CO = Select another sixth column
  From 2010 12 04
    To 2011 01 04          Sum:                 8.078.162 33.694.608  15.380.645
                                Total                                Changes
   Tablespa Part Date     Time rows          NACTIVE    Space (KB) since backup
-- -------- ---- -------- ---- ------------- ---------- ---------- ------------
__ SYSRTSID    * 20110104 1008   MASS DELETE    985.153  4.697.280   8.510.632
__ SYSRTSIX      20110104 0353    98.096.139  4.043.375 16.185.696            0
__ SYSRTSTD    * 20110104 1006   MASS DELETE    615.307  3.051.360   6.870.013
D_ SYSRTSTS      20110104 0403    78.936.940  2.434.327  9.760.272            0
   ******************************* Bottom of data *****************************
```

# Viewing RTS-Data – History Daily



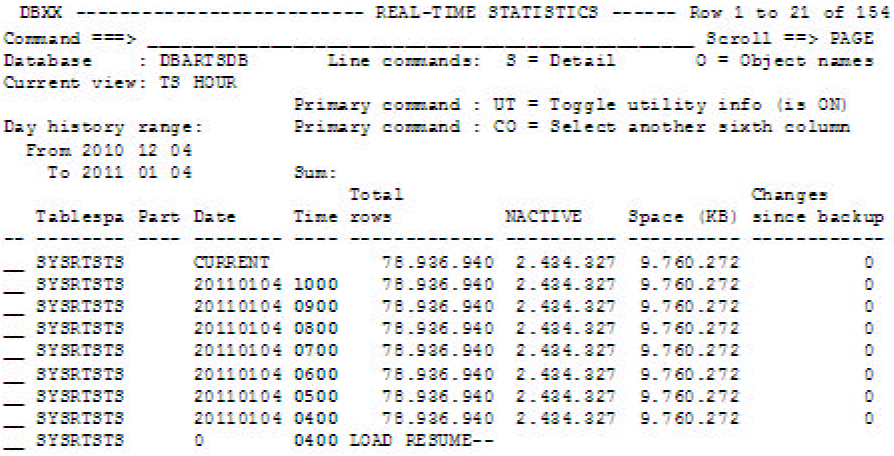
```
 DBXX -------------------------- REAL-TIME STATISTICS ------- Row 1 to 21 of 61
Command ===> ________________________________________________ Scroll ==> PAGE
Database    : DBARTSDB      Line commands:  S = Detail       O = Object names
Current view: TS DAY
                          Primary command : UT = Toggle utility info (is ON)
Day history range:        Primary command : CO = Select another sixth column
  From 2010 12 04
    To 2011 01 04         Sum:
                               Total                                Changes
   Tablespa Part Date     Time rows          NACTIVE    Space (KB) since backup
-- -------- ---- -------- ---- ------------- ---------- ---------- ------------
__ SYSRTSTS      CURRENT          78.936.940  2.434.327  9.760.272            0
__ SYSRTSTS      20110104         78.722.211  2.427.412  9.717.072            0
__ SYSRTSTS      0             LOAD RESUME--
__ SYSRTSTS      20110103         78.507.561  2.420.413  9.717.072            0
__ SYSRTSTS      0             LOAD RESUME--
__ SYSRTSTS      20110101         78.292.911  2.413.470  9.674.592            0
__ SYSRTSTS      0             LOAD RESUME--
```

# Viewing RTS Data

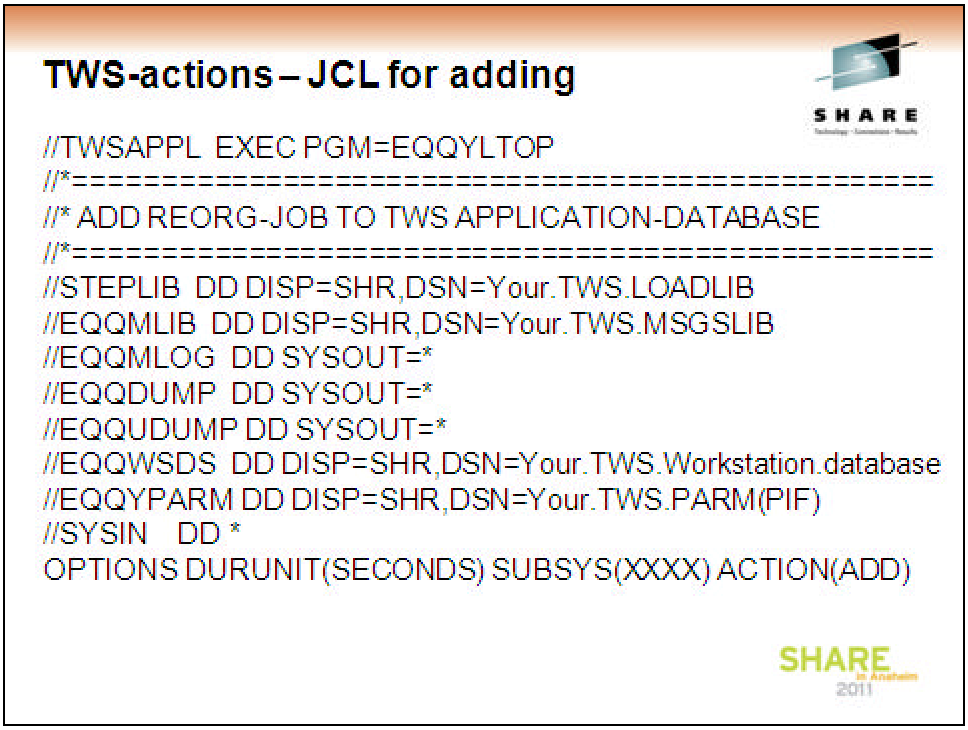


```
DBXX -------------------------- REAL-TIME STATISTICS --------- Row 1 to 4 of 4
Command ===> _________________________________________________ Scroll ===> PAGE
Database    : DBARTSDB       Line commands: PT = Partitions  IX = Index
Current view: TABLESPACE                     D = Day history  H = Hour history
                                             S = Detail       O = Object names
Day history range:        Primary command : CO = Select another sixth column
  From 2010 12 04
    To 2011 01 04         Sum:                8.078.162 33.694.608  15.380.645
                               Total                                Changes
   Tablespa Part Date     Time rows          NACTIVE    Space (KB) since backup
-- -------- ---- -------- ---- ------------- ---------- ---------- ------------
__ SYSRTSID    * 20110104 1008   MASS DELETE    985.153  4.697.280   8.510.632
__ SYSRTSIX      20110104 0353    98.096.139  4.043.375 16.185.696           0
__ SYSRTSTD    * 20110104 1006   MASS DELETE    615.307  3.051.360   6.870.013
H_ SYSRTSTS      20110104 0403    78.936.940  2.434.327  9.760.272           0
   ******************************* Bottom of data ******************************
```

# Viewing RTS Data – Detail hourly



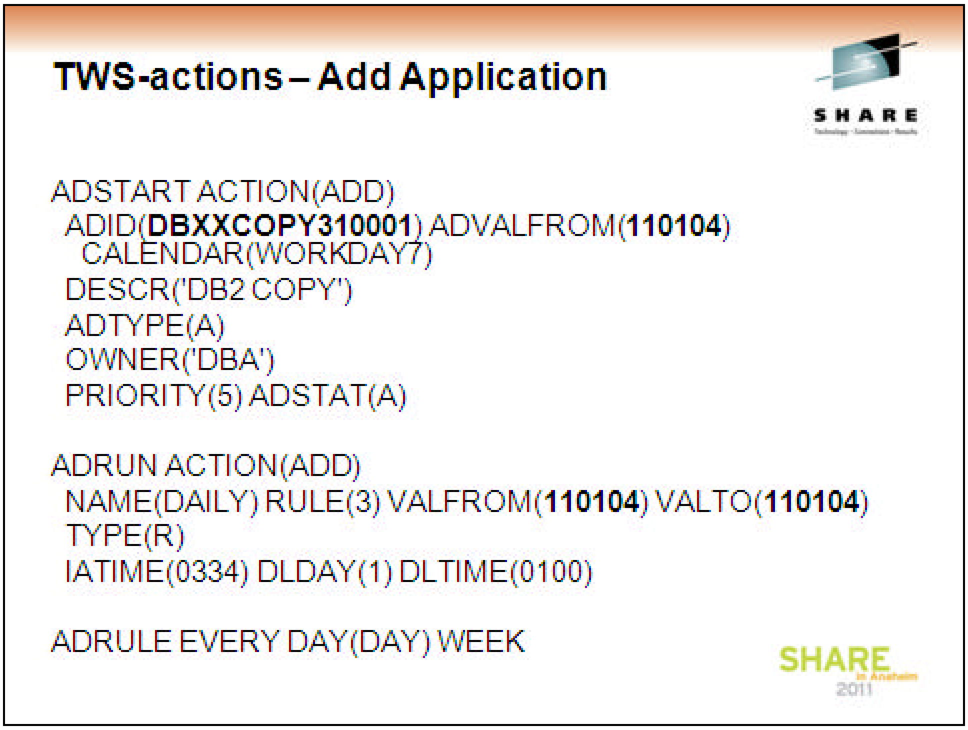
```
 DBXX -------------------------- REAL-TIME STATISTICS ------ Row 1 to 21 of 154
Command ===> ________________________________________________ Scroll ==> PAGE
Database    : DBARTSDB      Line commands:  S = Detail       O = Object names
Current view: TS HOUR
                          Primary command : UT = Toggle utility info (is ON)
Day history range:        Primary command : CO = Select another sixth column
  From 2010 12 04
    To 2011 01 04         Sum:
                               Total                                 Changes
   Tablespa Part Date     Time rows          NACTIVE    Space (KB) since backup
-- -------- ---- -------- ---- ------------- ---------- ---------- ------------
__ SYSRTSTS      CURRENT            78.936.940  2.434.327  9.760.272            0
__ SYSRTSTS      20110104 1000      78.936.940  2.434.327  9.760.272            0
__ SYSRTSTS      20110104 0900      78.936.940  2.434.327  9.760.272            0
__ SYSRTSTS      20110104 0800      78.936.940  2.434.327  9.760.272            0
__ SYSRTSTS      20110104 0700      78.936.940  2.434.327  9.760.272            0
__ SYSRTSTS      20110104 0600      78.936.940  2.434.327  9.760.272            0
__ SYSRTSTS      20110104 0500      78.936.940  2.434.327  9.760.272            0
__ SYSRTSTS      20110104 0400      78.936.940  2.434.327  9.760.272            0
__ SYSRTSTS      0        0400 LOAD RESUME--
```



## TWS-actions – JCL for adding

```
//TWSAPPL  EXEC PGM=EQQYLTOP
//*====================================================
//* ADD REORG-JOB TO TWS APPLICATION-DATABASE
//*====================================================
//STEPLIB  DD DISP=SHR,DSN=Your.TWS.LOADLIB
//EQQMLIB  DD DISP=SHR,DSN=Your.TWS.MSGSLIB
//EQQMLOG  DD SYSOUT=*
//EQQDUMP  DD SYSOUT=*
//EQQUDUMP DD SYSOUT=*
//EQQWSDS  DD DISP=SHR,DSN=Your.TWS.Workstation.database
//EQQYPARM DD DISP=SHR,DSN=Your.TWS.PARM(PIF)
//SYSIN    DD *
OPTIONS DURUNIT(SECONDS) SUBSYS(XXXX) ACTION(ADD)
```

The JCL above shows what you have to run in order to define the requirements for the Automated utilities



## TWS-actions – Add Application

```
ADSTART ACTION(ADD)
 ADID(DBXXCOPY310001) ADVALFROM(110104)
   CALENDAR(WORKDAY7)
 DESCR('DB2 COPY')
 ADTYPE(A)
 OWNER('DBA')
 PRIORITY(5) ADSTAT(A)

ADRUN ACTION(ADD)
 NAME(DAILY) RULE(3) VALFROM(110104) VALTO(110104)
 TYPE(R)
 IATIME(0334) DLDAY(1) DLTIME(0100)

ADRULE EVERY DAY(DAY) WEEK
```

**ADSTART**

DBXX is the Datasharing group

COPY is the Utilityname

31 is the day-number

0001 is the sequence number

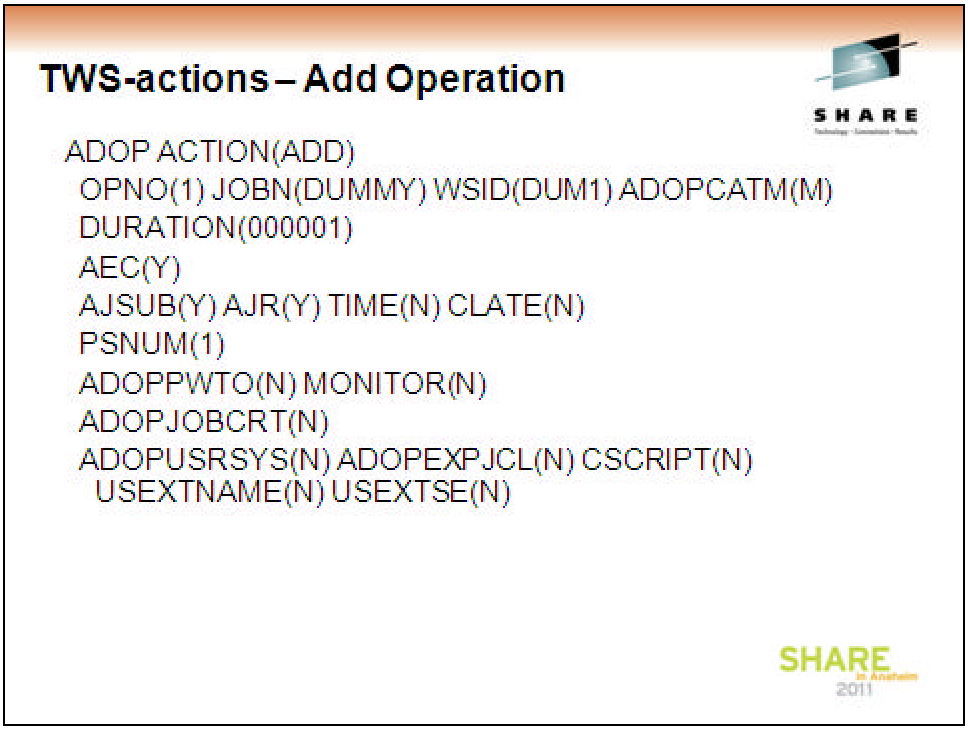
080731 is the actual date from which the application is valid

**ADRUN**

When do the application run from/to which is defined as 1 day by the system

**ADRULE**

Runs every day of the week



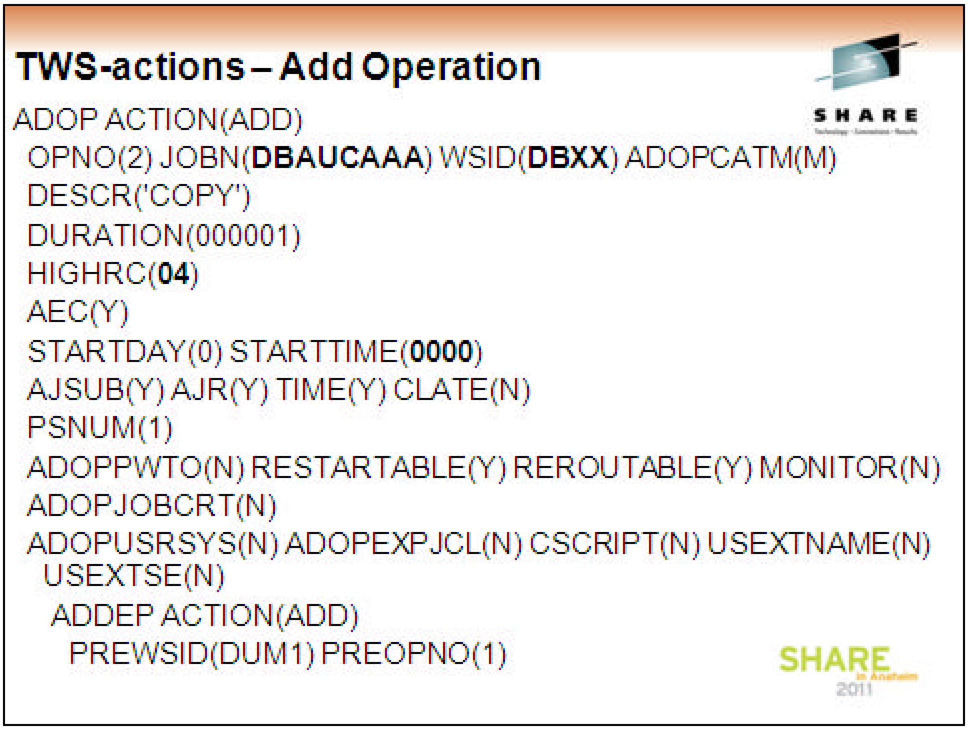
## TWS-actions – Add Operation

```
ADOP ACTION(ADD)
  OPNO(1) JOBN(DUMMY) WSID(DUM1) ADOPCATM(M)
  DURATION(000001)
  AEC(Y)
  AJSUB(Y) AJR(Y) TIME(N) CLATE(N)
  PSNUM(1)
  ADOPPWTO(N) MONITOR(N)
  ADOPJOBCRT(N)
  ADOPUSRSYS(N) ADOPEXPJCL(N) CSCRIPT(N)
    USEXTNAME(N) USEXTSE(N)
```

Each job is defined as operations in TWS, where we tell TWS the following

**JOBN** The start operation is always dummy in TWS ;-)

**WSID** The default for the Dummy-operation



## TWS-actions – Add Operation

```
ADOP ACTION(ADD)
 OPNO(2) JOBN(DBAUCAAA) WSID(DBXX) ADOPCATM(M)
 DESCR('COPY')
 DURATION(000001)
 HIGHRC(04)
 AEC(Y)
 STARTDAY(0) STARTTIME(0000)
 AJSUB(Y) AJR(Y) TIME(Y) CLATE(N)
 PSNUM(1)
 ADOPPWTO(N) RESTARTABLE(Y) REROUTABLE(Y) MONITOR(N)
 ADOPJOBCRT(N)
 ADOPUSRSYS(N) ADOPEXPJCL(N) CSCRIPT(N) USEXTNAME(N)
  USEXTSE(N)
   ADDEP ACTION(ADD)
    PREWSID(DUM1) PREOPNO(1)
```

Each job is defined as operations in TWS, where we tell TWS the following

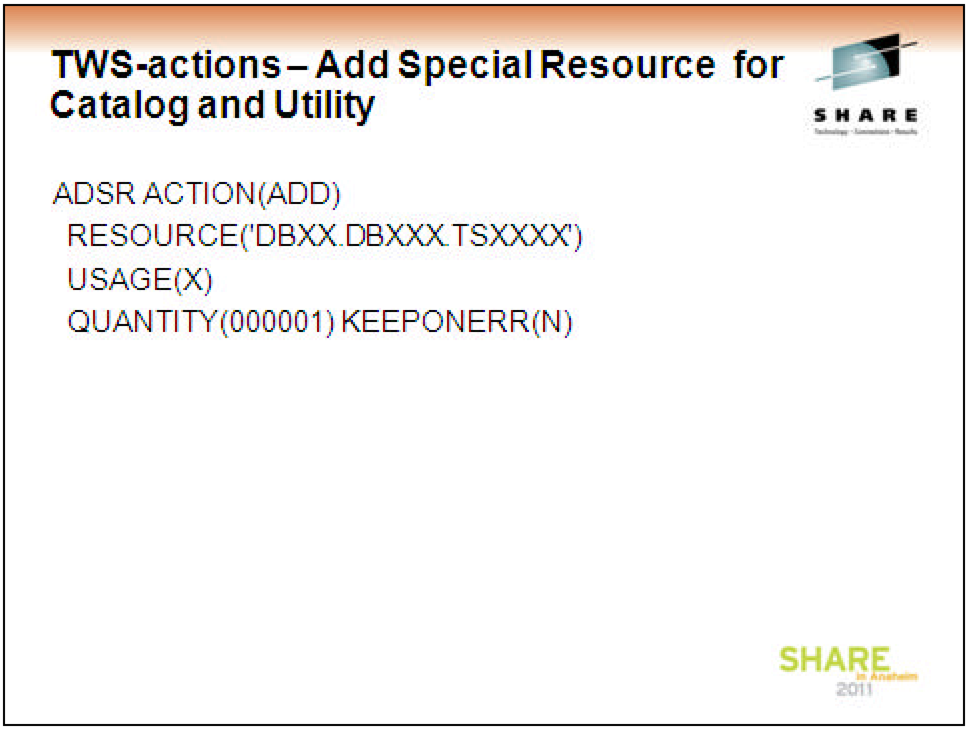
**JOBN** The generated JOBname from the automation system

**WSID** The Datasharing group

**DESCR** REORG defines that this is a reorg Operation

**HIGHRC** Tells TWS that the return Code of 4 is accepted without putting the job on the error queue

**STARTTIME** Tells TWS that this Operation can start at 0000 AM



## TWS-actions – Add Special Resource for Catalog and Utility

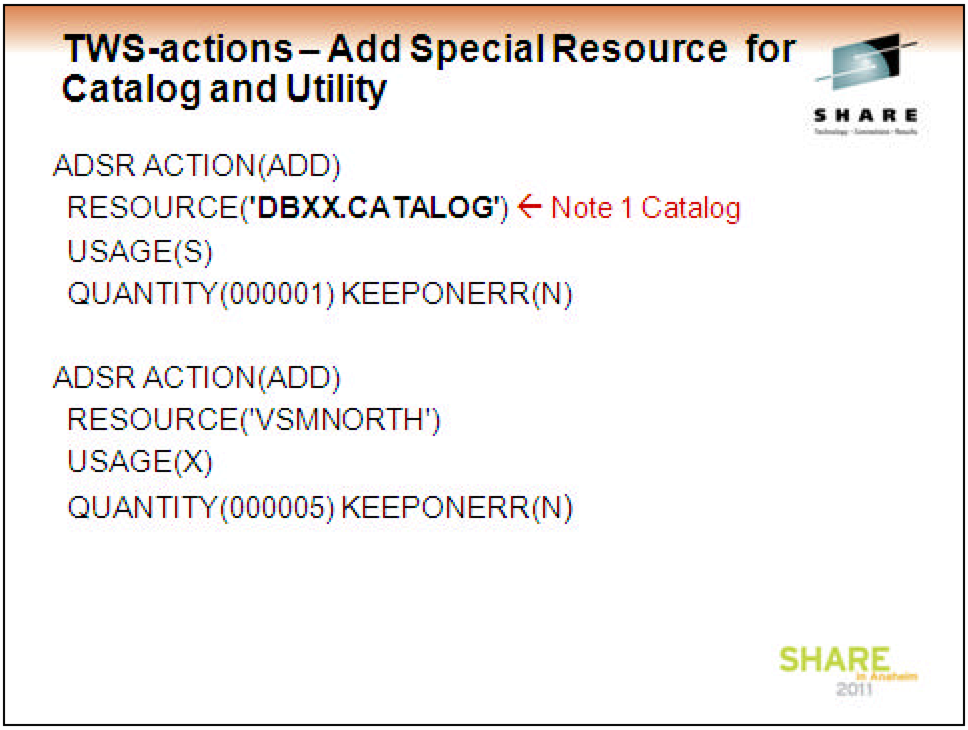
```
ADSR ACTION(ADD)
 RESOURCE('DBXX.DBXXX.TSXXXX')
 USAGE(X)
 QUANTITY(000001) KEEPONERR(N)
```

**ADSR** defines special resources that are used bnoth in this automated system and is added to utilityjobs that run against the same resource

The Resources are build on <Data sharing group name>.<Database name>.<Table space name>

USAGE(X) tells TWS that jobs using this ressource have eXclusive control

If the job fails the ressource is released.(KEEPONERR(N))



## TWS-actions – Add Special Resource for Catalog and Utility

```
ADSR ACTION(ADD)
 RESOURCE('DBXX.CATALOG') ← Note 1 Catalog
 USAGE(S)
 QUANTITY(000001) KEEPONERR(N)

ADSR ACTION(ADD)
 RESOURCE('VSMNORTH')
 USAGE(X)
 QUANTITY(000005) KEEPONERR(N)
```

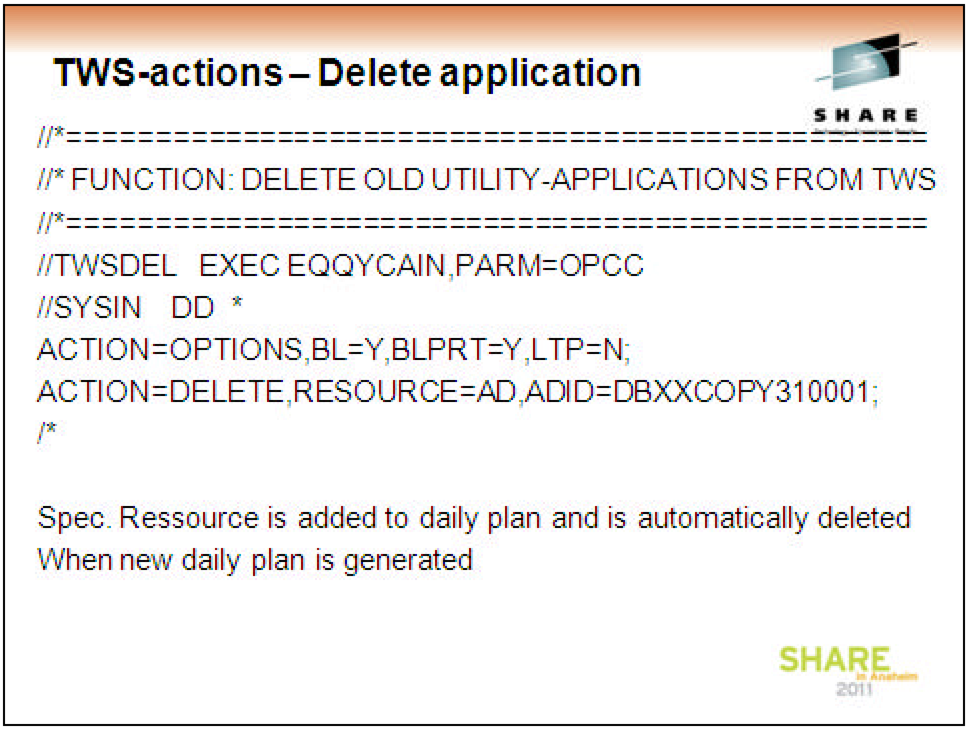
The Ressource DBXX.CATALOG is added ind order to tell TWS that the utilities are allowed to run at the same time as the DB2 catalog utilities are run (normally backup).

But the Catalog utility cannot start when this ressource is in use (it has defined the ressource with USAGE(X)i

The second ADSR tells DB2 that only 5 virtual tapestations can be in use at any given time

Note 1 Catalog

All the Copy jobs can run concurrently, but the DB2 Catalog Utility-jobs can nnot run when a utility-operation is in progress.



## TWS-actions – Delete application

```
//*================================================
//* FUNCTION: DELETE OLD UTILITY-APPLICATIONS FROM TWS
//*================================================
//TWSDEL   EXEC EQQYCAIN,PARM=OPCC
//SYSIN    DD  *
ACTION=OPTIONS,BL=Y,BLPRT=Y,LTP=N;
ACTION=DELETE,RESOURCE=AD,ADID=DBXXCOPY310001;
/*
```

Spec. Ressource is added to daily plan and is automatically deleted
When new daily plan is generated

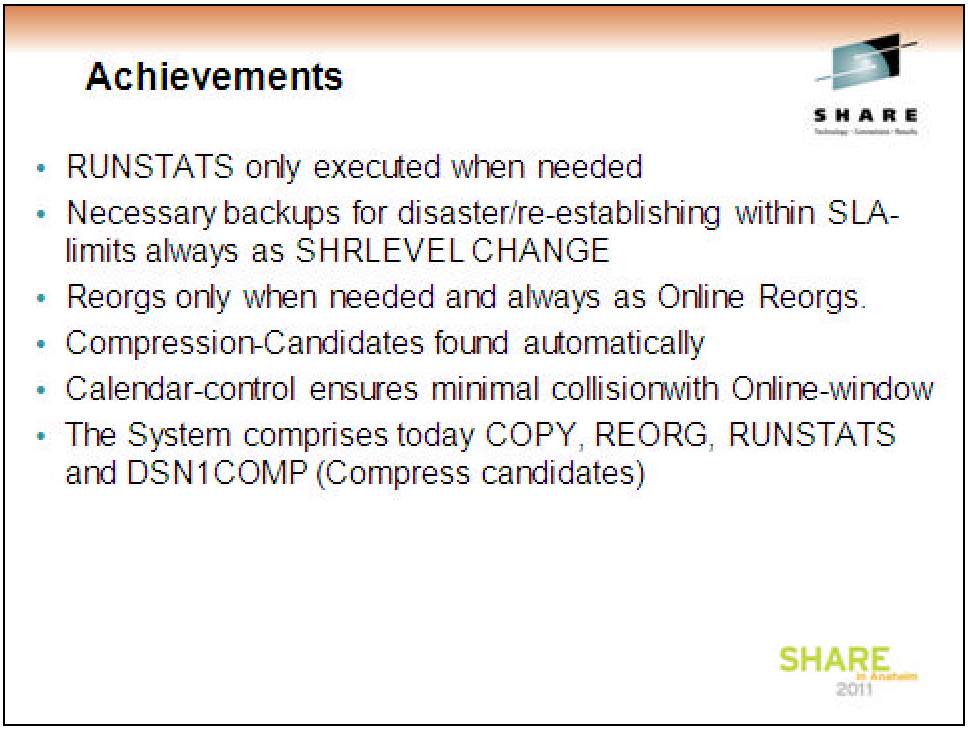
The Ressource DBXX.CATALOG is added ind order to tell TWS that the utilities are allowed to run at the same time as the DB2 catalog utilities are run (normally backup).

But the Catalog utility cannot start when this ressource is in use (it has defined the ressource with USAGE(X)i

The second ADSR tells DB2 that only 5 virtual tapestations can be in use at any given time

Note 1 Catalog

All the Copy jobs can run concurrently, but the DB2 Catalog Utility-jobs can nnot run when a utility-operation is in progress.



## Achievements

- RUNSTATS only executed when needed
- Necessary backups for disaster/re-establishing within SLA-limits always as SHRLEVEL CHANGE
- Reorgs only when needed and always as Online Reorgs.
- Compression-Candidates found automatically
- Calendar-control ensures minimal collisionwith Online-window
- The System comprises today COPY, REORG, RUNSTATS and DSN1COMP (Compress candidates)

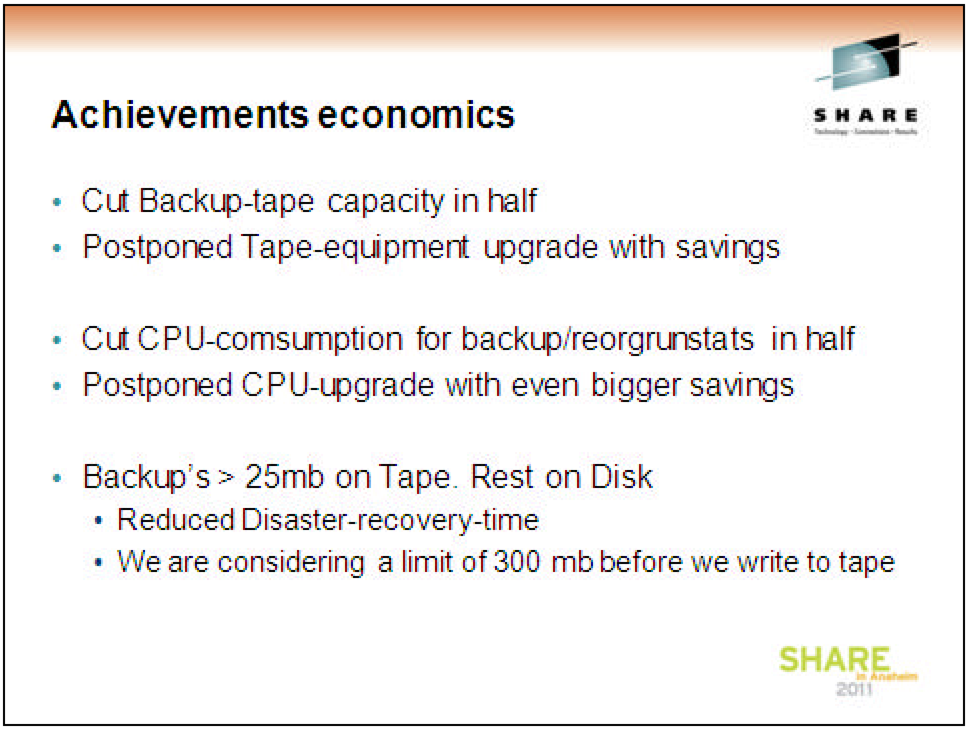
By automating all utility-operation all new databases and table spaces are automatically handled by this system and is without having to buuild separate jobflows etc.

Runstats are applied only when activity in the RTS-tables tells the system, that it is required

Reorg’s are based on activity and parameters telling that data are disorganised

Compression-candidates are spotted based on size

By going from FULL COPY YES to FULL COPY NO (Incremental) we have saved a lot of backup-space

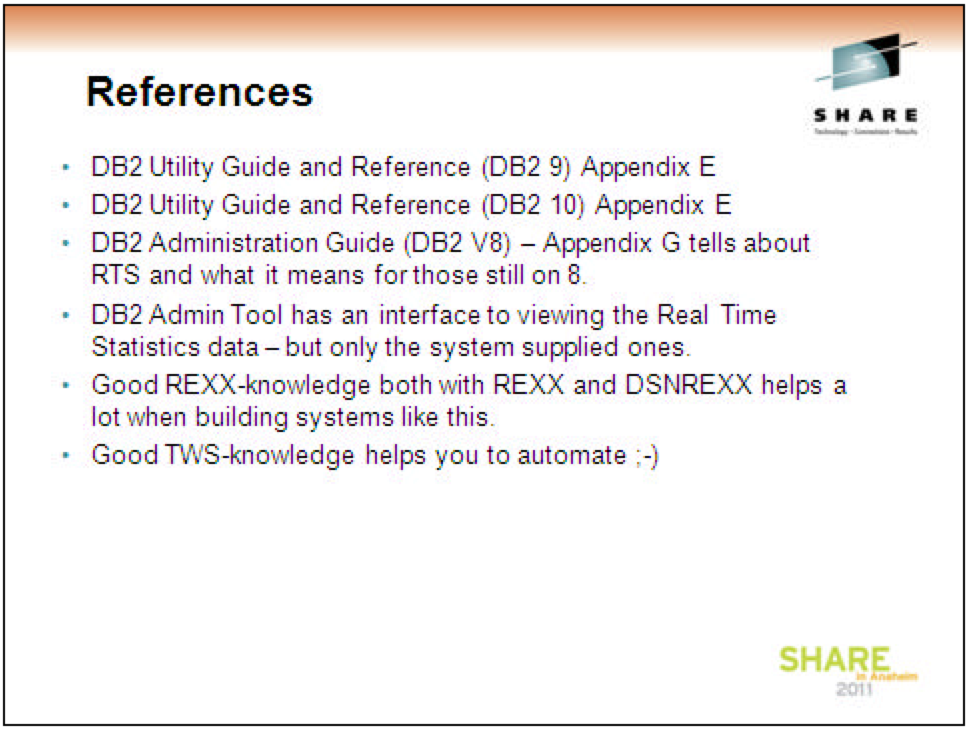


## Achievements economics

- Cut Backup-tape capacity in half
- Postponed Tape-equipment upgrade with savings

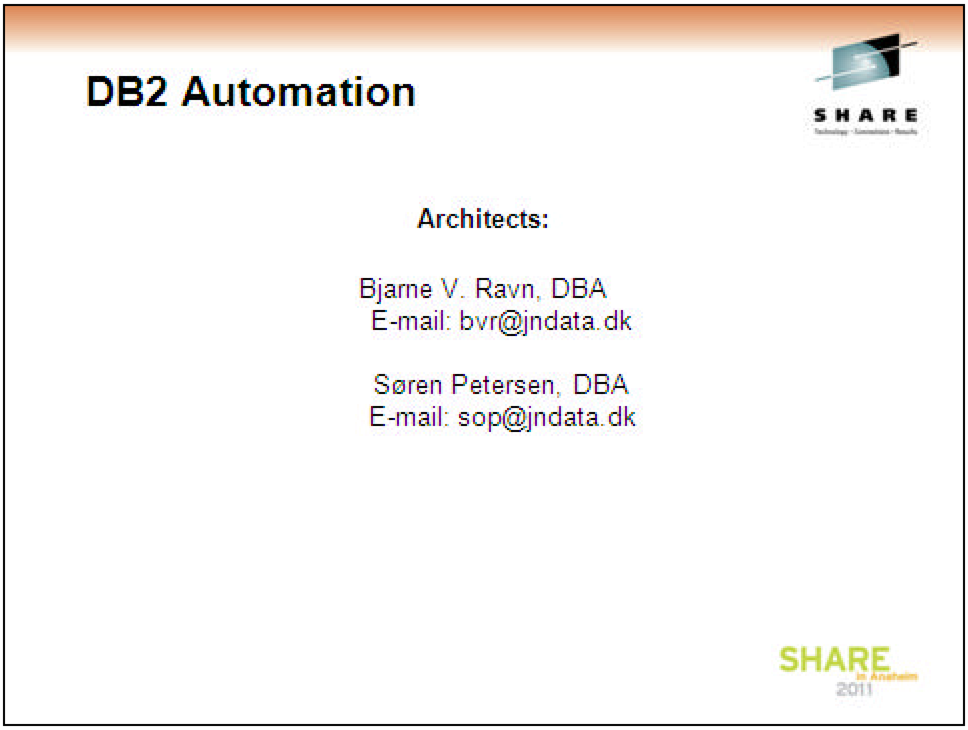
- Cut CPU-comsumption for backup/reorgrunstats in half
- Postponed CPU-upgrade with even bigger savings

- Backup's > 25mb on Tape. Rest on Disk
  - Reduced Disaster-recovery-time
  - We are considering a limit of 300 mb before we write to tape



## References

- DB2 Utility Guide and Reference (DB2 9) Appendix E
- DB2 Utility Guide and Reference (DB2 10) Appendix E
- DB2 Administration Guide (DB2 V8) – Appendix G tells about RTS and what it means for those still on 8.
- DB2 Admin Tool has an interface to viewing the Real Time Statistics data – but only the system supplied ones.
- Good REXX-knowledge both with REXX and DSNREXX helps a lot when building systems like this.
- Good TWS-knowledge helps you to automate ;-)



# DB2 Automation

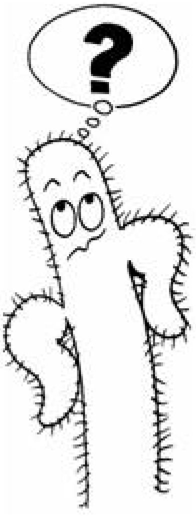
**Architects:**

Bjarne V. Ravn, DBA
E-mail: bvr@jndata.dk

Søren Petersen, DBA
E-mail: sop@jndata.dk

SHARE
Technology · Connections · Results
SHARE
in Anaheim
2011

# Questions ?



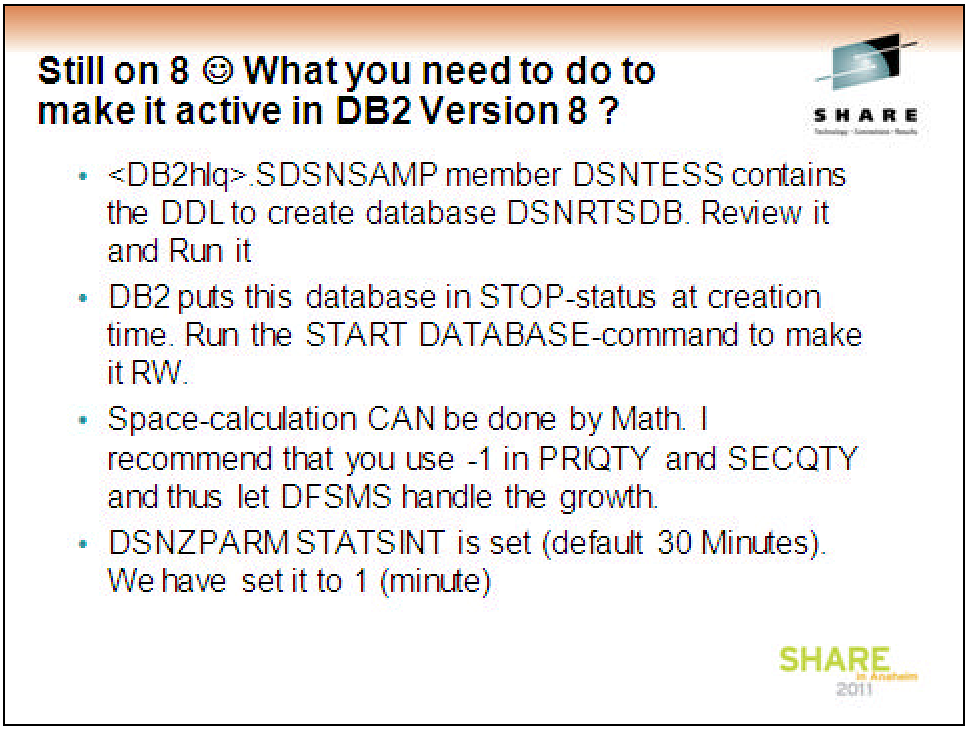
## Implementing Real Time statistics for utility automation

**Svenn-Aage Sønderskov**

JN Data A/S, Denmark

sas@jndata.dk

SHARE in Anaheim 2011

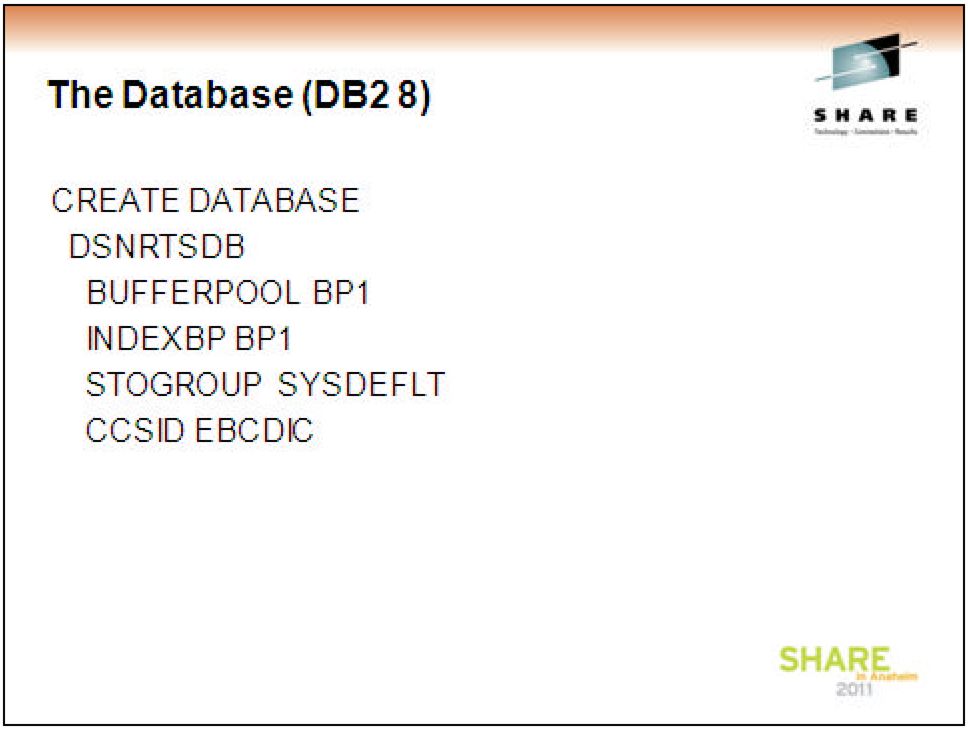


## Still on 8 ☺ What you need to do to make it active in DB2 Version 8 ?

- <DB2hlq>.SDSNSAMP member DSNTESS contains the DDL to create database DSNRTSDB. Review it and Run it
- DB2 puts this database in STOP-status at creation time. Run the START DATABASE-command to make it RW.
- Space-calculation CAN be done by Math. I recommend that you use -1 in PRIQTY and SECQTY and thus let DFSMS handle the growth.
- DSNZPARM STATSINT is set (default 30 Minutes). We have set it to 1 (minute)

The Installation SDSNSAMP-library contains the basic RTS-definitions.

If you want to have shadow tablesets for viewing growth etc you have to build them yourself. You can yuse the table-ddl from tha basic ones and add a timestamp which you set when you load from the basic table

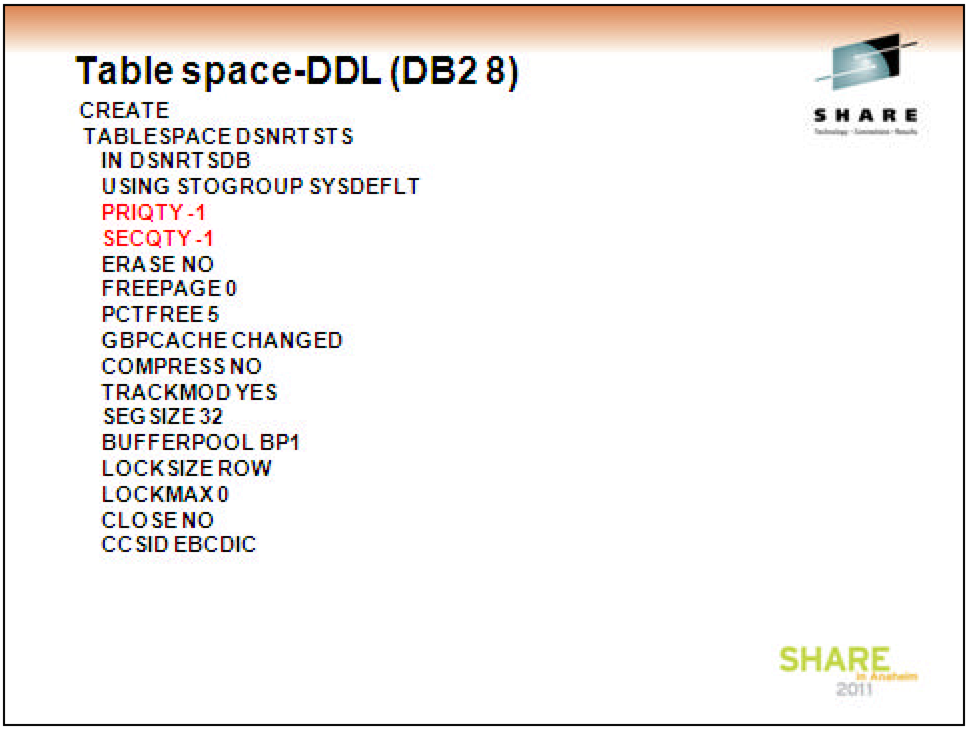


## The Database (DB2 8)

```
CREATE DATABASE
  DSNRTSDB
    BUFFERPOOL BP1
    INDEXBP BP1
    STOGROUP  SYSDEFLT
    CCSID EBCDIC
```

The Above DDL is what it takes to define the database that comprises the Real Time Statistics tablespaces/tables and indexes

Use another Bufferpool than the catalog's  (we use BP0 for tha Catalog)

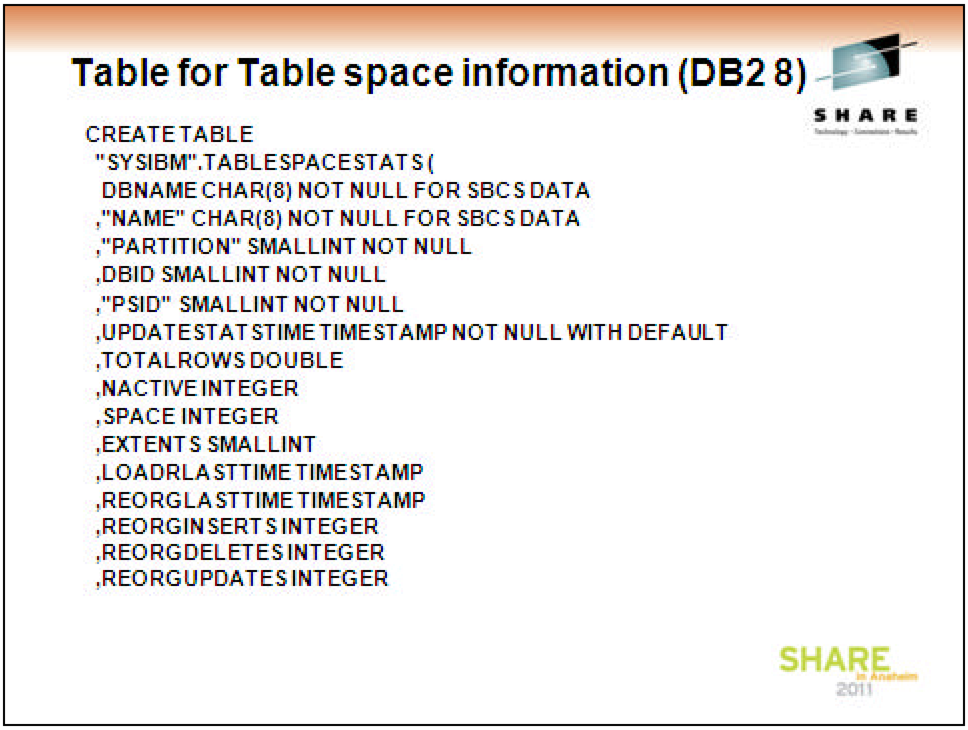


## Table space-DDL (DB2 8)

```
CREATE
 TABLESPACE DSNRTSTS
   IN DSNRTSDB
   USING STOGROUP SYSDEFLT
   PRIQTY -1
   SECQTY -1
   ERASE NO
   FREEPAGE 0
   PCTFREE 5
   GBPCACHE CHANGED
   COMPRESS NO
   TRACKMOD YES
   SEGSIZE 32
   BUFFERPOOL BP1
   LOCKSIZE ROW
   LOCKMAX 0
   CLOSE NO
   CCSID EBCDIC
```

The Real Time Statistics tables are defined in a segmented tablespace with Segsize 32

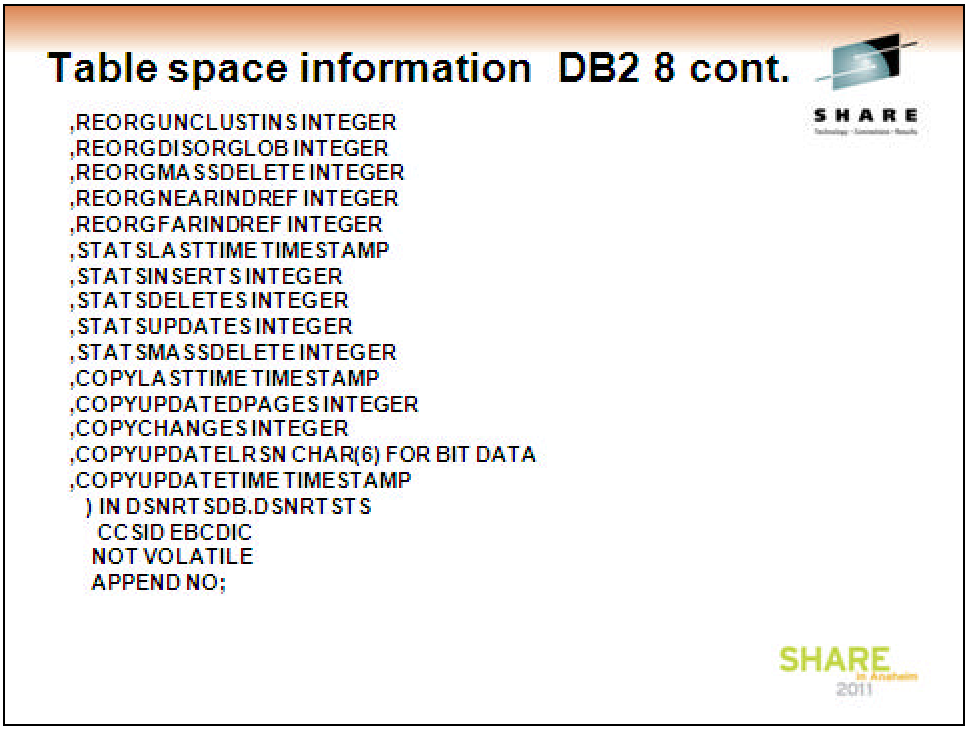
We use automatic space management, where DB2 and SMS decides on the growth this is indicated as the -1 in bothe SECQTY and PRIQTY



## Table for Table space information (DB2 8)

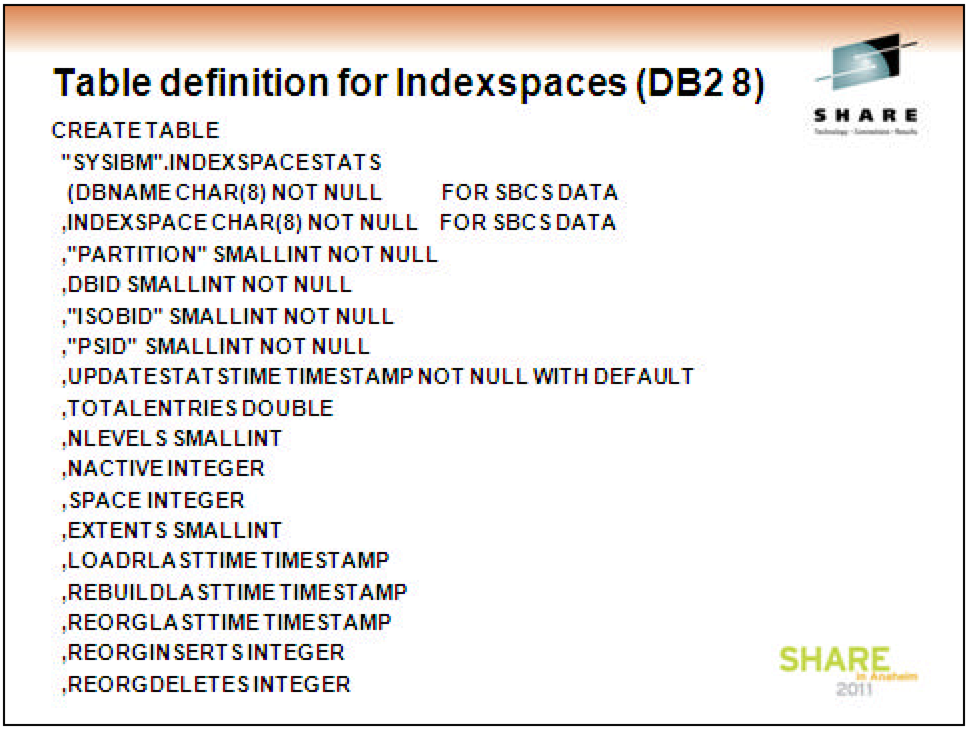
```
CREATE TABLE
 "SYSIBM".TABLESPACESTATS (
  DBNAME CHAR(8) NOT NULL FOR SBCS DATA
 ,"NAME" CHAR(8) NOT NULL FOR SBCS DATA
 ,"PARTITION" SMALLINT NOT NULL
 ,DBID SMALLINT NOT NULL
 ,"PSID" SMALLINT NOT NULL
 ,UPDATESTATSTIME TIMESTAMP NOT NULL WITH DEFAULT
 ,TOTALROWS DOUBLE
 ,NACTIVE INTEGER
 ,SPACE INTEGER
 ,EXTENTS SMALLINT
 ,LOADRLASTTIME TIMESTAMP
 ,REORGLASTTIME TIMESTAMP
 ,REORGINSERTS INTEGER
 ,REORGDELETES INTEGER
 ,REORGUPDATES INTEGER
```

The Above DDL and the continuation on thje next foil is the definition for the Tablespace information table



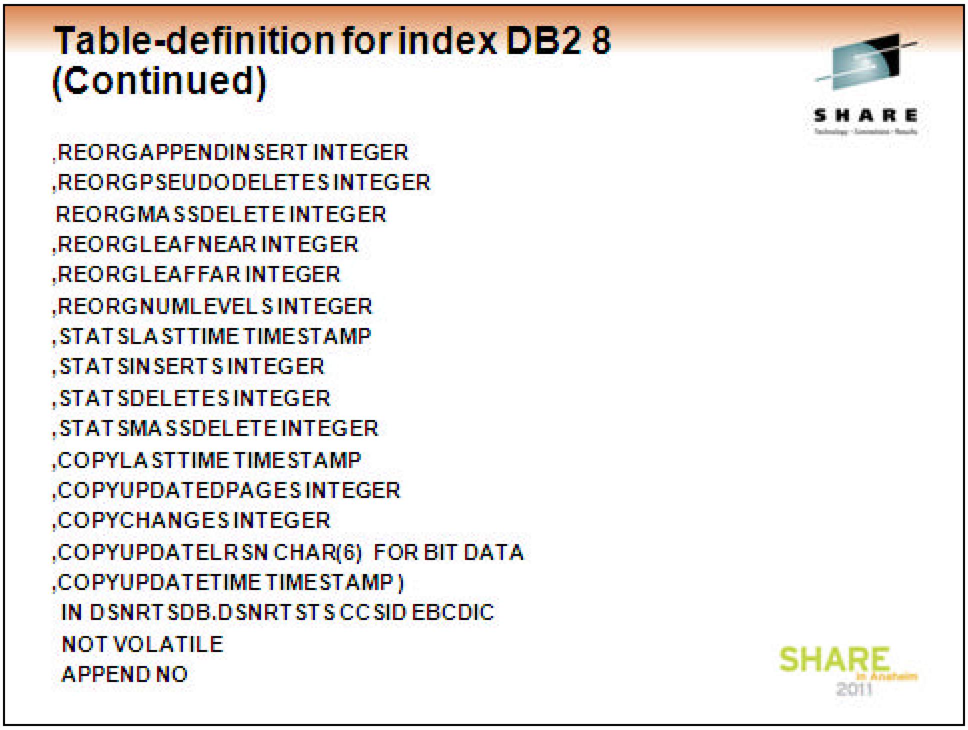
## Table space information  DB2 8 cont.

```
,REORGUNCLUSTINS INTEGER
,REORGDISORGLOB INTEGER
,REORGMASSDELETE INTEGER
,REORGNEARINDREF INTEGER
,REORGFARINDREF INTEGER
,STATSLASTTIME TIMESTAMP
,STATSINSERTS INTEGER
,STATSDELETES INTEGER
,STATSUPDATES INTEGER
,STATSMASSDELETE INTEGER
,COPYLASTTIME TIMESTAMP
,COPYUPDATEDPAGES INTEGER
,COPYCHANGES INTEGER
,COPYUPDATELRSN CHAR(6) FOR BIT DATA
,COPYUPDATETIME TIMESTAMP
  ) IN DSNRTSDB.DSNRTSTS
    CCSID EBCDIC
   NOT VOLATILE
   APPEND NO;
```



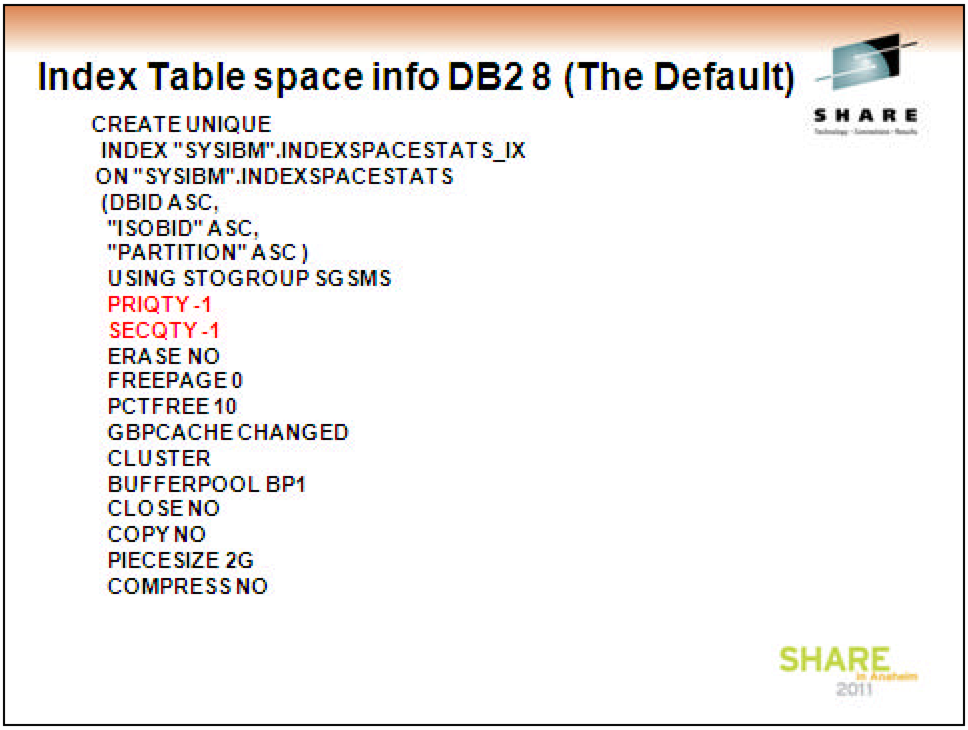
## Table definition for Indexspaces (DB2 8)

```
CREATE TABLE
 "SYSIBM".INDEXSPACESTATS
  (DBNAME CHAR(8) NOT NULL        FOR SBCS DATA
 ,INDEXSPACE CHAR(8) NOT NULL    FOR SBCS DATA
 ,"PARTITION" SMALLINT NOT NULL
 ,DBID SMALLINT NOT NULL
 ,"ISOBID" SMALLINT NOT NULL
 ,"PSID" SMALLINT NOT NULL
 ,UPDATESTATSTIME TIMESTAMP NOT NULL WITH DEFAULT
 ,TOTALENTRIES DOUBLE
 ,NLEVELS SMALLINT
 ,NACTIVE INTEGER
 ,SPACE INTEGER
 ,EXTENTS SMALLINT
 ,LOADRLASTTIME TIMESTAMP
 ,REBUILDLASTTIME TIMESTAMP
 ,REORGLASTTIME TIMESTAMP
 ,REORGINSERTS INTEGER
 ,REORGDELETES INTEGER
```



## Table-definition for index DB2 8 (Continued)

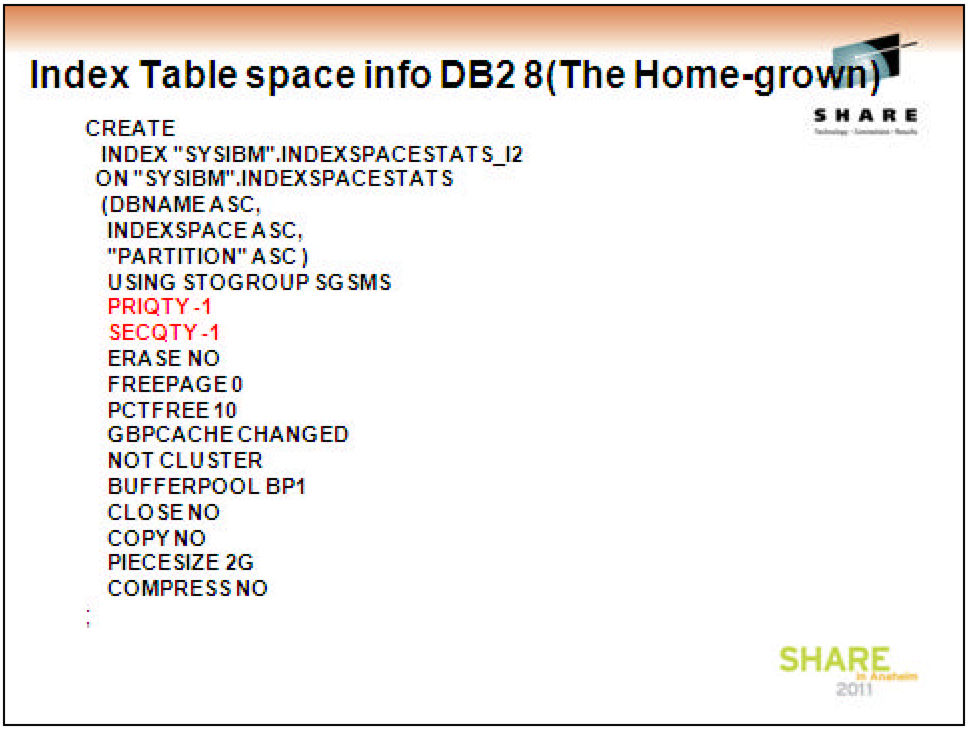
```
,REORGAPPENDINSERT INTEGER
,REORGPSEUDODELETES INTEGER
 REORGMASSDELETE INTEGER
,REORGLEAFNEAR INTEGER
,REORGLEAFFAR INTEGER
,REORGNUMLEVELS INTEGER
,STATSLASTTIME TIMESTAMP
,STATSINSERTS INTEGER
,STATSDELETES INTEGER
,STATSMASSDELETE INTEGER
,COPYLASTTIME TIMESTAMP
,COPYUPDATEDPAGES INTEGER
,COPYCHANGES INTEGER
,COPYUPDATELRSN CHAR(6)  FOR BIT DATA
,COPYUPDATETIME TIMESTAMP )
 IN DSNRTSDB.DSNRTSTS CCSID EBCDIC
 NOT VOLATILE
 APPEND NO
```



## Index Table space info DB2 8 (The Default)

```
CREATE UNIQUE
 INDEX "SYSIBM".INDEXSPACESTATS_IX
 ON "SYSIBM".INDEXSPACESTATS
 (DBID ASC,
  "ISOBID" ASC,
  "PARTITION" ASC )
  USING STOGROUP SG SMS
  PRIQTY -1
  SECQTY -1
  ERASE NO
  FREEPAGE 0
  PCTFREE 10
  GBPCACHE CHANGED
  CLUSTER
  BUFFERPOOL BP1
  CLOSE NO
  COPY NO
  PIECESIZE 2G
  COMPRESS NO
```

This index the default one



## Index Table space info DB2 8(The Home-grown)

```
CREATE
  INDEX "SYSIBM".INDEXSPACESTATS_I2
 ON "SYSIBM".INDEXSPACESTATS
  (DBNAME ASC,
   INDEXSPACE ASC,
   "PARTITION" ASC )
   USING STOGROUP SG SMS
   PRIQTY -1
   SECQTY -1
   ERASE NO
   FREEPAGE 0
   PCTFREE 10
   GBPCACHE CHANGED
   NOT CLUSTER
   BUFFERPOOL BP1
   CLOSE NO
   COPY NO
   PIECESIZE 2G
   COMPRESS NO
;
```

We have build our own index which gives us the oppertunity to use the names in stead of the numbers for searching the RTS-tables

The Indexspace-table and indexes are build over the same kind of template.