IBM System z Technology Summit

Charles Lewis – How to Streamline Your DB2 for z/OS Utility Processing

January 27, 2011

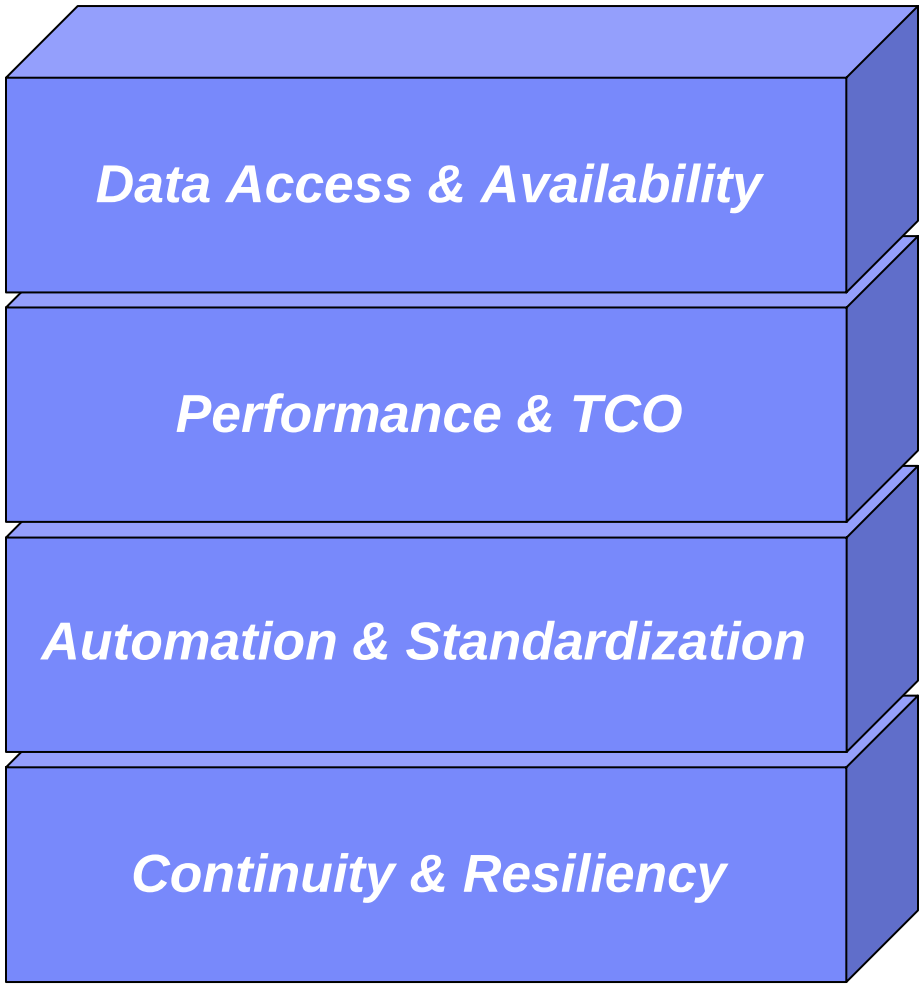
Topics

- **Trends in Database/Utility Management**
- **IBM's Investment in Utility Management**
  - New Products
  - New Features
  - Day 1 support for new releases of DB2 for z/OS
  - Combined technology of products
- **Detailed Examples**
  - Avoid REORGs Using the DB2 10 for z/OS Automation Tool Exceptions
  - New DB2 Utilities Enhancement Tool Utility Syntax Monitor feature
  - DB2 10 for z/OS Online REORG of LOBs Using DB2 Automation Tool
- **Questions**

Trends in Database/Utility Management

- Data growth puts pressure on IT infrastructure, SLAs, staff, and performance
- According to IDC, the amount of data is exploding. Structured data is growing 32% per year, unstructured data is growing 63% and replicated data is growing 49%. Companies are compelled to take the right steps to protect their valuable data and maintain high database availability
  - Average data growth per year is approximately 30%
  - Large critical application data growth rate is > 50%
- In the last 10 years the number of objects needing performance management has increased:
  - The number of objects that need management has increased 3X
  - the number of objects per DBA has increased 4X
- Focus on reducing CPU and elapsed time
- Running multiple databases on a server has become the norm
- 90% of customers have more than one DBMS ➔ Resource/skill issues, consistent administration efforts, increased cost in administration, greater need to automate routine operations

IBM Investment Areas for Managing Utilities



- **Data Access & Availability**
  - Fast retrieval of information
  - Reducing the amount of down time or minimizing batch window for maintenance
- **Performance & TCO**
  - Meeting or exceeding SLA's and/or chargeback
  - Reducing CPU and ET to achieve lowest TCO
- **Automation & Standardization**
  - Reducing repeated tasks, manual effort and error
  - Ensuring consistency at company level
- **Continuity & Resiliency**
  - Ensuring data integrity
  - Ensuring Day-1 support of new versions of DB2 for z/OS

How is the investment realized by IBM in the Utility Management space?

- **New Products that take advantage of existing investment to provide options for those customers with special needs**
  - Ex: DB2 Sort for z/OS
- **New features in existing products that address pain points for customers**
  - Ex: Utility Syntax Monitor in DB2 Utilities Enhancement Tool
- **Day 1 support for new releases of DB2 for z/OS**
  - Ex: DB2 10 for z/OS
- **Combining strengths of existing products to take advantage of new features**
  - Ex: REORG enhancements with Automation Tool
  - Ex: LOAD Presort with DB2 Utilities Enhancement Tool and DB2 Sort

DB2 Sort for z/OS v1.1

- DB2 Sort provides high speed utility sort processing for data stored in DB2 for z/OS. It improves sort performance while optimizing overall system efficiency by exploiting the advanced facilities of the z/OS operating system and System z.
- DB2 Sort leverages the strengths of the System z platform, DB2 for z/OS and the DB2 Utilities Suite to drive:
  - Significant savings in elapsed time and CPU during utility sort processing, especially LOAD, REORG and RUNSTATS
  - Relief from application constraints of large volumes of data in highly-transactional workloads performing numerous insert, update and delete operations against DB2 for z/OS databases
  - Continued commitment from IBM to deliver DB2 solutions to provide the highest level of ROI
- DB2 Sort provides an alternative, high performance sort engine that can be utilized by the DB2 Utilities

DB2 Sort Performance

- **Customers using DB2 Sort V1.1* may see during certain utility sort processing:**
  - Up to 30% in reduction of elapsed time
  - Up to 50% reduction of CPU
  - Up to 30% zIIP offload of remaining CPU
- **IBM DB2 Utilities where you'll see performance benefits**
  - LOAD, REORG, RUNSTATS, REBUILD INDEX, CHECK INDEX and CHECK DATA
- **Workloads where there is more likely to be a benefit from utility sort processing and DB2 Sort V1.1, such as:**
  - Highly-transactional workloads performing lots of insert, update, delete operations requiring RUNSTATS and REORG
  - Applications that are performing frequent or large volumes of loading data requiring LOAD and REBUILD INDEX

**Customer results may vary. Results based on analysis done at SVL lab*

Benefits of DB2 Sort v1.1

- Will provide relief if you
  - Have large amounts of data
  - Have utility batch window constraints
  - Have to execute utility maintenance during peak business hours that may affect elapsed time and/or CPU
  - Have Sort Capacity Exceeded problems running utilities
  - Have purchased utilities from ISVs, requiring
    - Paying for multiple sets of utilities
    - Managing multiple sets of utilities
- Once installed and enabled, is used by all utility sorting
- Requires no changes to utility jobs
- Improves/reduces resource consumption for single and parallel sorts
- Can result in higher degree of utility parallelism
- Gives greater resilience with respect to inaccurate sort estimates

Monitoring Utility Syntax

- **DB2 Utilities Enhancement Tool provides new Utility Syntax Monitor**
  - Can establish and enforce company-wide utility syntax practices
  - IT staff can control who executes which IBM DB2 utilities with what parameters on which objects
  - Can also purposely fail utility if rules are violated
  - Supports DB2 V8, DB2 9 and DB2 10 Utility Syntax

- **Delivered via PTF**
  - UK60173 for all versions of DB2 for z/OS

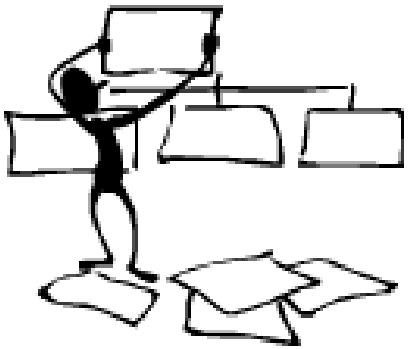
Support for DB2 10 for z/OS and More!

- **All DB2 Utilities and Utility Management tools provided Day 1 DB2 10 for z/OS support**
  - DB2 Utilities Suite
  - High Performance Unload for DB2
  - DB2 Automation Tool
  - DB2 Sort
  - DB2 Utilities Enhancement Tool

- **Example: DB2 10 support of FlashCopy support in COPY, RECOVER, REORG, LOAD and REBUILD INDEX**
  - Ex: Can now run COPY utility online, create an image copy data set to be transaction-consistent image copy data set w/ NO application outage!
    - Keeps DB2 applications available and reduces CPU

Did you know?

- **Hash support in DB2 10 enhanced by support in Utilities/Utility Management Tool**
  - REORG immediately after table converted to/out of hash format
  - Maximizing DB2 Sort and DB2 Utilities Enhancement Tool (UET) by using PRESORT option in LOAD utility (via APAR PM22685)
    - Ability to presort data based on hash key
    - Results in much faster LOAD using fewer system resources
    - Improves application availability
- **REORG Force Option and DB2 UET**
  - DB2 UET can add parameter to your utility via the Utility Syntax Monitor to ensure that online REORGs are always run with Force Option
  - Enhances online REORG use to increase application availability
  - Many customers avoided using online REORG because they could not obtain the necessary drain to process writers in a timely manner. This enhancement in the REORG utility allows the cancellation of the writer just before the switch phase.



DB2 Automation Tool for z/OS

REORG AVOIDANCE WITH DB2 10 for z/OS EXCEPTIONS

Avoid REORGs with DB2 10 for z/OS Automation Tool Exceptions

- **The best REORG is one you don't have to run!**
- **Only REORG what needs it**
  - Which applications' performance are being impacted?
  - What is the cause of frequent REORGs?
- **Re-evaluate thresholds used to determine when REORGs are done**
  - New Exceptions within DB2 Automation Tool assist DBAs in evaluation process
- **Re-evaluate design of database objects**
  - Is your index efficient, and is it being used?
  - Would a hash table be better than an IX?

RUNSTATS versus REALTIME STATS

- **RUNSTATS was used for two things:**
  - To update information for the optimizer
  - To update the DBA on the status of an object
- **REALTIMESTATS helps you instantly know the status of an object, eliminating the need to run RUNSTATS**
- **REALTIMESTATS Exceptions in DB2 Automation Tool include:**

- DAYS_SINCE_HASH
- DATAISMORETHANHASH
- UNCLUST_INS
- UNCLUST_INS_PCT
- CLUSTERSENS
- HASHACCESS
- SCANACCESS
- INDEXACCESS

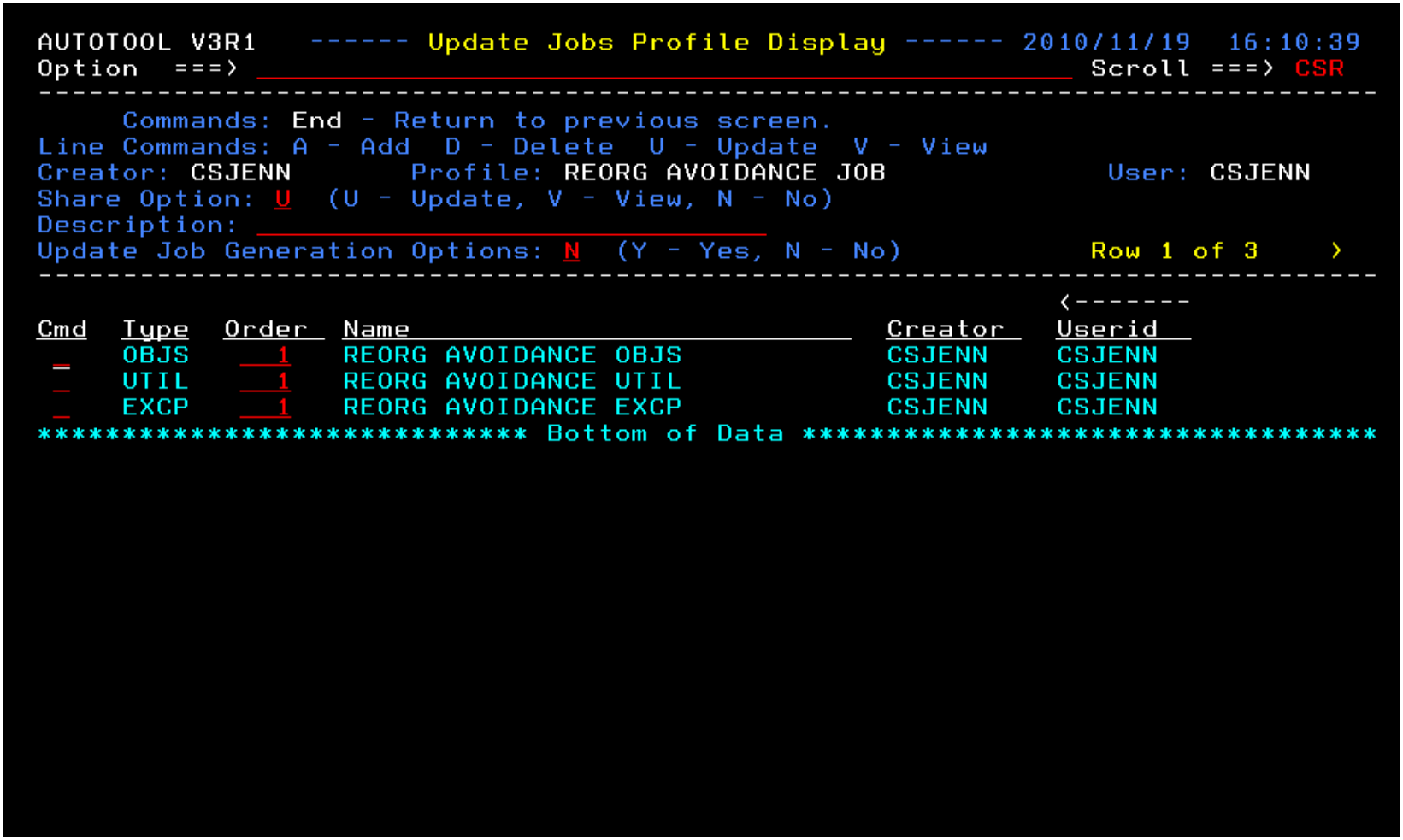
Exception Options to Determine a REORG

- **DAYS_SINCE_HASH**
  - Number of days since hash access was used for SELECT|FETCH|UPDATE|DELETE or used to enforce RI constraints.
- **DATAISMORETHANHASH**
  - Trigger Exception if the DATASIZE is larger than the HASHSPACE for a HASH Organized TS.
- **UNCLUST_INS**
  - The number of inserted records since the last REORG or LOAD REPLACE that were placed more than 16 pages from the ideal candidate page with respect to the clustering IX.
- **UNCLUST_INS_PCT**
  - Percentage of Inserted rows placed > 16 pages away from the ideal candidate page with respect to the clustering IX.

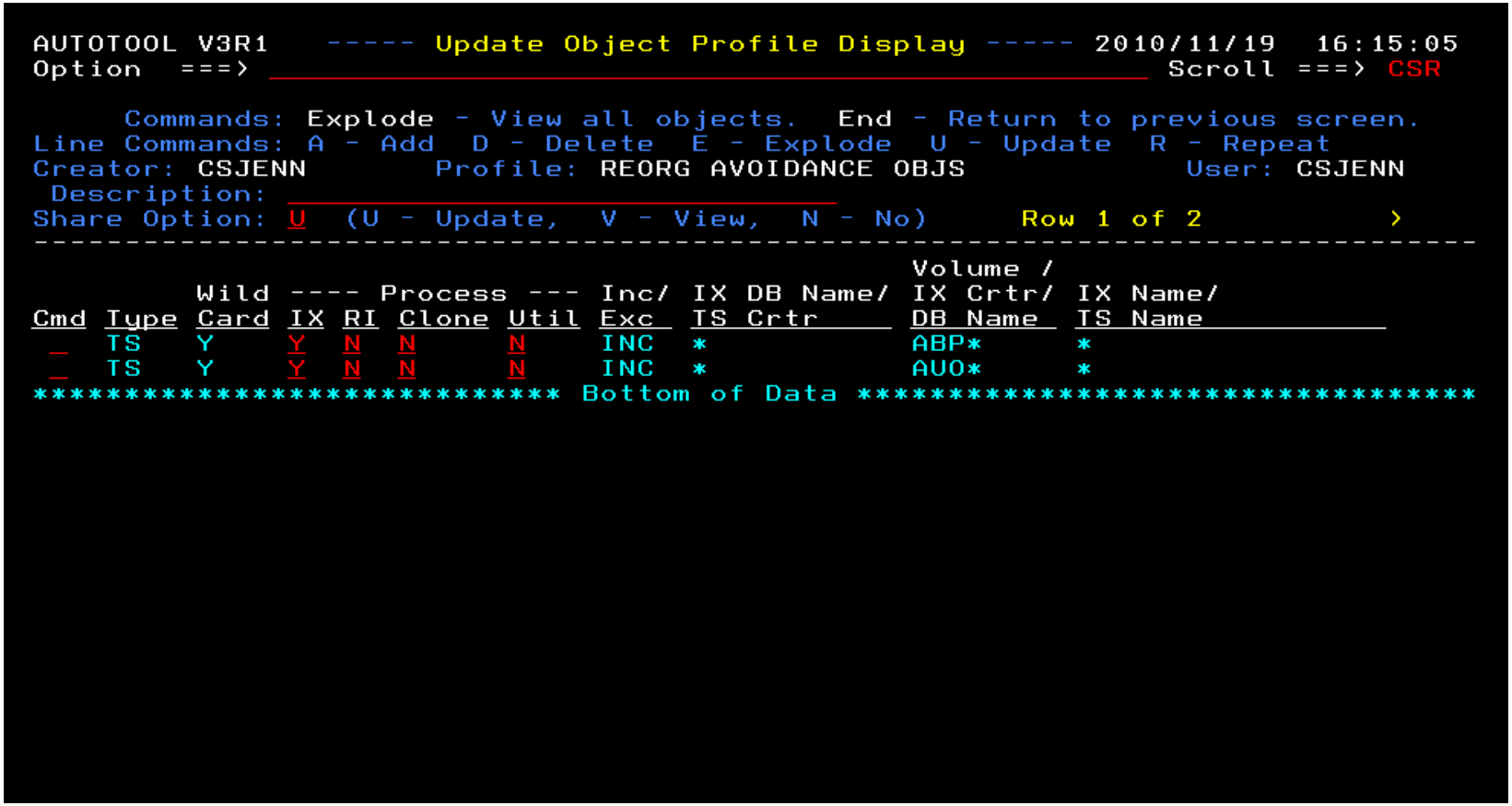
Exception Options to Determine a REORG

- **CLUSTERSENS**
  - The number of times data has been read by SQL statements sensitive to the clustering sequence of the data since the last REORG or LOAD REPLACE.
- **HASHACCESS**
  - Number of times data was accessed using hash access since the last CREATE, LOAD REPLACE or REORG.
- **SCANACCESS**
  - Number of times data was accessed using a TS scan for SELECT, FETCH, searched UPDATE, searched DELETE or used to enforce RI constraints since the last CREATE, LOAD REPLACE or REORG.
- **INDEXACCESS**
  - The number of times the index was used for SELECT, FETCH, searched UPDATE, searched DELETE, or used to enforce RI constraints.

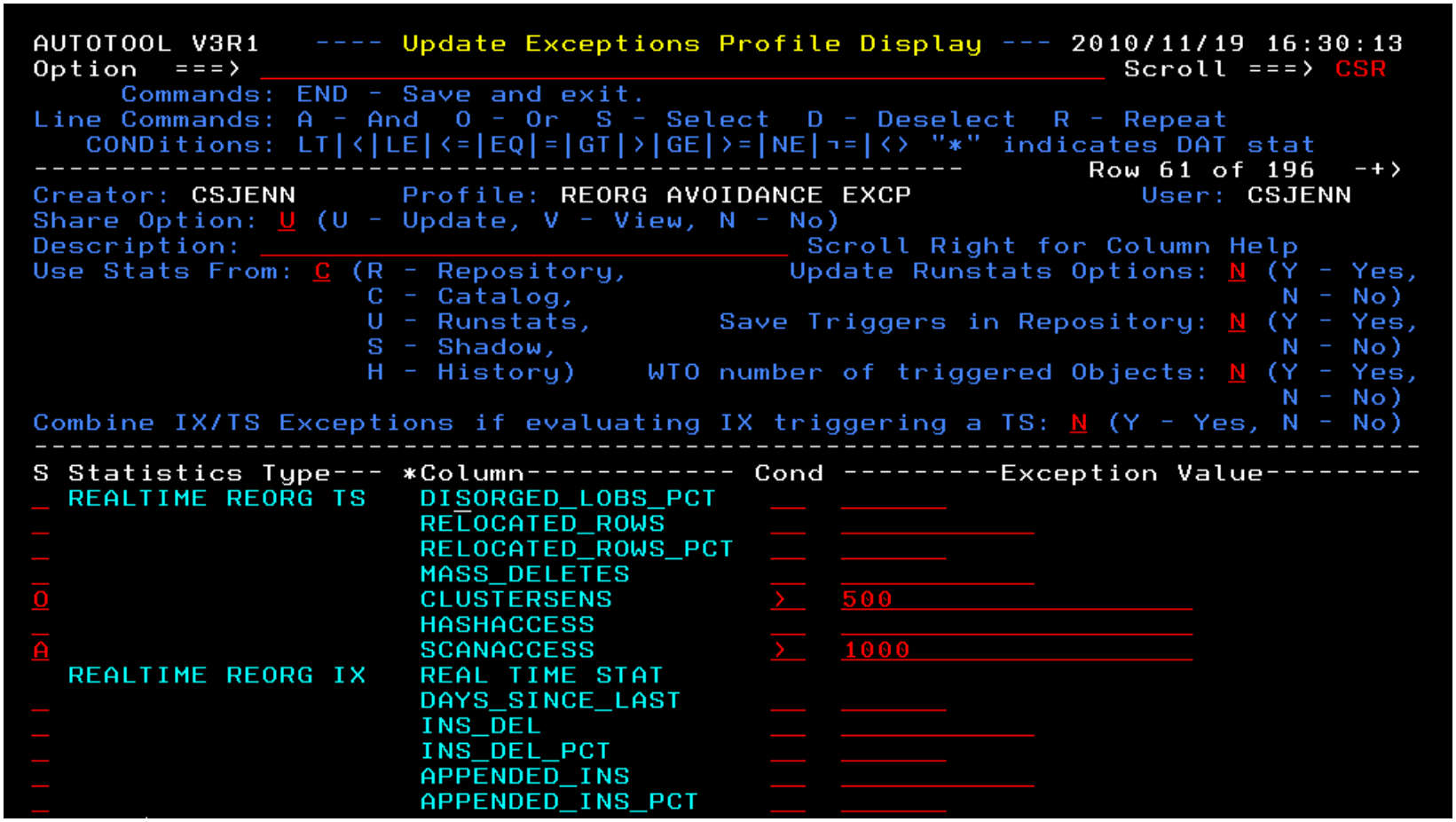
REORG Avoidance Example



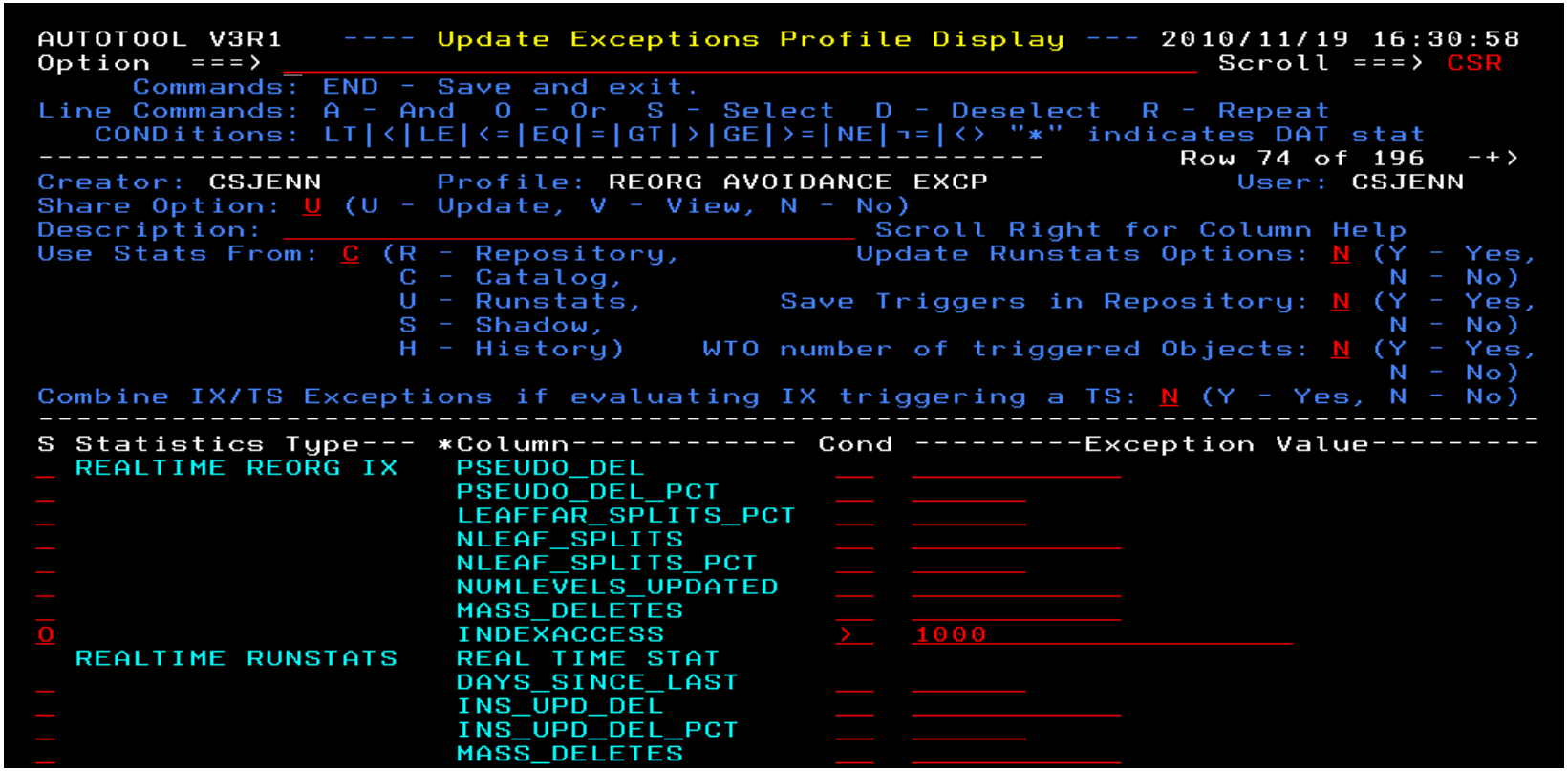
REORG Avoidance Example



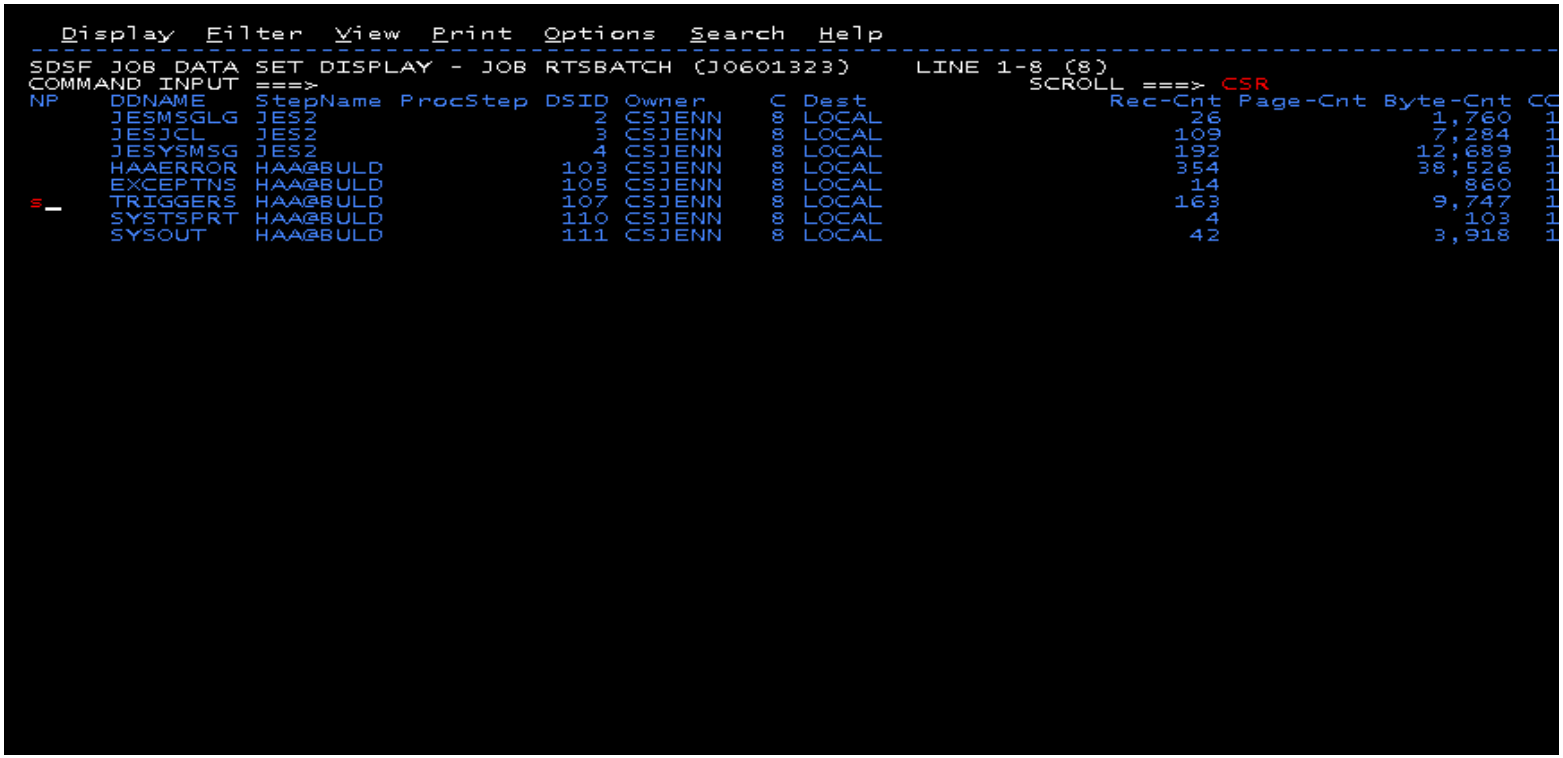
REORG Avoidance Example



REORG Avoidance Example



REORG Avoidance Example



REORG Avoidance Example

```
  Display  Filter  View  Print  Options  Search  Help
-------------------------------------------------------------------------------
SDSF OUTPUT DISPLAY RTSBATCH J0601323  DSID   107 LINE 0       COLUMNS 02- 133
COMMAND INPUT ===>                                          SCROLL ===> CSR
******************************** TOP OF DATA ********************************
1IBM Shared Profile Support -- Print Exception Triggers --  V03.10 Run Date 2010/11/19 Run Time 16:31:53

 26 Triggers created...

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  CLUSTERSENS        BGINT  >   500

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value-
 AUOVRDB  AUOCOPY                                                        0 356030

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  SCANACCESS         BGINT  >   1000

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value-
 AUOVRDB  AUOCOPY                                                        0 368810

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  CLUSTERSENS        BGINT  >   500

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value-
 AUOVRDB  AUODBASE                                                       0 3069275

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  SCANACCESS         BGINT  >   1000

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value-
 AUOVRDB  AUODBASE                                                       0 4792985

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  CLUSTERSENS        BGINT  >   500

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value-
 AUOVRDB  AUOGPAUT                                                       0 895

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  SCANACCESS         BGINT  >   1000

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value-
 AUOVRDB  AUOGPAUT                                                       0 35993
```

REORG Avoidance Example

```
  Display  Filter  View  Print  Options  Search  Help
-------------------------------------------------------------------------------
SDSF OUTPUT DISPLAY RTSBATCH J0601323  DSID   107 LINE 0       COLUMNS 02- 133
COMMAND INPUT ===> f indexaccess                              SCROLL ===> CSR
******************************** TOP OF DATA ************************************
1IBM Shared Profile Support -- Print Exception Triggers --  V03.10 Run Date 2010/11/19 Run Time 16:31:53

 26 Triggers created...

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  CLUSTERSENS        BGINT  >   500

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value--
 AUOVRDB  AUOCOPY                                                        0 356030

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  SCANACCESS         BGINT  >   1000

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value--
 AUOVRDB  AUOCOPY                                                        0 368810

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  CLUSTERSENS        BGINT  >   500

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value--
 AUOVRDB  AUODBASE                                                       0 3069275

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  SCANACCESS         BGINT  >   1000

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value--
 AUOVRDB  AUODBASE                                                       0 4792985

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  CLUSTERSENS        BGINT  >   500

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value--
 AUOVRDB  AUOGPAUT                                                       0 895

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  SCANACCESS         BGINT  >   1000

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value--
 AUOVRDB  AUOGPAUT                                                       0 35993
```

REORG Avoidance Example

```
  Display  Filter  View  Print  Options  Search  Help
------------------------------------------------------------------------------
SDSF OUTPUT DISPLAY RTSBATCH J0601323  DSID   107 LINE NO CHARS  'INDEXACCESS'
COMMAND INPUT ===>                                  SCROLL ===> CSR
******************************** TOP OF DATA ***********************************
1IBM Shared Profile Support -- Print Exception Triggers --  V03.10 Run Date 2010/11/19 Run Time 16:31:53

 26 Triggers created...

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  CLUSTERSENS        BGINT  >   500

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value-
 AUOVRDB  AUOCOPY                                                        0 356030

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  SCANACCESS         BGINT  >   1000

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value-
 AUOVRDB  AUOCOPY                                                        0 368810

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  CLUSTERSENS        BGINT  >   500

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value-
 AUOVRDB  AUODBASE                                                       0 3069275

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  SCANACCESS         BGINT  >   1000

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value-
 AUOVRDB  AUODBASE                                                       0 4792985

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  CLUSTERSENS        BGINT  >   500

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value-
 AUOVRDB  AUOGPAUT                                                       0 895

 Statistics Type--- Column------------ Type- Cond ---------Exception Value--------
 REALTIME REORG TS  SCANACCESS         BGINT  >   1000

 DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index----- TBOWNER- IXCRTR-- PART# ---------TRIGGER Value-
 AUOVRDB  AUOGPAUT                                                       0 35993
```

REORG Avoidance Example

- **Sample object AUOVRDB.AUODBASE needs one of the following:**
  - Should the index be changed to be more efficient?
  - Should the index be REORG'd?
  - Should I use a hash table instead?
- **Change Exception values and re-build Job Profile**
- **Select different Exceptions and re-build Job Profile**
- **Evaluate the objects in TRIGGERS DD in job output to make an intelligent decision on how to avoid REORGs**

Only REORG What Needs It

- **Let DB2 Automation Tool intelligently select objects to include in REORG JCL**
- **Use DB2 Automation Tool's 180+ Exceptions**
  - RTS Exceptions
  - DB2 Catalog Exceptions
  - MVS Catalog Exceptions
- **Optionally evaluate indexes independently of the tablespace**

DB2 Utilities Enhancement Tool for z/OS and DB2 Sort for z/OS

Using the Utility Syntax Monitor

DB2 Utilities Enhancement Tool – New! Utility Monitor

- **New with UK60173: Changes utility syntax at run-time based on Policy rules**
- **Passively enforce company IT policies**
- **Enables users to:**
  - ADD parameters that are not present in the utility syntax
  - REMOVE parameters that are present and should not be
  - SUBSTITUTE given parameters with different parameters
  - FAIL the utility based on object name, or user ID
- **Each action is logged or JOURNALED in UET's tables for future reference**
  - Audit who is doing what
  - See what syntax was originally specified
  - See what the original syntax was changed to

DB2 Utilities Enhancement Tool

- **New Policy Rules used to change utility syntax**
- **Optionally monitor utility syntax and/or cancel threads with the same Policy rules**
  - Default action is to cancel active threads
  - Monitor utility syntax with specific Policy parameters
- **Example: Add UET's PRESORT parameter to a LOAD utility and use DB2 Sort**
  - Utility Monitor will ADD parameter PRESORT to LOAD syntax
  - Use DB2 Sort to sort SYSREC data prior to LOAD utility running
  - Improves elapsed time, and reduces CPU consumption

Utility Monitor Policy Rules

```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Iest  Help
VIEW        CSJENN.ABP1074.SAMPLIB(ABPDB2AP) - 01.11          Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
000024 <!-- ************************************************************ -->
000025 <DSNUTILB_INTERCEPT>
000026     <PRACTICE NAME="LOAD_RULE">
000027       <UTILITY NAME="LOAD">
000028         <MONITOR>
000029           <SYNTAX ADD="PRESORT" JOURNAL="YES"/>
000030         </MONITOR>
000031       </UTILITY>
000032     </PRACTICE>
000033
000034     <POLICY>
000035       <DB2SYSTEM SSID="DA1A" ACTION="MONITOR_UTILITY">
000036         <USE_PRACTICE NAME="LOAD_RULE"/>
000037         <INCLUDE>
000038           <RULE UTILITY_COMMAND="LOAD"/>
000039         </INCLUDE>
000040       </DB2SYSTEM>
000041
000042       <DB2SYSTEM SSID="DA1A">
000043         <EXCLUDE>
000044           <RULE TABLESPACE="DB1543%.%"/>
000045         </EXCLUDE>
000046       </DB2SYSTEM>
000047
000048     </POLICY>
000049
000050 </DSNUTILB_INTERCEPT>
```

Original Utility Syntax

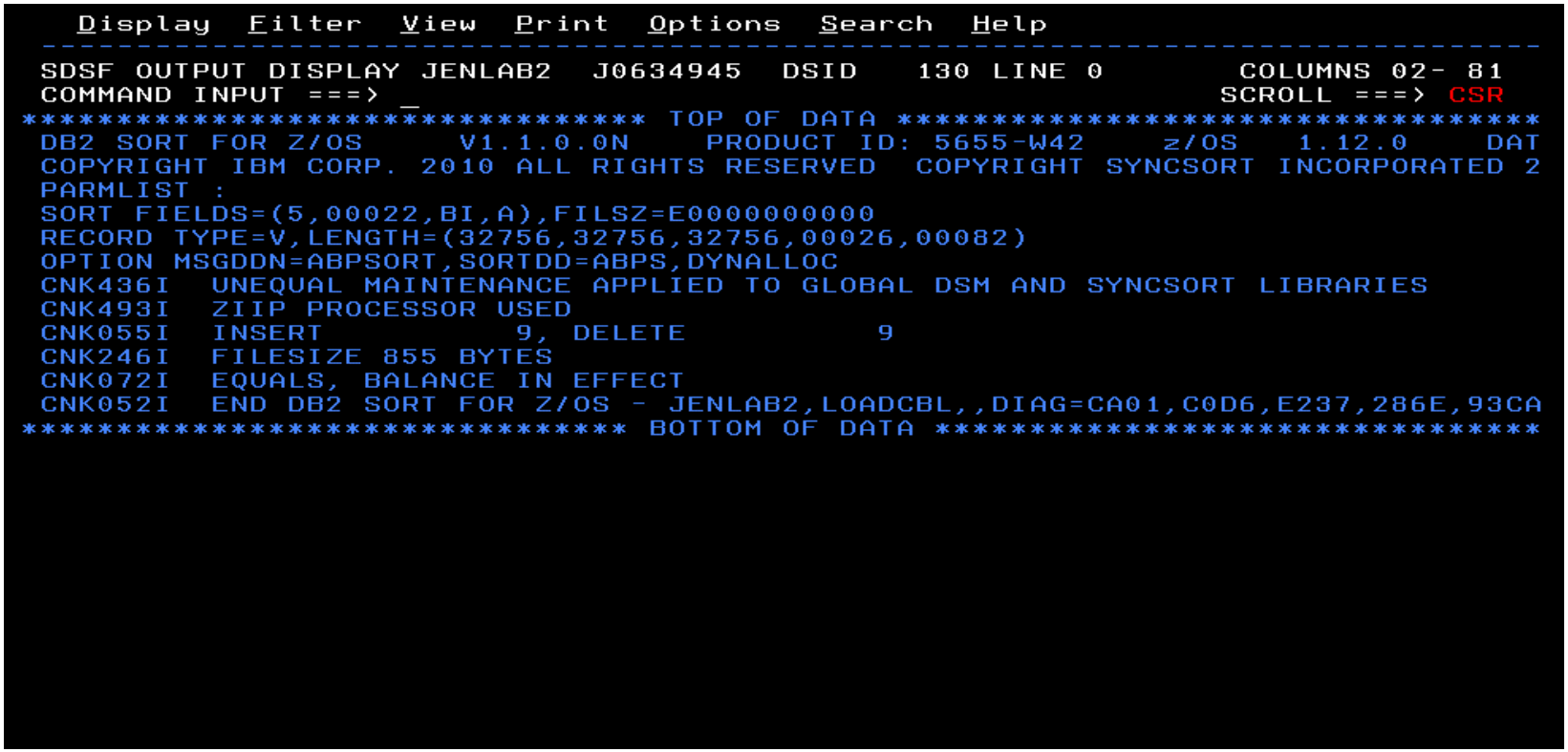
```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       CSJENN.ABP1074.TESTLIB(TC610L8) - 01.03          Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR
000184 //          DD DISP=SHR,DSN=VENDOR.DB2SORT.V110.SCNKLPA
000185 //          DD DISP=SHR,DSN=VENDOR.DB2SORT.V110.SCNKLINK
000186 //          DD DISP=SHR,DSN=DSN.VA10.SDSNLOAD
000187 //SYSPRINT DD SYSOUT=*
000188 //UTPRINT  DD SYSOUT=*
000189 //*
000190 //SYSREC   DD DSN=CSJENN.DA1A.JENDBL1.ABPTS2.SYSREC,
000191 //             DISP=SHR
000192 //*
000193 //SYSMAP   DD DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
000194 //             SPACE=(CYL,(10,10))
000195 //SYSUT1   DD DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
000196 //             SPACE=(CYL,(10,10))
000197 //SORTOUT  DD DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
000198 //             SPACE=(CYL,(10,10))
000199 //*
000200 //SYSIN    DD *
000201   LOAD DATA INDDN SYSREC LOG NO SHRLEVEL NONE RESUME YES
000202   INTO TABLE "JNABP610"."ABPTB1"
000203   ( "NAME"
000204    POSITION(  00004:00023) CHAR(00020)
000205   , "CABLE"
000206    POSITION(  00025:00044) CHAR(00020)
000207   , "STATE"
000208    POSITION(  00046:00065) CHAR(00020)
000209   )
000210 //* ------------------------------------------------------------------
```

PRESORT Added to Utility Syntax

```
  Display  Filter  View  Print  Options  Search  Help
 ------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY JENLAB2  J0634945  DSID   135 LINE 11      COLUMNS 27- 106
 COMMAND INPUT ===>                                          SCROLL ===> CSR
 Utility execution started. Step=1
 Original DSNUTILB syntax follows:
 LOAD DATA INDDN SYSREC LOG NO SHRLEVEL NONE RESUME YES INTO TABLE "JNABP610".
 "ABPTB1" ( "NAME" POSITION( 00004:00023) CHAR(00020) , "CABLE" POSITION( 00025:
 00044) CHAR(00020) , "STATE" POSITION( 00046:00065) CHAR(00020) ) PRESORT
 End of original DSNUTILB syntax listing.
 73 DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = LOAD.ABPTB1
 80 DSNUGTIS - PROCESSING SYSIN AS EBCDIC
 81 DSNUGUTC -  LOAD DATA LOG NO SHRLEVEL NONE RESUME YES
 :40.81 DSNURWI -  INTO TABLE "JNABP610"."ABPTB1"
 :40.81 DSNURWI -   ("NAME" POSITION(4:23) CHAR(20),
 :40.81 DSNURWI -    "CABLE" POSITION(25:44) CHAR(20),
 :40.81 DSNURWI -    "STATE" POSITION(46:65) CHAR(20)) INDDN ABPREC SORTKEYS 18
 95 DSNURPIB - NUMBER OF OPTIMAL SORT TASKS = 2, NUMBER OF ACTIVE SORT TASKS = 2
 95 DSNURPIB - INDEXES WILL BE BUILT IN PARALLEL, NUMBER OF TASKS = 4
 :41.04 DSNURWT - (RE)LOAD PHASE STATISTICS - NUMBER OF RECORDS=9 FOR TABLE JNABP
 :41.04 DSNURWT - (RE)LOAD PHASE STATISTICS - TOTAL NUMBER OF RECORDS LOADED=9 FO

 04 DSNURILD - (RE)LOAD PHASE STATISTICS - NUMBER OF INPUT RECORDS PROCESSED=9
 04 DSNURILD - (RE)LOAD PHASE COMPLETE, ELAPSED TIME=00:00:00
 :41.17 DSNURBXA - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=9 FOR INDEX JNABP610
 :41.17 DSNURBXA - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=9 FOR INDEX JNABP610
 18 DSNURPTB - SORTBLD PHASE STATISTICS. NUMBER OF INDEXES = 2
 18 DSNURPTB - SORTBLD PHASE COMPLETE, ELAPSED TIME = 00:00:00
 :41.18 DSNUGSRX - TABLESPACE JENDBL1.ABPTS2 IS IN COPY PENDING
 :41.18 DSNUGSRX - INDEX JNABP610.ABPTB1IX1 IS IN INFORMATIONAL COPY PENDING STAT
 :41.18 DSNUGSRX - INDEX JNABP610.ABPTB1IX2 IS IN INFORMATIONAL COPY PENDING STAT
```

DB2 Sort for z/OS Used to Sort Data



LOAD Syntax PRESORT Performance Example

- **1,000 Partition Tablespace**
- **10,000 rows of data**
- **Single SYSREC**
- **One Reader Task**

When using UET's PRESORT parameter and DB2 Sort:

45% Elapsed Time Savings
76% CPU Savings

Partitions	Rows per Part	Elapsed Time	Total CPU	
1,000	**10,000**	**09:41.70**	**01:07.80**	**Data in key order without UET**
1,000	**10,000**	**20:22.96**	**06:44.30**	**Data in random order without UET**
1,000	**10,000**	**11:11.08**	**01:35.17**	**Data in random order with UET PRESORT**

The Utility Monitor and DB2 Sort

- **New with DB2 Utilities Enhancement Tool's PTF UK60173:**
  - Utility Monitor changes utility syntax at run-time based on Policy rules
- **Passively enforce company IT policies**
  - ADD parameters that are not present in the utility syntax
  - REMOVE parameters that are present and should not be
  - SUBSTITUTE given parameters with different parameters
  - FAIL the utility based on object name, or user ID
- **DB2 Sort is fully supported in the DB2 Utilities Enhancement Tool**

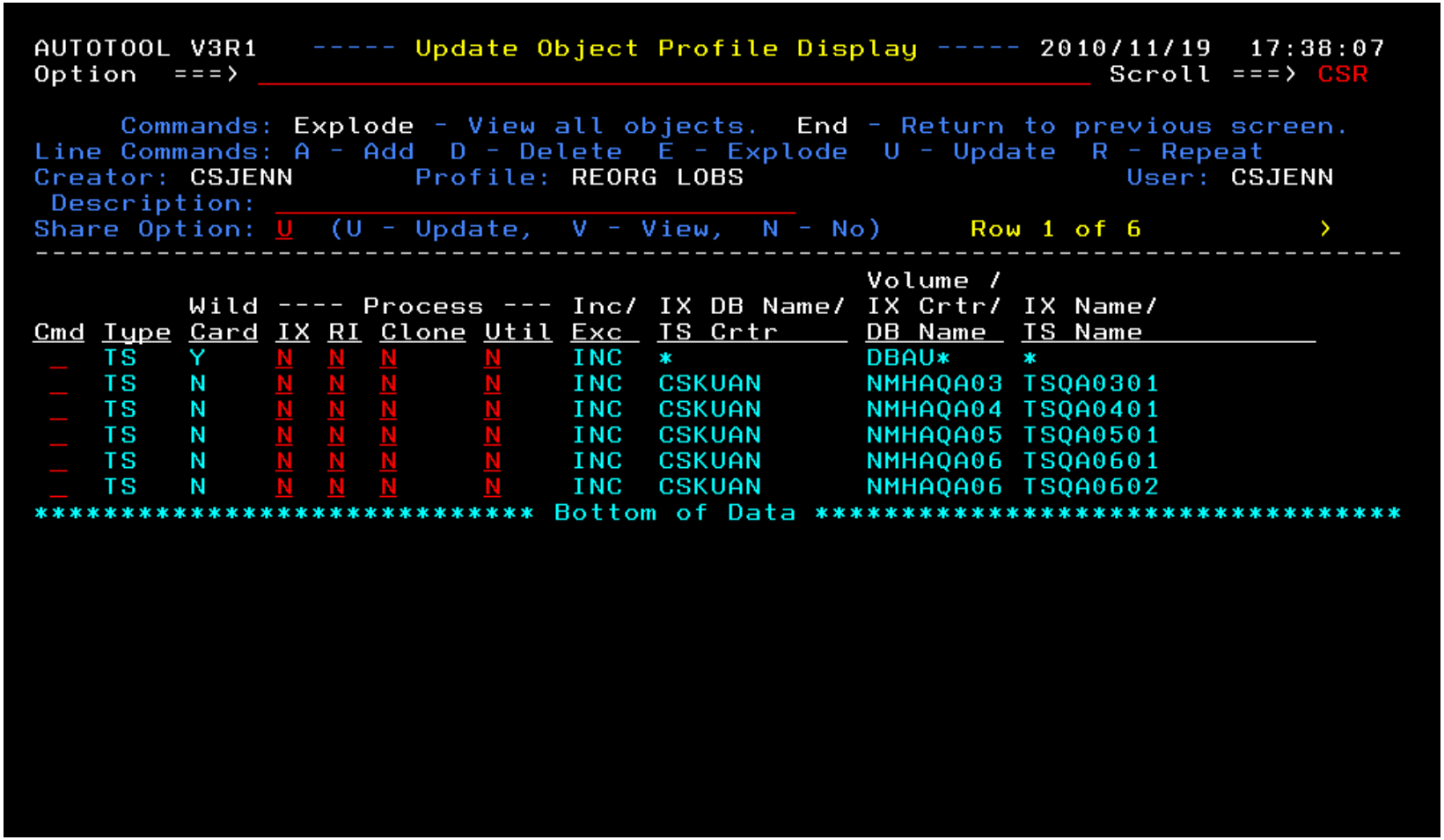
DB2 Automation Tool for z/OS v3.1

Managing complexity made easy – LOBs and REORG

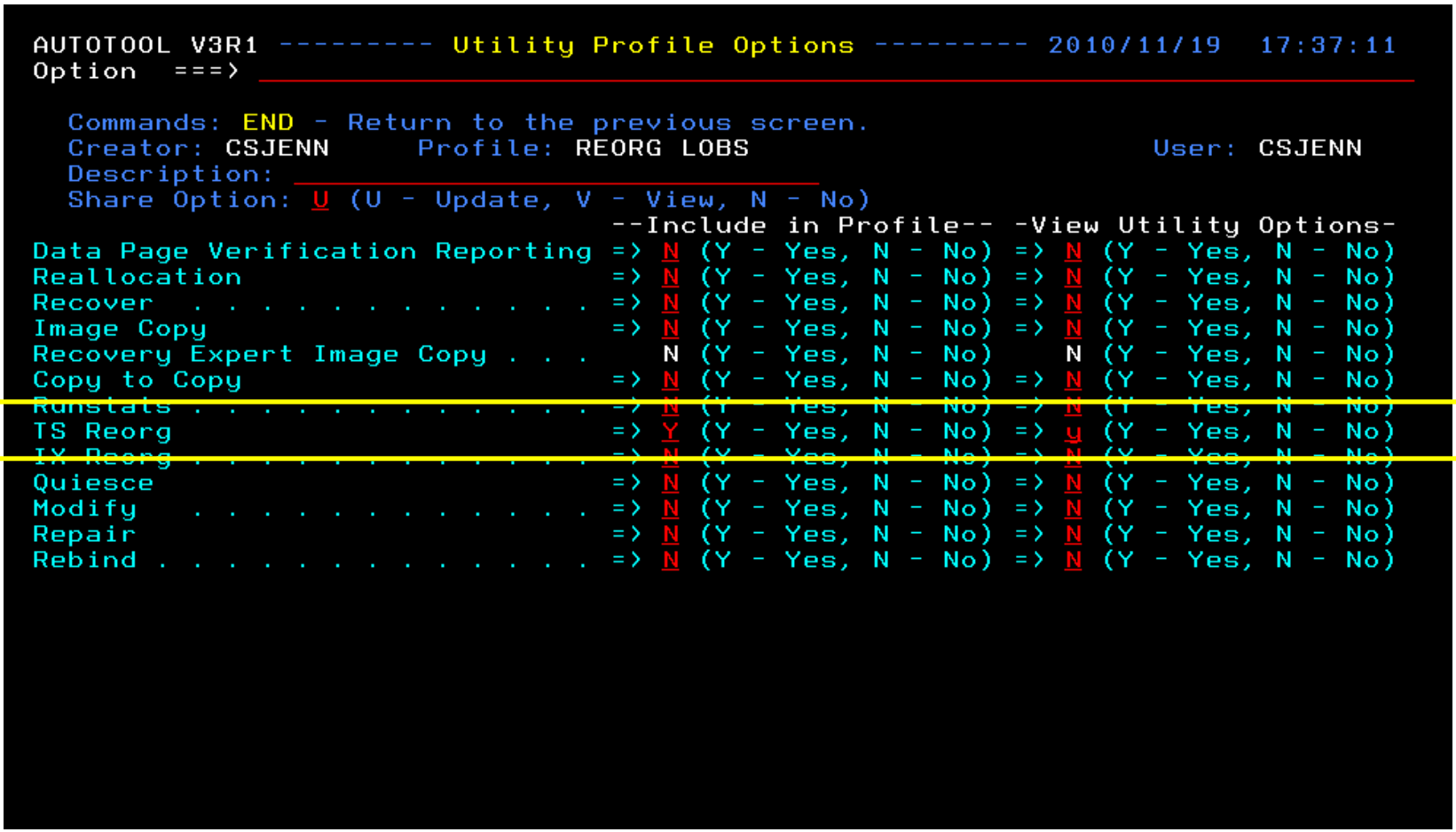
REORG LOBs Online with DB2 Automation Tool

- **What a great convenience!**
- **Reduce application downtime by reorganizing LOBs online**
- **Wildcard object names in Object Profiles without the worry of inadvertently including LOB objects**
  - No longer exclude specific LOB objects from Object Profile
  - No longer exclude LOBs using Exceptions Profile to exclude them
- **Can REORG SHRLEVEL CHANGE LOB table space:**
  - Independent of whether LOBs are LOG NO or LOG YES
  - No mapping table required
  - Base table space must be LOGGED

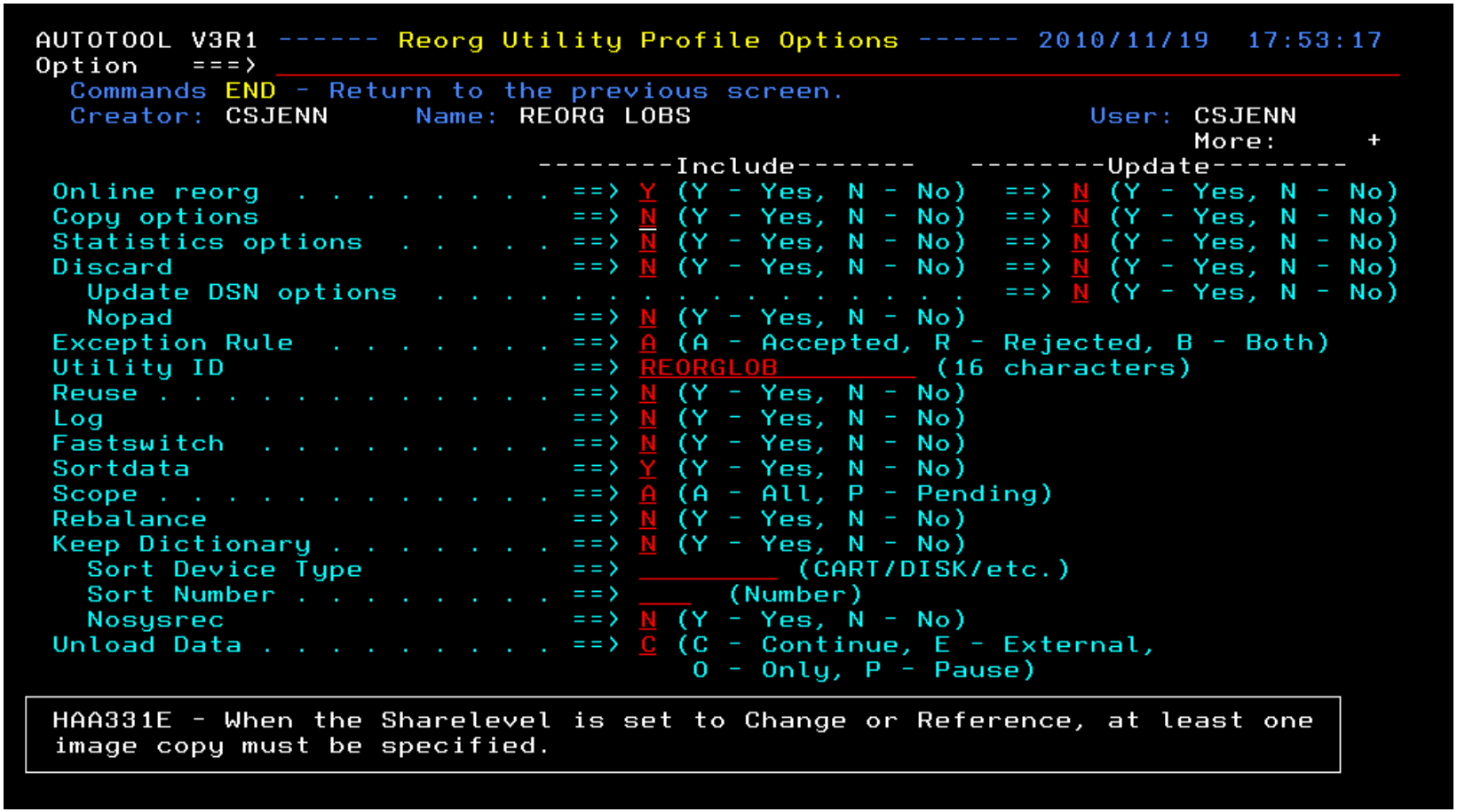
Online LOB REORGs



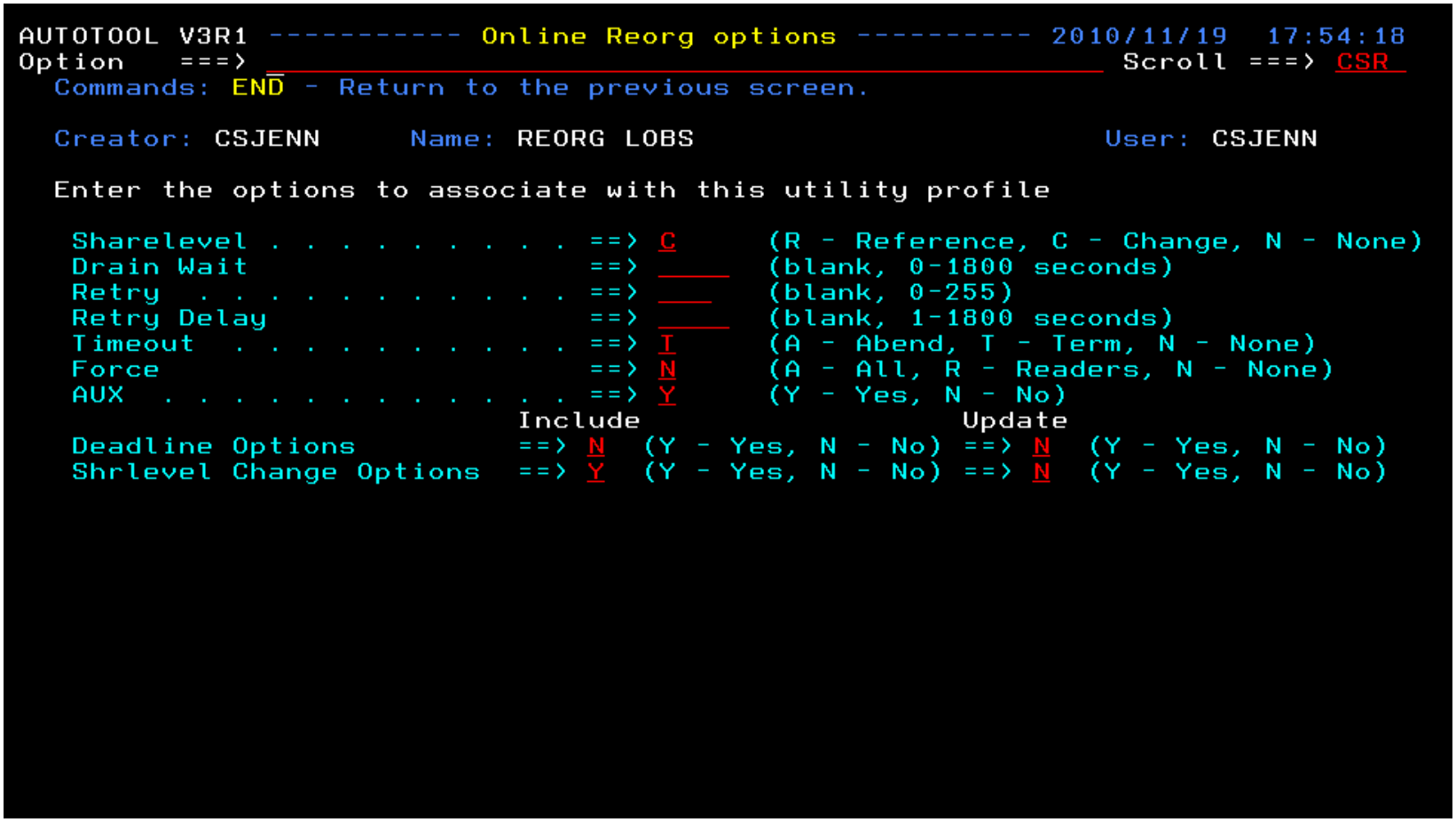
Online LOB REORGs



Online LOB REORGs



Online LOB REORGs



Online LOB REORGs

```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       CSJENN.HAA310.JCL(LOBREORG) - 01.00            Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR
000197 //SYSIN    DD *
000198    TEMPLATE R1LP0001
000199             UNIT     SYSDA
000200             DSN              'CSJENN.&DB..&SN..&SSID..&UNIQ.'
000201             SPACE    TRK
000202             MAXPRIME 00066666
000203             UNCNT    5
000204             DISP     (NEW,CATLG,CATLG)
000205
000206    LISTDEF REO11003
000207           INCLUDE TABLESPACE DBAUA203.TPAUA203
000208           INCLUDE TABLESPACE DBAUDIT.TSAUDIT
000209           INCLUDE TABLESPACE DBAUDIT2.TSAUDIT2
000210           INCLUDE TABLESPACE DBAUY203.TPAUY203
000211           INCLUDE TABLESPACE NMHAQA03.TSQA0301
000212           INCLUDE TABLESPACE NMHAQA04.TSQA0401
000213           INCLUDE TABLESPACE NMHAQA05.TSQA0501
000214           INCLUDE TABLESPACE NMHAQA06.TSQA0601
000215
000216    REORG TABLESPACE LIST REO11003
000217          SCOPE         ALL
000218          LOG           NO
000219          SORTDATA      YES
000220          COPYDDN      (R1LP0001)
000221          SHRLEVEL      CHANGE
000222          TIMEOUT       TERM
000223          MAPPINGTABLE "RTSBATCH"."REORG_CMP00102"
```

REORG LOBs Online with DB2 Automation Tool

- **Allow DB2 Automation Tool to generate JCL for both LOB and Non-LOB objects**
- **Wildcard object names in Object Profiles without the worry of inadvertently including LOB objects**
  - No longer exclude specific LOB objects from Object Profile
  - No longer exclude LOBs using Exceptions Profile to exclude them
- **DB2 Automation Tool will generate appropriate JCL for each kind of object**
- **Some restrictions apply:**
  - LOG NO is required for SHRLEVEL REF
  - Mapping tables are ignored for LOB tablespaces
  - SHRLEVEL REF requires inline image copy

Questions?