

## RealTime DBAExpert

INTELLIGENCE. INNOVATION. INTEGRITY

### **Presentation Abstract:**

Some of the Europe's largest computer centers are using RealTime DBAExpert to manage their DB2 database maintenance 24x7 in real-time. This presentation will focus on how the tool monitors all objects of an entire DB2 subsystem and supports just-in-time online utility generation based on real-time statistics (RTS). In this presentation, we show how to set up real-time maintenance, what to consider when setting up RTS for maintenance decisions, and how to eliminate unnecessary RUNSTATS. We will demonstrate the tools integration with typical job scheduling systems like TWS (aka OPC), CONTROL-M, CA-7 and UC4 through the Dynamic Job Scheduler Interface (e.g., Batch Loader). Live examples from the airline, insurance, banking and government installations are presented as well.

## Agenda:



- DB2 maintenance: cyclic, re-active, automated, on demand, or autonomic
- Real Time Statistic exploitation including COLGROUP
- REORG w/o BUILD 2 phase
- Universal Table Spaces (UTS)
- And more...

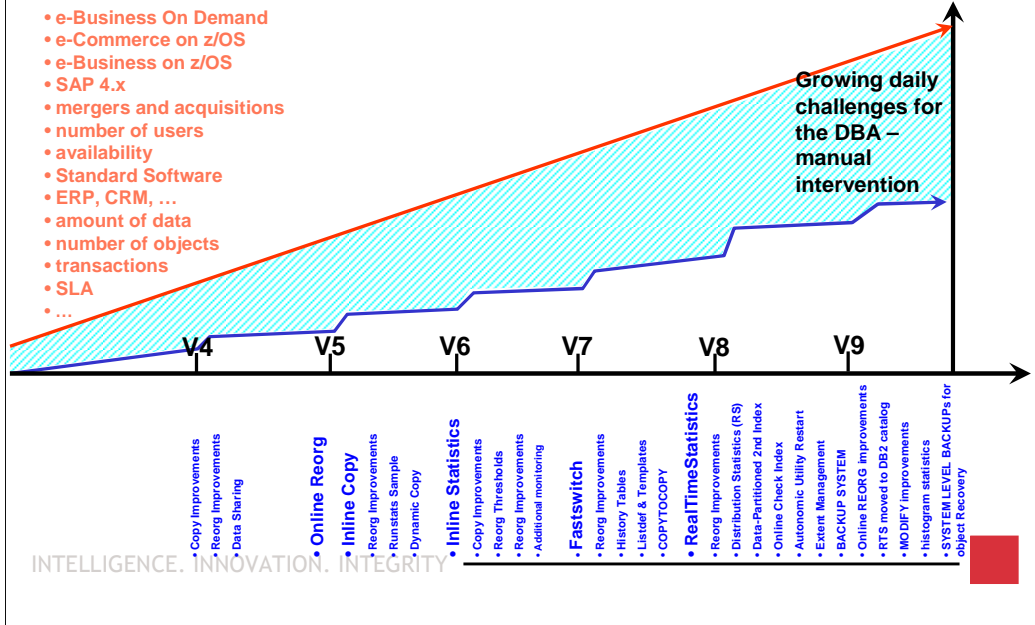
INTELLIGENCE. INNOVATION. INTEGRITY

**DB2 9 enhancements, use them or you'll forget them!**

**This presentation highlights the DB2 9 for z/OS enhancements that directly impact DB2 maintenance cycle and recoverability. Every release of DB2 offers new features and functionality that could make your maintenance cycle shorter and ensure your recovery time objectives are met. Don't leave these features and functions sitting around, put them to use and enjoy the benefits.**

**If your maintenance cycle is in need of a review and updates, your batch window continues to grow smaller, then this presentation should have something to offer you.**

# What drives database maintenance?



## What Drives Database Maintenance?

Business requirements such as availability and cost drive database maintenance. In the database maintenance area, it requires features that reduce the manual effort (such as thresholds) and features that improve availability (such as SHRLEVEL CHANGE capabilities, FASTSWITCH, etc.) and reduce costs. Business demands are rising, and new capabilities are provided to meet these demands. This is especially true of ERP and CRM applications that consist of more than 40,000 objects. Additionally, companies are running more and more DB2 systems. Significant improvements to REORG, RUNSTATS, COPY, and the other utilities are documented in each version of DB2. Occasionally, there are milestone enhancements that lead to a new level of database maintenance like Online/Inline Utilities, Fastswitch and especially Realtime Statistics.

This graph is also depicting what IBM has done to assist the DBAs with the different DB2 releases and the rising demands that Enterprises and customers face today. The shaded area represents the manual work that still must be done and you can see that even though IBM have done a lot of work to help us there is still a significant gap between what is automated and what is actually desired. Even with the DB2 9 there is STILL work for the DBA to do!

V9.1 introduces **Universal table spaces** a new type of table space—a *universal table space*. A universal table space is a table space that is both segmented and partitioned. Two types of universal table spaces are available: the partition-by-growth table space and the range-partitioned table space. A universal table space offers the following benefits: v Better space management relative to varying-length rows: A segmented space map page provides more information about free space than a regular partitioned space map page. Improved mass delete performance: Mass delete in a segmented table space organization tends to be faster than table spaces that are organized differently. In addition, you can immediately reuse all or most of the segments of a table. Those new objects make space management and administration easier, but I focus on the following topics of database maintenance tasks within the next slides.

- [RTS moved to DB2 catalog](#)
- [SYSTEM LEVEL BACKUPS for object Recovery](#)
- [Online REORG improvements](#)
- [Histogram Statistics](#)
- [MODIFY improvements](#)

## DB2 for z/OS maintenance levels covered by RealTime DBAExpert:



	<b>level 1 <i>cyclic</i></b>	<b>level 2 <i>re-active</i></b>	<b>level 3 <i>automated</i></b>	<b>level 4 <i>on demand</i></b>	<b>level 5 <i>autonomic</i></b>
<b>procedure</b>	static maintenance jobs for all objects	generated maintenance jobs for specific objects	generated, threshold based maintenance jobs	dynamic maintenance jobs based on real-time data	dynamic maintenance integrated via SLA's
<b>operation</b>	manual job building	manual job building	automated, threshold based job generation	continuous, threshold based Job generation	continuous, DB2 based generation
<b>execution</b>	total maintenance, manual execution	selective maintenance, manual execution	selective maintenance, static execution	selective maintenance, automated execution	selective maintenance automated execution
manual intervention					

INTELLIGENCE. INNOVATION. INTEGRITY

### What Drives Database Maintenance?

Business requirements such as availability and cost drive database maintenance. In the database maintenance area, it requires features that reduce the manual effort (such as thresholds) and features that improve availability (such as SHRLEVEL CHANGE capabilities, FASTSWITCH, etc.) and reduce costs. Business demands are rising, and new capabilities are provided to meet these demands. This is especially true of ERP and CRM applications that consist of more than 40,000 objects. Additionally, companies are running more and more DB2 systems. Significant improvements to REORG, RUNSTATS, COPY, and the other utilities are documented in each version of DB2. Occasionally, there are milestone enhancements that lead to a new level of database maintenance like Online/Inline Utilities, Fastswitch and especially Realtime Statistics.

Database maintenance can be characterized on these levels from one to five:

Level 1: cyclic database maintenance, a DBA writes a utility job for all objects of the database. This procedure occurs periodically.

Level 2: reactive maintenance, the procedure in level one has become too time-consuming. Only when an object required maintenance was a utility job built and executed. This reduced the number of utility jobs as well as run-time, CPU and I/O, but it's a reactive approach. An upcoming performance bottleneck can lead to an outage even before a DBA learns about it.

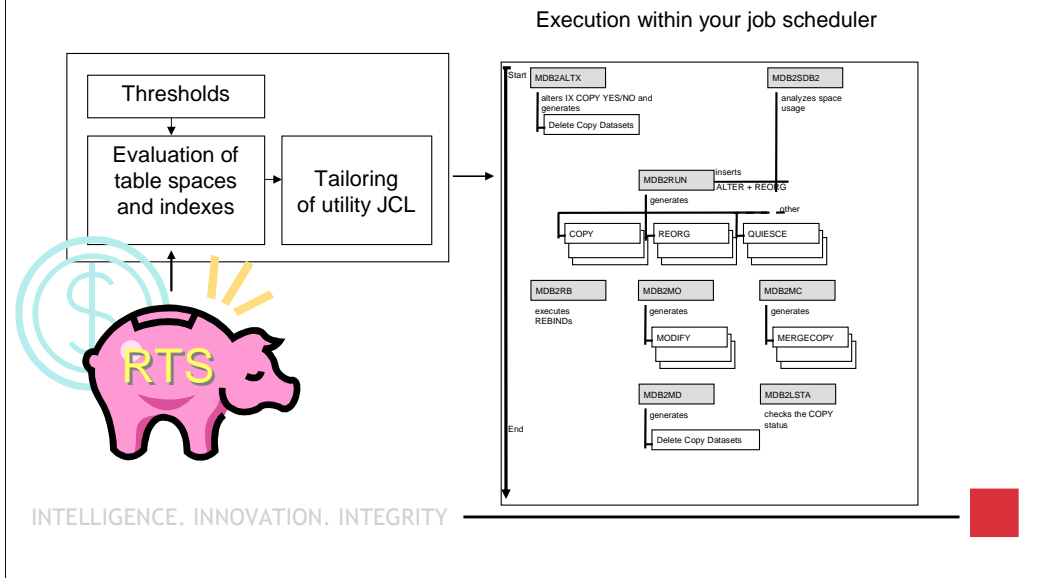
Level 3: automated maintenance, if there's a maintenance requirement, a utility is automatically generated and executed. Within a pre-defined, static job net, each single object is frequently checked for maintenance requirements based on defined thresholds. At this level, granular automation comes up with the database maintenance process. Utility job generation is included; no one has to start or execute the process steps. The entire procedure is defined and automated in the company's job scheduler. Automated maintenance is the most common way of performing database maintenance today, and most database maintenance solutions support it. Because this method involves frequently checking thresholds, automated maintenance can ensure an emerging bottleneck doesn't develop into a serious problem. However, automated maintenance may not be sufficient for today's requirements and its future may be limited.

This brings us to Level 4: On-demand database maintenance. This solves the existing problems by quickly reacting and fully exploiting Real-Time Statistics (RTS) and the latest technology enhancements. At this level, the objects are continuously monitored for maintenance requirements, and appropriate utility jobs are dynamically generated and executed based on workload considerations. However, before getting into the details, let's see what drives database maintenance and how we arrived at the current level.

## DB2 for z/OS Maintenance



- Level 3 maintenance (batch)



Most DB2 customers currently do DB2 database maintenance on Level 2 or 3 and a significant number of customers does that through a traditional database maintenance solution. Those solutions trigger utilities either in a defined interval (level 2) and/or threshold based (level 3). RTDX allows to keep those processes as is, but exploit RTS for example. With RTDX you even don't have the disadvantage of RTS activation. We initialize them to get the ready to run, without the need of exhaustive utility runs. In addition we provide a bunch of tools that assure that you can use the RTS as reliable as your former DB2 catalog source for threshold checks.

Here is an example Job net from RTDX. All traditional DB2 database maintenance tools usually have something similar to this.

## Real-Time Statistics (RTS)



- What is RTS ?
  - Collect real-time statistics for table and index spaces
  - V7 or later (**integrated into DB2 Catalog within DB2 9**)
  - Capture statistics in memory and periodically written to the RTS tables
    - SYSIBM.TABLESPACESTATS
      - Contain statistics for table spaces
      - One row per table space/partition
    - SYSIBM.INDEXSPACESTATS
      - Contain statistics for indexes
      - One row per index space/partition
  - Tables reside in a segmented table space (DSNRTSTS)
  - Table spaces reside in the DSNRTSDB database

INTELLIGENCE. INNOVATION. INTEGRITY



With DB2 Version 7, IBM introduced RTS, which significantly changes the process of threshold-based maintenance. With RTS, DB2 provides current statistics of all your DB2 objects without any extra costs; hence, eliminating RUNSTATS normally needed to collect statistics. Additionally, the statistics no longer influence the DB2 optimizer, as they're stored in specific RTS tables.

## Real-Time Statistics (RTS)



- Enabling RTS
  - Create objects via DSNTTESS
  - **A new DB2 9 already provides them**
  - Explicit START DATABASE(DSNRTSDB) enables real time statistics externalization

INTELLIGENCE. INNOVATION. INTEGRITY



By creating the RTS tables with the DSNTTESS sample job, you get a repository for the data being collected. If you are already running DB2 9 existing pre 9 RTS tables are moved to the DB2 catalog. Starting the RTS database explicitly triggers DB2 to start externalizing the data it has gathered.

## Real-Time Statistics (RTS)



- Working with RTS
  - DBM1 allocates RTS blocks for each updated object
    - At first update since the pageset/partition is opened
    - RTS runs in DBM1 Address Space
      - CPU time is included in DBM1's SRB time
      - The system task is created during START DB2
  - Free RTS blocks when
    - Pagesets/Partitions are closed
    - After statistics are written to RTS tables
  - In a data sharing system, statistics are collected by each member
  - In-memory, statistics are always collected even if RTS is not enabled

INTELLIGENCE. INNOVATION. INTEGRITY



RTS are gaining popularity. Customers using RTS reported no measurable overhead of DBM1 collecting the data. The RTS are collected in memory. The feature is integrated into the internal processes and can't be set on or off by a command or a ZPARM.

## Real-Time Statistics (RTS)



- Collecting RTS
  - CREATE/DROP TABLESPACE
  - Insert / Update / Delete
  - Rollback
  - REORG
  - RUNSTATS
  - COPY
  - LOAD
  - REBUILD
  - RECOVER
  - Triggers may cause statistics updated for other tables

INTELLIGENCE. INNOVATION. INTEGRITY



Whenever an update, insert, or delete occurs whether by SQL or by a utility DB2 collects statistics. And this makes clear why they are so efficient. Instead of processing data without notice and gathering statistics afterwards, DB2 just remembers what it does while it happens.

## Real-Time Statistics (RTS)



- Externalizing the RTS
  - RTS manager externalizes in-memory statistics to the RTS Tables on a timer interval
    - STATSINT in ZPARM - default 30 minutes (**DB2 9 15 min.**)
    - REAL TIME STATS in DSNTIPO install panel
  - STOP DATABASE SPACENAM command
    - Flush in-memory statistics for all target objects
  - STOP DATABASE(DSNRTSDB)
    - Flush all in-memory statistics
  - STOP DB2 MODE(QUIESCE)
  - A utility operation (e.g. LOAD, REORG, RUNSTATS, COPY, REBUILD, RECOVER)

INTELLIGENCE. INNOVATION. INTEGRITY

The externalization process can be customized. By default, the data is externalized every 30 minutes. With DB2 9, it's already reduced to only 15 minutes. RTS resolve a major weakness of Level 3 maintenance and the process is less expensive and time-consuming. Using RTS, you can check for a maintenance requirement without any preceding process to get current object statistics and is fully exploited by RTDX. This is faster and less expensive than any RUNSTATS and makes this feature compelling and gives hours back to your batch window. This proves the big advantage of RTS. With the DB2 catalog you only had consistent statistics after executing an expensive RUNSTATS or 3rd party equivalent.

For more insights on RTS, see Craig Mullins' article in the August/September 2006 issue of z/Journal titled "z/Data Perspectives: It's Time for Real Time Stats" and look for RTS-related presentations at local user groups and conferences like IDUG.

## Real-Time Statistics (RTS)



- Accuracy of the RTS if managed by RTDX
  - As timer interval
  - Loss of all in-memory statistics when DB2 is crashed or stopped in MODE(FORCE)
  - Only physical space statistics are maintained for DSNDB07 and the TEMP databases
  - Statistics counters will not be updated during DB2 Restart

INTELLIGENCE. INNOVATION. INTEGRITY



Although the Realtime Statistics are called realtime, the RTS tables represent the statistics of the last externalization. This interval becomes shorter and shorter. The IBM labs already tested intervals like 5 minutes and shorter in DB2 9 and haven't encountered problems. However, my suggestion is to use the default.

## Real-Time Statistics (RTS)



- RTS good practice
  - Avoid Timeouts or Deadlocks with RTS manager
  - Use Uncommitted Read lock isolation when accessing RTS tables
  - For Disaster Recovery
    - Recover RTS objects after DB2 catalog and directory objects are recovered
    - Explicitly issue START DATABASE(DSNRTSDB) after RTS objects are recovered

INTELLIGENCE. INNOVATION. INTEGRITY



If you plan to use the RTS for your own tasks keep these recommendations in mind. Important is to integrate these objects in general recovery procedures and process them after the DB2 catalog and directory. Explicitly starting them afterwards assures, that any subsequent recovery of objects will be considered.

## Real-Time Statistics (RTS)



- Weak points when enabling RTS without RTDX
  - Validate table space, table and index definitions for orphans, widows and null stats and inconsistencies
  - Data may not be accurate until a new REORG/RUNSTATS/COPY is done
  - Unable to capture statistics when DSNRTSDB is stopped or statistics tables are unavailable
  - Statistics could be inaccurate if running vendor utilities without flushing the in-memory statistics
  - Some stats may not be valid in a utility restart scenario
  - Use SHRLEVEL CHANGE when running utilities against RTS
  - Don't mix RTS objects with other user objects in a utility list operation

INTELLIGENCE. INNOVATION. INTEGRITY



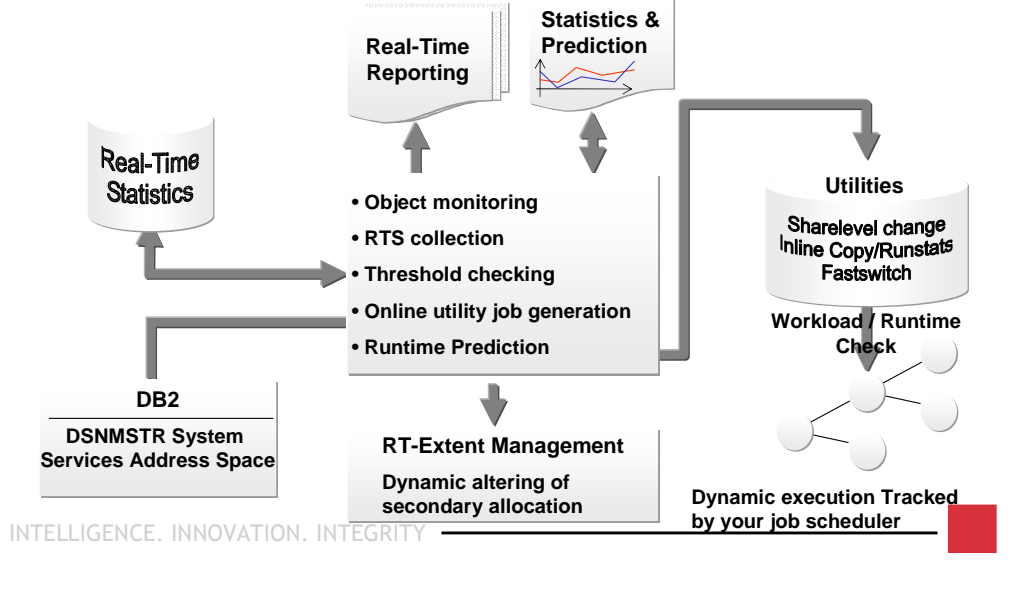
There are some disadvantages if you don't use RTS with a solution like RealTime DBAExpert, e.g. the data isn't initialized with RTS until after you execute a REORG, RUNSTATS, and COPY against all existing objects. However, customers of RTDX don't have to care for things like that, as RTDX does it for you. We do all the required stuff from initialization, through maintenance and housekeeping. We also provide consistency checks within our procedures.

If you use RTS within RTDX database maintenance procedures, you can be sure you can exploit the advantages without compromising the quality you know from your catalog.

## DB2 for z/OS Maintenance



- Level 4 maintenance (24x7)



RTS are a major component of Level 4 maintenance and made it even possible. However to do On Demand database maintenance requires additional technical improvements of the traditional processes. This is the “Full Picture” of our realtime maintenance tool RTDX and should give you an overview about the interaction of all parts. However, all components can be used on an as-needed basis. Especially the job execution is a general question. Primarily customers may reduce the costs by exploiting new features like RTS or Flashcopy technology, but keep their batch cycle. The direct costs can be prohibitive for database maintenance processes, as can be the indirect costs for backups and DB2 resource requirements due to delayed maintenance.

These problems seep in; they don’t emerge overnight. The resulting symptoms can be alleviated by adjusting the thresholds, spending more resources on CPU, I/O and run-time, or trying to keep up with the implementation of fast utilities. But the “faster way” defuses the symptoms and doesn’t eliminate the recurring problems.

Here are some strong indicators that you might have, requiring to switch to the next Level of database maintenance to adjust to the grown requirements of your shop:

- **Changed conditions:**

The maintenance window is shrinking even as the number of required utility jobs grows. Many shops already have to ensure non-stop system availability. But maintenance processes often can’t keep up with the increasing number of objects and transactions. This raises important questions about business continuity.

- **Time to react:**

The period of time between an upcoming requirement for nightly maintenance and its actual execution is too long. Today’s applications generate more database transactions than ever before. The static cycle no longer meets the requirements.

The second indicator usually affects recoveries first. Having mass updates over the day, but backups only in the night leads to growing recovery durations over the day due to more and more log apply. This can only be solved by On Demand maintenance. It assures optimal 24x7 operation and recoverability of dynamic systems with high workload. Within the next slides I will introduce the new components of On Demand database maintenance of Level 4 and show some examples how it improves recoverability and performance. Additionally I will address the resulting cost reduction.

## Controlled On-Demand Processing



```

RealTimeMaintain ----- Execution Modes ----- Row 1 of 5
Command ==> _____ Scroll ==> CSR
DB2: S910

Primary cmd: I(nsert), P(rint), L(ocate EXECUTION MODE), T(ime-Window Off)
Line cmd: S(elect), I(nsert), D(elete), E(xcludes)

0-+-2- 3-+-5- 6-+-8- 9-+-11 12+-14 15+-17 18+-20 21+-23 24 HOUR CLOCK
MON nnnnnn ***** **pp pp** **pppp pp** ***** **nnnn
TUE nnnnnn ***** **pp pp** **pppp pp** ***** **nnnn - Time of day
WED nnnnnn ***** **pp pp** **pppp pp** ***** **nnnn in 1/2 hour
THU nnnnnn ***** **pp pp** **pppp pp** ***** **nnnn steps
FRI nnnnnn ***** **pp pp** ***** **pp** ***** **nnnn
SAT wwwwww wwwwww wwwwww wwwwww wwwwww wwwwww wwwwww
SUN wwwwww wwwwww wwwwww wwwwww wwwwww wwwwww wwwwww

SP1 zzzzzz zzzzzz zzzzzz zzzzzz zzzzzz zzzzzz zzzzzz
SP2 _____
SP3 _____

EXECUTION MODE ID RO MAX CO MAX RS MAX IXSPACE EXCLUDES
-----
- NIGHT-TIME n 180000
- NON-PEAK-TIME * 45000 90000 180000 N Y
- PEAK-TIME p 30000 90000 90000 Y Y
- WEEKEND w N N
  
```

INTELLIGENCE. INNOVATION. INTEGRITY

On Demand processing is a strictly controlled and transparent process. This screen shot shows general “Time Window” configuration. The time window controls whether or not a utility is allowed to run if a regular threshold has been exceeded. It is extremely flexible and gives you the ability to define 30 minute slices of time. Each slice must be defined with a valid Execution Mode ID, which indicates the time of day or night, e.g., peak or non-peak.

Each Execution Mode has a set of characteristics defined. The first of these is a general object size limit for each utility. The related IXSPACE can be include when this space calculation is done. If this is exceeded the Utility is NOT generated. The Execution Mode can also contain an Exclusion List of objects for a specific time.

Usually a group definition is implemented that allows COPY and RUNSTATS more often than a REORG.

The time window control of RTDX also has a “look-ahead” to see if a Utility would indeed finish within an allowed time-slot. That means we not only check if a utility is allowed at a specific time, but also if it can be finished within the time slot. The check is done at generation time and a final cross check at utility initialization.

## Controlled On-Demand Processing



```
RealTimeMaintain ----- WLM Control -----
Command ==> _____ DB2: S910

Primary cmd: W(LM control ON), C(current system load)

WLM CONTROL . . . . . : ON
SHORT-TERM SYSTEM LOAD : 84 - (0..100) at 2008-03-14-09.35.48

Specify your system characteristics to calculate current system load:

WORKLOAD STABILITY . . . 6 - 0 .. 9, recommended values are 5 .. 7
IO BIAS . . . . . 3 - 0 .. 9, recommended values are 3 .. 5
PEAK TOLERANCE . . . . . D - 0 .. 9 or D, recommended is D(efault)
SRM WEIGHT . . . . . D - 0 .. 9 or D, recommended is D(efault)

Specify your system load levels and actions to take:

WARNING LEVEL . . . . . 63 - 50 .. 100, allow utility to run
ACTION . . . . . B - W(TO)/M(sg)/B(oth)/N(one)
EXCESSIVE LEVEL . . . . . - 50 .. 100, do not allow utility to run
ACTION . . . . . - W(TO)/M(sg)/B(oth)/N(one)
```

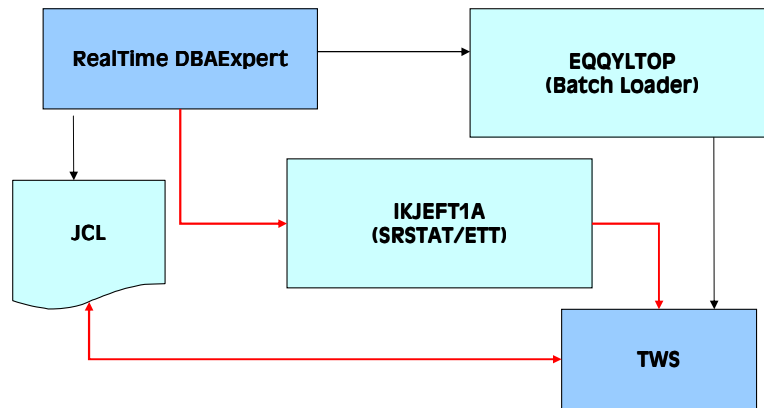
INTELLIGENCE. INNOVATION. INTEGRITY

The time slots represent a general rule set. In addition RTDX has an interface to WLM to check the real workload within the final check at utility initialization. This assures that we don't execute a utility if the workload is above a defined level.

## Dynamic Job Scheduler Interface



Job scheduler interface (OPC-TME/TWS sample) :



INTELLIGENCE. INNOVATION. INTEGRITY

If a dynamic utility execution is allowed based on the time table and WLM, we not only generate the JCL, but also hand it over to the job scheduler in place. Job schedulers usually provide a dynamic interface for that purpose. We support the most common job schedulers on the market like TWS, CA7, Control/M.

This has several great advantages.

- Utility execution can happen On Demand
- The job scheduler stays the controlling resource for jobs
- Job net definitions are no more required

Dynamic job execution sometime frightens people before they have tested it. After getting familiar with this makes it clear that the process itself doesn't change.

1st a utility is generated based on its own criteria

2nd a check is done weather execution is allowed and possible

3rd its passed to the scheduler for execution

This is the same as with automated maintenance procedures of prior database maintenance levels.

## Transparent On-Demand Processing



```
RT-InformationCenter ----- Reorg Tablespace Details -----
Command ==>
Primary cmd: R(eason list) DB2: S910

DETAILS
REASON . . . . . : REORG Unclustered inserts percentage (MULTI)
TIMESTAMP . . . . : 2008-03-15-08.41.32 UT STAT : Completed
EXCEPTION REASON . . : None MAXPAGES : 45000
EXEC MODE . . . . . : NON-PEAK-TIME EXEC ID : * ACTIVE : Y
EXEC MODE SET BY . . : BOXWELL AT . . . : 2008-02-03-10.47.07
LINKED GROUP ID . . . : * * RUNSTAT : Y COPY : Y

RTS DATA
DATABASE . . . . . : DATADB SPACE . . . . . : SPACE23
PARTITION . . . . . : 0 LAST REORG DATE . . : 0001-01-01-00.00.00
NBR CHANGED ROWS . . : 2014 PCT CHANGED ROWS . . : 17
NBR UNCLUSTINS . . . : 1532 PCT UNCLUSTINS . . . : 23
NBR DISORGLOB . . . . : 0 PCT DISORGLOB . . . : 0
NBR INDREF . . . . . : 0 PCT INDREF . . . . . : 0
NBR MASS DELETES . . : 10 EXTENTS . . . . . : 1
TOTAL NBR ROWS . . . . : 12014 NACTIVE . . . . . : 720

RTS THRESHOLDS
MIN PAGES . . . . . : 32 PCT CHANGED ROWS . . : 5
PCT UNCLUSTINS . . . : 10
PCT DISORGLOB . . . . : 10
PCT INDREF . . . . . : 10
FOR EXTENTS . . . . . :
PCT ACTIVE . . . . . :
NBR MASS DELETES . . : 0
```

INTELLIGENCE. INNOVATION. INTEGRITY

To keep control of the process is the first thing, but another important step is to be able to get any information of all processes whenever required. RTDX provides an Information Center that gives any information on the checks we did, the maintenance requirements that came up, the utilities and jobs that are generated, the status and the execution statistics.

Here is an example screen shot after drilling down to the detail level for a REORG of a table space. The screen is split into three information areas.

The first is the execution environment:

In this case you see the REASON and it has the word MULTI appended which means that more than one reason caused the Utility to be generated.

The middle area contains the RTS data:

It provides the statistics from the check and we highlight any thresholds with YELLOW for REGULAR and with RED for CRITICAL maintenance requirements.

The last part contains the thresholds:

It shows the values that are set at the time we did the comparison.

## Transparent On-Demand processing



```
RT-InformationCenter ----- Object Statistics Summary ----- Row 177 of 391
Command ==> _____ Scroll ==> CSR
DB2: S910

Primary cmd: L(ocate DATABASE), S(ort), P(rint)
Line cmd: E(xpand), R(eporting)
```

	DATABASE	DBID	SPACE	PSID/ ISOBID	PART	IX	CO	CO AV ELAPS	RS	RS AV ELAPS	RO	RO AV ELAPS
-	MDATA1OD	269	IXR0X221	118	0	Y	0	0	0	0	1	14
-	MDATA1OD	269	IXR0X241	128	0	Y	0	0	0	0	2	15
-	MDATA1OD	269	IXR0X251	133	0	Y	0	0	7	3	2	15
-	MDATA1OD	269	IXR0X252	135	0	Y	0	0	7	3	2	14
-	MLIVEEST	264	M790601	2	0	N	9	22	0	0	2	71
-	MLIVEEST	264	M790602	84	0	N	4	20	1	5	11	98
-	MLIVEEST	264	M790603	16	0	N	12	37	1	2	3	54
-	MLIVEEST	264	M790604	23	0	N	6	35	7	11	21	90
-	MLIVEEST	264	M790605	30	0	N	21	4	1	16	1	42
-	MLIVEEST	264	M790606	37	0	N	16	8	13	10	12	47
-	MLIVEEST	264	M790608	45	0	N	33	4	0	0	0	0
-	MLIVEEST	264	M790609	48	0	N	13	27	2	3	2	83
-	MLIVEEST	264	M790610	53	0	N	0	0	0	0	25	72
-	MLIVEEST	264	M790611	58	0	N	16	35	3	4	2	55
-	SAPDB1	264	SPACE02	63	0	N	6	30	7	10	21	82
-	SAPDB2	264	SPACE04	58	0	N	16	35	3	4	2	55
-	SAPDB3	264	SPACE02	63	0	N	6	30	7	10	21	8

INTELLIGENCE. INNOVATION. INTEGRITY

The InformationCenter provides all the historical data of the database maintenance processes. The ISPF interface is easy and intuitive to use. It provides several filters, sort and drill down capabilities. When a job has been executed, runtime statistics are captured by RTDX. This panel shows how many utilities were executed down to the object level. It also provides data on how long each type of Utility took. This data is input for the runtime prediction done to check if a utility fits a time window.

## Copy Utility



### Goal:

#### Recovery Time Objectives (RTO)

*>>The Recovery Time Objective is the time needed to recover or, saying it another way, how long you can afford to be without your objects, applications, or systems.<<*

INTELLIGENCE. INNOVATION. INTEGRITY



A major task of database maintenance is to assure recoverability by taking backups at the appropriate interval. It's a good topic to understand the advantages and disadvantages of the different database level. To be able to assess if a backup procedure fits the requirement, a clear goal and reason is mandatory. Most companies want to have the shortest recovery time as possible if an outage is encountered. Therefore you only have to take as many copies as possible and place them on a DASD. This makes clear that there is also another goal. The operation should also be as cheap as possible. Now we have two goals and they compete! From the regular operation point of view backups are useless and wasting resources. DB2 doesn't require them to process SQL queries and customers usually don't need them either. But if an outage occurs they are crucial to the survival of a company.

## Copies by RTDX



- Example: Fixed Interval

```
RealTime Maintain ----- COPY Parameters -----
Command ==>> _____ DB2: S910

DATABASE . . . : *      TABLESPACE . . . : *

COPY OBJECT . . . . V - TS/TP/V(ariable)
COPY TYPE . . . . 2 - F(ull)/I(ncremental)/V(ariable)/2(FLASHCOPY2)
                       R(ecovery based) (*)
COPY SHRLEVEL . . . C - R(eference)/C(hange)
COPY CONCURRENT . . N - Y(es includes QUIESCE before)/N(o)
MERGE INCR COPY . . - Merge incremental copies if this number is reached
COPY CHECKPAGE . . . Y - Y(es)/N(o)
IGNORE COPY YES IX . N - Y(es)/N(o)

(*) : COPY TYPES = F or I are not valid for on-demand processing.
```



The most simple way in triggering backups is a fixed interval (Daily/weekly/... backups) and this is how to set it up within RTDX. The implementation and alignment with Space Management and MODIFY is very easy with this backup concept. Through the ISPF interface some general copy criteria can be defined and...

## Copies by RTDX



- Example: Fixed Interval

```
RealTimeMaintain ----- Batch Calendar -----
Command ==> _____ DB2: S910

NOTE: This calendar is not used for on-demand processing.

DATABASE . . . : *      TABLESPACE . . . : *      UTILITY . . . : CO

Specify day interval to process IMAGECOPY entry for the above object group:
DAY INTERVAL . . . . 1__ - Number of days

Specify day(s) of week by marking the fields below with an S:
DAY OF WEEK . . . . MON TUE WED THU FRI SAT SUN
      - - - - -
Specify day(s) of month by marking the fields below with an S:
DAY OF MONTH . . . . 01 02 03 04 05 06 07 08 09 10
      - - - - -
      11 12 13 14 15 16 17 18 19 20
      - - - - -
      21 22 23 24 25 26 27 28 29 30 31
      - - - - -
```

INTELLIGENCE. INNOVATION. INTEGRITY



... a subsequent calendar panel allows to define the desired interval. Certainly the naming conventions can be defined as well as the device where the copy should be placed. It's as easy as it can be. You can select DASD, Tape or VTS. You can differentiate it for objects or object sizes. GDG are supported as well as Flashcopy technology.

Within large environments, the time and costs for backing up terabytes of unchanged data can't be accepted. Customers often try to solve this by either making the process faster or more granular. Intelligent storage technology like Flashcopy is an opportunity, thresholds are another one.

## Copies by RTDX



- Example: Update-based

```
----- COPY TYPE Variable -----
! Command ==> DB2: S910
! Primary cmd: S(witch off on-demand thresholds)
!
! COPY OBJECT MAX PAGES CLASS 1 1000 CLASS 2 5000 CLASS 3 > CLASS 2
!
! COPY TYPE INCR BATCH ON-DEMAND BATCH ON-DEMAND BATCH ON-DEMAND
! PCT UPDATED PAGES . 10 50 . 5 40 . 0 30
! PCT CHANGED ROWS . 20 . 20 . 0
! COPY TYPE FULL
! PCT UPDATED PAGES . 10 50 . 5 40 . 10 30
! PCT CHANGED ROWS . 20 . 20
! DAYS NO COPY . 1 3 n/a - If updates and no copy in this time
! DAYS UNCOND COPY . n/a n/a - If no updates and no copy
! MIN HOURS . n/a 12 n/a - Minimum hours between copies
! COPY SETTINGS FOR BOTH BATCH AND ON-DEMAND
! NBR INCR COPY . 5 - Full if this number of incremental is reached
! CHECK LOGRANGE . N - Also check logrange if no updates in RTS
! COPY SETTINGS FOR ON-DEMAND
! COPY PEND AS CRIT . Y - Treat any copy pending status as a critical
-----
```

INTELLIGENCE. INNOVATION. INTEGRITY

Reducing the amount of backups is usually very easy by considering to take backups only for changed objects. If an object isn't changed at all the last backup is still good. If only a few pages are changed take an incremental. If most of the data has changed take a full backup. This is the reason why threshold based maintenance can drastically reduces cost and time required for daily database maintenance. This panel shows how to define it and it gives a good impression how granular you can do it. A trigger to merge incremental copies after a number is reached is supported as well as considering exceptions. We allow the user to define different thresholds if the On-Demand processing is used.

## Copies by RTDX



- Example: Dependent Copies (Exceptions)

```
RealTime Maintain ----- REORG Dependent Actions -----
Command ==> _____ DB2: S910

DATABASE . . . : *      TABLESPACE . . . : *

REORG LOG . . . . . N      - Y(es)/N(o)/V(ariable)
  MAX PAGES LOG=YES . . . - Valid only if REORG LOG = V

BEFORE REORG: QUIESCE . N - Y(es)/N(o)  WRITE . . . . . - Y(es)/N(o)
                COPY . . N - Y(es)/N(o)  CONSIDER RI . . . - Y(es)/N(o)
                SHRLEVEL . . . . . - C(hange)/R(ef)
                TYPE . . . . . - F(ull)/I(ncr)
                CONCURRENT . . . . . - Y(es)/N(o)
                CHECKPAGE . . . . . - Y(es)/N(o)

WHILE REORG: COPY . . . Y - Y(es)/N(o)
              RUNSTATS . Y - Y(es)/N(o)

AFTER REORG: COPY . . . N - Y(es)/N(o)  CONCURRENT . . . - Y(es)/N(o)
              RUNSTATS . N - Y(es)/N(o)
              REBIND . . N - Y(es)/N(o)

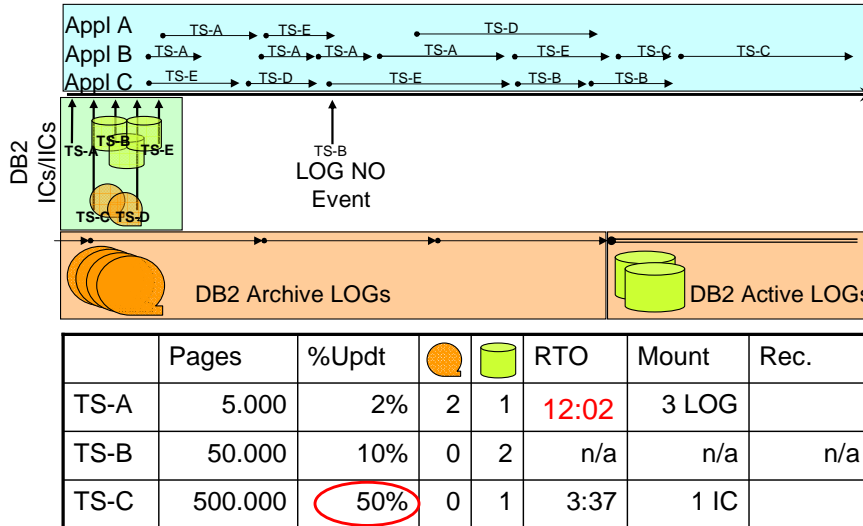
IGNORE COPY YES IX . . N - Y(es)/N(o)
REORG DEPENDENT . . . Y - Y(es to avoid redundant copies)/N(o)
```

INTELLIGENCE. INNOVATION. INTEGRITY

Exceptions and dependant actions are important triggers for backups. Whenever an exception is encountered DB2 requires a new backup independent from the amount of data that is affected. This is because DB2 cannot determine which data has changed. Usually the reason for this is a LOG NO event. If this is encountered the object is placed in copy pending.

If you do REORGs, copies are always suggested. Required options are always predefined as well as options that aren't allowed are checked when defining a strategy as well as at execution time. Taking a SHRLEVEL CHANGE REORG for example force the copy definition for an inline. These features make the definition and processing of database maintenance an easy task even for people not in depth with new features.

# RTO-Based Copies



	Pages	%Updt			RTO	Mount	Rec.
TS-A	5.000	2%	2	1	12:02	3 LOG	
TS-B	50.000	10%	0	2	n/a	n/a	n/a
TS-C	500.000	50%	0	1	3:37	1 IC	

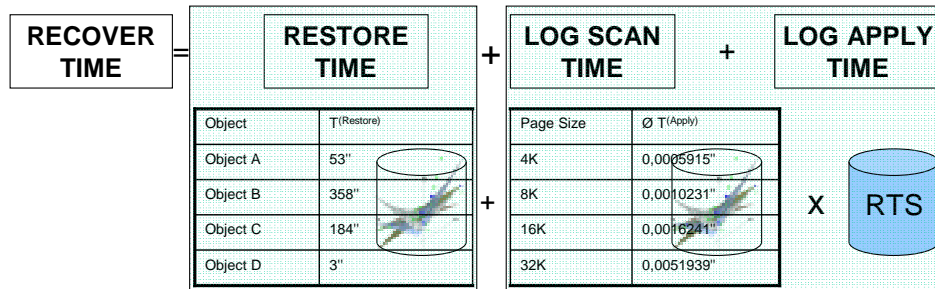
INTELLIGENCE. INNOVATION. INTEGRITY

Even if your database maintenance solution detects an exception like copy pending (which most solutions usually do) it's not acceptable to wait until the next database maintenance cycle. Today, this usually requires manual effort. Having an On-Demand solution triggers a backup when needed. This can also be true for any other copy trigger like regular thresholds. Usually the thresholds for the objects in a database are not all exceeded at the same time. And usually this is not only when the maintenance tool runs. This is the big advantage of an On Demand database maintenance. If a backup is required it should be taken ASAP instead of waiting until the next batch window. This is often managed by complex dependencies within the job net definition of a scheduler, triggering backups for objects after transactions that affects them.

Diving into the details of DB2 backup and recovery opens up a strategy that can make this complex-, time-intensive work obsolete.

This is the DB2 point of view of transactions affecting some database objects. It can represent a typical environment, but it is usually different with situations like month end, quarters end, years end or just increased workload. Additionally the recovery situation usually differs depending on the point in time. With a focus on the log ranges, we see something different than focusing on the updated pages. Transactions can span several archive logs with subsequent updates of one and the same single page, but the updated pages indicator wont tell you that. On the other hand you can have 50% updated pages that are all covered by an active log. Which recovery will take longer and which object will get a copy based on the common thresholds. And what does that cost you for such a benefit? Automated, threshold based database maintenance solutions will backup the wrong object as it only checks the updates. This won't give you any benefit and it wastes a lot of storage as especially big objects are copied too often. Unfortunately this makes a gamble of a recovery situation. If you are lucky an outage may be as expected, if you are unlucky an object is affected that has a lot of updates spread of different LOG datasets – this is often considered as Murphy's Law.

## RTO-Based Copies



INTELLIGENCE. INNOVATION. INTEGRITY

This shows how copies based on your Recovery Time Objectives (RTO) are triggered. Based on a systems analysis we have to do once, we have key performance indicators (KPIs) of the hardware in place. This data along with RTS and log distribution data gives us the ability to do RTO-based copies. These copies use the interpolated RTO itself as a copy trigger. You only specify how much time you can except for a recovery and if it is exceeded we trigger a copy. Certainly this process only makes sense when an On Demand database maintenance solution is in place.

The threshold check considers the restore time of the last backup and the subsequent updates. The tables gives an example about that. We have object A to D along with their current copy restore time. Additionally an average apply time according to the page size provided by our KPIs and is calculated by our interpolator also. Along with this data combined with the RTS, it is possible to monitor and trigger RTO base copies within the on demand processing.

Here is an example.

If 2 minutes for a recovery of object A are defined and restoring it's last backup already takes 53 seconds of this then one minute and 7 seconds are left for subsequent log processing. RealTime DBAExpert knows about that as well as about the average log apply time. It monitors the RTS and suggests to triggers a copy when more pages have been changed than they can be applied within one minute and seven seconds.

As I said it suggests. See the following slides how the actual execution is controlled.

## Copies by RTDX



- Example: RTO-Based

```
----- COPY OBJECT/COPY TYPE -----
Command ==> _____ DB2: S910

Primary cmd: S(witch off on demand thresholds), R(ecovery Time Objectives)

          BATCH ON DEMAND
COPY THRESHOLDS      REG  CRIT
RTO EXCEEDED . . . . Y   Y   T   - Y(es), N(o), or T(olerance)
UNREALISTIC RTO . . . 25  25  ___ - Percent of ideal recovery time
INDEX COPY REQD . . . Y   Y   N   - Y(es to check indexes), N(o)
COPY EXCEPTIONS
COPY PENDING . . . . n/a n/a Y   - Treat copy pending as critical
BACKUP MISSING . . . W   W   E   - W(arning) or E(rror and Warning)
LOG MISSING . . . . W   W   E   - W(arning) or E(rror and Warning)
COPY INTERVAL
MIN HOURS . . . . . n/a 36  ___ - Minimum hours between copies
DAYS NO COPY . . . . . ___ ___ - Full if updated but no copy
UNCOND COPY . . . . . ___ ___ - Full if not updated and no copy

COPY SWITCH TO FULL FOR BOTH BATCH AND ON DEMAND
NBR INCR COPY . . . . 70  ___ - Full if this incremental count is reached
COPY APPLY PCT . . . . 75  ___ - Full if copy apply time reaches this RTO pct
PCT UPDATES PAGES . . 30  ___ - Full if updates reach this percentage
```

INTELLIGENCE. INNOVATION. INTEGRITY

The most recovery oriented backup is triggering them RTO based. The Recovery Time Objective is the time that it takes to recover an object an is the common target of availability SLAs.

This screen shot shows how RTO based copies are defined within RealTime DBAExpert. Remember the RTOs defined within Recovery HealthChecks online interface? This is the counterpart of it, to set them within RealTime DBAExpert, our On-Demand database maintenance solution. A cross reference link is available via primary command R.

In general the only thing that has to be defined is to trigger copies when a RTO is EXCEEDED. Actually there are two possible settings for that. Y(es) means that a copy should be triggered when the RTO is no more met and T(olerance) forces RealTime DBAExpert to wait until the tolerance is exceeded either. Because RealTime DBAExpert takes several additional factors like the current workload into account, there are two possible assignments for ON Demand processing. The REGULAR assignment checks for those additional factors the CRITICAL assignment assures immediate execution via the dynamic interface of the job scheduler in place. The configuration of this example therefore leads to a copy if the RTO is exceeded, but still not violating the tolerance. Since this is a regular request, it is only executed when the workload allows this. Having T assigned to the critical on demand trigger forces RealTime DBAExpert to trigger a copy immediately when RTO and even the tolerance is violated. Shops that have to assure that a potential recovery never takes longer than a defined SLA use this critical trigger that assures that a recovery never takes longer independent from any change in workload update activity

## Copies by RTDX



- Example: RTO-Based

```
----- COPY OBJECT/COPY TYPE -----
Command ==> _____ DB2: S910

For the recovery service levels (RSL), you have to define
your recovery time objectives (RTO) and related tolerances:

          SIZE          RTO          TOLERANCE
-----
CLASS 1 . . . . . 10000 KB      5 min  20 5 .. 50 %
CLASS 2 . . . . . 50000 KB     8 min  20 5 .. 50 %
CLASS 3 . . . . . 100000 KB    15 min  10 5 .. 50 %
CLASS 4 . . . . . 500000 KB    40 min  10 5 .. 50 %
CLASS 5 > CLASS 4 . . . . . 60 min  10 5 .. 50 %
```

INTELLIGENCE. INNOVATION. INTEGRITY



To verify an interpolated RTO against the defined or expected one (the threshold), we provide five size ranges for the objects of a database. Along the desired RTO is defined along with a tolerance. The tolerance allows to wait if a violation is within a small range. For example, if a RTO of five minutes is defined for an object up to 10.000KB and it would currently take five minutes and 2 seconds to recover, RTDX will not trigger a copy as it is still within the tolerance of 20 percent.

RTO based copies are best fitting for cost savings as well as for recoverability. However if fixed time interval or update rate thresholds are preferred we still support this – opportunities only.

## REORG Utility



- Goal:

*>>REORG improves access performance and reclaims fragmented space.<<*

INTELLIGENCE. INNOVATION. INTEGRITY



Taking Backups is the most important database maintenance task from one perspective. The other one is optimizing performance. To assure an optimal access path requires to do REORG, RUNSTATs and REBIND (the 3Rs of DB2 like Roger Miller always says).

Doing a REORG additionally reclaims fragmented or unused space; thus, the goal of a REORG is to improve performance and to save wasted space. However, a REORG also costs a lot of CPU and I/O and it requires more or less exclusive access to the data.

## REORGs by RTDX



```
RealTimeMaintain ----- REORG Thresholds -----
Command ==>
DB2: S910
DATABASE . . . : *      TABLESPACE . . . : *
RTM      RTM
REGULAR  CRITICAL
REORG TABLESPACE
MIN PAGES . . . 32      - No reorg if OBJECT is smaller
PCT CHANGED . . .    - Percentage changed rows
PCT INDREF . . . 10     - Percentage FARINDREF+NEARINDREF
PCT UNCLUSTINS 10     - Percentage unclustered inserts
MASS DELETES . . . 0    - Number of mass deletes
REORG LOB
PCT DISORGLOB . 10     - Percentage disorganized LOBs
REORG INDEX
PCT CHANGED . . .    - Percentage changed rows
PCT APPENDED . 10     - Percentage appended rows
PCT LEAFFAR . . . 10  - Percentage LEAFFAR
PCT PSEUDODEL . 10   - Percentage PSEUDO_DELETED_ENTRIES
MASS DELETES . . . 0  - Number of mass deletes
LEVELS CHANGED 0     - Number of indextree level changes
```

INTELLIGENCE. INNOVATION. INTEGRITY

Deciding whether and when to do REORG is object dependant. Indexes usually benefit from a REORG LOBs usually don't. If a table in a tablespace is accessed sequentially a REORG can improve performance. If it is accessed through an index a REORG won't necessarily improve performance. Exceptions are again an important indicator. This is also true for the ADVISORY REORG PENDING state which you will get after doing online schema changes.

A REORG should always be done SHRLEVEL CHANGE. A good set up the switch options assures that you can do REORGs without compromising availability. Running a REORG against partitioned objects requires to take care of NPIs as they will not be available when processing on the partition level.

A good understanding of z/OS extent management like extent limits and extent consolidation as well as storage technology improvements like Parallel Access Volumes is important when deciding whether a REORG should be done from the space management perspective.

We consider the aspects and support setting up different thresholds for indexes, table spaces and LOBs. We consider exceptions through the DB2 Database Exception Table (DBET) and we assure availability by capsulation of utilities as well as BUILD2 avoidance.

This example shows the REORG thresholds like we have already seen for the COPY utility. The ISPF panels are all following the same flow. Object groups based on wildcards, or fully qualified objects are selected. Afterwards, thresholds and utility parameters are defined. Everything follows the IBM official DB2 thresholds and is assigned by the official recommendations. Detailed help panels explain the

## RUNSTATS Utility



Goal:

*>>The DB2 Optimizer gets the current statistics from DB2 catalog for the columns and tables identified in the SQL statements to select an access path.<<*

INTELLIGENCE. INNOVATION. INTEGRITY



The next two Rs to address are RUNSTATS and REBIND. The DB2 optimizer is a very good component of a DB2 database. It's decisions are usually resulting in the best access path and especially the V8 and DB2 9 improvements made it even better and more robust. The optimizer's decisions are based on DB2 catalog statistics that are set by the RUNSTATS utility. This makes clear that the optimizer decision can only be as good as the data that is provided.

It is important to understand that the optimizer doesn't make it's decisions based on RTS. It's still the DB2 catalog that is consulted for object statistics. Effectively this is an additional advantage of RTS as statistics for threshold checking and statistics for the critical optimizer source are now separated.

## RUNSTAT AVOIDANCE\* by RTDX



- Distinctive Sizes

```

Realtime Maintain for DB2 z/OS - RUNSTATS Avoidance Settings - Row 1 to 3 of 3
Command ==> _____ Scroll ==> CSR
                                         DB2: S910

Primary cmd: I(nsert), S(ategy overview)
Line cmd: S(elect), I(nsert)
DATABASE . . . : MIVPV810  TABLESPACE . . . : *

BATCH OPTIONS
REBIND AFTER . . . . . Y          - Y(es)/N(o)
RUNSTATS AVOIDANCE      NON-PART  PART      LOB          MEMBER CLUSTER
STRATEGY . . . . . AVOID

Select RUNSTATS avoidance strategies. All sizes are row counts:

STRATEGY  ! DOWN  ! VOLATILE ! STEPS
          ! DELAY % ! MIN MD/LR ! MIN  INTERMEDIATE  FACT MAX
----- ! ----- ! ----- ! -----
_AVOID    ! 48H  25 ! 5K   Y ! 100K  150K,1500K      3   80M
_DEFAULT  ! 1D   25 ! 5K   Y ! 20K   50K,150K,500K,1500K  3   -
 OftEN    ! 1H   10 ! -    N ! -     1K,3K,5K,10K      1.4 -
----- ! ----- ! ----- ! -----
    
```

\* (US Patent No. 11/0088,711)

INTELLIGENCE. INNOVATION. INTEGRITY

Realtime DBAExpert has a RUNSTATS AVOIDANCE feature that reduces RUNSTATS up to 95%. This is realized by switching from technical thresholds to Optimizer-Controlled Maintenance. “Optimizer-Controlled” are optimizer relevant utilities that are not triggered because of update rates for example, but of a thresholds that check/consults the optimizer for a resulting improvement. If a RUNSTATS will not lead to a new optimizer decision we don’t generate it.

Some circumstances force DB2 to do a AUTO REBIND this can be risky if an access path results that isn’t desired. IBM focuses on “old” plans/packages with this to get rid of old code.

Stand-alone, RUNSTATS only leads to a new access path for dynamic SQL as the dynamic statement cache gets invalidated. Static SQL requires a subsequent REBIND.

This is an example of how to set up Runstats Avoidance. Mainly its based on size classes. If a new class is reached, the optimizer will probably change the access path and a RUNSTATS is generated. A subsequent REBIND can be selected (follow the next slides about our REBIND check).

## RUNSTAT AVOIDANCE\* by RTDX



- Volatile Detection

```
Realtime Maintain for DB2 z/OS - RUNSTATS Avoidance Strategy - Row 1 to 4 of 4
Command ==> _____ Scroll ==> CSR
DB2: S910

Primary cmd: I(nsert)
Line cmd: I(nsert), D(elete)

STRATEGY . . . . . DEFAULT

VOLATILE: MD/LR . . . Y - No down scale for single table TS with
          MIN . . . 5 K - mass deletes, load-replace, or low row-count
DOWN SCALE: PERCENT . . 25 - Step tolerance before scaling down
          DELAY . . . 1 D - Minimum delay time before scaling down (*)
STEPS: MIN . . . 20 K - No RUNSTATS for smaller table space row counts
       FACTOR . . . 3 - Used to compute steps above the given ones
       MAX . . . M - No RUNSTATS for bigger table space row counts
          INTERMEDIATE UNIT (*) : Valid time units are M(inute),
          ----- H(our), D(ay), W(eek) and Y(ear).
          -----
          ----- 50 K
          ----- 150 K
          ----- 500 K
          ----- 1500 K
          -----
```

\* (US Patent No. 11/0088,711)

INTELLIGENCE. INNOVATION. INTEGRITY

RUNSTATS AVOIDANCE additionally provides VOLATILE CONTROL. Thereby we automatically detect volatile tables based on their usage.

In general, a RUNSTATS should always be as good as the fresh statistics from the optimizer. However, the truth can be different for example if a RUNSTATS is triggered for volatile objects when it is triggered when the object is currently empty or very slow. In DB2 V8, the volatile indicator for those objects is introduced and assures index access in any situation. Those objects don't need RUNSTATS following the common strategy.

This example shows the VOLATILE CONTROL part of Runstats Avoidance. We detect volatile indicators like mass deletes and load replace.

We recommend RUNSTATS AVOIDANCE for database maintenance, because of the resulting savings without compromising performance. However, if the traditional way of triggering RUNSTATS is preferred, this is still supported – opportunities only!

## ADVANCED RUNSTAT by RTDX



- COLGROUP support

```
RealTimeMaintain ----- COLGROUP ----- Row 1 of 3
Command ==> _____ Scroll ==> PAGE
                                      DB2: S91A
Primary cmd: L(ocate TABLE NAME), I(nsert), P(rint)
Line   cmd: D(elete), I(nsert), S(elect)

  TABLE NAME          FREQ  MBL  HIST  MIN
  COLGROUP              -----
-----
PTFD0420.S420PTFT001      42   B    Y    0
ALCCYL, ALCKB, ALCP6, ALCP6, ALCTYPE. ....
PTFD0420.S420PTFT008      42   B    Y    0
ALTERID, EXCLUDED. ....
R510.R510T01              10   M    Y   12
DBNAME, TSNAME, INDEXSPACE. ....
-----
```

INTELLIGENCE. INNOVATION. INTEGRITY

The RUNSTATS option COLGROUP collects correlation statistics for the specified column group. RUNSTATS can collect statistics on any single column or set of columns. If specified, **Cardinality** - the number of distinct values in the column or set of columns is determined.

If you want RUNSTATS to also collect distribution statistics, specify the FREQVAL option with COLGROUP. RUNSTATS then also collects the **Frequency** - the percentage of rows in the table that contain a value for a column or combination of values for a set of columns.

## ADVANCED RUNSTAT by RTDX



- COLGROUP support

```
RealTimeMaintain ----- COLGROUP Details-----
Command #44>
DB2: 591R
Enter, view, or change data. Press ENTER to update or PF3 to exit.

TABLE CREATOR . . . PTF09429
TABLE NAME . . . S42OPTTR01
-----
FREQVAL COUNT . . 25 . . - 0 .. 1098
FREQVAL PARAM . . . B . . . - M(ost)/L(east)/B(oth)
HISTOGRAM STATS . . Y . . . - Y(es)/N(o)
NUM QUANTILES . . 48 . . - 1 .. 100
MIN HOURS . . . 0 . . . - 0 .. 9999 Minimum hours between RUNSTAT runs
LAST USED . . . 2008-08-18-15.56.39
COLGROUP . . . . . @LCCYL,@LCKB,@LCPG,@LCSPC,@LCTYPE
```

INTELLIGENCE. INNOVATION. INTEGRITY

**COUNT integer** Indicates the number of frequently occurring values to be collected from the specified column group. For example, COUNT 20 means that DB2 collects 20 frequently occurring values from the column group. This can be done for the most, the least or even for both values.

**Better data for the DB2 9 optimizer with histogram statistics** Data distribution statistics are important for query optimization. DB2 chooses the best access path based on cost. The basic foundation of cost is predicate selectivity estimation, which relies heavily on data distribution statistics. Prior versions of DB2 for z/OS rely on frequency statistics that are collected (through the RUNSTATS utility) on single values, either from a single column or from multiple columns. Now, V9.1 supports histogram statistics, which provide better data for the optimizer. With histogram statistics, DB2 can improve access path selection by estimating predicate selectivity from value-distribution statistics that are collected over the entire range of values in a data set, unlike frequency statistics.

RUNSTATS TABLESPACE will ignore HISTOGRAM when processing XML table spaces and indexes. When you run RUNSTATS INDEX, histogram statistics can only be collected on the prefix columns with the same order. Key columns with a mixed order are not allowed for histogram statistics.

NUMQUANTILES Indicates how many quantiles that the utility is to collect.

## RUNSTATS Utility with REBIND control option



- Goal:
  - PreCheck – ask the Optimizer for a potential improvement  
→ Reduces & Secures REBINDs
  
- Optimizer-Controlled Maintenance (incl. Interface to X-Tune + DSC for explain analysis)

INTELLIGENCE. INNOVATION. INTEGRITY



Triggering or automating REBINDs is a question of philosophy. IBM says it's not a question whether to REBIND or not, but when. We are committed to assure production availability, thus pre-checking REBINDs is the way we do it. If a REBIND doesn't lead to an improvement it is won't be executed. An additional X-Tune interface allows further analysis if a REBIND would lead to access path degradation. This covers static SQL as well as dynamic SQL through a DSC capturing facility...

## Controlled REBINDs by RTDX



```
ImpactExpert for DB2 z/OS ----- Impact Filter Summary -----
Command ==>
DB2: S910
Primary cmd: END, J(obs)
Enter your selections. Then press ENTER to see the impact summary.
COLLECTION. . . . . *          PACKAGE. . . *
TIME FROM . . . . . 2008-05-06-13.37.47
TIME TO . . . . . 2008-05-06-13.37.47
INPUT SOURCE. . . . . R - A(11) / B(INDS) / R(EBINDs)
Select (x) one of the following from the impact summary to view details.
_ PACKAGES ANALYZED : 630          STATEMENTS ANALYZED : 3758
NUMBER OF PACKAGES.          NUMBER OF STATEMENTS.
- IMPROVED. . . . . 53          IMPROVED. . . . . : 108
- X WORSENERD. . . . . 17      WORSENERD. . . . . : 20
- UNCHANGED . . . . . 421      UNCHANGED . . . . . : 3567
- CHANGED . . . . . 28         CHANGED . . . . . : 42
- OTHER . . . . . 0
```

Processing Summary

INTELLIGENCE. INNOVATION. INTEGRITY

...using the online interface for further analysis is entered by an impact summary for all packages matching a filter criteria.

All results are nicely categorized into the number of packages whose access paths have been improved by a REBIND, as well as the number of packages whose access paths would have been unchanged or worsened. Here we can see, no big surprise, that 90% of the access paths for statements are unchanged. We waste absolutely no time looking at improved or unchanged categories and can focus our effort on the few packages that have degraded access paths.

We can select these packages with command S and they are listed on the next foil. Look now to see why these might be of interest. . .

## Controlled REBINDs by RTDX



```
ImpactExpert for DB2 z/OS ----- Worsened Packages ----- Package 9 from 26
Command ==> _____ Scroll ==> CSR
                                     DB2: S910
Primary cmd: END, L(ocate) -Package-
Line      cmd: S(statement), C(create REBIND)

      JOB SUBMIT TIME      COLLECTION/      PACKAGE/      STATEMENTS
      _____      _____      _____      _____
      _____      _____      _____      _____
      2008-05-06-09.53.58  IQA_COLLECTION_510  ADB2M101  NO      3      3      0  R
      -
      2008-05-06-09.53.58  IQA_COLLECTION_510  ADB2M101  NO      3      3      0  R
      -
      2008-05-06-09.53.58  IQA_COLLECTION_510  ADB2M101  NO      3      3      0  R
      -
      2008-05-06-09.53.58  IQA_COLLECTION_510  ADB2M101  NO      3      3      0  R
      -
      2008-05-06-09.36.12  IQA_COLLECTION_510  XDB2CN22  NO      0      1      0  R
      -
      2008-05-06-09.36.12  IQA_COLLECTION_510  XDB2CN01  NO      0      1      0  R
      -
      2008-05-06-09.36.12  ADB20410            SQLZU102  NO      0      1     10  R
      -
      2008-05-06-09.36.12  SQLZ0120            SQLZU102  NO      0      1     10  R
      -
      2008-05-06-09.36.12  ADB20410            PARSTYPE  NO      1      1      1  R
      -
      2008-05-06-09.36.12  MDB20330            O2DBIX    NO      4      2      0  R
      -
      2008-05-06-09.36.12  IQA0510             BATMN200  NO      1      1      0  R
      -
      2008-05-06-09.36.12  IQA_COLLECTION_510  ADB2SET1  NO      7      3      0  R
      -
      2008-05-06-09.36.12  IQA_COLLECTION_510  ADB2M101  NO      3      3      0  R
      -
      2008-05-06-09.36.12  ADB20410            ADB2DSTS  NO      0      3      0  R
      -
      2008-05-06-09.36.12  ADB20410            ADB2DSTB  NO      2      3      0  R
      -
      2008-05-06-09.36.12  ADB20410            ADB2DSTB  NO      0      3      0  R
      -
```

Statement overview for each package in the list

INTELLIGENCE. INNOVATION. INTEGRITY

The packages are listed for which no REBIND was done. For each package in this list, you see an overview of the number of statements whose performance could have been worsened, as well as the number of statements that might have been improved if a REBIND had been done.

In this case, even though some statements might have improved, **ImpactExpert is always better safe than sorry.**

To help you with this decision, you can drill down and take a close look at the statements for a specific package.

## Controlled REBINDs by RTDX



```
ImpactExpert for DB2 z/OS ----- REBIND Impact ----- Statement 1 from 23
Command ==> _____ Scroll ==> CSR
                                     DB2: S910

Primary cmd: END, SE(tup Analyze),
Line   cmd: S(elect), A(nalyze), D(ynamic Analyze), E(dit and Analyze)

Timestamp . . 2008-05-06-09.36.12.250000
Collection . . ADB20410
Package . . . ADB2DSTP
Version . . . 2007-11-05-17.18.16.515822
```

		BAD ACCESS TYPES			
STMTNO	IMPACT	BEFORE REBIND	WITH REBIND	COST	HINT
—	3478	EQ	SORT	SORT	29.56
—	3496	EQ	LP	LP	23.34
—	3516	EQ	SORT	SORT	23.34
—	3537	EQ			08.34
—	3559	EQ	SORT	SORT	23.34
S	3730	WRS	SORT	TS,SORT	124.15
—	5120	EQ			10.12
—	7584	EQ			0.18
—	7597	EQ			0.55

Presents the REBIND impact for each statement in the package

INTELLIGENCE. INNOVATION. INTEGRITY

For each statement, the REBIND IMPACT along with the *before* and *after* access path types are presented.

We only detail the bad types in this list. A BLANK indicates a good access type.

We see that the access type for this statement is a SORT, but would have changed to a SORT with a table space scan if rebound. If we select this statement . . .

## Controlled REBINDs by RTDX



```
ImpactExpert for DB2 z/OS ----- Comparison ----- LINE 0000077 COL 001 080
Command ==> _____ Scroll ==> CSR
                                         DB2: S910

Primary cmd: END, CAN(ce1)
Collection . ADB20410          StmtNo . . 3730
Package. . . ADB2DSVD          Stmtcost . 124.15

Statement Text + Access paths
-----
SELECT
  MAX ( X_HIST_TIMESTAMP )
FROM
  PARSVTAB )
ORDER BY
  X_NAME , X_SEQNO
FOR FETCH ONLY
Access path before REBIND -----! Access path with REBIND -----
TABLE          QB PN AC MA ME IX ! TABLE          QB PN AC MA ME IX
INDEX          TY CO TH ON !   INDEX          TY CO TH ON
-----
ADB2T071       1 1 I  1 0 N ! ADB2T071       1 1 R  0 0 N
  ADB2X0711    !
                1 2  0 3 N !
ADB2T071       2 1 I1 0 0 Y ! ADB2T071       2 1 I1 0 0 Y
  ADB2X0711    !   ADB2X0711
```

Presents the access path comparison

INTELLIGENCE. INNOVATION. INTEGRITY

. . .we can scroll through ALL of its statement text and see the actual access paths, which in this case still has the SORT but

the access path would have changed  
**from** a single column matching index  
**to** a table space scan

This gives you the opportunity to see why the rebind was not recommended. There is a lot of other investigative information that we provide but I don't have time to show that now.

We recommend to use this feature and gain all the improvements of REBINDs without the risk of a degradation. However if REBINDs are not desired or triggered for all affected PLANS and PACKAGES that should be this is still supported – opportunities only.

## Automated Utility Control by RTDX



```

BatchControl for DB2/Utilities ----- Rule DB2 Abend ----- Row 6 from 27
Command ==> _____ Scroll ==> PAGE
                                     DB2: S910
Primary CMD : I(nsert), S(ort), F(ilter), L(ocate), P(rint)   Script DEFAULT
Line CMD    : D(elete), U(pdate), I(nsert), R(estart), C(leanup)

CMD  UTILITY          SHRL PHASE    MVS-CC  MVS-RC  WAKEUP  RESTART  CLEANUP
-----
- REORG INDEX        NONE BUILD    S04E   *      NO      *        *
  DB2-NAME: *
  DESCRIPT: ABEND DURING REORG INDEX (BUILD PHASE)
-----
- REORG TABLESPACE *   *      S04E   *      NO      *        *
  DB2-NAME: *
  DESCRIPT: ABEND DURING REORG TABLESPACE
-----
- REORG TABLESPACE *   BUILD2  S04E   *      NO      *        *
  DB2-NAME: *
  DESCRIPT: ABEND DURING REORG TABLESPACE (BUILD2 PHASE, ONLINE REORG ONLY)
-----
- REORG TABLESPACE *   SWITCH   S04E   *      NO      *        *
  DB2-NAME: *
  DESCRIPT: ABEND DURING REORG TABLESPACE (SWITCH PHASE, ONLINE REORG ONLY)
-----
- REORG TABLESPACE NONE RELOAD  S04E   *      NO      *        *
  DB2-NAME: *
  
```

INTELLIGENCE. INNOVATION. INTEGRITY

This panel shows the utility control rule database. As we encapsulate the utilities at runtime, we are able to resolve any utility error or abend and to assure availability of the objects. Any problem that is unique and identifiable through either MVS condition code, return code or the utilities reason code can be handled by this component. We ship the rule database with all commonly known errors/abends. Any customization is possible, like adding new problems or change the resolving processing. New errors that should be corrected automatically just have to be defined by the utility, the phase and the error. Any correction can be triggered by selecting from utility restart, executing DB2 commands or even executing external procedures and programs.

## Space Management by RTDX



- Housekeeping
  - After MODIFY
  - Space downsizing
- Transparent Extent Management
- Proactive space monitoring
  - LPS
  - Storage groups
- Manage all maintenance dependant spaces
  - Work data sets
  - Sort data sets
  - Copy data sets
  - Mapping table

INTELLIGENCE. INNOVATION. INTEGRITY

My last topic for database maintenance is database space management. We provide housekeeping like deleting backups after a modify as well as downsizing of spaces due to fragmentation or due to the fact they shrink.

We also provide space monitoring features like extent monitoring and adjustment, Linear Page Set Audit or STORAGE GROUP Audit. This allows proactive work on growing objects or avoiding to hit over all space limit within store groups for DB2 objects, backups or LOGs.

Whenever a dataset is required, we calculate the size and allocate it. This is also true for GDG and Mapping Tables that have to be created. Any mixture of different devices is handled and considered for stacking and utility job distribution.

## Extent Management by RTDX



```

RT-ExtentManager ----- Extent Summary ----- Row 1 of 15
Command ==> _____ Scroll ==> CSR
DB2: S910

Primary cmd: L(ocate DATABASE), S(ort), F(ilter), P(rint)
Line cmd: E(xpand), F(irst details), L(ast details), R(eporting)

  DATABASE  DBID  SPACE  PSID/ PART I  DS  NUM  NUM !  .... CURRENT .....
            ISOBID  X  NUM ALTR WARN !  EXT  SQTY KB  ALLOC KB
-----
- DSNDB04    4  DSNR1VZS   50  0    1    1    0 !    3    12    144
- DSNDB04    4  DSNR130T  120  0 Y    1    1    0 !    3    12    144
- DSNDB04    4  PLAN1O43  177  0    1    1    0 !    6   2880   3312
- DSNDB04    4  PLAN1SS3  180  0 Y    1    1    0 !    4    12    288
- DSNDB04    4  RTMR1V9V  386  0 Y    1    0    0 !    2   7200   7920
- DSNDB04    4  RTMR13TA  396  0 Y    1    0    0 !    2   7200   7920
- M320TEST   264  M320S30   88  0    1    0    0 !    2  10284  10368
- M410D001   298  M410S06   33  0    1    0    0 !    2   3600   5760
- M410D001   298  M410S14   69  0    1    0    0 !    3    720   2160
- M410TEST   310  M410S02    7  0    1    0    0 !    2   9456  12240
- M410TEST   310  M410S06   33  0    1    0    0 !    2   9236  11520
- M410TEST   310  M410S14   69  0    1    0    0 !    2   9296  10080
- M410TEST   310  M410S29  144  0    1    0    0 !    2   9264   9312
- M410TEST   310  M410S30  147  0    1    0    0 !    2   9236   9312
- M410TEST   310  M410X021  10  0 Y    1    0    0 !    2   9364  10800
-----

```

INTELLIGENCE. INNOVATION. INTEGRITY

The details about Space Management is again provided through our online interface. This panel shows extent management details. If it detects that a data set will reach 251 (or a user given threshold) of extents BEFORE reaching its maximum data set size it dynamically issues an ALTER SECQTY based upon six user given progressive adjustment values. This enables a very granular SECQTY allocation and reduces the chance of very large secondary extents being allocated, but being practically empty.

## Extent Management by RTDX



```
RT-ExtentManager ----- Extent Expand Indexspace ----- Row 1 of 1
Command ==> _____ Scroll ==> CSR
                                     DB2: S910

Primary cmd: L(ocate TIMESTAMP), S(ort), Z(oom index), F(orecast), P(rint)
Line      cmd: D(etails), O(ptions), R(eporting)

  DATABASE . : DSNDB04      DBID . . :      4      MAX ALLOC . : 2097152 KB
  INDESPACE : PLAN1S$3     ISOBID . :     180     MAX EXTENTS :    251
  PARTITION . : 0         DSNUM . . :      1     MAX VOLUMES :     59
  IXCREATOR . : MDB2      IXNAME . . : PLAN_TABLE_INX_SE

          ALTER | . AUDIT SQTY . |
TIMESTAMP      EXT SQTY KB SQTY KB E | WARN % LASTEXT | ALLOC KB VOLS
-----|-----|-----|-----|-----|-----|-----|-----|
- 2008-04-15-14.15.57  4      12  2880 |      400      48 |      288      1
-----|-----|-----|-----|-----|-----|-----|-----|
```

INTELLIGENCE. INNOVATION. INTEGRITY



This is using another drill down feature provides more details.

## LPS Audit by RTDX



Object type:	Maximum # of data sets:
LOB TS without DSSIZE	254
Non partitioned TS (except ...)	32
Non partitioned IX on TS with: <ul style="list-style-type: none"> <li>• LARGE or DSSIZE</li> <li>• No more than 254 partitions</li> </ul>	254
Non partitioned IX on TS with: <ul style="list-style-type: none"> <li>• LARGE or DSSIZE</li> <li>• More than 254 partitions</li> </ul>	If PIECESIZE is used: 4096 If no PIECESIZE is used: if DSSIZE <= 32 GB 4096 if DSSIZE > 32 GB 2048
Non partitioned IX	32
Partitioned IX	1 (ignore)
Partitioned TS	1 (ignore)

INTELLIGENCE. INNOVATION. INTEGRITY


This is how we monitor the DB2 Linear Page Sets. Based on a customizable threshold we issue a warning to the console about exceeded limits. Thereby an adjustment like partitioning of a non part TS can be planned in advance without the risk of an outage.

## Optimizer-Controlled Maintenance - Do it the cost based way



- ROI and TCO
  - Nobody runs a data center for itself
    - All processes of RTDX are aligned to the business needs
  - Main goals
    - Production Assurance
    - Cost Savings
    - Exploit existing technology you already paid for
    - Intuitive and integrative



- See customer experiences e.g.  **Lufthansa Systems**
- Get RTDX ROI Tool to forecast YOUR savings

INTELLIGENCE. INNOVATION. INTEGRITY



Now we discussed RTO based Copies, REORGs that improve, RUNSTATS AVOIDANCE and prechecking REBIND. They all follow the same goal: Optimize process and align benefits and costs. Many of our customers are addressing cost savings as an important topic for their business. Sometimes there are areas of potential savings that are not expected. New database maintenance has always been a great opportunity to reduce costs for database operation. We focus on production assurance, best performance and recoverability with Realtime DBAExpert, but we also consider the costs. Integrating new or improved technology, optimizing processes and a critical cross check of the results of the maintenance processes lead to the features described above and saved our customers already a lot of real money. But important tasks like database maintenance of a production system that is available 24x7 are not changed in a big bang. Therefore we provide these features as opportunities and allow a smooth transition for their usage.

## Summary



- RealTime DBAExpert
  - Is easy to implement and easy to use
  - Follows YOUR needs
  - Covers Ad Hoc, Batch and On Demand Utilities
  - Allows to keep current strategies, but
  - Provides smooth transition/exploitation of technology enhancements
  - Aligns costs and benefits

INTELLIGENCE. INNOVATION. INTEGRITY



RealTime DBAExpert is the best solution for all maintenance strategies and lets you decide flexible what fits your needs. It strictly follows the IBM standards and easy implements into your environment, interfacing to your existing solutions and following your naming conventions. It assures that new features can be used without a risk and gives you great opportunities to improve existing traditional maintenance procedures to increase performance and recoverability with a strong focus on cost reduction.

Can you maintain your DB2 database on the fly?

