



CDB Software, Inc.

DB2 V8 FAQs

DIPSIES EXPLAINED

Q. What is a DPSI?

A. A “Dipsie” is a Dependent Partition Secondary Index (DPSI). This is a new type of index in DB2 V8. It is a Non-Partitioning Index (NPI), but there is a separate index dataset for each partition instead of a single large dataset for an NPI. For example, if you have defined a 200 partition tablespace with 2 DPSIs, then you will have 400 indexes in addition to any partitioning index (which, in DB2 V8, is optional).

Q. What are the benefits?

A. DPSIs solve the BUILD2 problem with IBM Reorg, especially Read-Write Reorg (REORG SHRLEVEL CHANGE). DPSIs allow IBM to eliminate the BUILD2 phase. In some circumstances, DPSIs may be more efficient than a single large NPI, but queries or updates must supply the partitioning key value to take advantage of a DPSI. Inserts should be more efficient because the partitioning key columns are always provided. Performance of deletes and updates will improve only if the partitioning columns are provided. Otherwise, the deletes and updates must access all DPSIs, severely impacting performance.

Q. How do they impact your applications?

A. DPSIs will drastically degrade application performance. As an example, suppose an application tablespace has 200 partitions using the customer number as the partitioning key and two DPSIs, one based on customer name and the other on customer address. An insert of a new customer would update only the DPSIs for the partition where the new customer resides. A query based only on customer address or customer name would need to search 200 indexes to find all hits (unless you already have the customer number, in which case the search

is irrelevant). An update or delete without using the customer number would also have to search all 200 indexes (DPSIs) to make the update or delete.

Note: DB2 V8 has increased the number of partitions allowable to 4096. This means the above examples would be multiplied. 2 DPSIs would require 8192 separate index datasets and queries would have to search 4096 indexes!

Q. Why did IBM invent them?

A. DPSIs were invented to solve the IBM Reorg problem. When NPIs are defined on partition tablespaces, the DB2 reorg requires the BUILD2 phase to update the NPIs. During this phase, the logical partitions of the NPI are in UTUT status, which is a restrictive state – no deletes, inserts, or updates (that would change the NPI) are allowed. The tablespace is effectively DOWN. This is not acceptable when 24x7 availability is a requirement. Given the basic design of the IBM Reorg, this problem is not solvable. Therefore, DPSIs were born. Since the DPSI is a separate index per partition, it can be reorged with the partition and partitioning indexes without affecting any other partitions and without compromising availability. In other words, to solve the utility problem, applications have to be changed or else they will perform poorly. This is not a customer-oriented solution.

Q. What are the Restrictions?

A. DPSIs are available in DB2 V8 only. Once DPSIs are created, you cannot fall back to DB2 V7 with those indexes. The DPSIs must be dropped first. Also, for obvious design reasons, DPSIs cannot be unique indexes. For example, how could uniqueness be enforced on an INSERT? DB2 would have to check all the partitions for the value before allowing the INSERT.

Q. Is there any alternative to using DPSIs?

A. Yes, CDB/Auto-Online Reorg has none of the problems that IBM Reorg has with partitions. CDB automatically processes partitions in parallel. You do not have to schedule multiple jobs to run the partitions in parallel. CDB automatically runs partitions in parallel, so multiple jobs do not have to be scheduled. CDB can automatically skip partitions that do not need to be reorged. There is 100% 24x7 online availability for applications. There is no Read-Only phase with CDB. Drains are acquired, but in an orderly fashion so the Drain will timeout instead of your applications. This makes the reorg completely transparent to applications. CDB does not access the DB2 Log. All changes are managed without Log access. NPIs are reorged as part of the normal reorg process while the object is in Read-Write status. There is no separate phase required. There is no BUILD2 phase required. With CDB, restart is automatic. Simply resubmit the Reorg job. No Recovery is needed. If there is any failure during the Reorg, the original objects are still in Read-Write status.

REAL-TIME STATS EXPLAINED

Q. What are Real-Time Stats (RTS)?

A. Real-Time Stats gather information about all the objects in a DB2 system. RTS are not a replacement for Runstats. Runstats gathers information and populates the DB2 Catalog table, providing information for the DB2 Optimizer. RTS, on the other hand, populates a new set of tables created for RTS. These values are not used by the DB2 Optimizer.

Q. What are the benefits?

A. RTS values can be used by the DBA to determine when a utility such as Reorg, Copy, Runstats, needs to be run on an object. This can eliminate unnecessary utility runs and application outages.

Q. How do they impact your applications?

A. Actually, your entire system is affected. Once activated, RTS gathers statistics constantly for all objects in your system. This adds processing overhead to DB2. In addition, the RTS tables require DASD, reducing the amount available to your applications. RTS does not solve any of the inherent problems with IBM utilities which affect application availability.

Q. Why did IBM invent them?

A. IBM introduced RTS to try to solve utility problems. Optimal application performance requires that Reorg, Copy and Runstats be run on the application data on a regular basis. Unfortunately, traditional IBM utilities are expensive in terms of resources (CPU time, elapsed time, DASD, manpower) and cause application outages. More applications for ebusiness, Web access or international business operations are requiring 24x7 data availability. Providing RTS is IBM's attempt to reduce the frequency of utility runs and thereby increase data availability. This is IBM's utility "pre-processor". This method only provides limited rule processing for triggering utilities and does not solve the underlying problem of application downtime.

Q. Is there any alternative to using RTS?

A. CDB provides a multi-dimensional solution. CDB's utilities provide true 24x7 data availability to applications, so the application downtime problem is solved. CDB's utilities run in a fraction of the time and with a fraction of the resources as IBM's, reducing costs. CDB's straightforward interface, wildcard capability and intelligent object management eliminate the need for DBAs to constantly generate and monitor utility jobs. CDB has a full intelligent object management facility built into all of the utilities which is not limited to some predefined set of statistics. CDB does not



require RTS or Runstats to be run before the utilities can do “intelligent” processing. Traditional trigger values are supplied by CDB, but many other factors can be considered also. Any rules a DBA follows manually to determine utility runs can be built into the CDB products. This allows a policy of “best practices” to be enforced.

IBM REORG REBALANCING EXPLAINED

Q. What is IBM Reorg Rebalancing?

A. IBM Reorg Rebalancing is a new feature of IBM Reorg in DB2 V8 which equalizes partitions of a partitioned tablespace.

Q. What are the benefits?

A. Rebalancing allows applications to benefit from partitioned tablespaces. Among other benefits, partitions increase application speed and reduce contention. If most of the data is in one or two partitions, there is less speed and more contention. Rebalancing can also prevent one partition from reaching maximum dataset size. IBM automatically calculates the new limitkey breakpoints and moves the data so that each partition has the same number of rows.

Q. Why did IBM invent it?

A. IBM Reorg Rebalancing solves part of the problem of managing partitioned tablespaces.

Q. What are the Limitations?

A. IBM controls the limitkey calculation. The new limitkeys are determined based on the number of rows. In a tablespace with varying row length, partitions may still be of vastly different physical size after the rebalancing. The user has no control over the limitkey selection.

Q. Is there any alternative to using IBM Reorg Rebalancing?

A. CDB/RePart is CDB’s solution to managing partitioned tablespaces. CDB will calculate the limitkeys based on number of rows or physical size, as well as allow the user to specify limitkeys. CDB can also convert non-partitioned to partitioned tablespaces or change the number of partitions of a partitioned tablespace, again giving options for limitkey selection.

About CDB Software

CDB Software, Inc. is a leader in data management solutions for DB2 z/OS. CDB focuses its business on DB2 for z/OS to provide unique and innovative solutions that enable companies to expand their DB2 system to meet business needs while controlling the overall cost of the mainframe. Founded in 1985, CDB is a privately held corporation based in Houston, Texas with offices worldwide.

For more information visit:

www.cdbsoftware.com



CDB Software, Inc.
11200 Richmond Ave.
Houston, TX 77082