



CDB Software, Inc.

CDB/Quiesce for DB2[®] z/OS[™]

Q) How does QUIESCE work? – What does it do?

A) QUIESCE “finds” a window in time, or point in time, where there are no transactions currently executing and records that time (referred to as either the RBA or LRSN) to the DB2 SYSCOPY catalog table. This point is called a QUIESCE point. This can be used by the RECOVER utility if the object needs to be recovered to a point in time. Along with recording the location in the DB2 log of a clean recovery point, QUIESCE forces unwritten records (pages) to be written from the DB2 bufferpools to the physical DB2 datasets. Partitioned tablespaces can be QUIESCED as a whole object or by individual partitions. While QUIESCE can be used to flush buffers, a keyword is available to not have the DB2 buffers flushed for the object being QUIESCED.

Q) Why run QUIESCE?

- A)** QUIESCE is run for two main reasons:
1. To flush the DB2® buffers for a particular tablespace or set of tablespaces.
 2. To establish a recovery point for a particular tablespace or set of tablespaces.

Typically, a customer would run QUIESCE to establish a recovery point. Generally speaking, ISVs are more interested in flushing DB2 buffers than customers are, but that does not preclude customers from running QUIESCE to flush DB2 buffers.

Customers need to establish recovery points for several reasons. Most of the reasons are based on the assumption that there might be a failure at some point and a Recover will require a consistency point for the data. By establishing a QUIESCE point (which is synonymous with “clean recovery point”) at different times during the day or week, shops can minimize the amount

of effort that has to be placed into a recovery scenario. The more “clean” recovery points available, the more granular their efforts can be and the more options they will have in the event of recovery.

Q) When should I run QUIESCE?

A) QUIESCE should be run whenever a clean recovery point is needed. Considering the other methods of obtaining a recovery point (stopping an object, stopping DB2, taking an image copy), taking a QUIESCE is the least obtrusive, least disruptive and the most efficient. QUIESCE should be run prior to and after batch update cycles and before and after utility operations. QUIESCE can be thought of as “cheap” insurance. When running CDB/QUIESCE instead of IBM QUIESCE, it can even be thought of as “free” insurance, since no application outage costs are associated with CDB/QUIESCE.

Q) What are the options when running IBM QUIESCE?

A) The options for running IBM QUIESCE are limited. You can choose whether or not DB2 buffers are flushed. You can choose to have a complete partitioned tablespace to be QUIESCED or to only QUIESCE a partition at a time. You may QUIESCE several tablespaces at once, as long as you explicitly name each one of them. If you have DB2 referential integrity rules defined, you may QUIESCE a “tablespace set” so that all objects that are related receive the same recovery point. You are limited to 32K bytes for the syntax of your QUIESCE statement, so it is not feasible to QUIESCE every object in a subsystem at any given point in time. On average the maximum number of objects you can QUIESCE in one job step is about 1000.

Q) What are the problems with IBM



QUIESCE?

A) IBM QUIESCE will do two things that cause problems in a DB2 shop. First, it will place the object(s) being QUIESCED into UTRO (utility mode, read-only), status. This means that applications cannot write (INSERT, UPDATE or DELETE) data into tables in the tablespace and can cause -904s (resource unavailable) errors. The second problem IBM QUIESCE will cause is to hold a drain for so long that application SQL times out (receives an SQLERROR code of -911 or -913) while waiting on the QUIESCE to complete. When an application gets a -904 or -911, the application fails and must return an error condition to the caller, or end user, of the application. A -904 or -911 is typically a devastating error from which typical application logic cannot recover.

Q) What is CDB/QUIESCE?

A) CDB/QUIESCE is a manager that sits on top of the IBM QUIESCE utility. CDB/Quiesce manages the IBM QUIESCE utility to prevent application failures due to resource unavailable conditions or timeouts due to drains being held for an excessive amount of time.

Q) What are the options when running CDB/QUIESCE?

A) The options for running CDB/QUIESCE really add a new level of control over the normal IBM QUIESCE process.

CDB/QUIESCE allows the use of wildcards, without using a "LIST" approach. For instance, to QUIESCE all tablespace in a data base, you would simple code "QUIESCE DBASE.%", along with any other keywords that you want to use. Along with wildcards to select objects, the EXCLUDE keyword may be used to exclude

objects from the selected set of objects. Objects may be QUIESCED individually, as a set, and if partitioned, partition by partition, or all partitions at once. Tablespace sets are also supported.

Similar to IBM, you may use the "WRITE" keyword to control whether or not buffers will be flushed from the DB2 buffer pools.

The "DRAINTIMEOUT" keyword controls the length of time CDB/QUIESCE will wait for a DB2 drain to be acquired. The "RETRYCOUNT" keyword controls the number of times a QUIESCE will be retried in the event that the QUIESCE fails or the drain timeout expires.

With CDB/QUIESCE, you can pick the format of the output report with the "REPORT" keyword. In the event that a QUIESCE fails, you have the option to allocate a special DDNAME, "QSCERROR DD", that CDB/QUIESCE will use to write out the failing QUIESCE statement.

Q) What problems does CDB/QUIESCE solve?

A) CDB QUIESCE solves both problems that IBM QUIESCE presents. CDB/QUIESCE will not place objects in UTRO status, but will leave objects in RW (read write) status. This not only lets application read requests (SELECT) to complete, but also write requests (INSERT, UPDATE, & DELETE) to complete. Also, since users can control the amount of time that CDB/QUIESCE will hold a drain, CDB/QUIESCE can be made to not cause -904, -911 and -913 errors.

Q) With CDB/QUIESCE, is a QUIESCE point guaranteed?

A) No. The only thing that CDB/QUIESCE



guarantees is that applications will not get -904s or -911s due to QUIESCE. CDB/QUIESCE manages the IBM QUIESCE Utility, so if the IBM QUIESCE cannot obtain a drain because applications do not commit frequently enough, QUIESCE will not run, thus CDB/QUIESCE will not run, and a recovery point cannot be established. However, unlike IBM QUIESCE, failed CDB/QUIESCEs may be written to a file to be run later.

Q) Does CDB/QUIESCE invoke IBM QUIESCE?

A) Yes. CDB/QUIESCE does not replace IBM QUIESCE. CDB/QUIESCE acts as a manager to make the IBM QUIESCE behavior more application-friendly.

Q) Can a point in time recovery be performed without a QUIESCE point?

A) Yes. Any point in the DB2 log may be used as a recovery point. However, when using an arbitrary point in the DB2 log, you are exposed to recovering to a point that might be in the middle of a UOW (unit of work). If you do this, DB2 would flag this situation and raise a condition of in-flight, in-commit, in-backout or in-doubt. Any of these conditions would not allow access to the DB2 object and an additional recovery action (or actions) would have to occur before normal DB2 operation could occur. If using point in time recovery, it is best to use a “clean” recovery point, such as those that can be obtained with CDB/QUIESCE.

Q) Can CDB/QUIESCE help in both (TOCOPY and TORBA) RECOVERY scenarios?

A) CDB/QUIESCE can help in point in time

(TORBA) recovery preparedness. CDB/QUIESCE cannot help in “to copy” recovery preparedness. Only taking an image copy can establish a “to copy” recovery point. However the CDB/QUIESCE technology is exploited in CDB/Auto-Online Copy as well.

Q) Can CDB/QUIESCE be used as part of a regular maintenance utility process?

A) Yes. Anytime a QUIESCE point, (a “clean” recovery point) can be established, the more flexibility you have in the event that a recover action is needed. Note that if you are using CDB Software’s “CDB/Auto-Utilities”, they already use this safe QUIESCE technology to obtain a recovery point.

Q) Can CDB/QUIESCE be used as part of a batch update process?

A) Absolutely. A good application of CDB/QUIESCE would be to run CDB/QUIESCE as part of a regular batch application update process. CDB/QUIESCE could be run prior to starting the application update process and after the application update process completes, and possibly midstream as well. By establishing a recovery point prior to the application update process, batch update transactions can easily be backed out in the event that the batch update process fails and data needs to be brought to a consistent state as quickly as possible. The same effect (a recovery point) could be achieved by taking an image copy prior to the batch update process, but there may be times when an image copy may not be feasible or even possible, due to availability issues or due to the amount of time required by an image copy process.

Establishing a recovery point after a batch update process also allows for a more granular



Whitepaper

and flexible approach to dealing with recover scenarios. An image copy may not be possible, so it makes perfect sense to obtain the next best alternative, which is a consistent, clean recovery point using CDB/Quiesce.

Function	IBM Quiesce	CDB/Quiesce
Flush DB2 Buffers	Yes	Yes
Create Sync Point for Log Apply	Yes	Yes
-911's (application timeout)	Yes	No
-913s (application timeout)	Yes	No
-904 (unavailable resource - application error)	Yes	No
Wildcard Support	No	Yes
Sync multiple objects to a single point or granularly at separate points	No (no wildcard support)	Yes
Automatic Error Handling and Notification	No	Yes
Report Output Format	No	Yes
Automation to exclude or include objects	No	Yes
Adjustable timeout value	No	Yes
Automatic Retry	No	Yes
Object Remains in RW	No	Yes
Ability to perform system wide Quiesce	Not Advisable	Yes

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