Garbage collection

Garbage collection is the ongoing cleaning of the database and is performed in the background around the clock. This constantly reorganizes the memory space used by the database. If you don't clean up, database performance will slowly but surely degrade.

When performing a garbage collection, Firebird/InterBase® does nothing other than remove outdated data sets and index files, which results in a smaller database. Outdated data sets are stored by Firebird/InterBase® for the following reason: Firebird/InterBase® are multi-generational databases. When a data set is altered, this alteration is stored in the database as a new copy. The old values remain in the database as a back version, which is the rollback protocol. If the transaction is rolled back after the update, the old value is ready to resume its function as the valid value. If the transaction is however committed, and not rolled back, this back version becomes superfluous. In databases with a lot of update operations this can result in a lot of garbage.

When garbage is collected in Firebird/InterBase®, not only the out-of-date update values are deleted, but all outdated and deleted data set versions, based on the Transaction Inventory Page (TIP).

A garbage collection is only performed during a database sweep, database backup or when a SELECT query is made on a table (and not by INSERT, ALTER or DELETE). Whenever Firebird/InterBase® touches a row, such as during a SELECT operation, the versioning engine sweeps out any versions of the row where the transaction number is older than the Oldest Interesting Transaction (OIT). This helps to keep the version history small and manageable and also keeps performance reasonable.

The sweep interval (i.e. at what interval (in number of transactions) a database sweep should be automatically conducted) for the garbage collection may be specified under the IBExpert Services menu item Database Properties.

Garbage collection is co-operative, meaning that all transactions participate in it, rather than a dedicated garbage team. Old versions, deleted records, and rolled back updates are removed when a transaction attempts to read the record. In a database where all records are continually active, or where exhaustive retrievals (i.e. non-indexed access) are done regularly on all tables, co-operative garbage collection works well, as long as the transaction mask stays current.

For databases in which all access is indexed, old records are seldom - or never - revisited and so they seldom - or never - get garbage collected. Running a periodic backup with gbak has the secondary effect of forcing garbage collection since gbak performs exhaustive retrievals on all tables.

The garbage collection may be performed during 24 hour operation online without any problems (i.e. the server does not need to be shut down). Performance may however decline during the database sweep which may not be desirable. If the sweep interval is specified at zero (0) (see Database Properties), the garbage collection is not performed automatically at all. It could then be carried out, for example, at night as a sweep or backup using GFIX and the at Windows command or the Linux chron command.

New to Firebird 2.0: Superserver garbage collection changes

Formerly, the Firebird Superserver performed only background garbage collection. By contrast, Classic performs “cooperative” garbage collection, where multiple connections share the performance hit of garbage collection. Superserver's default behavior for garbage collection is now to combine cooperative and background modes. The new default behavior generally guarantees better overall

performance as the garbage collection is performed online, curtailing the growth of version chains under high load.

It means that some queries may be slower to start to return data if the volume of old record versions in the affected tables is especially high. ODS 10 and lower databases, having ineffective garbage collection on indices, will be particularly prone to this problem. The GCPolicy parameter in firebird.conf allows the former behavior to be reinstated if you have databases exhibiting this problem.

Since Firebird 2.1 introduced its virtual system MON$ tables, the MON$GARBAGE_COLLECTION field in the MON$ATTACHMENTS table indicates whether garbage collection is allowed for a specific attachment (as specified via the DPB in isc_attach_database). Please refer to the Firebird 2.1 Release Notes chapter, Administrative features, for further information.

Firebird's Database housekeeping and garbage collection documentation includes the following definitions:

1. Garbage
2. Cooperative garbage collection
3. Background garbage collection
4. Combined garbage collection
5. Manual garbage collection

From:
https://www.ibexpert.com/docu/ - **IBExpert**

Permanent link:
https://www.ibexpert.com/docu/doku.php?id=01-documentation:01-13-miscellaneous:glossary:garbage-collection

Last update: **2023/08/16 19:59**