Two-phase commit

A transaction spanning multiple Firebird/InterBase® databases is automatically committed in two phases. A two-phase commit guarantees that the transaction updates either all of the databases involved or none of them - data is never partially updated.

In the first phase of a two-phase commit, Firebird/InterBase® prepares each database for the commit by writing the changes from each subtransaction to the database. This subtransaction is the part of a multi-database transaction that involves only one database. In the second phase, InterBase® marks each subtransaction as committed in the order that it was prepared.

If a two-phase commit fails during the second phase, some subtransactions are committed and others are not. A two-phase commit can fail if a network interruption or disk crash makes one or more databases unavailable. Failure of a two-phase commit causes in limbo transactions, i.e. transactions that the server does not know whether to commit or roll back.

It is possible that some records in a database are inaccessible due to their association with a transaction that is in a limbo state.

Note: The Borland Database Engine (BDE), as of version 4.5, does not exercise the two-phase commit or distributed transactions capabilities of Firebird/InterBase®, therefore applications using the BDE never create limbo transactions.

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

Permanent link:
https://ibexpert.com/docu/doku.php?id=01-documentation:01-13-miscellaneous:glossary:two-phase-commit

Last update: **2023/08/21 18:01**