Interbase Replication Suite
Database Recovery

Editor:

Tomáš Mandys, tomas.mandys@2p.cz (2p plus)

Home site:

http://www.2p.cz

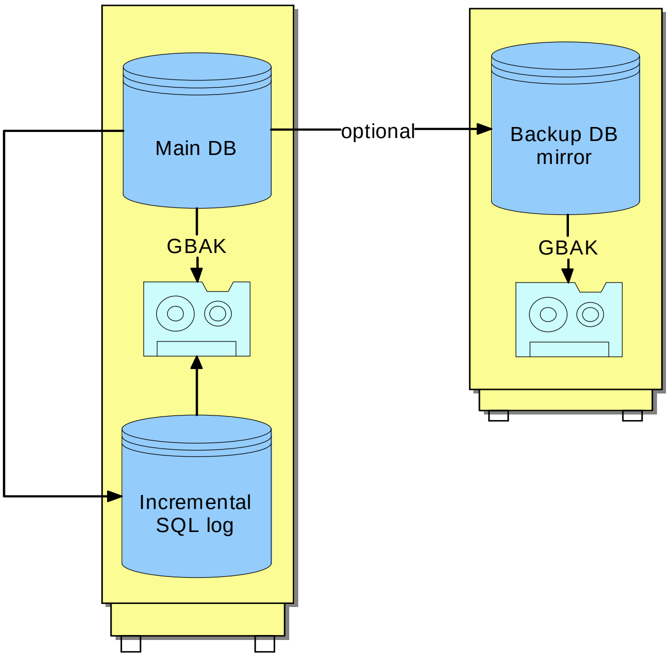
Document status:

Version 1.0 First release

This document describes how *Interbase Replication Suite* can help to recover crashed databases.

Suppose that a replication project meets following conditions:

- The first replication task replicates changes to a mirror database. The mirror database is due security located at remote place and is connected through WAN
- The optional second task writes changes to an incremental log, i.e. log of INSERT, UPDATE, DELETE SQL commands. The incremental log is located on other disk (or machine in LAN)
- both databases and the incremental log are backuped daily (databases using *GBAK* utility)
- Provided that replication into the incremental log is processed almost in online manner, faster than replication to the mirror (remote) database.



If the **source database has been damaged** it is possible to recover from daily backup and from the incremental log. It is necessary to apply all SQL commands performed since backup.

1. delete damaged source database
2. restore source database from last daily backup using *GBAK* utility
3. look to the *REPL$SNAPSHOT* table of the source database what is last *SEQID* and compare it to max *SEQID* from *REPL$LOG* table. We need obtain max of these two values (name it *seqid*)
4. now execute all SQL command from the incremental log having *SEQID* greater that *seqid* obtained above. *IBReplScr* utility does it automatically:

   ```
   ibreplscr <script.log> /S:<schemaid> /G:<groupid> /T:0 /Q1:<seqid+1>
   ```

5. now the source database should be in the same state as the incremental log

By similar way it is possible recover the **target (mirror) database**

1. delete damaged target database
2. restore target database from last daily backup using *GBAK* utility
3. look to *REPL$SNAPSHOT* table of the target database what is *SEQID* of last successfully replicated record (name it *seqid1*)
4. look to *REPL$LOG* table of the source database what is the lowest *SEQID* of record (in scheme) which has not been replicated to the target database. Such records has not null bit in *DBMASK* or *DBMASK_PENDING* field. These records will be replicated in regular way after database recovery (name it *seqid2*)
6. now execute all SQL command from the incremental log having *SEQID* greater than *seqid1* and lower than *seqid2* obtained above. *IBReplScr* utility does it automatically:

   ```
   ibreplscr <script.log> /S:<schemaid> /G:<groupid> /T:<tgtdbid> /Q1:<seqid+1>
   /Q2:<seqid2>
   ```

5. now the target database should be in the same state as before crash