



Firebird and IBExpert White Paper

IBExpert: How to do it?

Automatically synchronize metadata and data and a permanently active backup!

Holger Klemt, October 2022

IBExpert: How to do it?

Part 1: Automatic comparison of the database structure between several databases

Part 2: Automatic synchronisation of table contents between several databases as a permanently active backup

IBExpert: How to do it?

Automatic comparison of the database structure between several databases

IBExpert makes it very easy to compare alterations to one database with another. IBExpert automatically generates a script for this, which can then be run on the target database to update its metadata.

1. Using the script db1.sql from the IBExpert Demo Database package, create a reference database, e.g. in c:\db\db1.fdb
2. Register the database in IBExpert as alias db1.
3. Open the database.
4. Gain an impression of the existing tables, the CUSTOMER table does not yet contain any demo data.
5. Open the procedure INITALL, start it with F9 and enter 10000 as the parameter.
6. After this is completed, do not forget to press the COMMIT button above the tool button.
7. Another look at the CUSTOMER table now shows randomly created demo data.
8. Close the database connection.
9. Make a copy of the file c:\db\db1.fdb, for example: c:\db\db2.fdb.
10. Now register the newly created database db2 in IBExpert.
11. Although we still have 2 identical database files, we shall start comparing the database structures with Tools/Database Comparer.
12. Enter db1 as the source and db2 as the target.



13. Leave all the checkboxes in the dialogue ticked in the left and middle columns.
14. In the right column, all checkboxes are disabled, except for the Verbose checkbox, which should remain activated.
15. By clicking F9, all database structures of the source database db1 are selected and compared with those of the target database db2.
16. If there are any differences, IBExpert will generate a script to accommodate the differences.
17. With IBExpert's technology this almost always works without any loss of data and even for very strongly nested objects.
18. Following execution, the result shows "No difference", because both databases are completely identical.
19. To make a modification in the database db1, open the database db1 and execute this command in an SQL window

```
"create table test1(id bigint not null primary key, txt varchar(80))"
```
20. Using F9 followed by a COMMIT, this new table is now created in database db1, but not yet in database db2.
21. Close the SQL Editor again and restart the Database Comparer that is still open.
22. After running, this will show you the missing table TEST in the database db2.
23. During the run, the Database Comparer only generates a script internally to adjust the metadata, but does not execute it yet.
24. If something is different, you can transfer the script to the Script Executive using the toolbar (4th button).
25. This can now be executed in the Script Executive using F9.
26. After it has been executed, do not forget to COMMIT and then close the Script Executive again.
27. If you now start the Database Comparer again, "no difference" is displayed again because the structures of both databases are identical again.
28. How do we automate this now? Let's look at the IBEBlock tab in the Database Comparer.
29. In this IBEBlock window, IBExpert always compiles the script version for everything that can be compiled with IBExpert in the GUI. However, the IBEBlock tab is always read-only, but each checkbox always affects the parameters directly.
30. To make a few more adjustments, we copy the content of the IBEBlock page via the clipboard into a new SQL Editor (F12 or Tools/SQL Editor) to enter it there again.
31. Replace to the left of "-- Don't forget to specify file name for the result script" with a valid file name, e.g. 'c:\db\cmpres.sql' between the quotes.
32. The contents in the SQL Editor can now be saved by right-clicking - Save To File, e.g. as c:\db\cmp.sql.
33. The Database Comparer can now be closed.
34. For something to happen on the next run, we create another new table in db1 in the SQL Editor.

```
"create table test2(id bigint not null primary key, txt varchar(80))"
```
35. Load the script again from the file you have just created: \db\cmp.sql into the SQL Editor, e.g. with Ctrl+L.
36. Start the script with F9.
37. The result is now, as specified, in the file c:\db\cmpres.sql.
38. In order for this result script not just to be a simple SQL or IBEBlock, we need a Script Executive that can execute multi-line scripts. Start this by going to Tools/Script Executive or Ctrl+F12.
39. With Ctrl+L we load the script from the file c:\db\cmpres.sql. We see that the new table TEST2 is to be created, but it is not yet executed!



40. To automate the comparison, we create a very simple batch file `c:\db\cmp.bat` using Notepad with the following content

```
ibescript.exe cmp.sql
ibescript.exe cmpres.sql
```
41. IBEScript.exe is the command-line version of IBExpert and can execute all SQL and IBEBlock scripts without any interaction. IBEScript.exe is included in the IBExpert full version, as well as the low-cost IBExpert Server Tools version or, for distribution as a software producer, the IBExpert Distribution OEM version.
42. We start a `cmd.exe` DOS window, go to the path `c:\db` and run `cmp.bat`.
43. In the first step, the script creates a script file with which the database `db2` is compared and creates the comparison script as `cmpres.sql`.
44. In the second step, the script executes this file, i.e. immediately starts to modify database `db2` so that it has the same structure as database `db1`.
45. Following execution, we reopen the `db2` file in IBExpert and can now also see the new table `TEST2`.
46. We take a look at the script `c:\db\cmpres.sql` and see the `CREATE TABLE` command, but this has already been executed by the script.
47. We run the batch `cmp.bat` in the DOS window again.
48. If we take a look at the script `c:\db\cmpres.sql` we will see nothing more in it, because now there are no more differences; don't forget to finally close the script file.
49. Now the batch can be started daily by a scheduler or whenever needed.
50. Known limitations with metadata synchronization: actually none, IBExpert copes well, even with fairly abstruse database constructions.

IBExpert: How to do it?

Automatic synchronisation of table contents between several databases as a permanently active backup

Since Firebird 4, database replication is included in Firebird, but if moving to Firebird 4 is not an immediate option, very powerful database replication is also possible for Firebird 2.5 and 3.0. The procedure presented here can be set up very easily for the synchronisation of master-slave database configurations, without having to change anything in the metadata. The databases used, `db1` and `db2`, correspond to the example mentioned above.

1. We open the IBExpert menu item `Tools/Table Data Comparer`.
2. We select the database `db1` again as master, and the target is database `db2` again.
3. On the first page you can select the tables that you would like to be synchronous.
4. Important: the tables should contain a primary key or a unique constraint.
5. First we synchronise the tables `CUSTOMER`, `PRODUCT`, `TEST1` and `TEST2`.



6. To ensure that the synchronisation is performed as quickly as possible, the checkbox "online sync" is activated at the top.
7. When "online sync" is activated, the synchronisation starts immediately and there is no need to write an additional SQL result file, so we can remove the file name and path completely.
8. In the right column we select the generator ID, as this should also be synchronised.
9. On the second page of the Table Data Comparer you will find many options. Usually at least the top 5 checkboxes should be active, most of the rest is self-explanatory.
10. Again, we could start this interactively in IBExpert, but we will go back to the IBEBlock page and save the contents as C:\db\datacmp.sql.
11. We will create a new batch c:\db\datacmp.bat with this content

```
ibescript datacmp.sql
timeout 15
datacmp.bat
```
12. We execute the batch file datacmp.bat.
13. We open the database db1 and change something in a record in the table CUSTOMER, not forgetting to COMMIT.
14. After running the batch file, which will keep starting itself after a timeout, we look at the data in the db2 database and see the same values in the CUSTOMER table.
15. In the same way, this can now be tried with new data records in the tables TEST1 and TEST2.
16. The entire process works for all data types and also for blobs.
17. If you have complex triggers which generate data, you should deactivate them in the database db2 with ALTER TRIGGER inactive.
18. Foreign keys etc. should also be kept synchronous if appropriate. IBExpert recognizes the required sequence extremely well.
19. The above script is intended as a functional example and can also run in reality as a real-time additional backup, but you should keep an eye on the database size and number of records.
20. For very large databases with millions of data records, you will need to do this differently, but this is also possible with Firebird 2.5 or newer without any problems. We have years of project experience with this and offer transaction-safe real-time or near-time replication as individual customer projects.
21. For the above example, the IBExpert Server Tools may be needed as a runtime; the IBEScript.exe is included and can also run at the customer's site without the full version of IBExpert.
22. Practical application:
 - Copy your database to a second computer or to a different path on the same computer (it is advisable to stop the Firebird server service briefly beforehand and restart it after copying).
 - Using the above technique, create an IBEBlock script for all your tables with the production database as the source and the backup database as the target.
 - Deactivate all triggers in the backup database (right-click on Trigger in IBExpert).
 - Create the batch file and select the timeout that suits your purposes, e.g. 3600 = hourly.
 - The batch file now compares all tables that have a primary key or unique constraint in both databases, and you have a permanent live backup without any great effort.
 - If your production database is no longer accessible, you now only need to open the backup database, reactivate all triggers and you will have a database that is quite up-to-date.



- For especially important tables, you can set up an extra script with a very short timeout or no timeout at all. The differences detected there do therefore not have to wait for the other scripts, which for example, also adjust blobs, etc.
- For very large databases or tables with a large number of records, you will need to check the runtime behaviour. Rough estimate: Databases up to 1 GB are usually no problem, up to 10 GB the synchronisation should be divided into table groups and from 100 GB it is best to choose a different procedure or only synchronise a part.
- A slow server will certainly not become faster with this method. It is advisable to first run the IBExpert benchmark, to check whether your server delivers an acceptable speed at all, before you use this procedure on a lame machine.