Migration Framework User Guide

MQFS Migration Framework for Firebird / HQBird – v20240430.001

Summary

1 Steps to Migration

The migration framework will convert the Firebird / HQBird databases to the desired version, performing the backup/restore process, recompiling objects and adapting them to the new version.

The basic steps for use are:

1. Download and install application;
2. Add databases in the current version in the "IN" subfolder;
3. Adjust settings in the conf.ini file;
4. Run the "Migrate.exe" executable;
5. Analyze logs;
6. Get successfully converted databases from "/OUT/OK" folder.

2 Compatibility

Currently, the Migration Framework is compatible with the Windows operating system and can be run separately or integrated with HQBird v2024.

3 Application download and first run

The framework will allow a limited number of successful conversions for free, but this availability is not guaranteed and may be changed or interrupted at any time without prior notice.

The application can be downloaded via the website https://download.mqfs.com.br.

When running the application for the first time, it will be checked whether there is a license for use in the local folder. If you cannot find it, you will be asked for licensing information.

If you already have a registration key, you must enter it in the application when requested. Otherwise, with the data provided, a registration request will be made and an email will be sent with the licensing data.

After the first purchase of licenses, your purchase will be registered and made available automatically. If you have never registered with the app, you will be sent an email with your licensing details.

4 Installation

After downloading, the files can be unzipped into two directories:

- *If you have HQBird licensed and installed: C:\HQBird\Migr*

  - This installation will use Firebird binaries located at C:\HQBird\FirebirdXX, so both the source and target versions need to be installed via HQBird.

- *If it is a separate acquisition: C:\Migrate*

  - Inside the root folder, there will be the "FB" folder and subfolders named according to the versions required for source and destination.

5 Configuration

The application settings are in the root, file called conf.ini.

Inside this file we have all the parameters necessary for the migration, we will describe each one of them.

5.1 AUTH Session

Definition of authentication data for connecting to databases.

5.1.1 USER

User definition, by default you must always use SYSDBA.

5.1.1 PASS

Definition of the user password, the default password being masterkey.

Note that if you use HQBird binaries, you need to fill in the same SYSDBA password used in HQBird.

If you use the binaries distributed from the application's FB subfolder, the password will always be masterkey, since the connection is made onboard.

5.2 PARAM session

Defines general parameters for the entire migration process.

The migration steps are performed exactly in the sequence below, and for this reason they are in these positions within the configuration file:

1. "RUN_PRESCRIPT"
2. "UPGRADE_DBS"
3. "UPGRADE_DIALECT"
4. "REPLACE_DOUBLE_QUOTES"
5. "RUN_POSSCRIPT"
6. "RECOMPILE"

Other parameters are

5.2.1 RUN_PRESCRIPT

Allowed values: true | false

Executes all *.sql files that are within the "scripts\pre" subfolder.

The execution will occur in alphabetical order, we suggest placing a numerical sequence at the beginning of the file name to avoid mistakes.

To execute a list of scripts within a single file, respect the use of SET TERM within your script.

*NOTE1: The *.sql scripts in this folder will only be executed if they do not start with # and if the extension of the database contained in the "in" subfolder is ".FDB". The affected databases will be those in the "in" folder.*

NOTE2: It is possible to declare in which databases a script should be executed, adding the name of the database at the top of the script file, as follows: --bd=BANCO1.FDB,BANCO2.FDB,BANCO3.FDB. The application will only execute the script if the database being converted is in this list.

5.2.2 UPGRADE_DBS

Allowed values: true | false

If the extension of the database contained in the "in" subfolder is ".FDB", it will perform a backup using binaries from the original version and restore to the "out" subfolder using binaries from the new version.

If the extension of the database contained in the "in" subfolder is ".FBK", it will perform the restore to the "out" subfolder using binaries from the new version.

NOTE: If the recompilation of objects has not been commanded, the databases that are successful in the restoration will be moved to the "OK" subfolders within "in" and "out", and the process will be completed.

5.2.3 UPGRADE_DIALECT

Allowed values: true | false

In the restored database, the application will migrate to dialect 3, making corrections to the use of current_date for localtimestamp in the objects where this variable was used.

5.2.4 REPLACE_DOUBLE_QUOTES

Allowed values: true [only if UPGRADE_DIALECT is true] | false

The use of double quotes to delimit text in PSQL codes is common when using dialect 1. However, dialect 3 interprets double quotes as an object reference, not as text, causing a compilation error and requiring replacement with single quotes .

NOTE: Use only if you are sure that most of the texts are enclosed in double quotation marks.

5.2.5 RUN_POSSCRIPT

Allowed values: true | false

Executes all *.sql files that are within the "scripts\pos" subfolder.

The execution will occur in alphabetical order, we suggest placing a numerical sequence at the beginning of the file name to avoid mistakes.

To execute a list of scripts within a single file, respect the use of SET TERM within your script.

*NOTE1: The *.sql scripts in this folder will only be executed if they do not start with #. The affected databases will be those in the "out" folder.*

NOTE2: It is possible to declare in which databases a script should be executed, adding the name of the database at the top of the script file, as follows: --bd=DB1.FDB,DB2.FDB,DB3.FDB. The application will only execute the script if the database being converted is in this list.

5.2.6 RECOMPILE

Allowed values: true | false

This is the parameter that will recompile all objects, making several adjustments so that the database is compatible with the target version.

If there is any incompatibility that cannot be resolved automatically as a syntax error, the application will display the error in the log, which must be mapped, corrected and the correction script must be added to the "scripts/pre" or "scripts/pos" subfolder, activating the corresponding parameter for execution. Thus, in the next execution the correction will be done automatically in all databases.

If this parameter is set to "true" and the UPGRADE_DBS parameter is set to "false", the application will search for the bases contained in the "out" subfolder to only perform the recompilation.

5.2.7 CHARSET_ISQL

Allowed values: empty | existing charsets in Firebird

If informed, it defines which charset will be used to connect to ISQL, in order to run standard and customized scripts ("pre" and "pos").

If the defined charset is different from the base's default, there will be an alert in the log indicating which ones are different and which is the base's default charset.

5.2.8 CHARSET_SCRIPTS

Allowed values: empty | existing charsets in Firebird

If informed, it defines which charset will be used to work with PSQL source codes. Firebird normally uses UTF8 for newer databases and UNICODE_FSS for older ones.

Leave it empty so that the application automatically identifies your base pattern.

5.2.9 COLLATE_SCRIPTS

Allowed values: empty | existing charsets in Firebird

If informed, it defines which collate will be used to work with PSQL source codes. Firebird normally uses UTF8 for newer databases and UNICODE_FSS for older ones.

Leave it empty so that the application automatically identifies your base pattern.

5.2.10 BKP

Allowed values: the same parameters available for backup with gbak

Here you can enter the same parameters that exist in gbak to perform the backup, except the "-b", file path, username and password as the application will inform them automatically.

By default, it is recommended to use at least the "-g-z-v" parameters to leave the backup log with more details.

NOTE: If you use HQBird binaries, you can use the "-par N" parameter to define the number of cores you want to use for multi-thread backup.

5.2.11 RST

Allowed values: the same parameters available for restore with gbak

Here you can enter the same parameters that exist in gbak to perform the restore, except for "-c", file path, username and password as the application will inform them automatically.

By default, it is recommended to use at least the "-z-v" parameters to leave the backup log with more details.

NOTE: If you use HQBird binaries or are migrating to version 5.0 of Firebird or later, you can use the "-par N" parameter to define the number of cores you want to use for multi-thread restore.

5.2.12 VERSION_IN

Allowed values: 25 | 30 | 40 | 50

Defines the current version of the database to be migrated. All bases added to the "in" folder must be the same version.

If you have a version of Firebird prior to 2.5, keep the value "25" in this parameter to enable migration. If that doesn't work, try leaving just the FBK in the "in" folder, this way only the target version's binaries will be needed.

5.2.13 VERSION_OUT

Allowed values: 30 | 40 | 50

Defines which version the database should be migrated to.

The result will be an FDB file compatible with the desired version.

5.2.14 LANG

Allowed values: BR | EN

Defines in which language the framework will show logs. This parameter will be automatically defined based on the language chosen at the time of installation, and can be changed at any time.

6 Folder Structure

The migration application has the following folders.

6.1 FB

This folder contains the Firebird versions necessary to execute the migration process in an embedded manner.

If you have HQBird licensed, the Firebird binaries used will be from HQBird, so this folder will not exist.

6.2 IN

Add *.FDB or *.FBK files in the source version here. 1 or more files can be added, and the migration process will occur for all in a single run.

After the migration process is complete, only the bases that were completed with ERRORS will remain in this folder. Successful databases will be moved to the OK subfolder.

6.3 OUT

Here the *.FDB files resulting from the migration process to the desired version will be created.

After the migration process is complete, only the bases that were completed with ERRORS will remain in this folder. Successful databases will be moved to the OK subfolder.

6.4 IN/OK | OUT/OK

After the migration process is complete, only the databases that were completed SUCCESSFULLY will be moved to the OK subfolder, both IN and OUT, making it easier to separate what worked and what didn't.

6.5 SCRIPTS/PRE | SCRIPTS/POS

Here you can add scripts with the ".sql" extension to be executed during the migration process.

The scripts in the PRE folder will be executed **before** the backup, in the databases contained in the IN folder.

The scripts in the POS folder will be executed **after** the restore, in the databases contained in the OUT folder, but before the base is recompiled.

To be able to execute scripts from these folders, the *RUN_PRESCRIPT* parameter (described in item 5.2.1) and *RUN_POSSCRIPT* (described in item 5.2.5) must be enabled, respectively.

6.5.1 IMPORTANT SCRIPTS CONSIDERATIONS

- Scripts run in alphabetical order, so name them in the sequence you want them to run.

- Scripts with names starting with # (hash) will be ignored by the framework.

- All scripts are executed through ISQL, so if you are making metadata changes, you need to add to your script the change of the end-of-command character from "*semicolon*" to "*caret*" (*SET TERM ^;*), for example.

  - At the end of your PSQL code, you must add the caret to determine the end of the procedure, trigger, function, etc. This action is only necessary if you have more than one object being changed in the same script.

- When migrating several banks simultaneously, it may be necessary to choose to run scripts only on certain databases. To do this, **in the first line of your script**, add the base names separated by commas, without spaces between them and including the extension.

  - Example: File "*1_sp_test.sql*", to run it only on databases A.FDB and B.FDB, add the text to the first line of the file: *--bd=A.FDB,B.FDB*. The "*1_sp_test.sql*" script will only be executed on the bases declared in this parameter. The Framework will not attempt to run this script in the C.FDB, D.FDB, etc. bases.

6.5.2 MIGRATION OF SECURITYN.FDB

Firebird 3.0 onwards has a new authentication method, SRP.

If you use Firebird 2.5 or earlier and have users other than SYSDBA, you will need to do the following:

1. Add your security2.fdb to the framework's "*in*" folder;

2. Add to the "*scripts/pos*" folder the script provided in versions 3.0+, located in the "*FB root folder\misc\upgrade\security\security_database.sql*";

3. Add the SYSDBA user creation command (now in SRP) above the content of the *security_database.sql* file: *CREATE USER SYSDBA PASSWORD 'XXXXXX';*

4. Add two parameters **to the first line of the file**, as follows: *--bd=SECURITY2.FDB –secdb*

    1. The "*--secdb*" parameter will indicate to the framework that this script will be executed in a security database, and therefore requires particularities in the execution that the framework will provide.

5. After migration, SECURITY2.FDB in the OUT folder needs to be renamed according to the target version, example: SECURITY5.FDB for Firebird version 5.0;

6. Now, the SECURITY5.FDB file from the out folder is ready to replace the original of the new Firebird installation on the target server;

Important: it is impossible to obtain the original passwords for each user for recreation using the SRP method, so the new passwords generated for each user will be listed in the migrate.log file. You can customize the "security_database.sql" script to create a shorter password however you prefer.

7 UDFs

If you use UDF, it will be required by restore daabase process and by recompile objects. Therefore, you will need to add them to the UDF folder of the chosen target version.

Example: If you are migrating to version 5.0, add your UDFs inside the folder:

- FB\50W\UDF – for use of the framework on an individual basis;

- C:\HQbird\Firebird50\UDF – when using HQBird binaries;

Remember to check that the "UDFAcccess" parameter is set to "Restrict UDF" so that Firebird can read these dlls.

NOTE1: The distributed binaries are 64bit, so you need to have the UDFs also in 64bit architecture, otherwise they cannot be read and the process will have errors.

8 Logs

During the migration process, different log files are generated, including:

- **Backup Logs:** If the source files have the *.FDB extension, there will be a backup log for each file in the "IN" folder, with the format "<name_do_banco>_bkp.log".

- **Restore Logs:** There will be a restore log for each file in the "OUT" folder, with the format "<database_name>_rst.log".

- **Migration process log:** The entire migration process will be described in the "migra.log" file within the application's root folder, containing each of the steps to complete the process according to the selected configuration.

8.1 Interpretation of the migration log

After the process is complete, at the end of the migra.log file you will have a summary of what happened with each database, whether it ended successfully or with an error.

To see error details, search for the text "#ERRO#" (in Brazilian Portuguese) or "#ERROR#" (in English) within the file.

9 Acquisition of licenses

The acquisition of database conversion packages can be done through the website https://ib-aid.com/en/mass-migration-framework-for-firebird/, end of page.

Licensing does not include migration support, but can be contracted separately. To hire, request via this email: ak@ib-aid.com

10 Updates

The framework will check for updates with each migration, and at the end of the "migra.log" file it will inform if there is a version last.

The download of new versions will be available on the website https://mqfs.com.br/migrfmk, bottom of the page