

# Guia do usuário Framework de Migração

MQFS Migration Framework para Firebird / HqBird – v20241002.001

## Sumário

1	Passos para Migração.....	2
2	Compatibilidade.....	2
3	Download do Aplicativo e primeira execução.....	2
4	Instalação.....	2
5	Configuração.....	3
5.1	Sessão AUTH.....	3
5.1.1	USER.....	3
5.1.1	PASS.....	3
5.2	Sessão PARAM.....	3
5.2.1	RUN_PRESCRIPT.....	3
5.2.2	UPGRADE_DBS.....	4
5.2.3	UPGRADE_DIALECT.....	4
5.2.4	REPLACE_DOUBLE_QUOTES.....	4
5.2.5	RUN_POSSCRIPT.....	4
5.2.6	RECOMPILE.....	4
5.2.7	CHARSET_ISQL.....	5
5.2.8	CHARSET_SCRIPTS.....	5
5.2.9	COLLATE_SCRIPTS.....	5
5.2.10	BKP.....	5
5.2.11	RST.....	5
5.2.12	VERSION_IN.....	6
5.2.13	VERSION_OUT.....	6
5.2.14	LANG.....	6
5.2.15	ISQL_PAGE_LENGTH.....	6
6	Estrutura de Pastas.....	6
6.1	FB.....	6
6.2	IN.....	7
6.3	OUT.....	7
6.4	IN/OK   OUT/OK.....	7
6.5	SCRIPTS/PRE   SCRIPTS/POS.....	7
6.5.1	CONSIDERAÇÕES IMPORTANTES SOBRE SCRIPTS.....	7
6.5.2	MIGRAÇÃO DO SECURITYN.FDB.....	8
7	UDFs.....	8
8	Logs.....	9
8.1	Interpretação do log de migração.....	9
9	Aquisição de licenças.....	9
10	Atualizações.....	9

# 1 Passos para Migração

O framework de migração irá converter as bases de dados Firebird / HQBird para a versão desejada, realizando o processo de backup/restore, recompilando objetos e adequando-os para a nova versão.

Os passos básicos para utilização são:

1. Baixar e instalar aplicativo;
2. Adicionar bancos na versão atual na subpasta "IN";
3. Ajustar configurações do arquivo conf.ini;
4. Rodar o executável "Migrate.exe";
5. Analisar logs;
6. Obter bancos convertidos com sucesso da pasta "/OUT/OK".

## 2 Compatibilidade

Atualmente o Framework de Migração é compatível com sistema operacional Windows, podendo ser executado de forma avulsa ou integrada com HQBird v2024.

## 3 Download do Aplicativo e primeira execução

O framework permitirá um número limitado de conversões de sucesso de forma gratuita, porém essa disponibilidade não é garantida, podendo ser alterada ou interrompida a qualquer momento sem aviso prévio.

O download do aplicativo poderá ser feito através do site <https://download.mqfs.com.br>.

Ao rodar o aplicativo pela primeira vez, será verificado se existe licença para uso na pasta local. Caso não encontre, serão solicitados dados para o licenciamento.

Se você já possui uma chave de registro, deverá informar no aplicativo quando solicitado. Caso contrário, com os dados informados será feita a solicitação de registro e um e-mail será enviado com os dados do licenciamento.

Após a primeira compra das licenças, sua compra será registrada e disponibilizada automaticamente. Caso nunca tenha feito um registro no aplicativo, será enviado e-mail com os seus dados de licenciamento.

## 4 Instalação

Após download, os arquivos podem ser descompactados em dois diretórios:

- *Caso tenha HQBird licenciado e instalado: C:\HQBird\Migr*
  - Esta instalação utilizará binários do Firebird localizados em C:\HQBird\FirebirdXX, portanto as versões de origem e destino precisam ser instaladas via HQBird.
- *Caso seja aquisição avulsa: C:\Migrate*
  - Dentro da pasta raiz, haverá a pasta "FB" e as subpastas nomeadas de acordo com as versões necessárias para origem e destino.

## 5 Configuração

As configurações do aplicativo estão na raiz, arquivo chamado conf.ini.

Dentro deste arquivo temos todos os parâmetros necessários para a migração, vamos descrever cada um deles.

### 5.1 Sessão AUTH

Definição dos dados de autenticação para a conexão com os bancos.

#### 5.1.1 USER

Definição do usuário, por padrão deve-se utilizar sempre o SYSDBA.

#### 5.1.1 PASS

Definição da senha do usuário, sendo a senha padrão masterkey.

Observe que se você utiliza os binários do HQBird, precisa preencher com a mesma senha do SYSDBA utilizada no HQBird.

Se você utiliza os binários distribuídos da subpasta FB do aplicativo, a senha será sempre masterkey, já que se faz conexão embarcada.

### 5.2 Sessão PARAM

Define parâmetros gerais para todo o processo de migração.

Os passos da migração são executados exatamente na sequência abaixo, e por esse motivo estão nessas posições dentro do arquivo de configuração:

1. "RUN\_PRESCRIPT"
2. "UPGRADE\_DBS"
3. "UPGRADE\_DIALECT"
4. "REPLACE\_DOUBLE\_QUOTES"
5. "RUN\_POSSCRIPT"
6. "RECOMPILE"

Demais parâmetros são

#### 5.2.1 RUN\_PRESCRIPT

Valores permitidos: *true* | *false*

Executa todos os arquivos \*.sql que estiverem dentro da subpasta "scripts\pre".

A execução ocorrerá em ordem alfabética, sugerimos colocar uma sequência numérica no início do nome do arquivo para evitar equívocos.

Para executar uma lista de scripts dentro de um único arquivo, respeite o uso do SET TERM dentro do seu script.

*OBS1: Os scripts \*.sql desta pasta só serão executados se não iniciarem com # e se a extensão do banco contido na subpasta "in" for ".FDB". Os bancos afetados serão os da pasta "in".*

*OBS2: É possível declarar em quais bancos um script deverá ser executado, adicionando o nome do banco no topo do arquivo de script, da seguinte forma: --bd=BANCO1.FDB,BANCO2.FDB,BANCO3.FDB. O aplicativo só irá executar o script se o banco sendo convertido estiver nessa lista.*

## 5.2.2 UPGRADE\_DB

Valores permitidos: true | false

Caso a extensão do banco contido na subpasta “in” for “.FDB”, executará backup usando binários da versão original e restore para a subpasta “out” usando binários da nova versão.

Caso a extensão do banco contido na subpasta “in” for “.FBK”, executará o restore para a subpasta “out” usando binários da nova versão.

*OBS: Caso não tenha sido comandada a recompilação de objetos, os bancos que obtiverem sucesso na restauração serão movidos para as subpastas “OK” dentro de “in” e “out”, e o processo será concluído.*

## 5.2.3 UPGRADE\_DIALECT

Valores permitidos: true | false

No banco restaurado, o aplicativo fará a migração para o dialect 3, fazendo correções do uso de current\_date para localtime nos objetos onde esta variável foi utilizada.

## 5.2.4 REPLACE\_DOUBLE\_QUOTES

Valores permitidos: true [somente se UPGRADE\_DIALECT for true] | false

O uso de aspas duplas para delimitar texto nos códigos PSQL é comum quando se usa o dialeto 1. Porém o dialeto 3 interpreta aspas duplas como referência de objeto, não como texto, causando erro na compilação e fazendo-se necessária a substituição por aspas simples.

*OBS: Use somente se você tem certeza que a maior parte dos textos estão entre aspas duplas.*

## 5.2.5 RUN\_POSSCRIPT

Valores permitidos: true | false

Executa todos os arquivos \*.sql que estiverem dentro da subpasta “scripts\pos”.

A execução ocorrerá em ordem alfabética, sugerimos colocar uma sequência numérica no início do nome do arquivo para evitar equívocos.

Para executar uma lista de scripts dentro de um único arquivo, respeite o uso do SET TERM dentro do seu script.

*OBS1: Os scripts \*.sql desta pasta só serão executados se não iniciarem com #. Os bancos afetados serão os da pasta “out”*

*OBS2: É possível declarar em quais bancos um script deverá ser executado, adicionando o nome do banco no topo do arquivo de script, da seguinte forma: --bd=BANCO1.FDB,BANCO2.FDB,BANCO3.FDB. O aplicativo só irá executar o script se o banco sendo convertido estiver nessa lista.*

## 5.2.6 RECOMPILE

Valores permitidos: true | false

Este é o parâmetro que fará a recompilação de todos os objetos, fazendo várias adequações para que o banco fique compatível com a versão de destino.

Se houver qualquer incompatibilidade que não possa ser resolvida de forma automatizada como erro de sintaxe, o aplicativo exibirá o erro no log, que deverá ser mapeado, corrigido e o script de correção deverá ser adicionado na subpasta “scripts/pre” ou “scripts/pos”, ativando o parâmetro correspondente para a execução. Assim, na próxima execução a correção será feita de forma automatizada em todos os bancos.

Caso este parâmetro esteja definido como “true” e o parâmetro UPGRADE\_DBS esteja definido como “false”, o aplicativo irá buscar as bases contidas na subpasta “out” para executar apenas a recompilação.

### 5.2.7 CHARSET\_ISQL

*Valores permitidos: vazio | charsets existentes no Firebird*

Caso seja informado, define qual será o charset utilizado para conectar no ISQL, com objetivo de rodar scripts padrão e personalizados (“pre” e “pos”).

Caso o charset definido seja diferente do padrão da base, haverá um alerta no log indicando qual que estão diferentes e qual é o charset padrão da base.

### 5.2.8 CHARSET\_SCRIPTS

*Valores permitidos: vazio | charsets existentes no Firebird*

Caso seja informado, define qual será o charset utilizado para trabalhar com os códigos-fonte PSQL. Normalmente o Firebird utiliza UTF8 para bases mais novas e UNICODE\_FSS para mais antigas.

Deixe vazio para que o aplicativo identifique automaticamente o padrão da sua base.

### 5.2.9 COLLATE\_SCRIPTS

*Valores permitidos: vazio | charsets existentes no Firebird*

Caso seja informado, define qual será o collate utilizado para trabalhar com os códigos-fonte PSQL. Normalmente o Firebird utiliza UTF8 para bases mais novas e UNICODE\_FSS para mais antigas.

Deixe vazio para que o aplicativo identifique automaticamente o padrão da sua base.

### 5.2.10 BKP

*Valores permitidos: os mesmos parâmetros disponíveis para backup com gbak*

Aqui você pode informar os mesmos parâmetros existentes no gbak para realização do backup, exceto o “-b”, caminho dos arquivos, usuário e senha pois o aplicativo irá informar automaticamente.

Por padrão, recomenda-se utilizar pelo menos os parâmetros “-g -z -v” para deixar o log do backup com mais detalhes.

OBS: Se você utiliza os binários do HQBird, pode utilizar o parâmetro “-par N” para definir o número de núcleos que quer utilizar para backup multi-thread.

### 5.2.11 RST

*Valores permitidos: os mesmos parâmetros disponíveis para restore com gbak*

Aqui você pode informar os mesmos parâmetros existentes no gbak para realização do restore, exceto o “-c”, caminho dos arquivos, usuário e senha pois o aplicativo irá informar automaticamente.

Por padrão, recomenda-se utilizar pelo menos os parâmetros “-z-v” para deixar o log do backup com mais detalhes.

OBS: Se você utiliza os binários do HQBird ou está migrando para a versão 5.0 do Firebird ou posteriores, pode utilizar o parâmetro “-par N” para definir o número de núcleos que quer utilizar para restore multi-thread.

### 5.2.12 VERSION\_IN

Valores permitidos: 25 | 30 | 40 | 50

Define qual é a versão atual do banco de dados a ser migrado. Todas as bases adicionadas na pasta “in” precisam estar na mesma versão.

Caso você tenha versão anterior à 2.5 do Firebird, mantenha o valor “25” neste parâmetro para possibilitar a migração. Se isso não der certo, experimente deixar apenas o FBK na pasta “in”, dessa forma só serão necessários os binários da versão de destino.

### 5.2.13 VERSION\_OUT

Valores permitidos: 30 | 40 | 50

Define qual é a versão para a qual o banco de dados deve ser migrado.

O resultado será um arquivo FDB compatibilizado com a versão desejada.

### 5.2.14 LANG

Valores permitidos: BR | EN

Define em qual idioma o framework mostrará os logs. Este parâmetro será definido automaticamente baseado no idioma escolhido no momento da instalação, e poderá ser alterado a qualquer momento.

### 5.2.15 ISQL\_PAGE\_LENGTH

Valores permitidos: 0-100000

Define em qual o tamanho da paginação para exibição dos resultados de comandos executados no ISQL pelo framework. O efeito prático disso é que quanto maior a paginação, menor o número de vezes que serão exibidos os cabeçalhos das colunas dos relatórios do framework.

O valor padrão é 10000.

## 6 Estrutura de Pastas

O aplicativo de migração possui as seguintes pastas.

### 6.1 FB

Esta pasta possui as versões do Firebird necessárias para executar o processo de migração de forma embarcada.

Caso tenha HQBird licenciado, os binários do Firebird utilizados serão do HQBird, portanto esta pasta não existirá.

## 6.2 IN

Adicione aqui arquivos \*.FDB ou \*.FBK na versão de origem. Podem ser adicionados 1 ou mais arquivos, e o processo de migração ocorrerá para todos em uma única execução.

Após a finalização do processo de migração, apenas as bases que foram concluídas com ERROS permanecerão nesta pasta. As bases que obtiverem sucesso serão movidas para a subpasta OK.

## 6.3 OUT

Aqui serão criados os arquivos \*.FDB resultantes do processo de migração para a versão desejada.

Após a finalização do processo de migração, apenas as bases que foram concluídas com ERROS permanecerão nesta pasta. As bases que obtiverem sucesso serão movidas para a subpasta OK.

## 6.4 IN/OK | OUT/OK

Após a finalização do processo de migração, apenas as bases que foram concluídas com SUCESSO serão movidas para a subpasta OK, tanto para IN quanto para OUT, facilitando a separação do que deu certo e do que não deu.

## 6.5 SCRIPTS/PRE | SCRIPTS/POS

Aqui você poderá adicionar scripts com extensão “.sql” para que sejam executados durante o processo de migração.

Os scripts da pasta PRE serão executados **antes** do backup, nos bancos contidos na pasta IN.

Já os scripts da pasta POS serão executados **depois** do restore, nos bancos contidos na pasta OUT, porém antes da base ser recompilada.

Para que seja possível executar os scripts dessas pastas, o parâmetro *RUN\_PRESCRIPT* (descrito no item 5.2.1) e *RUN\_POSSCRIPT* (descrito no item 5.2.5) precisam estar habilitados, respectivamente.

### 6.5.1 CONSIDERAÇÕES IMPORTANTES SOBRE SCRIPTS

- Os scripts são executados em ordem alfabética, portanto nomeie-os na sequência que você deseja que sejam executados.
- Scripts com nomes iniciados com # (cerquilha) serão ignorados pelo framework.
- Todos os scripts são executados através do ISQL, portanto se você está fazendo alteração de metadata, precisa adicionar no seu script a alteração do caracter de fim de comando de “*ponto e vírgula*” para “*circunflexo*” (*SET TERM ^;*), por exemplo.
  - Ao final do seu código PSQL, deverá adicionar o circunflexo para determinar o final da procedure, trigger, function, etc. Esta ação só é necessária se você tiver mais de um objeto sendo alterado no mesmo script.
- Ao migrar vários bancos simultaneamente, pode ser necessário optar por rodar scripts apenas em determinados bancos. Para isso, **na primeira linha do seu script**, adicione os nomes das bases separados por vírgula, sem espaços entre eles e incluindo a extensão.

- Exemplo: Arquivo “1\_sp\_test.sql”, para que seja executado apenas nos bancos A.FDB e B.FDB, adicione na primeira linha do arquivo o texto:--bd=A.FDB,B.FDB. O script “1\_sp\_test.sql” será executado apenas nas bases declaradas neste parâmetro. O Framework não tentará rodar este script nas bases C.FDB, D.FDB, etc.

## 6.5.2 MIGRAÇÃO DO SECURITYN.FDB

O Firebird 3.0 em diante possui um novo método de autenticação, o SRP.

Se você utiliza o Firebird 2.5 ou anteriores e possui usuários além do SYSDBA, precisará fazer o seguinte:

1. Adicione o seu security2.fdb na pasta “in” do framework;
2. Adicionar na pasta “scripts/pos” o script fornecido nas versões 3.0+, localizado na “pasta raiz do FB\misc\upgrade\security\security\_database.sql”;
3. Acrescente acima do conteúdo do arquivo *security\_database.sql* o comando de criação do usuário SYSDBA (agora no SRP): `CREATE USER SYSDBA PASSWORD 'XXXXXX';`
4. Adicione dois parâmetros **na primeira linha do arquivo**, da seguinte forma:--bd=SECURITY2.FDB --secdb
  1. O parâmetro “--secdb” indicará ao framework que este script será executado num banco de segurança, e por isso exige particularidades na execução que o framework providenciará.
5. Depois da migração, o SECURITY2.FDB da pasta OUT precisa ser renomeado de acordo com a versão de destino, exemplo: SECURITY5.FDB para a versão 5.0 do Firebird;
6. Agora, o arquivo SECURITY5.FDB da pasta out está pronto para substituir o original da nova instalação do Firebird no servidor de destino;

**Importante:** *é impossível obter as senhas originais de cada usuário para a recriação no método SRP, por isso no arquivo migra.log serão listadas as novas senhas geradas para cada usuário. Você pode personalizar o script “security\_database.sql” para criar uma senha menor da forma que preferir.*

## 7 UDFs

Se você usa UDF, elas serão necessárias para restaurar a base e para recompilar objetos. Portanto, você vai precisar adicioná-las na pasta UDF da versão de destino escolhida.

Exemplo: Se você está migrando para a versão 5.0, adicione suas UDFs dentro da pasta:

- FB\50W\UDF – para uso do framework de forma avulsa;
- C:\HQbird\Firebird50\udf – para uso do framework usando binários do HQBird;

Lembre-se de verificar se o parâmetro “UDFAccess” está definido como “Restrict UDF” para que o Firebird possa ler essas dlls.

**OBS1:** *Os binários distribuídos são de 64bits, portanto você precisa ter as UDFs também na arquitetura 64bits, caso contrário não poderão ser lidas e o processo terá erros.*

## 8 Logs

Durante o processo de migração são gerados diferentes arquivos de logs, sendo eles:

- **Logs de Backup:** Caso os arquivos de origem sejam da extensão \*.FDB, haverá um log de backup para cada arquivo na pasta "IN", com formato "<nome\_do\_banco>\_bkp.log".
- **Logs de Restore:** Haverá um log de restore para cada arquivo na pasta "OUT", com formato "<nome\_do\_banco>\_rst.log".
- **Log do processo de migração:** Todo o processo de migração será descrito no arquivo "migra.log" dentro da pasta raiz do aplicativo, contendo cada um dos passos para completar o processo de acordo com a configuração selecionada.

### 8.1 Interpretação do log de migração

Após a finalização do processo, no final do arquivo migra.log terá um resumo do que aconteceu com cada banco de dados, se finalizou com sucesso ou com erro.

Para ver detalhes de erros, busque pelo texto "#ERRO#" (para português BR) ou "#ERROR#" (para inglês) dentro do arquivo.

Você pode observar os resultados dos scripts "s10.sys" que contém os resultados da recompilação dos objetos (em ordem de execução) e "s11.sys" que contém o resumo sobre tudo existe, o que foi analisado e o que teve resultado OK.

## 9 Aquisição de licenças

A aquisição de pacotes de conversão de bancos pode ser feita através do site <https://mqfs.com.br/migrfmk>, final da página.

O licenciamento não inclui suporte à migração, mas pode ser contratado de forma separada. Para contratar, solicite através deste e-mail: [edson.gregorio@mqfs.com.br](mailto:edson.gregorio@mqfs.com.br).

## 10 Atualizações

O framework verificará por atualizações a cada migração, e no final do arquivo migra.log informará se existe versão mais recente.

O download de novas versões estarão disponíveis no site <https://mqfs.com.br/migrfmk>, final da página.