GBAK - Firebird Backup and Restore

GBAK is Firebird's command line tool for online backup and restore of a complete database.
GBAK is able to perform a backup while the database is running. There is no need to shut down the database during a GBAK backup. GBAK will create a consistent snapshot of the database at the time it starts running. You will, however, notice a perfomance degradation during the backup, so it is a good idea to backup at night. As GBAK visits all pages of the database, so it will also perform a garbage collection on the database.

General Syntax

gbak <options> -user <username> -password <password> <source> <destination>

Backup

For backups, <source> is the database you want to back up, <destination> is the file name of the backup file. The usual extension for the backup file is .fbk for Firebird and .gbk for InterBase.
Only SYSDBA or the database owner can perform a backup. For multi-file databases, specify only the name of the *first* file as the database name.

Restore

For restores, <source> is the backup file and <destination> is the name of the database that is to be built up from the backup file. You will have to specify the -C option for restore.

General Options

-nodbtriggers	Suppresses Database Triggers from running [Firebird 2.1]
-pas[sword] <password>	Database password
-role <role>	Connect as role
-se[rvice] <hostname>:service_mgr	Backup: Creates the backup file on the database server, using the Service Manager. Restore: Creates the database from a backup file on the server, using the Service Manager.
-u[ser] <username>	Database user name
-v[erbose]	Verbose output of what GBAK is doing
-y <filename>	Redirect all output messages to <filename> The file *must not* exist before running GBAK!
-y suppress_output	Quiet mode

-z	Show GBAK version and server version

Backup Options

-b[ackup_database]	Back up. This switch is optional.
-co[nvert]	Converts external tables to internal tables
-e[xpand]	Creates an uncompressed backup
-fa[ctor] n	Blocking factor for tape device
-g[arbage collect]	Does not perform garbage collection (sweeping) during backup
-ig[nore]	Ignores checksum errors while backing up
-l[imbo]	Ignores limbo transactions while backing up
-m[etadata]	Only backs up metadata (schema). No table data will be stored.
-nt	Non-transportable format (use only when you know you will restore on same platform and database version)
-t[ransportable]	Creates a transportable backup (transportable between platforms and server versions). This is the default.

Restore Options

-bu[ffers]	Set cache size for restored database
-c[reate_database]	Restore to a new database (the target database file MUST NOT exist)
-i[nactive]	All indexes will be restored as INACTIVE

-k[ill]	Does not create shadows that are defined in the backup
-mo[de] read_write	Restores to a read/write database (This is the default)
-mo[de] read_only	Restores to a read-only database
-n[o_validity]	Does not restore validity constraints. So you can restore data that does not meet these constraints and could not be restored otherwise.
-o[ne_at_a_time]	Usually the restore takes place in one single transaction for the whole database. This switch puts a commit after each table. So you can use this to partially restore databases with corrupt table data.
-p[age_size] <size>	Sets page size of new database. <size> can be one of 1024, 2048, 4096, 8192. Default is 1024.
-r[eplace_database]	Restores over an existing database. This can only be performed by SYSDBA or the owner of the database that is overwritten. Do NOT restore over a database that is in use! [Firebird 1.0, 1.5]
-rep[lace_database]	New abbreviation for the old -replace_database [Firebird 2.0]
-r[ecreate_database] o[verwrite]	[Firebird 2.0] Restores over an existing database. This can only be performed by SYSDBA or the owner of the database that is overwritten. Do NOT restore over a database that is in use! -r is equivalent to -c. Only the "overwrite" option will restore over an existing database.
-use_[all_space]	Normally, on restore, database pages will be filled to about 80 %. With the use_all_space option, database pages will be filled to 100 %. (Useful for read-only databases which will see no more modifications.)

Examples

A "normal" Backup

gbak -v -t -user SYSDBA -password "masterkey" dbserver:/db/warehouse.fdb c:\backups\warehouse.fbk

Backup with output to a logfile

del c:\backups\warehouse.log
gbak -v -t -user SYSDBA -password masterkey **-y c:\backups\warehouse.log**
dbserver:/db/warehouse.fdb c:\backups\warehouse.fbk

A "normal" Restore

gbak **-c** -v -user SYSDBA -password masterkey c:\backups\warehouse.fbk dbserver:/db/warehouse2.fdb

Restore to an already existing database (Firebird 1.0, 1.5)

gbak **-c -r** -v -user SYSDBA -password masterkey c:\backups\warehouse.fbk dbserver:/db/warehouse.fdb

Restore to an already existing database (Firebird 2.0)

gbak **-r o** -v -user SYSDBA -password masterkey c:\backups\warehouse.fbk dbserver:/db/warehouse.fdb

Create a read-only database

gbak -c -v **-mode read_only -use_all_space** -user SYSDBA -password masterkey c:\backups\warehouse.fbk c:\files\warehousedb.fdb

Multi-file backups

Syntax for backup

gbak [options] <database> <target file 1> <size 1> <target file 2> <size 2> ... <target file n>
NOTE: Do not specify a size for the last file. It will always be filled to take up what is left over, no matter how large.
Size can be given in bytes (8192), kilobytes (1024k), megabytes (5m), or gigabytes (2g)

Syntax for restore

gbak -c [options] <source file 1> <source file 2> ... <source file n> <database>

Restoring to a multi-file database

gbak -c [options] <source file> <db file 1> <size 1> <db file 2> <size 2> ... <db file n>
NOTE: Do not specify a size for the last database file. It can always grow unlimited to take up the rest.
Size can be given in bytes (8192), kilobytes (1024k), megabytes (5m), or gigabytes (2g)

Restoring from a multi-file backup to a multi-file database

gbak -c [options] <source file 1> <source file 2> ... <source file n> <db file 1> <size 1> <db file 2> <size 2> ... <db file n>

Backing up and Restoring the Security Database

Firebird 1.0, 1.5

You can perform a "normal" backup of the security database. The security database resides in the Firebird directory. It is named

- *ISC4.gdb* in Firebird 1.0 and
- *security.fdb* in Firebird 1.5.

Firebird 2.0

Firebird 2.0 does not allow normal database access to the security database. Its name is now *security2.fdb*
The only way to access the security database is via the Service Manager. As GBAK can also use the Service Manager (Option -se), you can run a backup using this option. However, the backup file will also be written to the server machine.
General Syntax:

gbak <options> -user <username> -password <password> -se <servername>:service_mgr <sec-db-name> <backup-filename>

Example:

gbak -v -t -user sysdba -password masterkey **-se dbserver:service_mgr** c:\Programme\Firebird2\security2.fdb C:\Backups\Security2.fbk

(in this case, Security2.fbk will be written to the C:\Backups folder of dbserver)

When your database server listens on a non-default port:

gbak -v -t -user sysdba -password masterkey -se dbserver/**3051**:service_mgr c:\Programme\Firebird2\security2.fdb C:\Backups\Security2.fbk

Firebird 2.1

In Firebird 2.1 there is a new option -no_dbtriggers that suppresses database triggers from running during backup/restore. So you can suppress any unwanted behaviour for the connection that GBAK needs to establish for the database.

Restoring the Security Database

It is not possible to restore the security database while Firebird is running. In case your security database gets destroyed, this is what you can do:

- Stop the Firebird service/daemon
- Replace the current security database with a new one. If anything else fails, re-install the Firebird server
- You should now have a working SYSDBA login and password. If not, re-install the Firebird server ...
- Start the Firebird service/daemon
- Using GBAK, restore your backup of the security database to a temporary place (like C:\Temp\security.fdb)
- Stop the Firebird service/daemon again
- Now copy the file from the temporary place to the correct place in the Firebird folder
- Start the Firebird service/daemon
- Now you should have your "old" security database back.
- Good Luck! :-)

Upgrading a Security Database to Firebird 2.0

There is a special chapter "Dealing with the New Security Database" about this in the Firebird 2.0 Release Notes, which get installed to the *doc* subdirectory of your Firebird directory. You should also take a look at the *misc\upgrade\security\security_database.txt* file, which explains in detail how to do it.

Stefan Heymann, last edit 2009-02-27