

GBAK - Firebird Backup and Restore

GBAK is Firebird's command line tool for online backup and restore of a complete 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 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

-pa[ssword] <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 <i>must not</i> exist before running GBAK!
-y suppress_output	Quiet mode
-z	Show GBAK version and server version

Backup Options

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

Restore Options

<code>-bu[ffers]</code>	Set cache size for restored database
<code>-c[reate_database]</code>	Restore (mandatory)
<code>-i[nactive]</code>	All indexes will be restored as INACTIVE
<code>-k[ill]</code>	Does not create shadows that are defined in the backup
<code>-mo[de] read_write</code>	Restores to a read/write database (This is the default)
<code>-mo[de] read_only</code>	Restores to a read-only database
<code>-n[o_validity]</code>	Does not restore validity constraints. So you can restore data that does not meet these constraints and could not be restored otherwise.
<code>-o[ne_at_a_time]</code>	Restores one table at a time. You can use this to partially restore databases with corrupt table data
<code>-p[age_size] <size></code>	Sets page size of new database. <size> can be one of 1024, 2048, 4096, 8192. Default is 1024.
<code>-r[eplace_database]</code>	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!

`-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

```
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

```
gbak -c -r -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>
```

Stefan Heymann, 2004-04-12

This documentation is licensed under (choose your favorite): GPL, LGPL, CC, IDPL, GFDL, BSD,
(did I forget one?)