Backup Mode

Before you consider using other tools to take a Database Backup of your Firebird database, you must make sure that you know what the tools do, and how a running database will be affected by them. For example, if you use Winzip to create a compressed copy of a database and you do it when users are accessing the system, the chances of a successful database restore of that database are slim.

You should either use the GBAK or nbackup tools which know how the database works or use GFIX to shut down the database completely before you even attempt to backup the database file(s).

Why should I use GBAK to back up my database?

GBAK is a **powerful and reliable backup and restore tool** that is well-suited for use with InterBase and Firebird databases. By using GBAK to create regular backups of your database, you can ensure that your data is safe and secure and that you can quickly recover in the event of a disaster or data loss.

GBAK creates a consistent backup of the database by starting a transaction that spans the backup period. When the backup is complete, the transaction is ended and this means that the backup process can be run while users are working in the database. GBAK is optimized for performance, and **it creates backups of even very large databases quickly and efficiently**. It uses multithreading and other techniques to maximize the speed of the backup process. However, any **transaction started after the backup process begins, will not have any of the changed data written to the backup file.** The backup will represent a copy of the entire database at the moment the backup began.

The dump file created by a default gbak backup is cross platform(transportable) so a backup taken on a Windows server can be used to recreate the same database on a Linux server, or on any other platform supported by Firebird. The dump file which is created by GBAK is a backup file that contains a complete copy of an Interbase/Firebird database. It is a binary file that includes all the data, metadata, and structural information for the database, as well as any stored procedures, triggers, or other database objects. This is not true of the copies of your database taken (while the database was closed!) with tools such as WinZip etc. **Those copies should only ever be used to restore a database on the same platform as the one copied.**

It is **important** that you must always back up the database with the version of GBAK supplied with the running database server.

Even if the backup is complete and looks to have gone smoothly, you **cannot be certain that the backup file produced is useful unless you test restoring the file** on the same server, or better yet, a separate server. Only then can you be sure that the backup was successful.

The following example shows a backup being taken on a server named linux and used to create a clone of the database on another Linux server named tux to make sure that all was well. First of all, **the backup on linux:**

linux> gbak -backup -verify -y backup.log employee employee.fbk

```
linux> gzip -9 employee.fbk
```

Note: Note that the above gbak command can be written as follows, leaving out the -b[ackup] switch as gbak defaults to running a backup when no other suitable switches are specified:

```
linux> gbak -verify -y backup.log employee employee.fbk
```

Then, **on the tux server:**

```
tux> scp norman@linux:employee.fbk.gz ./
```

Using keyboard-interactive authentication.

Password:

```
employee.fbk.gz | 19 kB | 19.3 kB/s | ETA: 00:00:00 | 100%
```

```
tux> gunzip employee.fbk.gz
```

```
tux> gbak -replace -verify -y restore.log employee.fbk employee.restore.test
```

At this point, the restore has worked and has overwritten the previous database known as employee.restore.test.

The actual location of the database for the database employee.restore.test is defined in the aliases.conf file in /opt/firebird on the server. In this test, it resolves to /opt/firebird/databases/employee.restore.fdb.

For further proof of reliability, the application may be tested against this clone of the live database to ensure all is well.

Speeding up the backup