Changing the database mode

Databases can be set to run in one of two modes, *read only* - where no updates are permitted, and *read/write* - where both reading and writing of data is permitted. By default, Firebird creates read/write databases and as such, all read/write databases must be placed on a file system which allows writing to take place.

Should you wish to put a Firebird database on a CD, for example, you wouldn't be able to do so. After a new database has been populated with data it can be changed to read only mode, and then used on a CD (or other read only file systems) with no problems.

Note: Firebird uses SQL internally to maintain its internal structures with details about transactions, for example, and this is the reason that a database must be placed on a read/write file system regardless of whether only SELECT statements are run or not.

Note: Only databases in dialect 3 can be changed to read-only mode.

The command to set the required mode for a database is:

```
gfix -mo[de] MODE database_name
```

The command takes two parameters, the MODE which must be one of the following:

- read_only - the database cannot be written to.
- read_write - the database can be written to.

The meaning of the two modes should be quite meaningful.

The second parameter is a database name to apply the mode change to.

The following example shows how to put a database into read-only mode, and then change it back again. The example also shows what happens when you try to update the database while running in read only mode.

```
linux> gfix -mode read_only my_employee

linux> isql my_employee
Database: my_employee

SQL> create table test(stuff integer);
Statement failed, SQLCODE = -902
Dynamic SQL Error
-attempted update on read-only database

SQL> quit;

linux> gfix -mode read_write my_employee

linux> isql my_employee
Database: my_employee
```

```
SQL> create table test(stuff integer);

SQL> show table test;
STUFF INTEGER Nullable

SQL> quit;
```

If there are any connections to the database in read/write mode when you attempt to convert the database to read only, the attempt will fail as shown below with Firebird 1.5:

```
linux> gfix -mode read_only my_employee
lock time-out on wait transaction
-lock time-out on wait transaction
-object my_employee is in use

linux> echo $?
0
```

Warning: As with many failures of gfix, the response code returned to the operating system is zero.

Under Firebird 2, the error message is more self explanatory:

```
linux> gfix -mode read_only my_employee
lock time-out on wait transaction
-object /opt/firebird/databases/my_employee.fdb is in use

linux> echo $?
0
```