Maintaining a Firebird Server - part 2

Michaël Van Canneyt

October 22, 2011

Abstract

In a previous article, a backup schedule script for Firebird servers was presented. In this article, a website designed to accompagny this script is presented.

1 Introduction

In a previous article, a set of scripts was presented which maintain a daily backup of a set of firebird database on a server. These scripts are complemented with a small website, consisting of a single CGI binary and a HTML page.

The website allows near-complete management of the databases on the server:

- Maintain the database list.
- View log files of daily backups.
- Backup and restore of databases.
- Upload and download of backups.
- Create archives from backups or unpacking of backups.
- View database statistics or mend databases.
- Run an SQL statement on a database.

The website allows non-technical users to maintain the database server, once it has been set up.

The CGI program is written in Free Pascal, and should work on any Unix system for which FPC can compile (it currently will not run on windows). The Browser side is programmed using ExtJS, meaning that it should work with any browser supported by ExtJS (IE 6-8, Gecko-based browsers such as Firefox, Opera and Safari). The GUI is built using an adapted version of freely available ExtJS classes made by Jozef Sakáloš.

2 Installation

For installation, a webserver is needed. The following explanation shows how to configure it for Apache, but it should work equally well with any server capable of executing a CGI binary. No PHP or any other modules need to be installed for the website to work.

First, ExtJS must be downloaded and installed. ExtJS comes under a dual license, with LGPL 3.0 as one of the available licenses. The site has been programmed in version 3 of ExtJS, which can be downloaded from

```
http://www.sencha.com/products/extjs3/download/
```

The zip file can be unpacked in the `DocumentRoot` directory of the website.

To make it easier to refer to it, an alias `ext` can be created in the Apache configuration file as follows:

```
Alias /ext /var/www/ext-3.4.0
```

The above assumes /var/www/ is the `DocumentRoot` directory. Alternatively, a symbolic link from ext-3.4.0 to ext can be made.

The website assumes that ExtJS is available as `/ext` in the website, so one of the above scenarios must be realized.

The Database management website can be likewise extracted in the `DocumentRoot` directory:

```
gzip -d dbmanage.tar.gz
tar xf dbmanage.tar
```

The website will now be in a directory manage below the documentroot directory, and contains a HTML page and some Javascript files, as well as a set of icons used when displaying the directory tree.

Since it uses a `CGI` application, the manage/server directory must be configured so it allows execution of cgi binaries:

```
<Directory /var/www/manage/server>
  Options +ExecCGI
  AddHandler cgi-script .cgi
</Directory>
```

After this, the website will be available from the following URL:

```
http://your.host.com/manage
```

3 CGI binary compilation

The archive contains a cgi program binary for Linux 64-bit systems in the directory server. For other systems, the cgi program must be compiled from the sources in the src directory. The cgi application is written in Free Pascal, using the Free Pascal framework for web applications. To compile it, release 2.4.4 should be enough. The latest Lazarus release (0.9.30) can also be used.

Prior to compiling, the file cfgdbmanage.pp can be edited, and some of the variables and constants in the interface section can be set to appropriate values. The values of these constants and variables can also be set in the configuration file, so this step is not absolutely necessary, but in particular the password may be set in the binary, so it is not available in readable form in the configuration file (more about the configuration options below).

The following command-line should compile the cgi-bin application:

```
fpc -S2 dbmanage.lpr -o../server/dbmanage.cgi
```

It assumes that Free Pascal is installed, and that the firebird client library is installed, as well as the accompagnying development package. On Ubuntu systems, the packages are called firebird2.1-dev and firebird2.1-common for version 2.1 of firebird.

Note that the binary must be placed in the server directory, and should have extension .cgi in accordance with the Apache configuration. It is recommended to move the sources to a directory outside webserver `DocumentRoot`.

4 Configuration

The CGI application will display 2 file trees: one with backups, one with databases. In order to do this, it must know the location of the databases and of the backups; These locations can be specified in the configuration file, which is a simple `.ini` file, with `Name=Value` pairs:

```
[Global]
Databases=/home/firebird/
Backups=/home/backup/backups/
BackupFileList=/home/backup/databases.list
Logs=/home/backup/logs/
BackupsInSubdirs=1
UserName=SYSDBA
Password=masterkey
```

The meaning of these entries is quite straightforward:

Databases The directory in which all Firebird databases are located. There can be subdirectories.

Backups The directory in which backups of the Firebird databases are placed. Here also, there can be subdirectories.

BackupFileList The list of databases to be backed up. The CGI script must be able to read it, but also allows to manipulate this file.

Logs the directory with the daily backup logs.

BackupsInSubdirs A boolean. Set to 0 or 1 to create backups in subdirectories (1) or not (0).

UserName The firebird user name to be used when performing database operations.

Password The password of the user to be used when performing database operations.

The values for these variables should correspond to the variables set in the firebird daily backup script. The CGI application looks for these settings in the file /etc/dbmanage.cfg.

The CGI application allows to create archives in various formats. It does this using the standard Unix commands, which it will look for in standard locations. The configuration file can be used to specify alternate paths for these tools in the `Tools` section:

tar The standard unix tar command to compress/decompress file.tar archives. default, it is searched in /bin.

bzip The bzip2 command tool to compress/decompress file.bz2 archives. By default, it is searched in /bin.

gzip The gzip (compress) tool to compress/decompress file.gz archives. By default, it is searched in /bin.

zip The zip tool to create .zip files. By default, it is searched in /usr/bin.

unzip The unzip tool to unpack .zip files. By default, it is searched in /usr/bin.

rar The rar tool to create .rar files. By default, it is searched in /usr/bin.

unrar The unrar tool to unpack .rar files. By default, it is searched in /usr/bin.

5 Security and permissions

The CGI process will be executed by the webserver process, and will therefor run as the same user as used by the webserver process - this is `www-data` on an Ubuntu installation. This has some consequences:

1. The configuration file needs to be readable for the webserver user.
2. The databases directory must be readable by the the webserver user. It should not be able to write there.
3. The logs directory with the daily backup logs must be readable by the webserver user.
4. The backups directory must be readable and writable by the webserver user. The daily backup script can change the ownership of the backup files, so it should be configured to set the ownership to the webserver user.
5. The file with the list of databases must be readable. If the CGI program allows editing the list, then it must also be able to write to this. By far the easiest is to make it owned by the webserver user.

A second part of security is website security. As the webpage allows to upload/download files, create backups and restore them, it is not advisable to publish this page to the wide world. If the website is in a private LAN, nothing needs to be done. If it is on a database server in some data center, it is advisable to add some form of HTTP authentication to the website. One way of doing this in Apache is to add the following to the configuration file:

```
<Directory /var/www/manage>
  AuthType Basic
  AuthName "My DB management site"
  AuthUserFile /etc/manage-passwords
  Require valid-user
</Directory>
```

After this, any valid user found in the file /etc/manage-passwords will be able to use the website. A user can be added to this file with the htpasswd command:

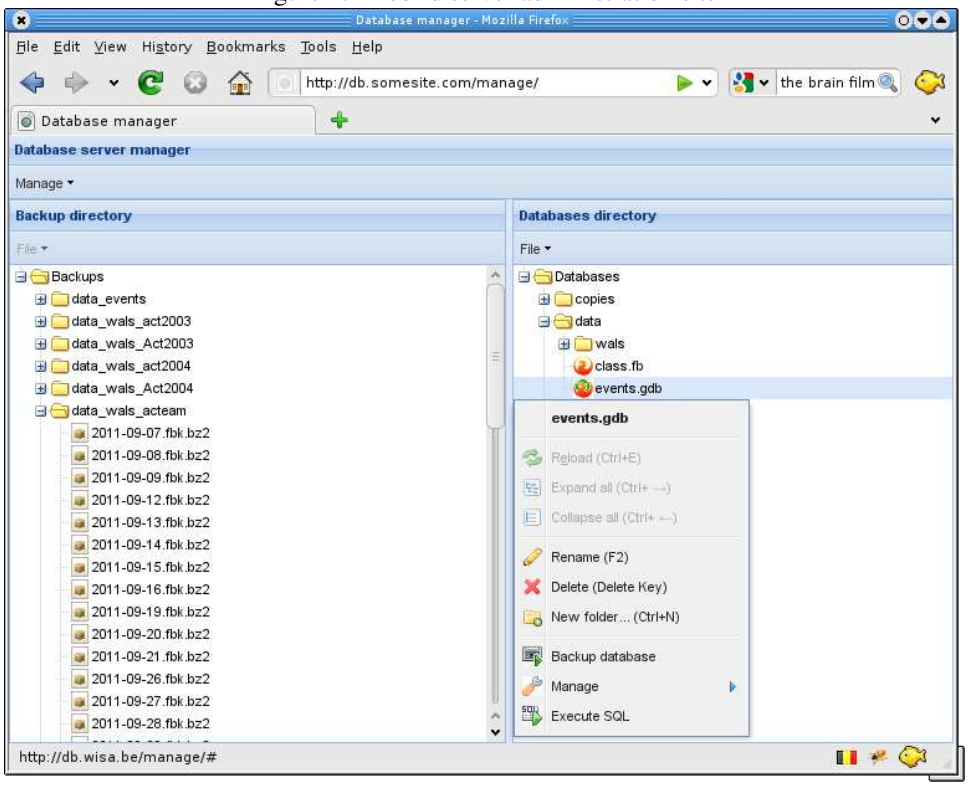
```
htpasswd /etc/manage-passwords anewuser
```

The tool will then prompt for a password for the new user, and write it to the file.

6 Usage

After all this preparation, the site is ready for use. Using the above configuration, the site might look as in figure 1 on page 5 . The site has 2 panes: The left one presents a view

Figure 1: Firebird server administration site



of the backup directory. The right one presents a view of the databases directory. The panes work more or less as a file explorer. When selecting a file, a context menu (available through a right-click) can be pulled up which presents the user with a set of options. The menu can also be shown with the 'File' button above each panel.

The site makes an attempt at showing recognizable icons for many file types, including the Firebird databases, archives and backups. If a Firebird database is included in the list of databases to backup, then the icon is colored differently, offering a visual indication of which databases are under control of the automated backup process.

There are some common file operations, and for the database pane, some extra Firebird database related menu items are available:

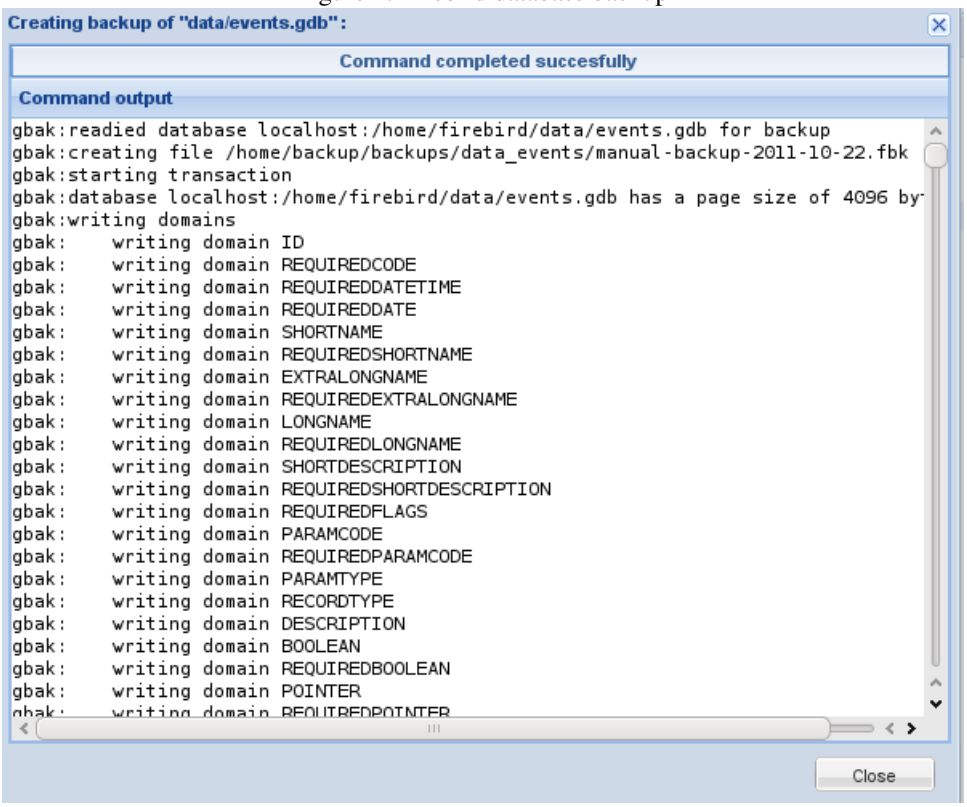
Backup database This will pop up a dialog which allows to take a backup of the Firebird database. The dialog supports all of the gbak options: The CGI application uses the gbak tool of Firebird to create the backup. The output of the gbak tool is shown on the screen as the backup is being taken.

Execute SQL this opens a window where an SQL statement can be types, and which will be executed on the database.

Manage Has some items to view some extra database information (using the Firebird gstat tool), and to add or remove the selected database from the list of databases to backup. Note that this will only work if the CGI application has write access to the backup file list.

The output of the backup tool is shown in figure 2 on page 6. When finished, the backup will show up in the backup directory pane: the CGI program makes sure the backup is

Figure 2: Firebird database backup



created in the directory for backups: the backup filename is always relative to the backup directory.

Likewise, the backup directory pane on the left has a context menu. In addition to the usual file operations (rename, delete, open and upload), there are 3 backup-specific items in this menu:

Restore backup This will pop up a dialog which allows to restore the backup to a new or existing Firebird database.

Create archive This menu item allows to create an archive from a file: this is useful in case one wants to download a backup from a database: first it can be compressed and then downloaded. The menu will be disabled if the file is already compressed.

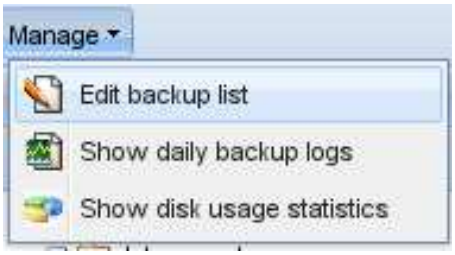
Extract archive After uploading a compressed file (normally a compressed backup) with the upload mechanism, this menu can be used to extract (uncompress) the archive and then restore the backup.

Finally, the `Manage` menu item at the top of the website (figure 3 on page 7) offers some general tools:

Edit backup list This pops up a dialog in which the contents of the backup list can be edited directly. Changes can be made, but can be saved only if the CGI application has write access to the backup file list.

Show daily backup log This will show a dialog with a list of daily backup logs, allowing to select one which can then be viewed. In case a backup failed and a notification was sent, this can be used to inspect the cause of the failure.

Figure 3: Manage menu



Show disk usage Finally, this menu item presents a small dialog which gives an overview of the disks and how much free space the disks still have available.

7 Conclusion

The life of a database server administator does not need to be restricted to command-line tools: the scripts presented in a previous article paired with the website presented here makes daily administration tasks easy, even for the people not so comfortable with the command-line. While it is just an administration tool - as opposed to a complete database design tool - it can easily be extended with additional options to create and design firebird databases.