

Getting started with Firebird

Michaël Van Canneyt

November 5, 2007

Abstract

Firebird is a robust open source enterprise-class database which has been existing much longer than most other open source databases. In this article the installation and initial configuration of Firebird on a Linux system will be discussed.

1 Introduction

Firebird is the open source offspring of Interbase, the Relational Database by Borland. Since the code of Interbase was made open source in 2000, the open source variant of this database has evolved at a very fast pace, building further on it's rock-solid versioning architecture, which has been around since the mid-eighties, far longer than most other open source (or even commercial) databases.

As a result of this heritage, Firebird is a very robust and scalable database, which is virtually maintenance free, and yet has a very small installation footprint: the same installation can be used for an embedded system as well as an enterprise-class installation: the system can be fine-tuned for both situations after the installation is finished. It's compactness does not limit it's feature range: it has all one would expect from a relational database: atomicity, transactional and relational integrity, a strong stored procedure language, views, triggers, generators and the possibility to define external functions. The upcoming 2.1 release will again add more features to the already long feature list.

It runs on a variety of platforms, covering most installations that one is likely to find in a company these days. It comes under a very free license: Firebird can be used freely, even for commercial software, and can be downloaded from the following locations:

<http://www.firebirdsql.org/>

<http://www.ibphoenix.com/>

The second link is on the website of IBPhoenix, a company providing commercial support for Firebird. Similar support can be obtained from the firebird foundation, a foundation which promotes the use of Firebird.

2 Installation

When installing firebird, there are 2 (actually 3) flavours to choose from:

Classic Server is a firebird server which spawns 1 process per connection to a database.

This version can take advantage of multi-processor or multi-core architectures.

SuperServer is a firebird server which runs as a single server instance, but runs a thread per connection. However, the threads run essentially in a linear fashion (requests are

handled one after the other), and no advantage is taken of SMP processing capabilities of the host system.

Embedded is a version of firebird which is a simple loadable library: it is usable for a single process exclusively accessing one or more databases. On Linux, this version is installed together with the classic version.

In terms of server functionality, these 3 versions are the same. Only the way in which multiple connections are handled differs. If SMP functionality is needed (when heavy server load is expected), the best option is to install the classic version.

The current stable version is 2.0.3, but an older 1.5.4 version (a version which contains backported security fixes) is also available. For new installations, using version 2.0.3 is recommended.

The installation procedure differs from system to system. Some Linux distributions come with pre-packaged version of firebird:

Ubuntu has a package for firebird, a simple

```
sudo apt-get install firebird2-super-server firebird2-utils-super firebird2-d
```

will install the firebird 2 superserver or classic server (despite the deceptive package name, you will be presented with the choice) and the client and development tools as well as the example database.

Debian works essentially in the same way as Ubuntu.

Mandriva has packaged firebird in their distribution, so it can be simply installed with the system itself. The package is called firebird.

OpenSuSE has no default installation of firebird, but support for the 32-bit version firebird has been added to the OpenSuse Build services (at least in version 10.3). Adding the 'OpenSuSE build service - Databases' repository to the installation sources of YaST allows to install Firebird 2.0.0 directly from YaST. Normally, installing the more up-to-date Mandriva RPMs should also work.

If no package is available for the particular system, an archive with an install script can be downloaded from the abovementioned website, where also a generic rpm is available.

The archive provides a very simple install process. After downloading the archive, the following commands will install Firebird 2.0.3 superserver for a AMD64 system:

```
tar xzf FirebirdSS-2.0.3.12981-1.amd64.tar.gz
cd FirebirdSS-2.0.3.12981-1.amd64
sh install.sh
```

The install script then shows something like the following:

```
Firebird super 2.0.3.12981-1.amd64 Installation

Press Enter to start installation or ^C to abort
Extracting install data
Updated /etc/services
firebird          0:off  1:off  2:on   3:on   4:off  5:on   6:off
Starting Firebird
Please enter new password for SYSDBA user: masterkey
Install completed
```

This will prompt for a password for the SYSDBA firebird user. The SYSDBA user is the database administrator account of Firebird, not the operating system user with whose credentials the server is running. The SYSDBA user has unlimited access to all Firebird databases on the system (and hence can destroy them), so it is vital that a good password for this database user is chosen.

Automatic installers generate a random password, which is stored in the `SYSDBA.password` file in the Firebird installation directory. This file is readable only by root.

Whatever the installation method chosen, the installation will create a firebird user. The firebird server process will run with the credentials of this user, so it is best to set a password for this user.

The installer will also configure the runlevels so the firebird server will be started whenever the system is booted. The firebird server can also be started or stopped with the runlevel scripts:

```
/etc/init.d/firebird start
```

or

```
/etc/init.d/firebird stop
```

The exact name and location of this script depends on the system. On Ubuntu, the script is e.g. called `firebird2`

After the installation was performed, the installation can be tested by connecting to the sample database. This database is called `employee.fdb` and is installed in the directory `examples/empbuild`. Connecting to a database can be done by the command line interactive query tool, `isql`:

```
cd /opt/firebird/examples/empbuild/  
isql -USER SYSDBA -PASSWORD masterkey employee.fdb
```

if all goes correctly, then `isql` will respond with

```
Database: employee.fdb, User: SYSDBA  
SQL>
```

And waits for a SQL command. The command `SHOW TABLES` will produce the following:

```
SQL> SHOW TABLES;  
          COUNTRY                CUSTOMER  
          DEPARTMENT             EMPLOYEE  
          EMPLOYEE_PROJECT        JOB  
          PROJECT                 PROJ_DEPT_BUDGET  
          SALARY_HISTORY           SALES
```

Note that each command must be ended with a semicolon. The `HELP` command gives a list of available commands. The `EXIT` command will exit the interactive SQL tool.

If all this worked without error, the firebird database is installed without errors and is ready to be used.

Note: the `UnixODBC` package also installs a `isql` command tool. If this binary comes first in the `PATH`, then this tool will be started instead of the firebird `isql` tool. In order to disambiguate, it may be best to create an alias for the firebird tool. In the C shell this would look something like this:

```
alias fisql /opt/firebird/bin/isql
```

Another method - regardless of the shell used - is to create a symbolic link:

```
ln -s /opt/firebird/bin/isql /usr/bin/fisql
```

3 Configuration

In fact, there is not much which needs to be configured in firebird. After the installation, the system is ready to go. However, there are some things which should be considered if one wants more strict control over what the server is allowed to do.

Firebird has 1 configuration file `firebird.conf`, in which various parameters can be set. This file is located in the firebird installation directory. It is a simple text file which can be edited with any text editor. When changes are made to this file, the server should be restarted. Classic server will re-read relevant parts of the file when a new connection is made, but for some settings the server must be restarted. Below, some of the Important security-related configuration parameters are discussed.

The `DatabaseAccess` parameter determines where databases may be located on the system. It can have the following values

Full meaning databases can be anywhere on the filesystem where the firebird user has write access. This is the default value.

Restricted this value, followed by a semicolon-separated list of directories, restricts database locations to the listed directories. If relative paths are used, they are relative to the installation directory of firebird.

None only databases defined in the `alias.conf` file can be accessed. The contents and meaning of this file will be described below.

Obviously, the most secure option is `None`. For a dedicated server serving only a couple of databases, this is the safest option.

Firebird allows to declare tables in external files. The `ExternalFileAccess` parameter determines where these files may be located on the system. It can have the same values as `DatabaseAccess`, except that the `None` value disallows the use of tables in external files.

Firebird can define UDF (User Defined Functions) in a database. The `UdfAccess` parameter determines where the libraries containing these functions can be located. It also can have the same values as the `DatabaseAccess` parameter.

Finally, the `TempDirectories` parameter determines in which directories any temporary files made by Firebird can be located. By default, they are placed in the system temporary directory. It's good practice to have them located somewhere else, as they can fill up the disk.

With the exception of the UDF directory, all directories mentioned here should be writable by the firebird user, as the database server will be writing files to these directories.

The last parameter of interest in the configuration file is the `DefaultDbCachePages` parameter: it determines the number of database pages Firebird keeps in memory. For Classic Server, this number is a per-connection (and hence per process) setting, defaulting to 75. For SuperServer this number is per opened database, defaulting to 2048.

Increasing this parameter means faster access to the database, but also increases memory usage. Setting it too high may degrade performance if the firebird process(es) start using more memory than is available when the number of connections grows.

There are many more parameters available to fine-tune the server, but the above ones are the main ones one should initially care about as they affect the security of the server.

The Firebird Server allows to define aliases for the databases it controls. Aliases make it much more easy to connect to databases on a server, as the user does not need to remember the full path of the database file. It also gives the database administrator the ability to move the files to another location, without the need to reconfigure all clients. If the `DatabaseAccess` parameter in the `firebird.conf` configuration file is set to `None`, only database names in the `aliases.conf` file can be accessed.

Defining aliases can be done in the `aliases.conf` file in the firebird installation directory. This file has a very simple format, it consists of lines of the form

```
Alias=filename
```

The alias should not contain spaces, and the filename is the full path to a firebird database file. Lines starting with the `#` character are treated like comments.

4 PHP support for firebird

PHP is a common web server scripting language. On most linux systems it comes pre-installed with MySQL support (the famous LAMP stack) but not with Firebird support. Adding support for firebird to PHP must be done manually.

Ubuntu users can use the package system to install Firebird support for PHP:

```
sudo apt-get php5-firebird
```

Mandriva users can use the `php-firebird rpm` to install firebird support in their PHP installation. The OpenSUSE build services also contain a `php-firebird` package, which works only with Firebird 2.0 on a 32-bit system.

If none of the above approaches works, then the firebird module (actually the interbase module) must be compiled manually for PHP. To do this, the PHP sources should be downloaded and unpacked. When firebird support must be added to an existing installation, then the sources corresponding to the installed PHP version must be downloaded.

After downloading, the following steps can be done to compile a PHP version with support for Firebird:

```
tar xjf php-5.2.4.tar.bz2
cd php-5.2.4
./configure --enable-shared --with-interbase=shared,/opt/firebird
make
```

If the configure step complains about not being able to find firebird, make sure the `firebird2-devel` package was installed (this is always so when using the firebird install script) and try again.

If a complete PHP is being installed, then the `make install` command will install the new PHP version to its location, including the interbase module.

If the firebird module needs to be added to an existing installation, then the following can be done:

```
cp ./modules/interbase.so /usr/lib/php5/extensions/
```

The location of the extensions directory may differ, as it depends on the system. The location is indicated in the `php.ini` file. (found, on a SuSE system in `/etc/php5/apache2/php.ini`)

After this, the `php.ini` file must be extended so it contains a line

```
extension=interbase.so
```

Restarting apache should then ensure that the firebird support is loaded. To verify this, the following php script can be used:

```
<?php
    phpinfo(INFO_MODULES);
?>
```

This will display the loaded PHP modules, and `interbase` should be one of them.

When the support for Firebird was added to PHP, then the `IBWebAdmin` package can be installed. This is a web-based Interbase and Firebird management tool, which allows managing almost all aspects of a Firebird server. It can be downloaded from

<http://www.ibwebadmin.net/>

Installation is very simple: simply unpack the archive in a location where the webserver can see it, and edit the `inc/configuration.inc.php` file: the location of the firebird user database and binaries must be set, as well as the location of the databases that may be configured using the web interface. When properly set up, something like figure 1 on page 7 will appear when the browser is first pointed to the `ibwebadmin` installation.

5 GUI management tools

There are 3 graphical management tools available for Firebird on Linux systems. The first one is `Flamerobin`, the second is `kfb` and the last one is `Pre-packaged binaries` exist for both, but many are outdated, so it is a better option to compile these tools manually. The installation of `Flamerobin` in this respect is the easiest.

The sources can be downloaded from its website:

<http://www.flamerobin.org/>

`Flamerobin` is cross-platform, and for this it needs the `wxWindows` widget set. This needs to be installed or compiled first, together with the contributed tools (`Flamerobin` needs the 'stc' contribution).

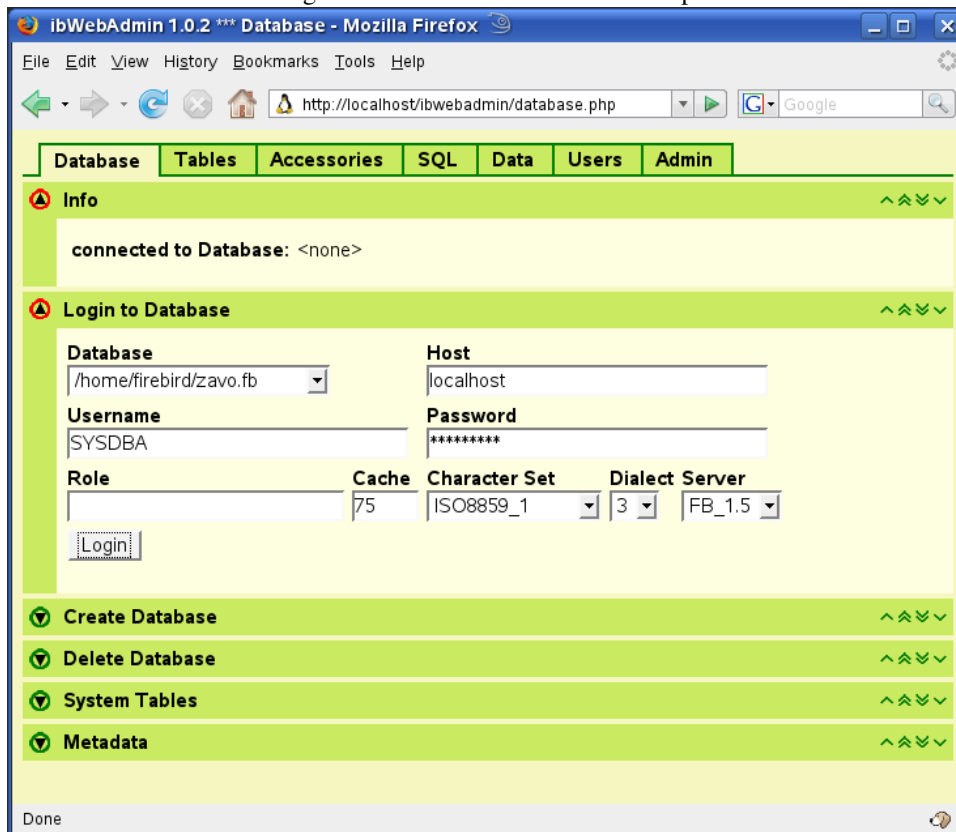
`wxWindows` comes pre-installed on many distributions. For those that do not have it installed, it can be compiled very easily. The sources can be downloaded from its website:

<http://www.wxwidgets.org/>

After downloading the sources to a suitable location, the following commands (run as root) will install `wxWidgets`:

```
tar -xzf wxGTK-2.8.6.tar.gz
cd wxGTK-2.8.6
./configure
make
```

Figure 1: IBWebAdmin on first startup



```
make install
cd contribs
make
make install
```

When the installation of wxWidgets was successful, flamerobin itself can be compiled and installed:

```
tar xzf flamerobin-0.8.1-src.tar.gz
cd flamerobin-0.8.1-src
sh configure
make
make install
```

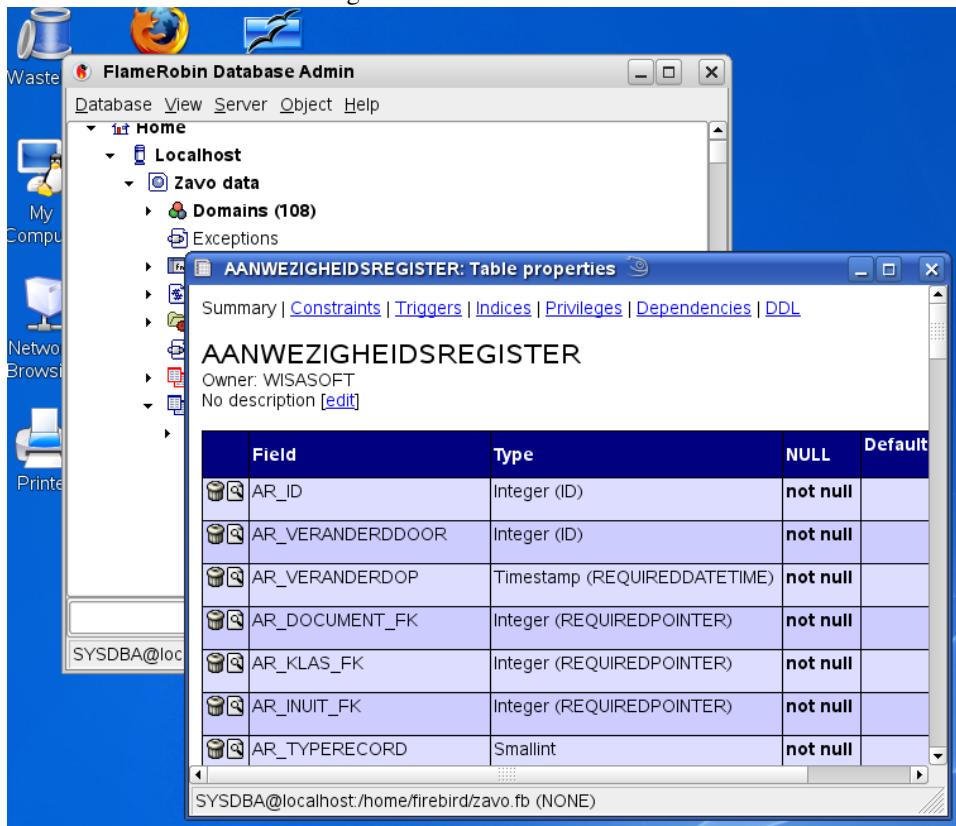
Starting it up with the 'flamerobin' command will show something like figure 2 on page 8. Compilation of kfb is more complicated, and the author did not succeed in getting a working binary compiled. Those readers that nevertheless would like to give it a try can download the sources from the home page.:

<http://kfbsite.googlepages.com/>

A simple sequence of `./configure` and `make` commands should in theory produce a working binary.

Finally, the sources for the `fenixsql` project (and the `fplib` component package) can be downloaded from

Figure 2: Flamerobin in action



<http://fbllib.altervista.com/>

Installation of the fbllib package in the Lazarus IDE will allow to open and compile the fenixsql.lpi project, producing a working fenixsql binary.

6 Conclusion

In this article, the first steps of a firebird installation were lined out, and minimal configuration was discussed. Support for firebird was added to PHP, enabling the installation of a web-based database management tool, and finally a nice GUI firebird management tool (flamerobin) was created. In a next contribution, further configuration issues such as user management and backup scheduling will be discussed.