



Firebird Shell Scripts

Norman Dunbar

Version 1.3, 20 June 2020

Table of Contents

1. Introduction	2
2. ChangeDBAPassword	3
3. CreateAliasDB	4
3.1. createAliasDB.sh Parameters	4
4. Fb_config	5
4.1. fb_config Options	5
5. ChangeRunUser	7
5.1. ChangeRunUser.sh Parameters	7
6. RestoreRootRunUser	9
7. ChangeGDSLibraryCompatibleLink	10
Appendix A: Document history	11
Appendix B: License notice	12

Chapter 1. Introduction

After a successful installation of Firebird, the `/opt/firebird/bin` directory will contain a number of useful shell scripts. This manual gives details of what these scripts are for and how they are used.

In the remainder of this manual, we will discuss the following scripts:

- `changeDBAPassword.sh`
- `createAliasDB.sh`
- `fb_config`
- `changeRunUser.sh`
- `restoreRootRunUser.sh`
- `changeGdsLibraryCompatibleLink.sh` (Classic server only.)



The above list is correct at the time of writing and, unless otherwise noted, only applies to Firebird 1.5, 2.0 and 2.1 Super Server installed on a Linux system. Other flavours of Unix may have different scripts. I am currently unable to discuss those potential changes as I do not have access to other Unix systems.

Chapter 2. ChangeDBAPassword

The `changeDBAPassword.sh` script allows the password for the SYSDBA user to be changed and various startup scripts etc to have their ownership changed accordingly. The script is run in non-interactive mode as part of the installation process to create an initial randomly generated password which is stored in the `/opt/firebird/SYSDBA.password` file.

Up until version 2.1 of Firebird, the password is subsequently used in the startup script `/etc/rc.d/init.d/firebird`, which also has a symlink set up to point to `/etc/init.d/firebird`.

The security database (`/opt/firebird/security.fdb` or `/opt/firebird/security2.fdb`—depending on your Firebird version) is also updated with the new password.



The script must be run as the root user, and when run, `changeDBAPassword.sh` will prompt you for the current SYSDBA password and then for a new password. Both of these will appear on the display so for enhanced security, don't allow anyone to look over your shoulder when you run the script.

After the script has been run, the file `/opt/firebird/SYSDBA.password` will contain the password in plain text, so make sure that this file is not readable by anyone except root.

The following is an example of running the script to change the SYSDBA password from 'masterkey' to 'biroguin' which, being a made up word, should be less crackable or guessable.

```
# cd /opt/firebird/bin
# ./changeDBAPassword.sh
Please enter current password for SYSDBA user : masterkey
Please enter new password for SYSDBA user : biroguin
GSEC> GSEC>
```

Running `ed` to modify `/etc/init.d/firebird`



Whenever you change the SYSDBA password using the `gsec` utility, you should also change the startup script file as well. To ensure a complete update, always use this script when changing the SYSDBA user's password.

This doesn't apply when using Firebird 2.1 and above as the startup script no longer requires the SYSDBA password.

In the example above, the last line of output will not be displayed on systems running Firebird 2.1 and above as the startup scripts are no longer changed when the SYSDBA password is modified.

Chapter 3. CreateAliasDB

The `createAliasDB.sh` script allows the creation of a new database to be carried out, and an alias for that database to be created in the file `/opt/firebird/aliases.conf`.



The `createAliasDB.sh` script must be run as the root user.

If your system is not set up properly, the database creation step may fail but the alias will still be added to the aliases file. This could leave you subsequently unable to add the alias properly, as the script checks to ensure that you do not overwrite an existing alias. You will have to manually edit the alias file to remove the broken alias.

All databases must be owned by the `firebird` user, and by the `firebird` group. The following shows a new directory being created by the root user to allow Firebird databases to be created.

```
# cd /u01
# mkdir databases
# chown firebird:firebird databases
```

At this point the directory `/u01/databases` is available for use as a repository for one or more Firebird databases. Obviously, in the above example, the `/u01` directory already existed.

3.1. createAliasDB.sh Parameters

To run the `createAliasDB.sh` script, use a command line similar to the following:

```
# createAliasDB.sh <new_alias> <database_name>
```

The script takes two parameters on the command line, both of which are mandatory:

new_alias

The first parameter is the new alias you wish to create. This alias must not already exist in the alias file, or an error will be displayed and no further action taken.

database_filename

The second parameter specifies the *full path* to the database file. You must not specify a relative path as this could lead to incorrect database filenames being used at connection time. The script will reject any attempt to pass a relative pathname instead of a full pathname.

A brand new empty database will be created provided that the name passed to the script doesn't already exist. If the database already exists, only the alias will be created and added to the alias file.

Chapter 4. Fb_config

The fb_config script is intended to be used from within a makefile to supply various settings and compiler options which are accurate and specific to the installation being used.



This script can be run by any user who has been given execute privileges to it. You do not need to be root to use this script.

4.1. fb_config Options

To run the fb_config script, use a command line similar to the following:

```
fb_config <option> [ <option> [...]]
```

The script takes one or more options on the command line:

--help

This parameter displays the list of allowed options. It should not be supplied in addition to other options.

--cflags

This option returns the list of directories where Firebird include files are stored. This is required by the C and C++ compilers to allow compiler #include directives to be correctly resolved. On my own system, this option returns '-I/opt/firebird/include'.

--libs

This option returns the list of directories where Firebird libraries are located and a list of those libraries that are required to be linked by the linker to create a client server Firebird application. This option returns '-L/opt/firebird/lib -lfbclient' on my system.

--embedlibs

This option returns the list of directories where Firebird libraries are located and a list of those libraries that are required to be linked by the linker to create an embedded Firebird application. This option returns '-L/opt/firebird/lib -lfbembed' on my system.

--bindir

On my system, this option returns '/opt/firebird/bin' as the full path to the Firebird /bin directory.

--version

This option returns a three part version string made up of the concatenation of the Firebird build version, a dash, the package version, a dot and the system architecture. My own laptop Linux system returns '1.5.0.4290-0.i686'.

The following is a brief excerpt from a makefile which shows how to define two macros, FBFLAGS and FBLIBS, and initialise them to the correct values using fb_config. Note the use of the back tick

character (`) rather than a single quote character (').

```
...  
FBFLAGS = `fb_config --cflags`  
FBLIBS = `fb_config --libs`  
...
```

Chapter 5. ChangeRunUser

Under Firebird 1.5 there are two versions of the `changeRunUser.sh` script, the one prefixed 'SS' is for Super Server installations and the one prefixed 'CS' is for Classic Server installations. In Firebird 2 onwards, the script is simply `changeRunUser.sh`.



The script should be run as root.

The `changeRunUser.sh` script allows the user and group, under which the Firebird server runs, to be changed. By default, this is now the `firebird` user and group, however, in previous versions the Firebird server ran as the root user which is undesirable from a system security point of view and allowed databases to be created all over the file system. With the new `firebird` user, restrictions can be placed on where databases can be created.

The script changes the owning user and group of a number of files in the Firebird installation directory, the logfile and also the startup script `/etc/rc.d/init.d/firebird` which is used to start and stop the Firebird server.

5.1. ChangeRunUser.sh Parameters

To run the script, use a command line similar to the following:

```
changeRunUser.sh <username> <groupname>
```

The script takes two parameters on the command line, both of which are optional as you will be prompted if both are omitted. If you only supply one parameter, it is assumed to be the username and you will be prompted for the groupname.

username

This parameter sets the username under which the Firebird server is to run. The supplied value is validated against entries in `/etc/passwd`.

groupname

This parameter sets the groupname under which the Firebird server is to run. The supplied value is validated against entries in `/etc/group`.

The following example shows the use of `changeRunUser.sh` to change the owning user and group to `firebird`. The `firebird` user and group is actually the default when Firebird is installed so there is no need for you to run the script unless you have changed these details already.

```
# cd /opt/firebird/bin
# ./changeRunUser.sh firebird firebird
Updating /opt/firebird
Updating startup script
Completed
```

If you are running Firebird 2.1 or higher, there is no longer a prompt if you run this script with no parameters. The script now defaults to firebird for the user and group if you do not supply any parameters. Running the script under 2.1 gives the following:

```
# cd /opt/firebird/bin
# ./changeRunUser.sh
Shutting down Firebird                                done

Change Firebird install for /opt/firebird to uid=firebird gid=firebird
(User or group options can be changed by editing this script)

Press return to continue - or ^C to abort

Updating /etc file(s)
Updating /opt/firebird
Starting Firebird                                    done
Completed.
```



When this script prompts you to "*Press return to continue ...*" any VNC sessions connected to the *database server* will have been killed. Please ensure that you have the server at a quiet period and not during normal running hours when you make this change. This affects Suse Linux Enterprise 10 but may affect other Linux distributions as well.

Chapter 6. RestoreRootRunUser

Under Firebird 1.5 there are two versions of the `restoreRootRunUser.sh` script. The one prefixed 'SS' is for Super Server installations and the one prefixed 'CS' is for Classic Server installations. In Firebird 2 onwards, the script is simply `restoreRootRunUser.sh`.



This script must be run as root.

This script simply restores the old style installation format whereby the Firebird Super Server runs as the root user and group.

Chapter 7. ChangeGDSLibraryCompatibleLink

This script, `changeGdsLibraryCompatibleLink.sh`, is available with Classic Server installations only, and is used to change the symlink `libgds.so` to point to the appropriate library for the installation. There are two possible libraries that the symlink can point to:

- `/opt/firebird/lib/libfbclient.so` for client server applications
- `/opt/firebird/lib/libfbembed.so` for embedded server applications.

After installation, the `libgds.so` symlink points to the client server library by default so if you are running an embedded application, you need to run this script to point `libgds.so` at the embedded library instead.



This script must be run as root.

The following example shows how this script is used to change from embedded server to client server use:

```
# cd /opt/firebird/bin
# ./changeGdsCompatibleLibraryLink.sh
For classic server there are two optional backward compatible client
libraries. These are libfbclient.so and libfbembed.so.

libfbclient.so) enables your client to be multithreaded but must
                 connect to a database via a server.
libfbembed.so)  allows the client to directly open the database file,
                 but does not support multithreaded access

Your current setting is:
/usr/lib/libgds.so -> /opt/firebird/lib/libfbembed.so

Which option would you like to choose (client|embed|remove)
                                     [client] client
#
```

The default option is `client` which will recreate the symlink to the client server library, `embed` will recreate the symlink to the embedded server, while `remove` will remove the symlink altogether.

There are no messages displayed to inform you of the success of the script, however, if you run it again, you will notice the current setting should be different to that displayed when you previously ran the script.

Appendix A: Document history

The exact file history is recorded in the firebird-documentation git repository; see <https://github.com/FirebirdSQL/firebird-documentation>

Revision History

0.1	22 Nov 2004	ND	Created as a chapter in the Command Line Utilities manual.
1.0	19 Oct 2009	ND	Updated for Firebird 2, minor changes to punctuation and then converted to a stand alone manual.
1.1	05 Jan 2010	ND	Spelling correction.
1.2	27 Apr 2010	ND	When changing the DBA password, it is no longer the case that the startup script is also amended. Other changed for version 2.1 added.
1.3	20 Jun 2020	M R	Conversion to AsciiDoc, minor copy-editing

Appendix B: License notice

The contents of this Documentation are subject to the Public Documentation License Version 1.0 (the “License”); you may only use this Documentation if you comply with the terms of this License. Copies of the License are available at <https://www.firebirdsql.org/pdfmanual/pdl.pdf> (PDF) and <https://www.firebirdsql.org/manual/pdl.html> (HTML).

The Original Documentation is titled *Firebird Shell Scripts*.

The Initial Writer of the Original Documentation is: Norman Dunbar.

Copyright © 2004–2010. All Rights Reserved. Initial Writer contact: NormanDunbar at users dot sourceforge dot net.