# Server configuration and management

There are several things you should be aware of – and take care of – before you start using your freshly installed Firebird server. This part of the manual introduces you to some useful tools and shows you how to protect your server and databases.

# User management: gsec

Firebird comes with a command-line user management tool called gsec. Although its functions can also be performed by a number of third-party GUI utilities, you should at least have a basic knowledge of gsec, since this is the official tool and it's present in every Firebird server installation. In the next sections you will use gsec to execute three tasks: changing the SYSDBA password, adding Firebird users and (optionally) appointing coadministrators. First though, some points of attention:

*Permission to run gsec*

With some Firebird installations, you can only run gsec if you are logged into the operating system as Superuser (root on Linux) or as the user the Firebird server process runs under. On Windows server platforms, you typically need to be in the Power User group or higher to run gsec successfully.

*Trouble running gsec*

If you have enough privileges but invoking gsec results in a message like "cannot attach to password database - unable to open database":

- You may be running Firebird on Windows and for some reason the local protocol isn't working. One rather common cause for this is running Windows with Terminal Services (Remote Desktop Services) enabled and connecting to the server from a different session. To enable the local protocol, open firebird.conf, uncomment the *IpcName* parameter and set it to Global\FIREBIRD. Then restart the server.

*Note*: In Firebird 2.0.1 and up, Firebird automatically prepends Global\ to the IPCname if the connection fails because of insufficient permissions, so this should not happen anymore.

- If the above doesn't apply to you, you can at least circumvent the problem by "tricking" gsec into using TCP/IP. Add the following parameter to the command line, adjusting the path if necessary:

```
database "localhost:C:\Program Files\Firebird\Firebird_2_5\security2.fdb"
```

The file security2.fdb is the security database, where Firebird keeps its user account details. It is located in your Firebird installation directory.

- Maybe your security database is a renamed security.fdb from Firebird 1.5 or earlier. Of course this can't be the case immediately after installation.

Someone (you?) must have put it there, in order to keep the existing accounts available. Consult the Release Notes for instructions on how to upgrade old security databases. If the error message starts

with “Cannot attach to services manager”, the server may not be running at all. In that case, go back to Testing your installation and fix the problem.

*Invoking gsec on Linux*

On *nix systems, if you call gsec from its own directory, you should type ./gsec instead of just gsec. The current directory is usually not part of the search path, so plain gsec may either fail or launch a “wrong” gsec.

back to top of page

### Changing the SYSDBA password

One Firebird account is created automatically as part of the installation process: SYSDBA. This account has all the privileges on the server and cannot be deleted. Depending on version, OS, and architecture, the installation program will either

- install the SYSDBA user with the password masterkey (actually, masterke: characters after the eighth are ignored), or
- ask you to enter a password during installation, or
- generate a random password and store that in the file SYSDBA.password within your Firebird installation directory.

If the password is masterkey and your server is exposed to the Internet at all – or even to a local network, unless you trust every user with the SYSDBA password – you should change it immediately using the gsec command-line utility. Go to a command shell, cd to the Firebird \bin subdirectory and issue the following command to change the password to (as an example) icuryy4me:

```
gsec -user sysdba -pass masterkey -mo sysdba -pw icuryy4me
```

Notice that you specify sysdba twice in the command:

- With the -user parameter you identify yourself as SYSDBA. You also provide SYSDBA's current password in the -pass parameter.
- The -mo[dify] parameter tells gsec that you want to modify an account – which happens to be SYSDBA again. Lastly, -pw specifies the type of modification: the password.

If all has gone well, the new password icuryy4me is now encrypted and stored, and masterkey is no longer valid. Please be aware that unlike Firebird user names, passwords are case-sensitive.

back to top of page

### Adding Firebird user accounts

Firebird allows the creation of many different user accounts. Each of them can own databases and also have various types of access to databases and database objects it doesn't own.

Using gsec, you can add a user account as follows from the command line in the Firebird \bin subdirectory:

```
gsec -user sysdba -pass masterkey -add billyboy -pw sekrit66
```

Provided that you've supplied the correct password for SYSDBA, a user account called billyboy will now have been created with password sekrit66. Remember that passwords are case-sensitive.

Firebird 2.5 also introduces SQL commands for user management. While attached to any database, SYSDBA (or co-admins, see below) can create, alter and drop users like this:

```
create user sonny password 'cher_ie'
alter user sonny password '9hgf72354b'
drop user sonny
```

Other parameters for CREATE/ALTER USER are FIRSTNAME, MIDDLENAME and LASTNAME. Like PASSWORD, they all take a string argument.

Ordinary Firebird users can alter their own account details with gsec (gsec -user toby -pass hEltoPay -mo toby -pw purgaToby) and with SQL (alter user toby password 'purgaToby'). Only the account name itself can never be changed, not even by SYSDBA.

back to top of page

#### Appointing co-administrators

Note: What follows here is not essential knowledge for beginners. You can skip it if you like and go on to the Security section.

In Firebird 2.5 and up, SYSDBA (and others with administrator rights) can appoint co-administrators. In gsec this is done by adding the -admin parameter:

```
gsec -user sysdba -pass masterkey -add bigbill -pw bigsekrit -admin yes

gsec -user sysdba -pass masterkey -mo littlejohn -admin yes
```

The first command creates user bigbill as a Firebird administrator, who can add, alter and drop users. The second command grants administrator privileges to the existing user littlejohn.

The SQL equivalents of these commands are:

```
create user bigbill password 'bigsekrit' grant admin role
alter user littlejohn grant admin role
```

To revoke administrator privileges with gsec, use -admin no. In SQL, use REVOKE ADMIN ROLE.

*Notes*:

- GRANT ADMIN ROLE and REVOKE ADMIN ROLE are not GRANT and REVOKE statements, although they look that way. They are parameters to the CREATE and ALTER USER statements. The actual role name involved here is RDB$ADMIN. This role also exists in regular databases; more about that in a minute.
- Every user who has received administrator rights can pass them on to others. Therefore, there

is no explicit WITH ADMIN OPTION.

back to top of page

### Differences between co-administrators and SYSDBA

- Co-admins can create, alter and drop users, but they have no automatic privileges in regular databases, like SYSDBA has.
- Unlike SYSDBA, co-admins must specify the extra parameter -role rdb$admin every time they invoke gsec to add, modify, drop or view users.
- Co-admins who want to use the SQL user management commands must specify the RDB$ADMIN role when connecting. However, since nobody can connect to the security database, this requires a little trickery. First, the co-admin has to be granted the RDB$ADMIN role in at least one regular database as well. This is done in the usual way:

```
grant rdb$admin to bigbill
```

Grantors can be the database owner, SYSDBA, and every other user who has the RDB$ADMIN role in that database and has specified it while connecting. Every RDB$ADMIN member in a database can pass the role on to others, so again there is no WITH ADMIN OPTION. Once the co-admin has obtained the role, he can connect to the (regular) database with it and use the SQL user management commands. It's not the most elegant of solutions, but that's how it works.

*Please remember:*

The RDB$ADMIN role in a database gives the grantee SYSDBA rights *in that database only*!

- If it is the security database, the grantee can manage user accounts, but gets no special privileges in other databases.
- If it is a regular database, the grantee can control that database like he was SYSDBA, but again has no special privileges, and has no user administation privileges.

Of course it is possible to grant a user the RDB$ADMIN role in several databases, including the security database.

back to top of page

# Security

Firebird 2.5 offers a number of security options, designed to make unauthorised access as difficult as possible. Be warned however that some configurable security features default to the old, “insecure” behaviour inherited from InterBase and Firebird 1.0, in order not to break existing applications.

It pays to familiarise yourself with Firebird's security-related configuration parameters. You can significantly enhance your system's security if you raise the protection level wherever possible. This is not only a matter of setting parameters, by the way: other measures involve tuning file system access permissions, an intelligent user accounts policy, etc.

Below are some guidelines for protecting your Firebird server and databases.

*Run Firebird as non-system user*

On Unix-like systems, Firebird already runs as user firebird by default, not as root. On Windows server platforms, you can also run the Firebird service under a designated user account (e.g. Firebird). The default practice – running the service as the LocalSystem user – poses a security risk if your system is connected to the Internet. Consult README.instsvc in the \doc sub-directory to learn more about this.

*Change SYSDBA's password*

As discussed before, if your Firebird server is reachable from the network and the system password is masterkey, change it.

*Don't create user databases as SYSDBA*

SYSDBA is a very powerful account, with full (destructive) access rights to all your Firebird databases. Its password should be known to a few trusted database administrators only. Therefore, you shouldn't use this super-account to create and populate regular databases. Instead, generate normal user accounts, and provide their account names and passwords to your users as needed. You can do this with gsec as shown above, or with any third-party Firebird administration tool.

*Protect databases on the file system level*

Anybody who has file system-level read access to a database file can copy it, install it on a system under his or her own control, and extract all data from it – including possibly sensitive information. Anybody who has file system-level write access to a database file can corrupt it or totally destroy it. As a rule, only the Firebird server process should have access to the database files. Users don't need, and should not have, access to the files – not even read-only. They query databases via the server, and the server makes sure that users only get the allowed type of access (if at all) to any objects within the database. An exception to this rule is the Windows Embedded Server, which requires that users have proper access rights to the database file itself.

*Disable Classic local mode on Linux*

Another exception to the above rule is the so-called local or embedded access mode of the Firebird Classic and Superclassic servers on Linux. Here too, users must have proper access rights to the database file itself. They also need read access to the security database security2.fdb. If this worries you (and it probably should), reserve file system access to the security database (and other databases, while you're at it) to the server process only. Users are then obliged to connect via the network layer. Don't remove the libfbembed library from your system, though: it contains the complete server engine used by your Classic or Superclassic server!

*Use database aliases*

Database aliases shield the client from physical database locations. Using aliases, a client can e.g. connect to frodo:zappa without having to know that the real location is frodo:/var/firebird/music/underground/mothers_of_invention.fdb. Aliases also allow you to relocate databases while the clients keep using their existing connection strings. Aliases are listed in the file aliases.conf, in this format on Windows machines:

```
poker = E:\Games\Data\PokerBase.fdb
```

```
blackjack.fdb = C:\Firebird\Databases\cardgames\blkjk_2.fdb
```

And on Linux:

```
books = /home/bookworm/database/books.fdb
zappa = /var/firebird/music/underground/mothers_of_invention.fdb
```

Giving the alias an .fdb (or any other) extension is fully optional. Of course if you do include it, you must also specify it when you use the alias to connect to the database. Aliases, once entered and saved, take effect immediately. There is no need to restart the server.

*Restrict database access*

The DatabaseAccess parameter in firebird.conf can be set to Restrict to limit access to explicitly listed file system trees, or even to None to allow access to aliased databases only. Default is All, i.e. no restrictions. Note that this is not the same thing as the file system-level access protection discussed earlier: when DatabaseAccess is anything other than All, the server will refuse to open any databases outside the defined scope even if it has sufficient rights on the database files.

*Choose your authentication model*

Firebird supports three authentication models when connecting to databases or using the tools:

1. *Native*: The user must identify him/herself with a Firebird user name and password, which the server checks against the security database.

2. *Trusted*: The user is automatically identified by his OS account name.

3. *Mixed*: The user either supplies a Firebird user name and password, or is logged in with his OS account name. On Linux, the mixed model is in effect. On Windows, native authentication is the default, but you can change it to trusted or mixed by setting the Authentication parameter in firebird.conf. Depending on your Windows system configuration and the way Firebird is used, *trusted* may be the most secure option.

*Consider whether Windows administrators should have SYSDBA rights*

In Firebird 2.1, if Authentication was *trusted* or *mixed*, Windows administrators would automatically receive SYSDBA privileges in all databases, including the security database. In Firebird 2.5, this is no longer the case. This reduces the risk that administrators with little or no Firebird knowledge mess up databases or user accounts. If you want to give individual administrators SYSDBA power in the security database and/or regular databases, you can grant them the RDB$ADMIN role as described in the section Appointing co-administrators. If, on the other hand, you want to restore the automatic SYSDBA mapping as it was in Firebird 2.1, read the following instructions. To give all the administrators automatic SYSDBA rights in the security database so they can manage Firebird user accounts, give the command:

gsec -user sysdba -pass masterkey -mapping set

You must do this as SYSDBA - a co-admin account won't do. To reverse the command, use -mapping drop. To give all the administrators SYSDBA rights in an ordinary database, log into the database as the owner, SYSDBA or someone who has the RDB$ADMIN role in that database, and issue the following SQL statement:

alter role rdb$admin set auto admin mapping

You must repeat this in every database where you want Windows administrators to have automatic SYSDBA rights. To turn the mapping off again, use DROP instead of SET. If automatic mapping is on, Windows administrators must not specify the RDB$ADMIN role when invoking gsec or connecting to a database – at least not if they want to make use of their SYSDBA rights. If they specify any role at all – even an unexisting one – the automatic mapping will not work.

There are more security parameters, but the ones not mentioned here are already set to an adequate protection level by default. You can read about them in the 1.5 through 2.5 *Release Notes* and in the comments in firebird.conf itself.

back to top of page

# Windows Control Panel applets

Several control panel applets are available for use with Firebird. Whilst such applets are not essential, they do provide a convenient way to start and stop the server and check its current status.

# Firebird Server Manager

The Firebird Server Manager applet is included in the Firebird distribution. The option to install this applet is only available for Superserver.

*Note*: The applet is also usable for (Super)Classic, provided that Firebird runs as a service, not as an application. Since the installation dialogue won't give you the option to include the applet with a Classic server, you must, if you really want it:

- install Superserver first;
- copy the applet Firebird2Control.cpl from the Windows system folder to a safe place;
- uninstall Superserver;
- install Classic;
- copy the applet back to the system directory.

This is a screenshot of the activated applet. Notice that the title bar says *Firebird Server Control*, although it is listed in the *Control Panel* as *Firebird Server Manager*.


This applet only works on Windows NT, 2000/3/8, XP, Vista and 7.

**Firebird Control Center**

For an alternative to the bundled applet, you can visit this webpage:

https://www.achim-kalwa.de/fbcc.phtml

...and download the Firebird Control Center (FBCC). Please note that, unlike the applet included with Firebird, the Firebird Control Center will not work with Classic or SuperClassic servers. This may change in the future.

The current version – 0.4.2 – should work well under Windows 2000 and up. It offers the same functionality as Firebird's own applet, and more. An older release, still downloadable at https://www.achim-kalwa.de/dl/fbcc-0.2.7.exe, also runs under Windows 9x, ME and NT. Notice however that these Windows versions are no longer actively supported by the Firebird project, even if the engine runs on it.

back to top of page

# Administration tools

The Firebird kit does not come with a GUI admin tool. It does have a set of command-line tools – executable programs which are located in the \bin subdirectory of your Firebird installation. One of them, gsec, has already been introduced to you.

The range of excellent GUI tools available for use with a Windows client machine is too numerous to describe here. A few GUI tools written in Borland Kylix, for use on Linux client machines, are also in various stages of completion.

Explore the Download > Tools > Administration page at https://www.ibphoenix.com for all of the options.

*Note*: Remember: you can use a Windows client to access a Linux server and vice-versa.