IBPhoenix Publications

Firebird Quick Start Guide

This section is an introduction for the complete newcomer to a few essentials for getting off to a quick start with a Firebird binary kit. For the fine details of configuring and running your server and tuning your installation, please refer to Chapters 4-6 of the *Using Firebird* manual, distributed on the IBPhoenix CD.

What is in the kit?

All of the kits contain all of the components needed to install the Firebird server:

- the Firebird server executable
- a client library located on the server machine
- the command-line tools
- the standard user-defined function libraries
- a sample database
- the C header files (not needed by beginners!)
- release notes—ESSENTIAL READING!

Default disk locations

The following table describes the default disk locations for the components on Windows and Linux. Information is provided in two version contexts:

- versions prior to Firebird 1.5
- versions from Firebird 1.5 onward

❖These differences are important. Versions prior to Firebird 1.5 use many of the same locations, component names and resource links as InterBase 6.x and prior versions of InterBase. Hence, it is not possible to run both a Firebird server and an InterBase server on the same machine with these versions.

In the major codebase revision starting at v.1.5, the old hard links to InterBase artifacts were removed and many of the major components were renamed. Firebird 1.5 may permit a running InterBase® server to be present. It is a certain objective for Firebird 2.

Table 1: Components of the Firebird installation

Platform	Component	File Name	Default Location
32-bit and 64-bit Windows	Firebird server	Pre v.1.5 ibserver.exe v.1.5 and onward fbserver.exe	c:\Program Files\Firebird\bin
	Command-line tools	gbak.exe, gfix.exe, gstat.exe, etc.	c:\Program Files\Firebird\bin
	Sample database	Employee.gdb	c:\Program Files\Firebird\examples
	User-defined function (UDF) libraries	ib_udf.dll & ib_utils.dll fbudf.dll	c:\Program Files\Firebird\udf
Windows NT, 2000, XP	Firebird client	Pre v.1.5 gds32.dll v.1.5 and onward gds32.dll, linking through to xxxxx.dll	c:\Winnt\System32
Windows 9x, ME	Firebird client	As above	c:\Windows

Table 1: Components of the Firebird installation

Platform	Component	File Name	Default Location
Linux and possibly other UNIX distros	Firebird server	Pre v.1.5 ibserver v.1.5 and onward fbserver	/opt/interbase/bin TBA
	Command-line tools	gbak, gfix, gstat, etc. Pre v.1.5 v.1.5 and onward	/opt/interbase/bin TBA
	Sample database	employee.gdb Pre v.1.5 v.1.5 and onward	/opt/interbase/bin TBA
	UDF libraries	Pre v.1.5 ib_udf.so v.1.5 and onward ib_udf.so fbudf.so	/opt/interbase/udf TBA
	Firebird client	Pre v.1.5 libgds.so.0 (binary) libgds.so (symlink) v.1.5 and onward TBA	/usr/lib

Installing Firebird

Installation drives

Firebird server—and any databases you create or connect to—must reside on a hard drive that is physically connected to the host machine. You cannot locate components of the server, or any database, on a mapped drive, a filesystem share or a network filesystem.

☆ You can mount a read-only database on a CD-ROM drive but you cannot run Firebird server from one.

Installation script or program

Although it is possible to install Firebird by a filesystem copying method—such as “untarring” a snapshot build file or decompressing a structured WinZip “.zip” file—it is strongly recommended that you use the distributed release kit the first time you install Firebird. The Windows executable installation script, the Linux rpm (RedHat Package Manager) program and the official tar.gz for other Posix platforms perform some essential setup tasks. Provided you follow the instructions correctly, there should be nothing for you to do upon completion but log in and go!

Windows platforms

On server platforms—Windows NT, 2000 and XP—the Firebird service will be running when the installation completes. Next time you boot up your server machine, it will be started automatically. To find out how to start and stop the server manually, see Running Firebird on Windows, p. 54 chapter 4 of *Using Firebird*.

The non-server Windows platforms—Windows 95, 98 and ME—do not support services. The installation will start Firebird server as an application, protected by another application known as the Guardian. If the server application should terminate abnormally for some reason, the Guardian will attempt to restart it.

☆ A copy of the client library—gds32.dll—will have been installed in your system directory.

Posix platforms

In all cases, read the release notes that pertain to the version of Firebird that you are going to install. There may be significant variations from release to release of any Posix operating system, especially the open source ones. Where possible, the build engineers for each Firebird version have attempted to document any known issues.

📖 If you do not find a copy of the release notes in your kit, go back to the Downloads page of the Firebird website at http://firebird.sourceforge.net and download a copy from there.

If you have a Linux distribution that supports **rpm** installs, consult the appropriate platform documentation for instructions about using RedHat Package Manager. In most distributions you will have the choice of performing the install from a command shell or through a GUI interface.

For Linux distributions that cannot process **rpm** programs, and for the various UNIX flavors, use the ".tar.gz" kit. You will find detailed instructions in the release notes.

Shell scripts have been provided. In some cases, the release notes may instruct you to modify the scripts and make some manual adjustments.

Testing your installation

If everything works as designed, the Firebird server process will be running on your server upon completion of the installation. It will start up automatically whenever you restart your server.

At this point, it is assumed that you will use the recommended TCP/IP protocol for your Firebird client/server network.

📖 For information about using NetBEUI protocol in an all-Windows environment, refer to Chapter 6, Network Configuration in the *Using Firebird* manual.

❖IPX/SX networks are not supported by Firebird

Pinging the server

Usually, the first thing you will want to do once installation is complete is ping the server. This just gives you a reality check to ensure that your client machine is able to see the host machine in your network. For example, if your server's IP address in the domain that is visible to your client is 192.13.14.1, go to a command shell and type the command

```
    ping 192.13.14.1
```

substituting this example IP address for the IP address that your server is broadcasting.

❖ If you get a timeout message, study the *Using Firebird* manual—Chapter 6, Network Configuration, and Chapter 7, Troubleshooting Connections—for further instructions.

Note that if you are connecting to the server from a local client—that is, a client running on the same machine as the server— you can ping the virtual TCP/IP loopback server:

```
ping localhost —or— ping 127.0.0.1
```

Checking that the Firebird server is running

After installation, Firebird server should be running as a service on Windows NT, 2000 or XP or on Linux.

Windows NT4, 2000 and XP

For the Windows server platforms, start the Services applet on the Control Panel.

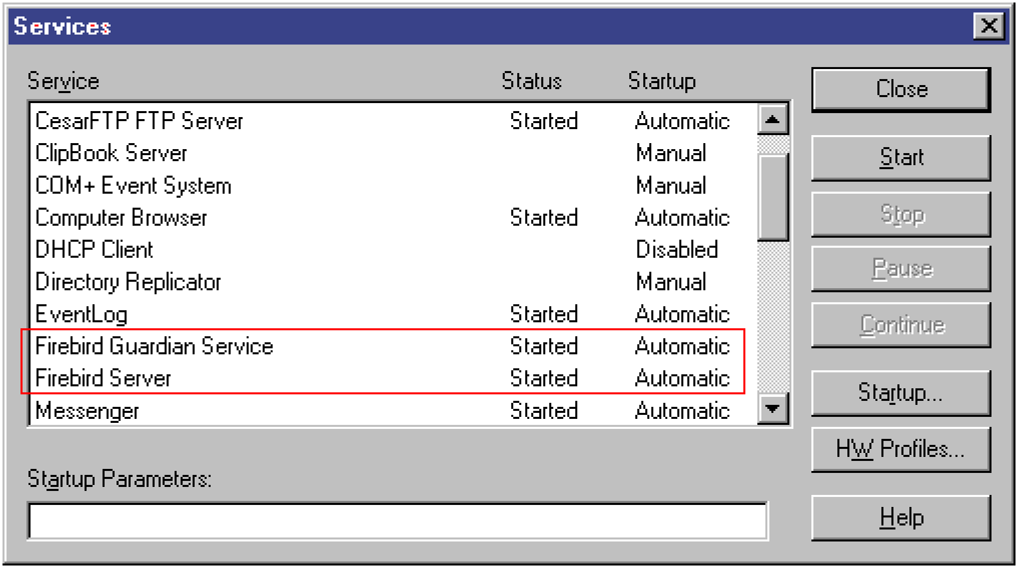
Firebird Server and The Guardian

This illustration shows the Services applet display on Windows NT 4. The appearance may vary from one Windows server edition to another.

<u>The Guardian</u>

If the Guardian is running (as shown in the screenshot, over) it may have a different service name because of version changes.

☆ On Windows 2000 and XP, the Guardian is a convenience rather than a necessity, since these two operating systems have the facility to watch and restart services. It is recommended that you keep Guardian active for other platforms if a SYSDBA is not available to restart the server manually in the event that it is stopped for some reason.

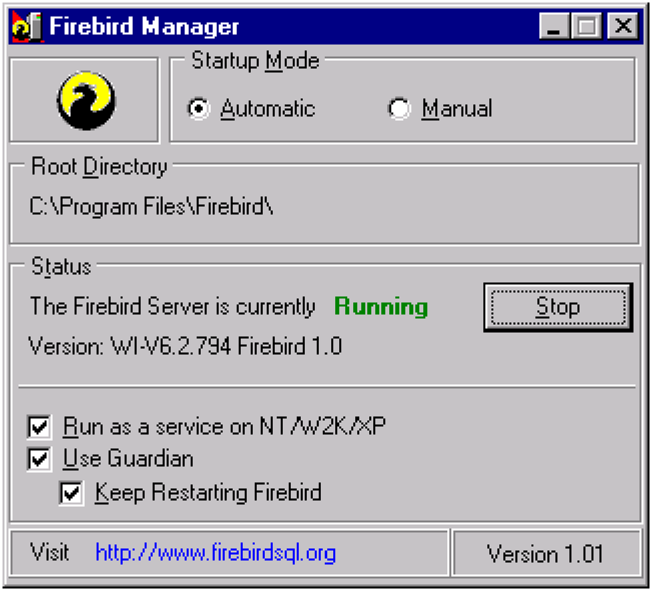


Service	Status	Startup
CesarFTP FTP Server	Started	Automatic
ClipBook Server		Manual
COM+ Event System		Manual
Computer Browser	Started	Automatic
DHCP Client		Disabled
Directory Replicator		Manual
EventLog	Started	Automatic
Firebird Guardian Service	Started	Automatic
Firebird Server	Started	Automatic
Messenger	Started	Automatic

Windows Control Panel applet

Firebird

Earlier distributions of Firebird for Windows kits installed the InterBase Manager applet into the Control Panel of operating systems that support services. It was omitted from Firebird 1.0 because a number of Firebird-specific ‘Firebird [Server] Manager’ implementations became available through open source channels. Whilst the applet is not essential, it does provide a convenient way to start and stop the service.



Startup Mode

● Automatic ○ Manual

Root Directory

C:\Program Files\Firebird\

Status

The Firebird Server is currently **Running**

Version: WI-V6.2.794 Firebird 1.0

☑ Run as a service on NT/W2K/XP

☑ Use Guardian

☑ Keep Restarting Firebird

Visit http://www.firebirdsql.org

Version 1.01

If you want a Firebird Manager applet and you do not find one installed in the Control Panel of your Windows server, download the Delphi source code from the addons branch of the Firebird CVS tree or a compiled version from www.ibphoenix.com.

The Control Panel applet will be similar in appearance to the illustration.

Windows 9x or ME

On Windows 9x or ME Firebird server should be running as an application, monitored by the Guardian. The Guardian's icon should appear in the tray with a green graphic. If the icon is flashing or showing as a red graphic, it indicates that Guardian is either attempting to start the server or has failed.

If you used an installation kit that installed but did not automatically start the Guardian and the Firebird server, you can set it up as follows:

1 Locate the executable file for the Guardian program (ibguard.exe) and create a shortcut for it in the Startup area of your machine's Start Menu.

2 Open the Properties dialog of the shortcut and go to the field where the command line is.

3 Edit the command line so it reads as follows:

```
ibguard.exe -a
```

4 Save and close the Properties dialog.

5 Double-click on the shortcut to start the Guardian. The Guardian will proceed to start ibserver.exe.

The Guardian should start up automatically next time you boot your Windows 9x or ME machine.

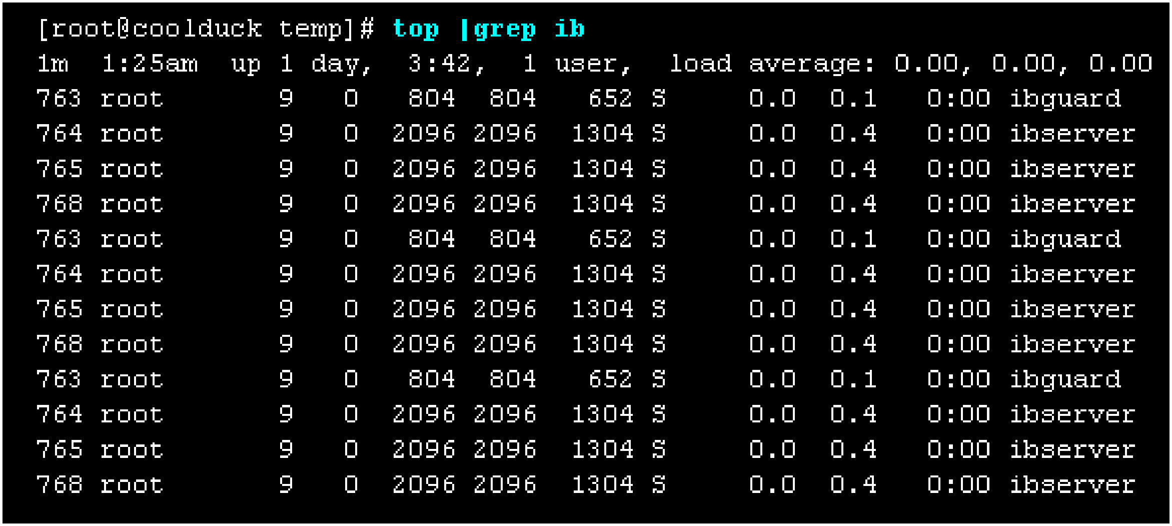
❖ There is currently no Control Panel applet which will detect the Firebird server application running on these non-server Windows versions.

Posix servers

Use the **top** command in a command shell to inspect the running processes interactively. If Firebird is running, you should see one process named **ibguard** (this is Guardian), one main and zero or more child processes named **ibserver** or **fbserver**—which one depends on the distribution of Firebird which you have installed.

The following screenshot shows the output of top, restricted by grep to show only processes with names starting with the characters ‘ib’, i.e.

```
top -A | grep ib
```



```
[root@coolduck temp]# top |grep ib
1m  1:25am  up 1 day,  3:42,  1 user,  load average: 0.00, 0.00, 0.00
763 root      9   0   804  804   652 S     0.0  0.1   0:00 ibguard
764 root      9   0  2096 2096  1304 S     0.0  0.4   0:00 ibserver
765 root      9   0  2096 2096  1304 S     0.0  0.4   0:00 ibserver
768 root      9   0  2096 2096  1304 S     0.0  0.4   0:00 ibserver
763 root      9   0   804  804   652 S     0.0  0.1   0:00 ibguard
764 root      9   0  2096 2096  1304 S     0.0  0.4   0:00 ibserver
765 root      9   0  2096 2096  1304 S     0.0  0.4   0:00 ibserver
768 root      9   0  2096 2096  1304 S     0.0  0.4   0:00 ibserver
763 root      9   0   804  804   652 S     0.0  0.1   0:00 ibguard
764 root      9   0  2096 2096  1304 S     0.0  0.4   0:00 ibserver
765 root      9   0  2096 2096  1304 S     0.0  0.4   0:00 ibserver
768 root      9   0  2096 2096  1304 S     0.0  0.4   0:00 ibserver
```

☆ If your Firebird server uses the Classic architecture, the process name is **gds_inet_server**. There will be one instance of this process running for each connection. Use `top -A | grep gds` to look at them.

Other things you need

A network address for the server

- If you are on a managed network, get the server's IP address from your system administrator

- If you have a simple network of two machines linked by a crossover cable, you can set up your server with any IP address you like except 127.0.0.1 (which is reserved for a local loopback server) and, of course, the IP address which you are using for your client machine. If you know the "native" IP addresses of your network cards, and they are different, you can simply use those

- If you are intending to try out a single-machine installation of both client and server, you should use the local loopback server address—localhost, with the IP address 127.0.0.1

> NOTE On Windows, it is possible to connect locally to the server, without using a local TCP/IP loopback. This is not a TCP/IP connection and it is not a thread-safe way to connect to a local server. For using single instances of the command-line tools (**gsec**, **gbak**, etc.) it works just fine.

Default user name and password

The SYSDBA user has all privileges on the server. The installation program will install the SYSDBA user with the password ***masterkey***.

☆ Actually, it is *masterke*, since all characters after the eighth one are ignored.

If your server is exposed to the Internet *at all*, you should change this password immediately using the **gsec** command-line utility.

How to change the SYSDBA password

You need to be logged into the operating system as Superuser (root on Linux) to run **gsec**. Let's say you decide to change the SYSDBA password to ***icuryy4me***

1 Go to a command shell on your server and change to the directory where the command-line utilities are located. Refer to Table 1 (above) to find this location.

2 Type the following (it is case-sensitive on all platforms except Windows):

```
gsec -user sysdba -password masterkey
```

You should now see the shell prompt for the gsec utility:

```
GSEC>
```

3 Type this command

```
GSEC> modify sysdba -pw icuryy4me
```

4 Press ENTER. The new password `icuryy4me` is now encrypted and saved and *masterkey* is no longer valid.

5 5. Now quit the **gsec** shell:

```
GSEC> quit
```

☆ Because Firebird ignores all characters in a password past the eighth character, `icuryy4m` will work, as will `icuryy4monkeys.`

An Admin tool

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 directory of your Firebird installation.

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.

Inspect the Downloads > Contributed > Admin Tools page at http://www.ibphoenix.com for all of the options.

☆ You can use a Windows client to access a Linux server and vice-versa.

Connecting to the sample database

In the ./examples directory of your Firebird installation is a sample database named *employee.gdb*. You can use this database to “try your wings”.

❖ If you are running the server on Windows XP or ME, be sure to rename this test database to employee.**fdb** to get around the System Restore utility, which targets files with a .gdb extension. There is more information about this problem in the release notes.

Server name and path

If you move the sample database, be sure you move it to a hard disk that is physically attached to your server machine. Shares, mapped drives or (on Unix) mounted SMB (Samba) filesystems will not work. The same rule applies to any databases that you create.

There are two elements to a TCP/IP connection string: the server name and the disk/filesystem path. Its format is as follows:

For a Linux server: `servername:/filesystem-path/database-file`
Example on a Linux or other Posix server named serverxyz:

```
    serverxyz:/opt/interbase/examples/employee.gdb
```

For a Windows server:`servername:D:\filesystem-path\database-file`

```
    serverxyz:C:\Program Files\Firebird\examples\employee.gdb
```

The CONNECT statement

Connecting to a Firebird database always requires the user to “log in” using a user name and a valid password—widely referred to as a *log-in* or *login*. Any user other than SYSDBA or root (on Posix systems only) needs also to have permissions to objects inside a database. For simplicity here, we will look at logging in as SYSDBA using the password *masterkey*.

Using isql

There are several different ways to connect to a database using isql. One way is to start **isql** in its interactive shell:
Go to the ./bin directory of your Firebird installation and, at that prompt, type the command **isql**

```
    C:\Program Files\Firebird\bin>isql Enter
    Use CONNECT or CREATE DATABASE to specify a database
    SQL>CONNECT "C:\Program Files\Firebird\examples\Employee.gdb" user
    'SYSDBA' password 'masterkey'; Enter
```

- ❖In isql, make sure that every statement ends with a semicolon. If you forget, the prompt will change from `SQL>` to `CON>` and you won't be able to continue until you type in the semicolon and press Enter again.
- ❖Although single quote symbols are the "norm" for delimiting strings in Firebird, double quote symbols were used with the database path string in the above example. This is sometimes necessary with some of the command-line utilities where the path string contains spaces. Single quotes should work for paths that do not contain spaces.

At this point, isql will inform you that you are connected:

```
DATABASE "C:\Program Files\Firebird\examples\Employee.gdb", User: sysdba
SQL>
```

You can now continue to play about with the employee.gdb database. The characters **isql** stand for *interactive SQL [utility]*. You can use it for querying data, getting information about the metadata, creating database objects, running data definition scripts and much more.

To get back to the command prompt type

```
SQL> QUIT;
```

For more information about **isql**, see chapter 10, Interactive SQL Utility (isql) (p. 152 of *Using Firebird*).

Using a GUI client

GUI client tools usually take charge of composing the CONNECT string for you, using server, path, user name and password information that you type into prompting fields. Use the elements as described in the preceding topic.

☆

- It is quite common for such tools to expect the entire server + path as a single string
- Remember that file names and commands on Linux and other Posix command shells are case-sensitive

Creating a database using isql

There is more than one way to create a database using **isql**. Here, we will look at one simple way to create a database interactively—although, for your serious database definition work, you should create and maintain your metadata objects using data definition scripts. There is a complete chapter in the *Using Firebird* manual discussing this topic.

Starting isql

To create a database interactively using the isql command shell, you need to be working on the server. Get to a command prompt in Firebird's ./bin directory and start isql as follows:

```
C:\Program Files\Firebird\bin>isql Enter
Use CONNECT or CREATE DATABASE to specify a database
```

The CREATE DATABASE statement

Now, you can create your new database interactively. Let's suppose that you want to create a database named test.fdb and store it a directory named "data" on your D drive:

```
SQL> CREATE DATABASE 'D:\data\test.fdb' page_size 8192 user 'SYSDBA'
password 'masterkey'; Enter
```

The database will be created and, after a few moments, the SQL prompt will reappear. You are now connected to the new database and can proceed to create some test objects in it.

To verify that there really is a database there, type in this query:

```
SQL> SELECT * FROM RDB$RELATIONS; Enter
```

The screen will fill up with a large amount of data! This query selects all of the rows in the system table where Firebird stores the metadata for tables. An "empty" database is not empty—it contains a database which will become populated with metadata as you begin creating objects in your database.

To get back to the command prompt type

```
    SQL> QUIT;
```

For more information about **isql**, see chapter 10, Interactive SQL Utility (isql) (p. 152 of *Using Firebird*).

Performing a client-only install

Each remote client machine needs to have the client library—libgds.so on Posix clients, gds32.dll on Windows clients—that matches the release version of the Firebird server.

Firebird versions from 1.5 onward require an additional client library named libfb.so or fb32.dll which contains the full library. In these newer distributions, the “gds” named files are distributed to maintain compatibility with third-party products which need these files. Internally, the libraries will “jump” to the correct access points in the renamed libraries.

Some extra pieces are also needed for the client-only install.

Windows

At the time of publication, no compact installation program was available to assist with installing the client pieces on a Windows client. If you are in the common situation of running Windows clients to a Linux or other Posix Firebird server, you need to download the full Windows installation kit that corresponds to the version of Firebird server you install on your Linux or other server machine.

Fortunately, once you have the kit, the Windows client-only install is easy to do. Run the installation program, just as though you were going to install the server—but select the CLIENT ONLY option from the install menu.

Linux and some other Posix clients

A small-footprint client install program for Linux clients is not available either. Additionally, some Posix flavors—even within the Linux constellation—have somewhat idosyncratic requirements for filesystem locations. For these reasons, not all *x distributions for Firebird even contain a client-only install option.

For most Linux flavors, the following procedure is suggested for Firebird versions lower than 1.5. Log in as root for this.

1. Look for libgds.so.0 in /opt/interbase/lib on the server where Firebird server is installed. Copy it to /usr/lib on the client.
2. Create the symlink libgds.so for it, using the following command:

   ```
   ln -s /usr/lib/libgds.so.0 /usr/lib/libgds.so
   ```

3. Copy the interbase.msg file to /opt/interbase
4. In system-wide default shell profile, or using **setenv()** from a shell, create the INTERBASE environment variable and point it to /opt/interbase, to enable the API routines to locate the messages.

Other things worth knowing

This chapter finishes with a small, eclectic collection of things other Firebird ‘newbies’ wished they had known about when they began.

Firebird SQL

Every database management system has its own idiosyncrasies in the ways it implements SQL. Firebird adheres to the SQL standard more rigorously than any other RDBMS except possibly its “cousin”, InterBase®. Developers migrating from products that are less standards-compliant often wrongly suppose that Firebird is quirky, whereas many of its apparent quirks are not quirky at all.

String delimiter symbol

Strings in Firebird are delimited by a pair of single quote symbols—`'I am a string'`—(ASCII code 39 **NOT** 96). If you used earlierversions of Firebird’s relative, InterBase®, you might recall that double and single quotes were interchangeable as string delimiters. Double quotes cannot be used as string delimiters in Firebird.

Double-quoted identifiers

Before the SQL-92 standard, it was not legal to have object names (identifiers) in a database that duplicated keywords in the language, were case-sensitive or contained spaces. SQL-92 introduced a single new standard to make any of them legal, provided that the identifiers were defined within pairs of double-quote symbols (ASCII 34) and were always referred to using double-quote delimiters.

The purpose of this “gift” was to make it easier to migrate metadata from non-standard RDBMSs to standards-compliant ones. The down-side is that, if you choose to define an identifier in double quotes, its case-sensitivity and the enforced double-quoting will remain mandatory.

Firebird does permit a slight relaxation under a very limited set of conditions: if the identifier which was defined in double-quotes 1) was defined as all upper-case, 2) is not a keyword and 3) does not contain any spaces, then it can be used in SQL unquoted as long as it is used in all upper-case.

Unless you have a compelling reason to define quoted identifiers, it is usually recommended that you avoid them. Firebird happily accepts a mix of quoted and unquoted identifiers—so there is no problem including that keyword which you inherited from a legacy database, if you need to.

❖ Some database admin tools enforce double-quoting of ALL identifiers by default. Try to choose a tool which makes double-quoting optional.

Apostrophes in strings

If you need to use an apostrophe inside a Firebird string, you can “escape” the apostrophe character by preceding it with another apostrophe.

For example, this string will give an error:

```
'Joe's Emporium'
```

because the parser encounters the apostrophe and interprets the string as ‘Joe’ followed by some unknown keywords.

To make this a legal string, double the apostrophe character:

```
'Joe''s Emporium'
```

Notice that this is TWO single quotes, not one double-quote.

Concatenation of strings

The concatenation symbol in SQL is two ‘pipe’ symbols (ASCII 124, in a pair with no space between). In SQL, the ‘+’ symbol is an arithmetic operator and it will cause an error if you attempt to use it for concatenating strings. The following expression prefixes a character column value with the characters ‘Reported by:’:

```
    'Reported by: ' || LastName
```

Take care with concatenations. Be aware that Firebird will raise an error if your expression attempts to concatenate two or more char or varchar columns whose potential combined lengths would exceed the maximum length limit for a char or a varchar (32 Kb).

See also the note below, Expressions involving NULL, about concatenating in expressions involving NULL.

Division of an integer by an integer

Firebird accords with the SQL standard by truncating the result (quotient) of an integer/integer calculation to the next lower integer. This can have bizarre results unless you are aware of it.

For example, this calculation is correct in SQL:

```
    1 / 3 = 0
```

If you are upgrading from a RDBMS which resolves integer/integer division to a float quotient, you will need to alter any affected expressions to use a float or scaled numeric type for either dividend, divisor, or both.

For example, the calculation above could be modified thus in order to produce a non-zero result:

```
    1.000 / 3 = 0.333
```

Expressions involving NULL

In SQL, NULL is not a value. It is a condition, or *state*, of a data item, in which its value is *unknown*. Because it is unknown, null cannot behave like a value. When you try to perform arithmetic on NULL, or involve it with values in other expressions, the result of the operation will always be NULL. It is not zero or blank or an “empty string” and it does not behave like any of these values.

So—here are some examples of the types of surprises you will get if you try to perform calculations and comparisons with NULL:

```
1 + 2 + 3 + NULL = NULL
if (a = b) then
MyVariable = 'Equal'
else
MyVariable = 'Not equal';
```

will return ‘Not equal’ if both a and b are null.

```
if (a <> b) then
MyVariable = 'Not equal'
else
MyVariable = 'Equal';
```

will also return ‘Not equal’ if both a and b are null.

```
FirstName ||' ' || LastName
```

will return NULL if either FirstName or LastName is NULL.

Backup

Firebird comes with its own utility for backing up and restoring your databases. Its name is **gbak** and it can be found in the ./bin directory of your Firebird installation. Firebird databases can be backed up whilst users are connected to the system and going about their normal work. The backup will be taken from a snapshot of the database state at the time the backup began.

Regular backups and occasional restores using **gbak** should be a scheduled part of your database management activity.

❖ Do not use external proprietary backup utilities or file-copying tools such as WinZip, tar, copy, xcopy, etc., on a database which is running. Not only will the backup be unreliable, but the disk-level blocking used by these tools can corrupt a running database.

💣 Study the warnings in the next section about database activity during restores!

How to corrupt a database

1. Modifying metadata tables yourself

Firebird stores and maintains all of the metadata for its own and your user-defined objects in—a Firebird database!   More precisely, it stores them in relations (tables) right in the database itself. The identifiers for the system tables, their columns and several other types of system objects begin with the characters 'RDB$'.

Because these are ordinary database objects, they can be queried and manipulated just like your user-defined objects. However, just because you ***can*** does not say you ***should***. The Firebird engine implements a high-level subset of SQL (DDL) for the purpose of defining and operating on metadata objects, typically through CREATE, ALTER and DROP statements.

It cannot be recommended too strongly that you use DDL—not direct SQL operations on the system tables—whenever you need to alter or remove metadata. Defer the "hot fix" stuff until your skills in SQL and your knowledge of the Firebird engine become very advanced. A wrecked database is neither pretty to behold nor cheap to repair.

2. Disabling forced writes on Windows

Firebird is installed with forced writes (synchronous writes) enabled by default. Changed and new data are written to disk immediately upon posting.

It is possible to configure a database to use asynchronous data writes—whereby modified or new data are held in the memory cache for periodic flushing to disk by the operating system's I/O subsystem. The common term for this configuration is *forced writes off* (or *disabled*). It is sometimes resorted to in order to improve performance during large batch operations.

The big warning here is—**do not disable forced writes on a Windows server**. It has been observed that the Windows server platforms do not flush the write cache until the Firebird service is shut down. Apart from power interruptions, there is just too much that can go wrong on a Windows server. If it should hang, the I/O system goes out of reach and your users' work will be lost in the process of rebooting.

☆ Windows 9x and ME do not support deferred data writes

Disabling Forced Writes on a Linux server

Linux servers are safer for running an operation with forced writes disabled temporarily. Do not leave it disabled once your large batch task is completed, unless you have a very robust fall-back power system.

3. Restoring a backup to a running database

One of the restore options in the **gbak** utility `(gbak -r[estore])` allows you to restore a gbak file over the top of an existing database. It is possible for this style of restore to proceed without warning while users are logged in to the database. Database corruption is almost certain to be the result.

❖ Be aware that you will need to design your Admin tools and procedures to prevent any possibility for any user (including SYSDBA) to restore to your active database if any users are logged in.

📖 For **gbak** instructions see chapter 21, Database Backup and Restore (p. 390 of *Using Firebird*).

For instructions about blocking access to users, see chapter 14, Getting exclusive access to a database (p. 283 of *Using Firebird*).

If is practicable to do so, it is recommended to restore to spare disk space using the `gbak -c[reate]` option and *test the restored database* using isql or your preferred admin tool. If the restored database is good, shut down the server. Make a filesystem copy of the old database and then copy the restored database file (or files) over their existing counterparts.

4. Allowing users to log in during a restore

If you do not block access to users while performing a restore using gbak -r[estore] then users may be able to log in and attempt to do operations on data. Corrupted structures will result.

Where to next?

How to get help

The community of willing helpers around Firebird goes a long way back, to many years before the source code for its ancestor, InterBase® 6, was made open source. Collectively, the Firebird community does have all the answers! It even includes some people who have been involved with it since it was a design on a drawing board in a bathroom in Boston.

- Visit the official Firebird Project site at http://firebird.sourceforge.net and join the user support lists.
- Visit the Firebird knowledge site at http://www.ibphoenix.com to look up a vast collection of information about developing with and using Firebird.
- Get the ***Using Firebird*** manual and its companion volume, ***The Firebird Reference Guide***. Both books ship on the IBPhoenix CD as e-books in a format similar to this one. They are fully cross-referenced.
- Read the *Firebird Reference Guide*— Resources and References (ch. 10 p. 399)— for a collection of useful resources about Firebird, SQL and database application development.

Using the books by IBPhoenix Publications

Using Firebird and *The Firebird Reference Guide* have been designed for easy use and access during your development work. A button at the top right-hand corner of each “content” page will cause Acrobat Reader to switch back and forth between the two volumes. Each content page also has a navigation bar with buttons to take you directly to the index listings for the selected character. All index listings are hyperlinked to their sources. If the reader is set to display at 125% resolution, one complete page and the hypertext side menu will be visible.

For greater detail about setting up your server and your network, refer to the early chapters of *Using Firebird*. Chapter 7 is a troubleshooting reference. The ensuing chapters deal in turn with design, language and development issues and provided detailed instructions for using the command-line tools.

The Firebird Project

The developers, designers and testers who gave you Firebird and several of the drivers are members of the Firebird open source project at Sourceforge, that amazing virtual community that is home to thousands of open source software teams. The Firebird project's address there is http://sourceforge.net/projects/firebird. At that site are the source code tree, the bug tracker and a number of technical files which can be downloaded for various purposes related to the development and testing of the codebases.

The Firebird Project developers and testers use an email list forum—firebird-devel@lists.sourceforge.net—as their "virtual laboratory" for communicating with one another about their work on enhancements, bug-fixing and producing new versions of Firebird.

Anyone who is interested in watching progress can join this forum. However, user support questions are a distraction which they do not welcome. Please do not try to post your user support questions there!

Alphabetical Index