

What is Happening with Firebird?

by Paul Reeves



InterBase was released as open source software over a year ago. It should have been a great occasion, but it was surrounded with controversy - that part of the story has been told before. However, in the last fifteen months work has been going on to create a database server out of the ashes of InterBase - Firebird!

Firebird is based on the original InterBase source code that was released back in July 2000. For the most part applications that are written to work against an InterBase database server will work against a Firebird server. Currently the incompatibilities between Firebird and InterBase are minor and I'll list them later in the article.

Given that Firebird (Fb) and InterBase (IB) are almost the same it is fair to ask 'What are the differences?' and 'Why use Firebird?'. Let's take a look at each of these issues in turn.

What is the difference?

Perhaps the main and most important difference is that Firebird is a true Open Source Software project. The Firebird project is open to anyone to join. Discussion about development is open to all - members and non-members. Fixes and enhancements to the code can be contributed easily. Binaries are available for numerous platforms and are constantly being updated. Much work has been done to enable anyone to build a working server from the code.

By comparison, the Borland code base is still virtually unbuildable outside of Scott's Valley. Neither is there any real community based around their source tree so, even if you wanted to work on the InterBase code, you won't find many to talk to about it. Borland have publicly committed to keeping new features secret - you have to pay for the 'server' edition and the code for it is unlikely to be released. Indeed, we are now faced with the scenario of Borland having forked their own code base. At the level of transparency, community support and clear understanding of where each project is headed, Firebird wins hands down.

Why use Firebird?

Probably not because it is free, although that is a bonus. Setting up any software project and (often) maintaining it long after its intended life cycle means that the cost of the database is only a minor component. Firebird brings all the benefits of open source software. You have the code. You can build your own binaries. There is no pressure to upgrade. If bug fixes come out, you can easily see what has been fixed and how. In short, there is no longer a black box that works mysteriously to subvert the intentions of your application.

Firebird also continues the long InterBase tradition of a single code-base providing cross-platform support. There are currently five server platforms that are well supported by active project leaders: Win32, Linux, Solaris (Sparc and X86), and Mac OS X. In addition, there are another five platforms that are less actively supported: FreeBSD,

NetBSD, AIX, HP-UX and SCO. InterBase 6, on the other hand is only available on three platforms - Win32, Linux and Solaris. Platform availability aside, perhaps the real reason for using Firebird is because it is quite clearly under constant development. Let's look at what has been happening in more detail.

What is in new in Firebird 1.0?

Fb 1.0 is more a 'line in the sand' than a major software release. You could almost say it is just InterBase 6.0 with some minor enhancements and lots of bug fixes. In some ways its main feature is that it represents the work of a true, independent, Open Source (OS) software project. More importantly it represents the work of a development team that could hardly build the code, let alone fix bugs or make enhancements when it started fifteen months ago. Firebird 1.0 shows just how far that team has come. So, what has that team been doing in the last year? Here are some of the highlights:

- **Select First M skip N**
This has been much requested by developers using Firebird as a back-end to a web-server.
Very flexible. It allows you to select the first M records, or skip a bunch of N records and select the next M records.
- **Recreate table**
Drops a table and creates it in a single statement.
- **16k page sizes**
This allows for bigger databases, as well as improved performance tuning.
Using this feature will of course make the database unreadable by an InterBase Server.
- **64-bit file support**
With the sudden increase in hard drive sizes over the last few years, the 2Gb/4Gb file size limit that 32-bit file pointers imposed has led to quite a bit of database corruption.
This limit has now been removed from Firebird. However, it does require support by the host Operating system.
- **Descriptors in UDFs**
These use existing internal mechanisms to pass data structures from the engine to the User Defined function (UDF). The most dramatic impact of this is that NULLs can now be passed to and from a UDF.
There is also a new UDF library available with Firebird that uses this feature.
- **Convert strings to blobs in insert/update statements**
This is a simple but effective feature.

- **Forced Writes are ON by default on Win32**
It was never documented, but Forced Writes were turned off for Win32 for the release of InterBase 6.0. This led to a lot of corrupt databases, especially in environments where InterBase was running on Win9x or without a UPS.

There is also a host of minor new features

- Port 3050 is now the default, thus making installation easier by removing the need to have an entry in the clients SERVICES file.
- ‘—’ can be used for single line comments in SQL scripts.
- Group By can now use UDFs
- There is a new BREAK keyword for use in triggers and stored procedures
- A Drop Generator command has been added
- An internal substring function has been put into the engine.

The list goes on and, to go through everything here, would be to risk boring you. There are, of course, release notes that fully document these and many other new features.

Bug fixes

Bug fixes are legion. The speed with which bugs are found and fixed is, without a doubt, one of the major benefits of Open Source Software. Again, the Fb 1.0 release notes document the main fixes.

Overall it is difficult to pick out the bug fixes that will matter to you. For the most part they are quite obscure, as is usually the case with database bug fixes.

For instance, it was discovered that ‘DROP VIEW mytable’ would allow a table to be dropped if ‘mytable’ was indeed a table. Canny developers, forged by years of defensive programming tend to know not to do that sort of thing, so most just won’t have come across the problem.

Likewise it has been discovered that Dialect 3 databases would allow object identifiers of ‘’. Obviously an error, but not one that most of us are likely to stumble across.

Using Generators in computed fields was found to return wrong results and even corrupt the database. Most people don’t use generators in such a way, so they have never noticed. But that isn’t the point. Someone found the problem and either they or someone else was able to look at the code and fix it - often with just a change to one or two lines. With the result that anyone else stumbling across the problem can immediately benefit.

What next?

Work has started on Firebird 2.0. The main change, so far, has been to move the code from C to C++. This step in itself may not appear much, but is fundamental to cleaning up the code ready for the major enhancements that are planned for the future.

The build process is also a lot cleaner, conforming to host platform standards where possible.

Developers have largely been holding off from more

ambitious work on version two while they finalize getting version one out. As with any open source project, it is difficult to plan precisely what will feature in an upcoming release. This really does depend on what comes along.

Two important areas that will see improvements are Performance and Security. For other feature requests that are under consideration, please see <http://firebird.sourceforge.net/download/OpenFeaturesDetailed.PDF>. If you have features you would like suggest for the next version, I would recommend that you read that first.

While the focus of development will shift to Fb 2.0, it is anticipated that bug fixes and some minor enhancements will be made to the Fb 1.0 code base, so it is anticipated that there will be an Fb 1.n release. After all, a formal release of Fb 2.0 is likely to be a year away.

How can I get Firebird 1.0?

Firebird is, of course, available for download from both the Firebird and IBPhoenix sites (see below) in binary and source formats. The binaries come with installation routines for most platforms. [We will also be including at least the Windows binaries on DIL. Tech Ed]

If you want to make a financial contribution to help further the development of Firebird, you can purchase a CD from IBPhoenix. Not only do you get all the binaries and source code in one handy place, you also get stacks of documentation, a searchable knowledge base, all sorts of drivers, a third-party tools section and much more. Pricing is geared towards enabling IBPhoenix to invest in further Firebird development. There are several Firebird developers who would like to spend more time on development and a chunk of the profits will be used to pay them to do just that.

Useful addresses:

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

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



Paul Reeves is currently looking after European Technical Support for IBPhoenix, so he has to declare an interest in the success of Firebird. When not doing support he is one of the many working to make Firebird a better InterBase. He can be contacted via email on: preeves@ibphoenix.com.