

Introduction

Firebird is a full-featured SQL database engine. It is powerful, yet lightweight, with minimal configuration and administration needs. It's easily scalable from single-user, single-database usage up to company-wide enterprise purposes. A single Firebird server can handle multiple independent databases, each with multiple client connections. And best of all: it's true *open source*, thus free of any license fees even for commercial use.

Benefits of using Firebird

Why would you want to use Firebird? There are five key reasons:

- **Capable.** Unlike most other open source database systems, Firebird is full featured. It is fast. It is reliable. It scales. What more could you wish for?
- **Compact.** A minimal install of Firebird only takes up about 3MB of disk space and can run in as little as 32MB of memory. This makes Firebird the smallest of the full featured database engines by far: the "Express" editions of Oracle, DB2 and SQLServer typically take up more than 100MB and offer only restricted functionality.
- **Easy.** Firebird is easy. It is easy to develop with, it is easy to deploy, it is easy to manage and administrate. A demo install at an OSDC conference in Melbourne took 53 seconds from hitting Enter to start the installer to opening a database using a command-line utility, stopping en route to change the SYSDBA password.
- **Free.** Firebird is free. You can download it at no cost and use it. You can redistribute it with your application, without having to pay license fees or having to open source the entire application. You get the full source code, so you can change it should you want to.
- **Community.** The strength of Open Source software lies in the community. Firebird has a large and vibrant community, ensuring that the code base will continue to be developed. The Firebird lists are renowned for their fast and helpful responses. Commercial support is available from multiple sources for those users that need it.

There is no other database system on earth that shares all the above five qualities with Firebird.

Connectivity

A database system is no use unless applications can connect with it. Firebird connects with clients through all popular technologies and through quite a few less well known technologies as well.



C/C++

The Firebird native client library offers C/C++ programmers full access to the database server and to the security features. It is a easy to deploy library: you can simply pack it with your applications and it will work (on Windows no changes to the Registry are needed). For C++ development, the IBPP wrapper offers an object-oriented interface to Firebird.



ODBC

Like it or not, ODBC is still a workhorse of database system connectivity. Many clients written in Visual Basic or MS Access use it to connect. The Firebird ODBC driver has full support for the ODBC 3.0 specification and is available for both Windows and Linux. The driver works flawlessly with all releases of Microsoft Office, including Access.



Java

Java clients use the JDBC standard to connect to database servers. The Firebird JDBC driver – more commonly known as 'JayBird' – has full support for the JDBC3 standard and plans support for the recently announced JDBC4 specification. Jaybird can operate both as a type 4 and as a type 2 driver and works with all popular makes of virtual machine (Sun, IBM, BEA, etc.). JayBird is fully compatible with all major J2EE frameworks, such as JBoss, WebLogic and WebSphere.



dotNet

Microsoft's dotNet uses the ADO.Net standard for connecting to database servers. Firebird was among the first non-Microsoft databases to support dotNet and as a result the driver is mature and full featured. The Firebird ADO.Net driver has support for both dotNet 1.1 and dotNet 2.0, and also supports the Linux equivalent of dotNet, Mono 1.1 and Mono 2.0.



Delphi / Lazarus

Firebird shares a history together with Delphi, having been 'room mates' at Borland for about a decade. From its launch till today, Delphi has been bundled with Interbase and Firebird is at the hart of millions of Delphi-based software installations. There is large pool of database connectivity frameworks for Delphi, all of which support Firebird. A 2004 survey by Evans showed that Firebird was more popular than MSDE (SQLServer Express) for the 'edge applications' that people write in Delphi or VB. The open source Lazarus system also supports Firebird, through the "sqldb" driver.



PHP

With some 1 million developers using it and driving about 20 million websites, PHP has taken on a prominent role in connecting data to the web. Firebird has native drivers for both PHP4 and PHP5. Firebird also supports the new DAO data access layer.

Next to the above technologies, Firebird supports connectivity to a host of other languages and systems, such Perl, Python and Ruby. The driver for Python is called Kinterbasdb.

Downloading Firebird

Starting with Firebird is easy, you just have to download it from

<http://www.firebirdsql.org/index.php?op=files&id=engine>

Don't forget to read the Firebird 1.5 Quick Start Guide at <http://www.firebirdsql.org/index.php?op=devel&sub=doc> to help you through your first steps.

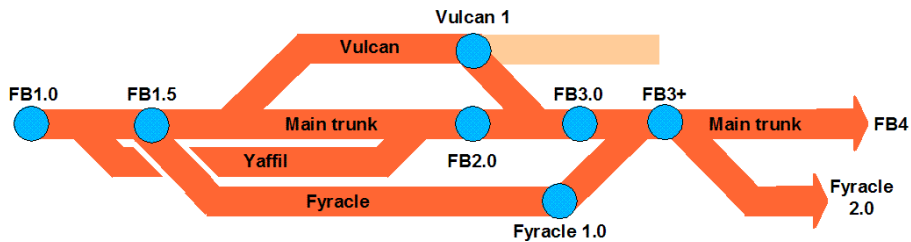
You can also try Firebird with a real demo application, "The DVD Store". The DVD Store application is a complete online e-commerce test application, with a backend database component, a web application layer, and driver programs. All instructions and explanation are here : <http://www.ibexpert.info/documentation/1.%20Database%20technology/Transaction%20Processing%20Performance%20with%20IBEBenchmarks/124344.html>

Another fast way to test Firebird is to setup Phoenix PHPServer (a four click setup of MyServer, PHP and the Firebird client) from <http://www.fyracle.org/phpserver.html> and to continue to play testing Bitweaver an application framework for content management written in PHP (<http://www.bitweaver.org>).

Roadmap

Firebird development is moving at a brisk pace, often in several concurrent branches. Firebird 1 was released in 2002 and fixed countless bugs in the InterBase 6 Open Edition code base. The Yaffil branch added features and improved embedded usage. Firebird 1.5 was released in 2004 and moved the code base to C++, laying the ground work for major work. The Fyracle branch added features, notably support for stored procedures written in Java or dotNet and PL/SQL.

At the time of writing (March 2006), Firebird 2 was about to be released. Firebird 2 removes many limits that existed in the 1.5 code base and makes major enhancements to the optimizer,



The next release will be Firebird Vulcan, which moves the code base to a fine grained locking model, which allows for near linear scaling on SMP hardware. Firebird Vulcan is planned for the first half of 2006.

The roadmap continues with Firebird 3 and Firebird 3+, which will consolidate all advances into a the main development branch.

SQL

SQL standards. Firebird implements most of the core ANSI SQL-99 standard with several extensions from the SQL200x standard. An overview chart showing SQL200x compliance is available on the project's main web site. On top of this, Firebird has several unique additional features which are not covered by the standard.

Data types. Firebird offers the full range of common datatypes and has a full implementation of data domains.

BLOB's. Firebird implements typed binary large objects ('blobs'). Text blobs, a.k.a. "clobs", can be used directly as character data in many cases. The engine supports "blob filters" for converting between compatible typed blobs and includes several of its own. Blob filters can be user-defined so that, for example, the database programmer can make an XML-to-RTF filter available to applications.

Internationalization. Firebird is fully internationalized. In fact, most of its user base is outside of the English speaking world, with Brazil, Europe, Russia and Japan being important markets / having important communities.

Oracle mode. Firebird has a third party extension, called "Fyrracle". This extension adds several features to the core engine and offers an oracle-compatibility mode:

- SQL syntax
- PL/SQL stored procedures and triggers
- Packages

Transaction processing

ACID. Firebird is fully ACID compliant. All ANSI isolation levels except Dirty Read are supported. Dirty Read isolation violates ACID principles and is not supported by Firebird.

Locking. In 1984 Firebird ancestor was the world's first database system to use multi-version concurrency control, a technology that avoids the use of user locks to achieve multi-user concurrency. Locking policy is optimistic and at row-level. Except in extreme transaction configurations, readers and writers do not block each other. This technology has since been imitated by Oracle in the late 80's and by SQLServer, Postgres and InnoDB (MySQL) in their most recent releases. Firebird offers an optional row level lock in the SELECT statement, for applications that need row locks, using a syntax similar to Oracle's. It is also possible to impose a table lock for applications that need such a feature.

Transaction control. Firebird offers full support for commit, rollback and savepoint, including nested savepoints. Additionally, both COMMIT [WITH] RETAIN and ROLLBACK [WITH] RETAIN are supported, enabling the resources allocated for a transaction to be re-used for repetitive tasks.

Distributed transactions . Firebird has offered two-phase commit and XA since its earliest releases.

Transaction programmability. Developers can select the isolation level on a transaction-by-transaction basis. Deadlock detection and resolution is automatic (and infrequent, see 'Locking' above). It is possible to specify whether deadlock notification is immediate or deferred for a configurable timeout period. Most locking conflicts are not deadlocks. Each Firebird transaction can be configured according to several parameters to determine when and how conflicts are resolved.

Embedded use

Embedded server. Because the Firebird server is only a few MB in size, offers a Services API and has low maintenance, it is an ideal system to include with an application. Starting the server involves nothing more than spawning a Firebird server process from the main application.

In-process library. Firebird offers an embedded model in which the entire database engine is located in a single library file ("dll" or "so", depending on platform) less than 2MB in size. This mode has exactly the same functionality as the full-server counterpart, with the exception that databases can only be opened by a single application at a time.

Backup. The backup/restore functionality of the in-process library server is exactly the same as that provided by the full server models.

Administration. Firebird has a "Services API", which makes it easy for the hosting application to access administration functionality, such as managing users, access rights and performing backup and restore.

Firebird was designed to be self-managing and self-tuning. Other than backup and restore it does not need any DBA intervention to keep running smoothly. This quality has been a major driver behind the success of Firebird.

Programmability

Constraints. Firebird has full support for constraints, including primary key, foreign key, check constraint, unique, default and not null. Referential integrity is completely implemented, including SQL-standard CASCADE and SET NULL options.

Triggers. Firebird supports triggers on tables and views. Triggers can act on update, delete and insert with full access to both the "old" and "new" column values for all columns. Uniquely, Firebird allows multiple triggers for each combination of phase (BEFORE/AFTER) and event (INSERT/UPDATE/DELETE), for execution in a specified sequence. It also supports multi-event triggers, i.e. a single trigger module that can be executed conditionally BEFORE | AFTER INSERT OR UPDATE OR DELETE.

Stored Procedures. Firebird supports stored procedures. The procedure bodies can be written in Firebird's powerful native procedural SQL language, "PSQL".

Fyracle offers a plugin interface for stored procedures and trigger bodies in other languages. Currently plugins are available for Java, dotNet and C/C++. Fyracle can also handle Oracle syntax PL/SQL stored procedures and triggers.

User-defined external functions. Firebird supports UDF's. UDF's can be written in any language that can provide an entry point through the "cdecl" interface, although C/C++ and Delphi are the most common choice. The Firebird distributions come with two UDF libraries and many more are available through community channels, with or without open source code.

Views. Firebird supports views. Views are 'first class citizens', with their own user access control, the possibility for views on views, support for WITH CHECK OPTION, etc.

Procedural views. Firebird directly supports another form of virtual table, the selectable stored procedure. This is a procedure that returns a result set of individually calculated rows. Oracle recently added a similar feature under the name "pipelined table function".

Sub-queries. Firebird supports sub-queries in many contexts: in the select list, in the from list ('derived tables'), as source for updates or inserts, in where clauses, as (potentially recursive) common table expressions, etc. Sub-queries can be nested to any depth within the maximum of 254 sub-queries per main query.

Federated queries. The Firebird the core engine does not support federated queries itself. Distributed queries can be implemented using third-party tools, such as one distributed by EasySoft.

Connectivity. Next to its native C/C++ interface, Firebird has full featured drivers for ODBC, Delphi, Java JDBC, dotNET and PHP. Firebird has drivers for many smaller languages, such as Perl, Python and Ruby.

ESQL precompilers. Firebird offers an ESQL precompiler for C, C++, ADA, COBOL and Fortran.

Database availability

Online reorganization. Firebird was designed to be self-managing and self-tuning. Incremental reorganisations are part of query execution. Larger reorganisations are handled by a background thread and do not stall queries and/or transactions.

Replication. Firebird does not support replication in the core engine. However, it is supported through a range of third party products, including support for bi-directional replication. The most well-known product is IBReplicator.

High-availability. Firebird offers a range of high-availability techniques. It can be configured with a guardian process to restart the database server after failure. It can be configured as 'one process per client', to isolate failures. To deal with failing disk hardware, Firebird can maintain a hot backup, or 'shadow', on a second disk. On Unix/Linux server platforms, the shadow can be located on a separate, second system.

Online backups. Firebird can take a backup of a database without shutting down. Making backups does not halt or delay transactions and/or queries. Backups are database level. From version 2.0 onward, incremental hot backup is an option also.

Disaster recovery. Firebird has a 'careful write' design, which always leaves the database in a consistent, ACID state. As a result, recovery is very fast. This quality has led to several design wins for Firebird, including usage by research ships in Antarctic waters and by the US military in field tanks. Firebird's architecture does not require WAL or retrospective logging to recover from disaster. For those who must have it, third-party plug-ins are available.

Security

Authentication and authorization. Authentication is through user name and encrypted password.

Encryption. Firebird does not encrypt the database or network communication. For secure communication across a network a companion product, Zebedee, is frequently used. Backups are not encrypted.

Auditing. Firebird does not offer auditing built into the core engine. Auditing is offered through a third party companion product, IBLogmanager.

Administration

Installation. The install requires only a few mouse clicks and typically completes in a few minutes. Even a first time user would complete the default install in under 5 minutes.

Disk space. A typical install requires some 25MB of disk space. A minimal install would require about 3MB of disk space. Firebird is by far the smallest of the full featured SQL database engines.

Memory. For a minimal configuration, some 16..32MB would suffice. As a rule of thumb, one should allow for 10MB of free memory per concurrent connection/user, using the installation defaults for configurable memory settings.

On-going administration. The install leaves the database system up and running. Other than backup/restore, DBA attention is not needed. All tasks, start/stop, backup/restore, etc. can be handled entirely programmatically.

Native administration tools. Firebird's native tools are powerful command line tools, that work identically across all supported platforms. Typically, these tools are also scriptable. GUI tools are offered by a variety of third party vendors and open source projects.

Third-party GUI tools. Many GUI tools are offered for Firebird -- too many to count accurately! Two toolsets that are widely used throughout the Firebird world are IBExpert and Database Workbench, for the breadth and depth of their access to all of Firebird's features. The open source FlameRobin toolset is a cross-platform GUI application that is winning adherents, even in beta.

Platforms

Server platform. Firebird 1.5 is available for Windows, Linux, Solaris (SPARC and Intel), FreeBSD, MacOSX and HP/UX.

Client platform. As above. Any supported client platform can access any supported server platform. The main determinant for application development lies in language driver support for the Firebird API that is exposed by the client library.

64-bit support. Firebird 2.0 will support 64-bit Linux and Windows, with Solaris and MacOSX following later. Firebird 1.5.3 (the current release version) supports 64-bit file access but must be used in 32-bit mode on 64-bit CPU cores. Because of data-alignment issues, that will be resolved for the Firebird 2.0 release, it is unlikely that a 64-bit core support version of v.1.5.x will be implemented.

Data warehousing

Data partitioning. Firebird does not support physical partitioning. However, the data can be spread over multiple files/disks if needed. Users report good performance on data warehouses 50-100GB in size.

Index partitioning. Firebird handles index partitioning internally and no deterministic DDL is available to change the way the engine maintains the index trees. Firebird 2 further optimizes the way partitioning is carried out for multi-segment indexes.

Bitmap indexes. Firebird automatically selects and optimizes indexes. Firebird has supported bitmap indexes since its earliest releases. Firebird will create bitmap indexes on-the-fly if the optimizer judges this to be useful. Firebird 2 removes hard restrictions on column width and enhances the run-time choices available to its cost-based optimizer.

Parallel query. Firebird does not support parallel queries in the sense of a single query's steps being broken down by the engine and allocated to separate threads/CPU's for parallel processing. Utilization of SMP hardware depends on the deployment model chosen for development and deployment, with clear benefits in the "Classic" server model. The Firebird Vulcan release allows full use of the processing power of SMP hardware.

ETL. Firebird can do high speed concurrent data load/unload. The source and target set[s] can optionally be flat text files on the local file system, which are defined to the database as external tables.

Freedom and Cost

License. Firebird is covered under the Initial Developer Public License and the Interbase Public License. These are essentially Mozilla ("MPL") licenses, with modifications to remove the inappropriate proprietary claims of Netscape that are inherent in the MPL.

Development support cost. Firebird is free, including for commercial use. Most users opt for support through the highly responsive community support forums. Support contracts are available from several companies around the world. Pricing varies according to locality and the level of support needed.

Production support cost. The Firebird software itself is free of any production usage fees. Firebird is often included with an application, and the vendor or reseller typically supports Firebird as part of the application support; otherwise the same parameters apply to production support as to developer support.

It is very unusual for a Firebird site to have a DBA-in-residence, since there is nothing much to do! In many locations it is very common for medium or large sized sites to retain a Firebird "expert" under some form of contract, to address any support issues that arise.

Access to product features. Firebird has no "paying customers". Everyone gets everything there is, free of any license or deployment fees. Some customers/vendors opt to create private builds to cater to their specific needs, but these are just that: private builds.

Registration. Commercial developers and users are encouraged to register with the Firebird Foundation and make voluntary payments to support further Firebird development. The suggested payment is \$50 per concurrent user per year.

Community

Governance. Control is held by the Firebird Project community, specifically centred in the Firebird Admin team, which is convened by consensus as the group of core engine developers and others most closely involved in product architecture, development, production and support.

Its legal and funding interests are in the hands of a non-profit organisation, the Firebird Foundation Incorporated, which is legally incorporated in Australia. No commercial entity has control over the Firebird Project. Businesses are encouraged to exert their "influence" by sponsoring specific projects, joining the Foundation and/or contributing their code to the project.

Source code control. Developers get update authority through a peer review, merit based process - a developer must earn the right to post updates. Any updater must verify his/her own changes before posting by building the engine in a sandbox and subjecting it to the entire QA battery.

All updates, no matter from which developer, pass through a screening process to check for quality, style, effect downstream, etc. Sources are strictly controlled using the CVS system hosted by Sourceforge.net.

History. The Firebird project has been running for almost 6 years and it has not slowed down. Firebird began in 2000 from the InterBase 6 source code that was released as open source. The roots of InterBase go back to 1984, and even before that to DEC's "Rdb" product.

Installed base. Firebird is distributed through a variety of means, including downloads directly from the project's website and as both source code and binaries in some Linux distributions. Taking version upgrades into account, the project estimates the installed base of Firebird 1.5.x, at between one and two million development and production servers. For all versions, we estimate about 3 million copies.

Vendor adoption. Since no commercial licensing is involved, a precise number is hard to give, but we estimate the number of application builders that include Firebird as part of their commercial deployments to be in the thousands. Government agencies, both military and civil, that incorporate Firebird into their systems are becoming noticeable, too.

It's worth noting that Firebird benefits from its ancestor, InterBase, having been bundled with Borland's developer tools (Delphi/C++Builder) for over a decade and being deployed over many years in thousands of midmarket client-server applications.

Customer base. Again, a precise number is hard to give, but a conservative estimate would be around the 100K mark. These appear to be typically departmental workgroup solutions running Windows workstations with Windows client applications or browser clients over a Firebird back-end on Linux or Windows. In Russia, Eastern Europe and Brazil, Firebird has seen significant uptake in the financial world (in some of those countries, Firebird has rapidly become the second most popular choice after Oracle).

Community activity. The community is very active, with dedicated, highly responsive forums for support, development, various language drivers, etc., in many languages. User conferences occur every 9 months or so, alternatively hosted in the EU and Brazil. There are many community news blogs/services.

License partners. The Firebird license allows for free, unqualified redistribution of binaries, whether downloaded or built from source code. There is no such thing as a "license partner". The licensing requires that all modifications to the Firebird source code be made publicly available under the same licenses as the Firebird code (IPL and IDPL).

Training

Publications. About twenty hard copy books at the last count. Most titles are in German, Japanese, Portugese, etc. "The Firebird Book" is devoted wholly to Firebird and has been published in English, Russian and Brazil Portuguese, so far. The English version is published in both hardcopy and eBook formats.

Classroom training. No specific classes that the project either controls or counts. Several support companies offer Firebird training, often alongside training programmes in Borland's InterBase and its language products.

Web-based training. Free support is readily available, and not just for the RDBMS. The Firebird community also has active support lists for Java, PHP and Python, as well as for a range of Delphi component solutions.

On-site training. On site training is offered by a small range of companies, such as IBPhoenix, H-K Software, iBase.ru, Firebase.com.br and others.

Services and support

System integrators. Although not limited in capabilities, Firebird is deployed most typically in mid-market scenarios where the application vendor also acts as system integrator. Again, since we don't sell licences, we don't have any way to keep account of the numbers.

Commercial support. End-user support is typically offered as part of application support for applications using Firebird. Specific Firebird support is offered by a small range of companies, such as IBPhoenix, H-K Software, Janus Software, iBase.ru, FireBase.com.br and others.

24x7 mission-critical support. Internet communication in its various forms is very important to both the support companies and their clients, since it enables them to cover wide ranges of time, expertise and geography.

