

Server versions and differences

Firebird is available for various platforms, the main ones are currently 32-bit Windows, Linux (i586 and higher, and x64 for Firebird 2.0 on Linux), Solaris (Sparc and Intel), HP-UX (PA-Risc), FreeBSD and MacOS X. Main development is done on Windows and Linux, so all new releases are usually offered first for these platforms, followed by other platforms after few days (or weeks).

There is also a choice of server architecture: [Classic server](#) or [Superserver](#). If you're not sure after reading this chapter, whether the Classic server or the Superserver better meets your needs, then install the Superserver. A comparison of the two server types can be found in the *Firebird 2 Quick Start Guide* chapter, [Classic or Superserver](#).

Classic server

The Firebird Classic server offers multiple processes per connection and [SMP \(Symmetric Multi-Processing\)](#) support. Each connection uses one process. It supports multi-processor systems but no shared cache. I.e. each user connecting and requesting [data](#), will have his/her [data pages](#) loaded into the cache, regardless of whether other users' requests have already caused the server to load these pages. Which of course leads to a higher RAM necessity. However, as RAM and cache requirements are relevant to the size of the [database file](#) and the drive on which it is stored, the effects of this cache connection architecture doesn't necessarily have to be a bad thing.

Classic can be a good choice if the host server has multiple CPUs and plenty of RAM. Should you have sufficient working memory, we recommend you use the Classic Server and [set the cache](#) per user somewhat lower.

Further information regarding the Classic server can be found in the *Classic Server versus Superserver* article, in the [InterBase® Classic architecture chapter](#).

Superserver

The Firebird Superserver has one process and multiple threads, but no SMP (Symmetric Multi-Processing), i.e. a dual-core machine. It serves many clients at the same time using threads instead of separate server processes for each client. Multiple threads share access to a single server process, improving database integrity because only one server process has write access to the database. The main advantage is however that all connected users share the database cache. If a data page has already been loaded for one user and the second user needs to access data on the same page, it doesn't need to be reloaded a second time into the cache.

Superserver's [shared page cache](#), properly configured, can be beneficial for performance where many users are working concurrently. On the other hand, Superserver for Windows does not “play nice” with multiple CPUs on most systems and has to be [set for affinity](#) with just one CPU, as requests are serialized per database internally by the engine. However using Superserver with simultaneous connections to several databases *can* be distributed across multiple CPUs.

For further information regarding the Superserver, please refer to the *Classic Server versus Superserver* article, in the [InterBase® Superserver architecture](#) chapter.

Embedded server

Firebird 2.5 Embedded server

The Embedded server in Firebird 2.5 is now based on the [Superclassic](#) architecture and includes all [SMP](#) advantages. There is no longer an exclusive lock on the database file, which opens up the potential fields of application. A mixture of multiple embedded processes and regular Classic and Superclassic servers is possible, by which the same database file can be accessed in both read and write mode. The necessary synchronization occurs via a global lock table that can be found, for example in Windows, under C:\Documents and settings\All users\Application data\Firebird. [Superserver](#) cannot be used in this multi-connect context, because it requires an exclusive lock on the database file.

Thread-safe means that an independent database connection per thread is no longer required, but one connection handle can be used by several threads. The connection level synchronization routines necessary for this can be found in the Embedded server or in the client libraries, i.e. the client libraries (gds32.dll, fbclient.dll etc.) are now thread-safe. However a thread synchronization is necessary in the client application, in order to avoid unwanted side effects, for example when a transaction handle is collectively used by multiple threads. One independent connection and transaction per thread will avoid any unwanted side effects. Furthermore the Superclassic SMP ability can only be fully utilized in Embedded with a separate connection per thread.

Pre-2.5 Embedded server

The Embedded server is based on the Superserver architecture and allows only one local process per database, which of course means that it is unsuitable for a web server! The Firebird 2.1 Embedded Server version provides a useful enhancement: the client library is embedded in the server, this combination performing the work of both client and server for a single attached application. Only a few files are required without installation. It mainly consists of a slightly larger fbclient.dll, which is capable of providing the database server service to all installations. It is not necessary to install or start anything. This is particularly advantageous, for example, in the following situation:

You have an accounting application in an old 1997 version that you need to start today to view old data that was created and processed using this version. Normally you would have to search for the old version, install it, and - if for whatever reason it doesn't work anymore (or maybe you never managed to find it in the first place!) - you can't get to your data. Solution: pack your accounting application onto a DVD together with the correct Firebird embedded version. You can then start the application directly from the DVD without having to search and install anything. This is particularly useful when archiving data.

Firebird is, by the way, one of the few database systems that can read a database on a [read-only](#) medium.

For details regarding installation of the Embedded server, please refer to the Firebird Migration and Installation Guide chapter, [Windows Embedded](#).

[back to top of page](#)

Firebird 4.0

Firebird 4.0 alpha was officially released on the 24 August 2017. Below you can find a list of [Firebird 4.0 features already implemented in IBExpert](#).

The Firebird 4.0 Release Notes (21 August 2017 - Document v.0400-07 - for Firebird 4.0 Alpha 1) can be referred to here: [PDF](#).

Firebird 4.0 features already implemented in IBExpert

Since IBExpert version 2017.10.01 and Firebird 4 Alpha, IBExpert supports the following:

- DECFLOAT, BINARY, VARBINARY data types.
- DECFLOAT types are now bound to CHAR. SET DECFLOAT BIND CHAR is performed just after the connection to database is established.
- Parameter tips for COMPARE_DECFLOAT, NORMALIZE_DECFLOAT, TOTALORDER, QUANTIZE functions in code editors.
- Extensions to the IDENTITY type in the Table and Column editors (autoincrement fields).
- SQL SECURITY clause is supported in the corresponding object editors (tables, views, triggers, procedures, packages).
- Extract Metadata supports new data types, SQL SECURITY clause and extensions to the IDENTITY type.

Since then, more features have been added gradually, such as:

Support of NUMERIC/DECIMAL data types with a maximum precision of 34 digits (Firebird 4) in the **Database object editors**; support for cumulative roles and default roles in the **User Manager**; initial support of time zones, support of **BINARY/VARBINARY data types**; support of Firebird 4 functions in the parameters tips in **Code editors**; support of Firebird 4 syntax extensions in **PSQL object editors**; support of management statements in the **PSQL debugger**; support of DECFLOAT values in **Export data into a script**; support of TIME/TIMESTAMP WITH TIME ZONE columns in **Test Data Generator**; support of system privileges in the **Grant Manager**; support of DECFLOAT and NUMERIC(38,x) variables and parameters in the **PSQL Debugger**; support of DECFLOAT and NUMERIC(38,x) parameters in the SQL Editor; bug fixes for missing SQL SECURITY clause in lazy mode in the Trigger Editor and for the missing USAGE privileges while working with a Firebird 4 database in **Grant Manager**; initial support of DECFLOAT/NUMERIC(38) fields in the Test Data Generator; support of DECFLOAT and NUMERIC(38) fields; sorting, filtering and aggregate functions implemented in the **Data Grid**; parameter hints for new Firebird 4 functions (HEX_ENCODE, HEX_DECODE, MAKE_DBKEY, RDB\$GET_TRANSACTION_CN) have been added, as well as highlighting of new Firebird 4 keywords and reserved words.

Most (if not all) new features of Firebird 4 are also available in [IBEBlock](#), so you might want to update your scripts and gain advantage of these new features.

[back to top of page](#)

Firebird 3.0

Tutorial

Firebird 3.0 alpha was officially released on the 2nd August 2013. A summary of all new Firebird 3.0 features can be referred to below.

The current Firebird 3.0 Release Notes can be referred to here: [Firebird 3 Release Notes](#).

Summary of Firebird 3.0 features

The following is an excerpt from the [Firebird 3.0 Release Notes \(27 January 2014 - Document v.0300-08 - for Firebird 3.0 Alpha 2\)](#) chapter, Summary of Features.

The following list summarises the features and changes, with links to the chapters and topics where more detailed information can be found.

Unification of the Firebird executable is complete

With the completion of true SMP support for Superserver, the Firebird core is now a unified library that supports a single ODS, loadable either as an embedded engine or by the “network listener” executable. Choice of server model is determined by settings for two new configuration parameters defining the locking and cache models, respectively: SharedDatabase and SharedCache. They can be specified at either global level (in firebird.conf) or “per database” (in databases.conf).

By default, SharedDatabase = false and SharedCache = true, thus meaning SuperServer.

Note: The previous aliases.conf is replaced by databases.conf, now including not just aliases for databases but also (optionally) configuration parameters to enable configuration of databases and/or alternative security databases individually.

The changes are described in more detail in the chapter Changes in the *Firebird Engine*.

True SMP support for SuperServer

In Superserver mode, the engine now makes use of multiple CPUs and cores when spawning connections.

Tracker: [CORE-775](#)

Implemented by V. Khorsun

New, object-oriented C++ APIs

Object-oriented C++ APIs enable external code routines to plug in and run safely inside Firebird

engine space, including (but not limited to):

- Stored procedures, triggers and functions written in Java, C++, ObjectPascal, etc.
- Encryption schemes for data
- User authentication schemes, including secure key exchange
- ...

New Data Type Support

A true BOOLEAN type (True/False/Unknown), complete with support for logical predicates, e.g.,

```
UPDATE ATABLE
SET MYBOOL = (COLUMN1 IS DISTINCT FROM COLUMN2)
```

See [BOOLEAN Type](#).

IDENTITY type, spawning unique identifiers for the defined column from an internal generator. See [IDENTITY-Style Column](#).

Support for SQL Packages

For details, refer to [Packages](#).

DDL Triggers

Now, triggers can be written to execute when database objects are modified or deleted. A typical use is to block unauthorised users from performing these tasks.

For details, refer to [DDL Triggers](#).

'Window' functions in DML

A whole new series of analytical functions to work with multiple subsets in DML. See [Window \(analytical\) functions \(Firebird 3.0\)](#).

[back to top of page](#)

Firebird 2.5

Firebird 2.5 was officially released on 4 October 2010. It introduces a number of interesting new features and numerous improvements over previous Firebird versions. 2.5 is the first step into the [Classic](#), [Superserver](#) and [Embedded](#) unification with codebase modernization, a common threadable architecture and a more efficient multi-threading support. Further information can be found below, under [Superclassic architecture](#) and in the online article, [Firebird 2.5 has arrived!](#).

Other new features and improvements in this release include:

Administrative enhancements

- System audit tracing and [user trace sessions](#) via the Services [API](#), making it possible to monitor and analyse everything going on in a database in real time.
- New system role [RDB\\$ADMIN](#) in the ODS 11.2 database allows SYSDBA to transfer its privileges to another user on a per-database basis.
- More information in the [monitoring tables](#).
- Asynchronous cancellation of connections.
- Capability for ordinary users to monitor any of their own attachments as well as [CURRENT_CONNECTION](#).

Other SQL language additions and enhancements

- Regular expression support using the [SIMILAR TO](#) predicate.
- ALTER COLUMN for computed columns.
- Autonomous transactions within a [PSQL](#) module ([stored procedure](#), [trigger](#) or dynamically executable [PSQL block](#)).
- Enhanced access to stored procedures in [view](#) definitions.Gray Colored Text
- Optional [GRANTED BY](#) or [GRANTED AS](#) for [DCL - Data Control Language](#)[[GRANT](#)]] and REVOKE statements, enabling the grantor to be a user other than the [CURRENT_USER](#) (the default).
- [REVOKE ALL](#) syntax to dispose of all privileges for a user or [role](#) at once.
- Support for [WHERE SOME_COL = ? OR ? IS NULL](#) predications.
- Removal of “reserved” status for all but a handful of keywords that are not reserved in the SQL standard.

Data-handling enhancements

- New built-in functions for converting UUID CHAR(16) OCTETS strings to RFC4122-compliant format and vice versa.
- Ability to pass 32-bit and 64-bit integers as hexadecimal in numeric literal and X-prefixed binary string literal formats.

API additions

- Statements now return an SQL-2003 standard 5-alphanumeric [SQLSTATE](#) completion code.
- New constant [DSQL_unprepare](#) available for use with [isc_dsql_free_statement](#) for efficient unpreparing of statements.

International language support

- Default [COLLATE](#) clause for [CREATE DATABASE](#).
- Ability to change the default [COLLATE](#) for a used [character set](#).
- [GBAK](#) restore switches [FIX_FSS_DATA](#) and [FIX_FSS_METADATA](#) to restore legacy databases with

UNICODE_FSS data and/or [metadata](#) correctly without resorting to scripts and manual techniques.

- Accent-insensitive collation for Unicode.

Details of all new features can be referred to in the [Firebird 2.5 has arrived!](#) article and in the [Firebird 2.5 Release Notes: Firebird 2.5 new features](#).

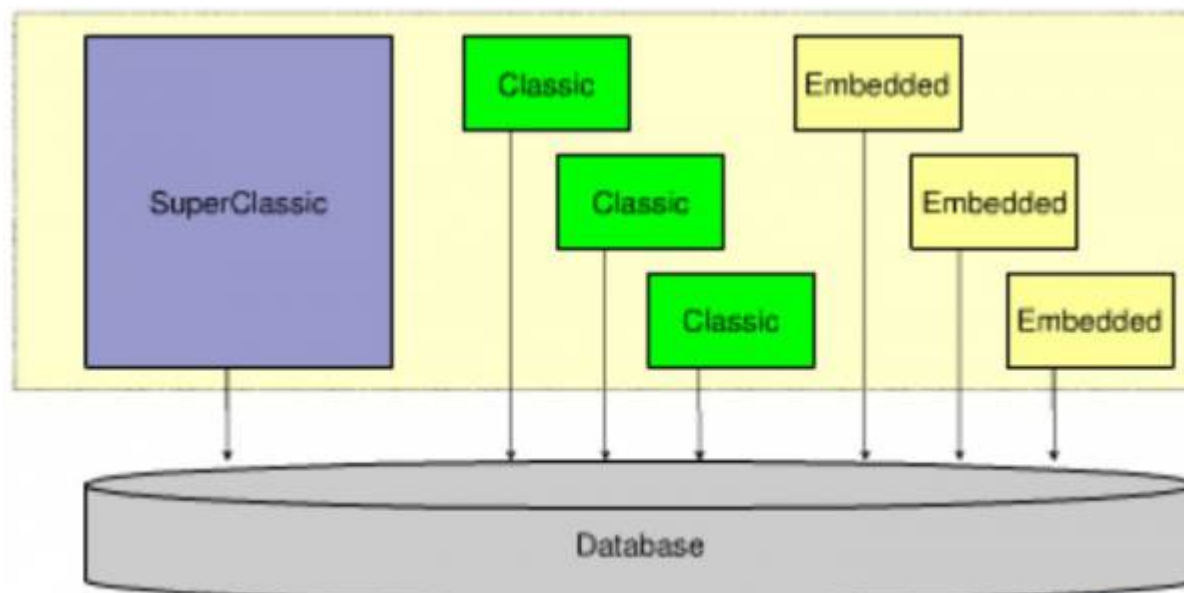
As IBExpert is currently still a 32-bit application, if you wish to use IBExpert with the 64-bit Firebird version, you will need to install a 32-bit client library. Please refer to [Install a 64-bit Firebird for use with IBExpert](#) for further information.

Superclassic architecture

Perhaps the most significant development in Firebird 2.5 is the Superclassic architecture, combining the best of both the [Classic](#) and [Superserver](#) structures to improve its support of multi-processor systems ([SMP](#)). Superclassic can use multiple CPUs with multiple simultaneous connections to a database. The new architecture can be characterized as follows:

- A dedicated server process ([fb_inet_server -m](#) on Windows or [fb_smp_server](#) on POSIX) for all connections (identical to Superserver).
- Private page and metadata cache. This implies the identical calculation of the maximum main memory usage for the coaching as with Classic server.
- SMP compatibility for simultaneous connections to one or more databases.
- SMP compatibility for sweeps and Services [API](#) requests.
- No exclusive locking on the database file (identical to Classic server)
- Cached connection to the user database for a quicker connection establishment (identical to Superserver).
- Secure shutdown of all database connections by ending the dedicated server process (identical to Superserver).
- Automatic shutdown of all database connections in the event of a server crash (identical to Superserver – a disadvantage).
- Target platform: of course deployable on 32-bit systems; however use on 64-bit systems makes more sense, as here there are no limits with regard to the maximum main memory addressing per process (identical to Superserver).

Database access can be safely shared among Classic, Superclassic and [Embedded](#) processes:



(Image copyright Dmitry Yemanov)

Compared to the Classic server Superclassic uses less kernel resources (but not memory), it achieves better performance due to local calls inside the lock manager and other in-process optimizations (25% performance boost in TPC-C). Compared to Superserver, Superclassic offer better scalability (the number of connections is not explicitly limited), better concurrency on SMP/multi-core hardware and it is more responsive under a high load.

For further information regarding the Superclassic server, please refer to the [Firebird 2.5 Release Notes](#).

See also:

- [Firebird 2.5 has arrived!](#)
- [Firebird 2.5 Release Notes](#)

[back to top of page](#)

Red Database server

The Red Soft Corporation has already developed and released their Superclassic server. The Red Database 2.1.x engine is based on Firebird 2.0 and also contains the following new features above the Firebird 2.0 base line:

- Multi-threading architecture - Superclassic server
- External [stored procedures](#)
- [Debug procedure, trigger, function \(IBExpert Debugger\)](#)|[Stored procedures and triggers debugger](#)
- Full-text search
- Improved security subsystem, providing the full set of tools for:
- authentication,
- fine-grained authorization controls, at row and field level,

- keeping detailed audit trail for users' access to data,
- system and data integrity checking,
- analysis of security incidents.

New features in the security subsystem include:

- Integrated cryptographic module
- Complete [DDL](#) operations access control
- Complete [DML](#) operations access control
- The [roles](#) assigned to a user have cumulative effect on his (her) access rights
- The roles can be defined globally for the server
- The support for access policies
- Multi-factor authentication
- Audit system with capability to save log in binary format and analyze it via SQL statements
- Access controls for database service functions ([backup](#), consistency validation, etc.)
- Security clean-up for deallocated memory
- Integrity checking for server files and data protection facilities
- Special tools for [metadata](#) and data integrity checking using digital signature

The integrated role for security and system administration enables you for example to give someone, who is neither the database owner nor the SYSDBA, the ability to create procedures.

This database (Red Database Community Edition) can be downloaded free of charge from <https://www.red-soft.biz/en>. If you wish to distribute this as part of your own software package, you will however require a Distribution License. For further information please mail info@ibexpert.biz.

[back to top of page](#)

Firebird 2.1 new features

Database triggers	Database triggers are user-defined PSQL modules that can be designed to fire in various connection-level and transaction-level events.
Global temporary tables	SQL standards-compliant global temporary tables have been implemented. These pre-defined tables are instantiated on request for connection-specific or transaction-specific use with non-persistent data, which the Firebird engine stores in temporary files.
Common table expressions (CTEs)	Standards-compliant common table expressions, which make dynamic recursive queries possible.
RETURNING clause	Optional RETURNING clause for all singleton operations, UPDATE , INSERT and DELETE operations.
UPDATE OR INSERT statement	New UPDATE OR INSERT for MERGE functionality: now you can write a statement that is capable of performing either an update to an existing record or an insert, depending on whether the targeted record exists.
LIST() function	A new aggregate function LIST() retrieves all of the SOMETHINGs in a group and aggregates them into a comma-separated list.
New built-in functions	Dozens of built-in functions replacing many of the UDFs from the Firebird-distributed UDF libraries.

Text BLOBs can masquerade as long VARCHARs	At various levels of evaluation, the engine now treats text BLOBs that are within the 32,765-byte size limit as though they were VARCHAR . String functions like CAST , LOWER , UPPER , TRIM and SUBSTRING will work with these BLOBs, as well as concatenation and assignment to string types.
Define PSQL variables and arguments using domains	PSQL local variables and input and output arguments for stored procedures can now be declared using domains in lieu of canonical data types.
COLLATE in PSQL	Collations can now be applied to PSQL variables and arguments.
Windows security to authenticate users	Windows “Trusted User” security can be applied for authenticating Firebird users on a Windows server platform host.
CREATE COLLATION command	The DDL command CREATE COLLATION has been introduced for implementing a collation, obviating the need to use the script for it.
Unicode collations anywhere	Two new Unicode collations can be applied to any character set using a new mechanism.
New platform ports	Ports to Windows 2003 64-bit (AMD64 and Intel EM64T) Classic , Superserver and Embedded models; PowerPC, 32-bit and 64-bit Intel Classic and SS ports for MacOSX.
Database monitoring via SQL	Run-time database snapshot monitoring (transactions , tables, etc.) via SQL over some new virtualized system tables . Included in the set of tables is one named MON\$DATABASE that provides a lot of the database header information that could not be obtained previously via SQL: such details as the on-disk structure (ODS) version, SQL dialect , sweep interval , OIT and OAT and so on. It is possible to use the information from the monitoring tables to cancel a rogue query.
Remote interface	The remote protocol has been slightly improved to perform better in slow networks once drivers are updated to utilise the changes. Testing showed that API round trips were reduced by about 50 percent, resulting in about 40 per cent fewer TCP round trips.

Please also refer to the [Firebird 2.1 Release Notes](#).

Note: If you are upgrading from an older Firebird version to the new 2.1 version, it is also important that you upgrade all your clients accordingly. The Firebird 2.1 client can communicate much more effectively with the Firebird 2.1 server, which can mean performance improvements of up to 40%!

From:
<http://ibexpert.com/docu/> - IBExpert

Permanent link:
<http://ibexpert.com/docu/doku.php?id=02-ibexpert:02-01-getting-started:downloading-install-firebird:server-versions-and-differences>

Last update: 2023/08/23 04:07

