

# Firebird Version 1



## Release Notes

March 2002

### **Contents**

[General Notes](#)

[Compatibility with Older Versions](#)

[New Features](#)

[Language Enhancements](#)

❑ [New Reserved Words](#)

[ISQL Features](#)

[External Functions \(UDFs\)](#)

❑ [ib\\_udf library](#)

❑ [FBUDF UDF Library \(using Descriptors\)](#)

[New Configuration Parameters](#)

[API Enhancements](#)

[Installation Notes](#)

❑ [Windows 32-bit](#)

❑ [Linux/UNIX](#)

❑ [Solaris](#)

❑ [MacOS X](#)

❑ [FreeBSD](#)

❑ [Pure Java JCA-JDBC XA-Capable Driver](#)

❑ [InterClient JDBC Driver](#)

[Further Information](#)

[Documentation](#)

[Fixed Bugs](#)

[Known Issues](#)

### **General Notes**

The Firebird database engine has been developed by an independent team of voluntary developers from the InterBase(tm) source code that was released by Borland under the InterBase Public License v.1.0 on 25 July 2000.

This release was built from code which has been subjected to extensive cleanup, bug-fixing and beta testing during the year between the release of the code and this release. The On-Disk Structure is still ODS 10. Certain enhancements requiring changes to the ODS have been deferred to a future release.

A list of bugs fixed appears later in this document.

Several new language features have been added. For syntax, please refer to the Language Enhancements section later in this document.

The Firebird binaries can be downloaded through the FirebirdSQL.org domain -  
<http://www.firebirdSQL.org>

Please refer to the Documentation section at the end of these release notes for locations of recommended documentation.

## **Compatibility**

If you are planning to "play" with Firebird using an existing InterBase database with the intention of reverting to InterBase later, please take all precautions to back up your current version and use a restored copy for experiments.

Please note the changes to functions in the **ib\_udf user-defined function library**, described in the Language Enhancements section following this.

Any existing InterBase database will be 100% upwardly compatible with this software provided you follow the usual rules when migrating databases currently operating under an InterBase 4.x or 5.x server (ODS lower than 10), viz.

- ❑ Create a TRANSPORTABLE InterBase Backup file using the version of the program gbak (gbak.exe in Windows) that is correct for the ODS of the existing database. It is recommended that you give this file a ".gbk" extension. For Windows versions 4 and 5, the program ibmgr32.exe provides a graphical interface for gbak.exe in the Tasks | Backup menu. Do NOT use this program to work on a Firebird or an InterBase 6 (ODS 10) database.
- ❑ If necessary, the backup file can be zipped up and moved via transportable media. Remember that files saved to and then retrieved from a CD-ROM will need to have their Read-Only attribute unset.
- ❑ Using the version of gbak from your Firebird installation, restore the gbk file to the disk location where your Firebird database will be run. It must be located on the same physical machine as the ibserver program.
- ❑ Your converted database is now ready to run as a Dialect 1 Firebird database.

The Operations Guide from the documentation set contains the command syntax for the gbak backup and restore program.

Some utility programs, such as IBConsole, provide a graphical user interface tool for ODS 10 backup and restore. The recommended tool is IBBackup, a freeware binary by Martin Schmid <M.Schmid@EQUITANIA.de>, which can be downloaded from <http://www.equitania.de/interbase/downloads/ibBackup.zip>.

## **Database Desktop Client Programs**

In the months leading up to the release of the independent open source Firebird release, the Borland InterBase community has closed ranks with respect to development of its open source IBConsole program. The IBConsole "code cardinals" have declined to engage in discussion of accepting patches to support the bug-fixes and enhancements that Firebird has acquired during 19 months of development. Because of this, there is no guarantee that all of the functionality of IBConsole will work with Firebird.

Several excellent alternatives are listed in the Contributed Downloads page at <http://www.ibphoenix.com>. Some are open source, some are freeware, others are established commercial products.

Two free desktop programs are recommended:

- ❑ Jason Wharton's IB\_SQL is available free-of-charge at <http://www.ibobjects.com>. It provides a large range of tools for database administration, scripting, data-pumping, security maintenance, etc.
- ❑ Michael Mutl's IBQuery, also available free-of-charge, provides an SQL toolbox with an elegant GUI and a small footprint. It can be obtained from the Torrys website at <http://www.torry.net/apps/utilities/database/mitecibquery.zip>.

Other free tools:

- ❑ IBAccess, also free, is built with IBX components and has both Windows and Linux versions. It can be found at [www.ibaccess.org](http://www.ibaccess.org). Get in touch with its author Toni Martir (address at the web site) if you bump into any compatibility problems. It is an open source project without Borland affiliations, so incompatibility is likely to be addressed.
- ❑ Marathon 2 is under open source development under the directorship of Tilo Muetze. See the project website at <http://www.alanti.net/firebird/marathon/>

### **Delphi/Borland C++Builder components**

Similar difficulties, for similar reasons, may be expected with the Borland InterBaseXpress (IBX) and DBExpress (DataSnap in versions 6) components for Delphi and C++Builder. It is also considered unlikely that Borland will repair or enhance its BDE InterBase 6 driver to support the Firebird language and API enhancements.

Delphi and C++Builder users have two powerful alternatives to IBX and the BDE, both well supported with developer and peer support:

- ❑ Jason Wharton's IB Objects at <http://www.ibobjects.com>
- ❑ FIBPLus at <http://www.devrace.com>

### **Version Strings for Firebird Releases**

Version strings remain the same for Firebird 1.0 and return much as they have before, i.e. "LI-V6.2.0.nnn Firebird" (where nnn is the build number).

This prepares the way for two connection mode options in the forthcoming Firebird 2:

- ❑ In native Firebird mode the server will listen on a different port (probably 3051) and return a version string like "LI-V2.0.0.XXX Firebird".
- ❑ It will have a "compatibility option" by which it listens on port 3050 and returns a compatibility version string (nominally "LI-V6.2.0.368 Firebird". This compatibility string will probably be user-configurable via Registry setting/conf file.

New items in the `isc_database_info` parameter - `isc_vendor` info and `isc_info_fb_version` - will provide an additional means for a client application to detect whether it is connecting to a Firebird or Borland Interbase database and to which Firebird version it is connecting.

### **InterClient and JDBC Connectivity**

The Firebird-Java team has implemented a pure Java driver for Firebird. This driver is based on both the new JCA standard for application server connections to enterprise information systems and the well known JDBC standard. For details, please refer to the Installation section.

### **Linux Versions**

All Firebird Linux versions are built with glibc 2.2. Older Linux versions like RH 6.2, Debian etc., will need to be upgraded. Paul Reeves and Pavel Cisar are investigating options to provide 32-bit I/O builds that will work with glibc 2.1 too, to offer as an "additional" release for these older Linux platforms.

A **thread-capable `gdslib.so`** (client library for Linux) is included in latest 1.0 builds.

## ***New Features***

### **Enhanced Dependency Checking**

UDFs and generators are now tracked, so that it will not be possible to drop one if it is in use by a procedure or computed column.

### **Large Database File Support**

(Sean Leyne, David Trudgett)

The logical database can comprise a primary file and numerous secondary files (the standing record for a single FB/IB database is 980GB!). However, this enhancement concerns the size of a single physical database file.

The maximum file size limit involves three separate factors:

- the implementation (coding) of the FB/IB engine I/O functions
- operating system support for 64-bit file operations
- filesystem limits on file size

### **Engine support for 64-bit file I/O**

All Firebird/IB versions prior to FB v1.0 (including IB v6.0) used both 32-bit API/OS calls and 32-bit (integer) file pointer structures. The engine's capability was thus designed to handle files with a maximum file size of 4Gb. In the Unix code the file pointer structures were defined as signed integers, thus limiting Unix and Linux implementations to file sizes of 2Gb.

### **MS Windows**

By design, all versions of Win32 (Win9x/ME/NT/2000) provide full support for 64-bit pointer file operations, provided the right file pointer structure is passed. To add large file support for Firebird, it was necessary only to change the file pointer structure to `LARGE_INTEGER` and to modify the I/O calls themselves slightly.

## Linux

For Linux, 64-bit file operations are supported with version 2.2 of glibc and above, using the new 2.4 kernel. Version 2.1.3 of glibc contained limited but broken support for large files. Red Hat Linux 7.1 ships with glibc 2.2.2-10 and has been tested to work with files over 2Gb (using the ext2 native Linux filesystem, which is the default filesystem in nearly all Linux distributions). Generally speaking, any recent Linux distribution that uses the 2.4 kernel, ext2 filesystem, and glibc 2.2 or higher, will handle files larger than the previous 32-bit/2Gb upper limit.

Because 64-bit large file support is not ubiquitous among Linux distributions and commercial Unix platforms, the Firebird Unix ports come in two flavours: 32-bit and 64-bit I/O. Choose the version you know your platform can support.

The basic requirement is glibc 2.2 (+ support in core utilities) and the 2.4 kernel (for VFS). All these conditions are met in RH 7.1, MDK 8.0 and SuSE 7.2 and newer versions. But beware, some distributions have a choice to install old components! For example Mandrake 8.x allows installation of the 2.2 kernel in advanced setup.

## MacOS X 10.0

Firebird does not work with MacOS X 10.0 due in part to 10.0's bug in 64 bit I/O support.

## Filesystems

Although the engine can now support 64-bit operations, it is important to remember that not all filesystems support large files.

In the case of Win32, the following \*file\* size (not to be confused with partition/volume) limits apply:

- FAT16 for Win9x/ME, max file size = 2GB - 1byte
- FAT16 for WinNT/2000, max file size = 4GB - 1byte
- FAT32 for Win9x/ME/2000, max file size = 4GB - 1byte
- NTFS for WinNT/2000, max file size = 16,384GB - 1byte

For Linux and Unix flavors, because there are so many different filesystems, you should consult the following URL for the exact details:

[http://www.suse.de/~aj/linux\\_ifs.html](http://www.suse.de/~aj/linux_ifs.html)

The good news is that all the major Linux distributions use the ext2 filesystem, which fully supports large files.

It really boils down to:

1. Use kernel 2.4 and above
2. Use glibc 2.2 and above
3. Use ext2 filesystem, or other filesystem that supports the kernel's LFS (large file system) interface.

(David Trudgett)

4. It should work on ReiserFS and Ext3 (but it is not verified). (Pavel Cisar)

## Support for 16kb pages is enabled

The engine and GBAK routines were modified to allow for PAGE\_SIZE beyond the previous 8kb maximum. It will allow for a greater database size provide more options for tuning databases to fit the hardware environment.

## Win32 Forced Writes Default = ON

With the introduction of IB 6.0 the default FORCED WRITES setting for newly created databases, under Win32, was changed to OFF, ostensibly to improve database performance but at the expense of database integrity/recovery. The default has been reverted to ON.

A problem arising from the "bad connection string" fix in RC 1, preventing you from switching forced writes in multi-user mode, was resolved.

## Limit raised on External Table handles (NT)

The former limit of the number of external tables that can be in use simultaneously on NT is the default number of *fopers* allowed minus some files we read, or about 510.

The Firebird change increases that total number of *fopers* on NT to the highest value allowed on the operating system, 2048. The actual limit on the number of external tables that can be open simultaneously is untested, but believed to be about 2040, depending on the number of databases open, the number of files per database, etc.

## External files now closed when external table is dropped

A small change to the [Deferred Work Handler module \(dfw.e\)](#) now closes external files when the external table that attached to them is dropped. This gets rid of the annoying problem of dropping the table and being unable to delete the associated file without shutting down the server.

## Specify the Listening Server Port in the Connection String

Feature id 1468, SF ID 447400

The ability to specify a port is a new Firebird feature. To connect to a server listening on a port that is not the standard 3050, you can now include the alternative port in the connection string. The syntax is as follows:

Win32:

```
Server/Port:Drive\Directory\Database.gdb
```

Linux and other Unix-syntax-compliant ports:

```
Server/Port:/device/Directory/Database.gdb
```

Notice the slash instead of a colon between the server name and the port. You still need the colon before the physical path string.

## Connection now defaults to Port 3050

The client and server will now use TCP port 3050 if no entry in the Services file is found. This should solve a common client/server installation problem.

## GBAK has new COUNTER option for -V(erbose) switch

The -V (Verbose) option of GBAK now allows for a 'counter' value to be specified i.e.

```
GBAK ... -V 20000
```

This value will provide additional user feedback, as the data rows are backed-up or restored. Once GBAK has processed the defined number of rows (either backed-up/restored or re-indexed), GBAK will print a message indicating the running row count value.

By default, the counter value is 10,000.

## Ambiguous JOIN statements are now rejected

InterBase does not prevent you from submitting a statement like this:

```
SELECT A.FIELD_A, B.FIELD_A  
FROM A JOIN B  
ON FIELD_X = FIELD_Y  
WHERE FIELD_A="99"  
ORDER BY FIELD_A
```

Such statements return unpredictable output sets. Now Firebird, in Dialect 3, will return an error if there are any unqualified column identifiers in join statements. In Dialect 1, it will return a warning but the ambiguous query will be allowed to proceed.

## **Language Enhancements**

### CURRENT\_USER and CURRENT\_ROLE

These two new context variables have been added to reference the USER and (if implemented<sup>1</sup>) the ROLE of the current connection context.

```
CREATE GENERATOR GEN_USER_LOG;  
CREATE DOMAIN INT_64 AS NUMERIC(18,0);  
COMMIT;  
CREATE TABLE USER_LOG(  
    LOG_ID INT_64 PRIMARY KEY NOT NULL,  
    OP_TIMESTAMP TIMESTAMP,  
    LOG_TABLE VARCHAR(31),  
    LOG_TABLE_ID INT_64,  
    LOG_OP CHAR(1),  
    LOG_USER VARCHAR(8),  
    LOG_ROLE VARCHAR(31));  
  
COMMIT;  
  
CREATE TRIGGER ATABLE_AI FOR ATABLE
```

```

ACTIVE AFTER INSERT POSITION 0 AS
BEGIN
  INSERT INTO USER_LOG VALUES(
    GEN_ID(GEN_USER_LOG, 1),
    CURRENT_TIMESTAMP,
    'ATABLE',
    NEW.ID,
    'I',
    CURRENT_USER,
    CURRENT_ROLE);
END

```

CURRENT\_USER is a DSQL synonym for USER that appears in the SQL standard. They are identical. There is no advantage of CURRENT\_USER over USER.

<sup>1</sup> If you insist on using an InterBase v.4.x or 5.1 database with Firebird, ROLE is not supported, so current\_role will be NONE (as mandated by the SQL standard in absence of an explicit role) even if the user passed a role name. If you use IB 5.5, IB 6 or Firebird, the ROLE passed is verified. If the role does not exist, it is reset to NONE without returning an error.

This means that in FB you can never get an invalid ROLE returned by CURRENT\_ROLE, because it will be reset to NONE. This is in contrast with IB, where the bogus value is carried internally, although it is not visible to SQL.

## DROP GENERATOR

Enables unused generators to be removed from the database. Storage will be freed for re-use upon the next RESTORE. Available in SQL and DSQL.

```
DROP GENERATOR <generator name>;
```

## GROUP BY UDF

It is now possible to aggregate a SELECT by grouping on the output of a UDF.  
e.g.

```

select strlen(rtrim(rdb$relation_name)), count(*) from rdb$relations
group by strlen(rtrim(rdb$relation_name))
order by 2

```

A side-effect of the changes enabling grouping by UDFs is that, whereas previously you could not call built-in Firebird functions in GROUP BY, now, by creating a dummy UDF wrapper, you can do:

```

select count(*)
from rdb$relations r
group by bin_or((select count(rdb$field_name) from rdb$relation_fields f
where f.rdb$relation_name = r.rdb$relation_name),1)

```

IT WOULD BE GOOD TO GET SOME EXAMPLES HERE, PLEASE - Editor



## RECREATE PROCEDURE

This new DDL command lets you create a new stored procedure with the same name as an existing procedure, replacing the old procedure, without needing to drop the old procedure first. The syntax is identical to CREATE PROCEDURE.

Available in SQL and DSQL.

## RECREATE TABLE

This new DDL command lets you create a new structure for an existing table without needing to drop the old table first. The syntax is identical to CREATE TABLE.

Observe that RECREATE TABLE does not preserve the data in the old table.

Available in SQL and DSQL.

## SELECT [FIRST (<integer expr m>)] [SKIP (<integer expr n>)]

Retrieves the first  $m$  rows of the selected output set. The optional SKIP clause will cause the first  $n$  rows to be discarded and return an output set of  $m$  rows starting at  $n + 1$ . In the simplest form,  $m$  and  $n$  are integers but any Firebird expression that evaluates to an integer is valid. A identifier that evaluates to an integer may also be used in GDML, although not in SQL or DSQL.

Parentheses are required for expression arguments and optional otherwise.

They can also bind variables, e.g. SKIP ? \* FROM ATABLE returns the remaining dataset after discarding the  $n$  rows at the top, where  $n$  is passed in the "?" variable. SELECT FIRST ? COLUMNA, COLUMNB FROM ATABLE returns the first  $m$  rows and discards the rest.

The FIRST clause is also optional, i.e. you can include SKIP in a statement without FIRST to get an output set that simply excludes the rows appointed to SKIP.

Available in SQL and DSQL except where otherwise indicated.

Examples:

```
SELECT SKIP (5+3*5) * FROM MYTABLE;
```

```
SELECT FIRST (4-2) SKIP ? * FROM MYTABLE;
```

```
SELECT FIRST 5 DISTINCT FIELD FROM MYTABLE;
```

### ***A Gotcha with SELECT FIRST***

This

```
delete from TAB1 where PK1 in (select first 10 PK1 from TAB1);
```

will delete all of the rows in the table. Ouch! the sub-select is evaluating each 10 candidate rows for deletion, deleting them, slipping forward 10 more...ad infinitum, until there are no rows left.

Beware!

## **SUBSTRING( <string expr> FROM <pos> [FOR <length>])**

Internal function implementing the ANSI SQL SUBSTRING() function. It will return a stream consisting of the byte at <pos> and all subsequent bytes up to the end of the string. If the option FOR <length> is specified, it will return the lesser of <length> bytes or the number of bytes up to the end of the input stream.

The first argument can be any expression, constant or identifier that evaluates to a string.

<pos> must evaluate to an integer.

<pos> starts at 1, like other SQL commands.

Neither <pos> nor <length> can be query parameters.

Because <pos> and <length> are byte positions, the identifier can be a binary blob, or a sub\_type 1 text blob with an underlying one-byte-per-character charset. The function currently does not handle text blobs with Chinese (2 byte/char maximum) or Unicode (3 byte/char maximum) character sets. For a string argument (as opposed to a blob), the function will tackle ANY charset.

Available in SQL and DSQL.

```
UPDATE ATABLE
SET COLUMNB = SUBSTRING(COLUMNB FROM 4 FOR 99)
WHERE ...
```

Please refer also to the section on External Functions (UDFs) following this, for details of changes and additions to external substring functions in the standard UDF library.

## **Case Insensitive Hungarian Collation Set**

Added case insensitive Hungarian collation set, developed and tested by Sandor Szollosi (ssani@freemail.hu).

## **Charset ISO8859-2 added**

Firebird supports character set ISO-8859-2 (for Czech language).

## **New Comment marker**

**for use in scripts, DSQL, stored procedures and triggers.**

```
-- This is a comment
```

This new marker can be used for "commenting out" a single line of code in a script, DDL/DML statement, stored procedure or trigger.

The logic to ignore characters is as follows:

1. Skip '--' if it is found as the first character pair following an end-of-line marker (LF on Linux/Unix, CRLF on Windows)
2. Continue skipping characters until the next end-of-line marker

This logic is NOT intended for mixing with the block comment logic ( /\* a comment \*/ ). In other words, don't use the '--' style of commenting within a block comment and don't use the block-style of commenting within a '--' line.

**INTERACTIVE ISQL SESSIONS:** Keep this in mind when working in an interactive **isql** session. **isql** will accept pieces of a statement in separate continuation segments, displaying the 'CON>' prompt until it receives the terminator symbol (normally ';'). If you type a '--' pair at the start of a continuation line, the ignoring logic will finish at the end-of-line marker that is printed to the screen or your OUTPUT file when you press Enter. There is potential for errors if you subsequently add a continuation, expecting it to be ignored.

The problem with isql arises because it has its own special commands that should be parsed only by isql. If they are not recognized due to tricky placement of "--", then they are passed to the engine. Obviously, the engine doesn't understand isql's SET and SHOW commands and rejects them.

### Alter Trigger no longer increments the change count on table

When the count of metadata changes on any single table reaches the maximum of 255, the database becomes unavailable. Backup and restore are required in order to reset the change count and make the database once again available. The intention of this feature is to enforce a database cleanup when table structures have undergone a lot of changes, not to inhibit useful capabilities in the engine.

Previously, each time a trigger was set ACTIVE|INACTIVE by an ALTER TRIGGER statement, the change count for the associated table would be incremented. This affected the usefulness of disabling and reenabling trigger code for regular operations, since it would cause the change count to rise quickly.

### ***New Reserved Words***

The following **new Firebird keywords** should be added to the list of reserved words published for InterBase 6.0.1. Those marked with asterisk (\*) are reserved for planned future use.

ABS *	BOTH *	BREAK
CASE *	CHAR_LENGTH *	CHARACTER_LENGTH *
COALESCE *	CURRENT_ROLE	CURRENT_USER
DESCRIPTOR	FIRST	IIF *
LEADING *	NULLIF *	OCTET_LENGTH *
RECREATE	SKIP	SUBSTRING
TRIM *	TRAILING *	

The following **new InterBase 6.5 keywords** (not reserved in Firebird) should also be treated as if they were reserved, for compatibility:

PERCENT	ROWS	TIES
---------	------	------

## ISQL Features

### New PLANONLY option for statements

Support for PLANONLY setting, allows for a statement/query to be submitted to the engine and the plan retrieved, without executing the statement/query. This is one of the "SET" commands that only isql understands.

```
SET PLANONLY; /* toggles the state */
SET PLANONLY ON|OFF;
```

The query is submitted to the engine and is prepared, but not executed. It is equivalent to the Prepare command of IB\_SQL and IBConsole and IB\_SQL's Prepare command which displays the plan chosen by the engine.

A thing PLANONLY can do without causing unnecessary errors is to allow testing of parameterized statements without needing to assign values:

```
SQL> SET PLANONLY ON;
SQL> select first ? fld from tbl;
```

**NOTE** :: Initially, PLAN and PLANONLY are off. (The problem with PLAN is that you need to execute the statement to know what the engine's plan)

- SET PLANONLY (or SET PLANONLY ON) will activate both PLANONLY and PLAN
- SET PLANONLY (or SET PLANONLY OFF) will deactivate PLANONLY but not PLAN. (PLAN remains turned ON).

SET => which returns a list of the current settings, now includes the PLANONLY state.

### Interactive row ADD command activated in isql (with flaws)

Using a new isql keyword ADD it is now possible to insert rows, including blobs loaded from a file, into a table interactively:

```
SQL> ADD aTable ;
```

This command adds a new record to aTable and begins prompting for values for one field at a time. It ends when you respond to a field prompt by pressing ENTER on its own.

When it sees a blob, it will offer to edit or to pump a file:

- EDIT causes it to open a temporary file and calls the editor that you have defined for BLOBVIEW. Upon saving the file you create in the editor, it takes the results and binds the newly-created blob to the record being inserted.
- If you type a string other than EDIT, isql's add\_row() function assumes it is the the name of a file and simply opens it and pumps it inside the engine, to be bound to the record being inserted.

Caveat: the addrow() function is flawed in some respects:

- After you press ENTER upon an entry, it appends a line-feed to the entry. Thus, when you press ENTER after the last field entry for a row, you need to complete the input with a Ctrl-Z in addition to the final ENTER.

- It mangles date input [ firebird-Bugs-518343 ] by not correcting the year and so makes it possible to enter invalid dates that are accepted by the engine itself. addrow() uses the tm struct for date fields but the implementor forgot that the year in this struct is 1900-based, causing the stored year to shift forward by 1900 years. For example,

```
SQL> create table t(d timestamp);
SQL> add t;
Enter data or NULL for each column. RETURN to end.
Enter D as M/D/Y>1/1/2002
Enter D as M/D/Y>
Bad date
SQL> select * from t;
D
=====
3902-01-01 00:00:00.0000
```

### isql double quotes requirement on-x and -a options relaxed

When extracting object names for an script with either option -x or -a, isql always took the easy path and put double quotes around identifiers. This was deemed annoying by several users. It will now use double quotes only if the name cannot be expressed without them.

An identifier doesn't need to be surrounded in double quotes if it contains:

- a) Only ASCII A-Z (uppercase only)
- b) 0..9 digits
- c) underscore ( \_ ) and \$ provided that in the first position, only a) is met.

The engine cannot distinguish between BLANK identifiers and zero-length identifiers. The reason is that system tables use CHAR not varchar so any trailing blank is not significant. DSQL cannot discern the difference between a field defined as blank and an output field identifier that is blank because it comes from an expression like a math operation. To prevent confusion, blank and zero-length identifiers were forbidden. FB can only generate such blank names from inline expressions in a SQL statement, because it does not create an artificial identifier for the "computed on the fly" column.

Ideally, you should use the AS keyword and explicitly name the output field. If you omit it, you are effectively creating a blank (or zero) identifier and it must be represented by a pair of double-quotes.

## API Enhancements

### Additions and Changes to *isc\_info\_database*

Three new items added:

*isc\_info\_db\_provider*, values are:

```
enum info_db_provider
{
isc_info_db_code_rdb_eln,
isc_info_db_code_rdb_vms,
isc_info_db_code_interbase,
isc_info_db_code_firebird
};
```

*isc\_info\_db\_class*, values are:

```
enum info_db_class
{
isc_info_db_class_access = 1,
....
isc_info_db_class_cache,
isc_info_db_class_classic_access,
isc_info_db_class_server_access
};
```

*isc\_info\_firebird\_version*, values are set in *inf.c*

One item renamed:

*isc\_info\_isc\_version*

which was *isc\_info\_version*.

The old name is #defined to the new, so everything works exactly as it always has. In the future we can change *isc\_info\_version* to point to *isc\_info\_firebird\_version*.

*Inf.c* has code to handle the new items.

*GDS\_VERSION* was redefined in *license.h* to reflect Firebird simultaneously being 6.2 and 1.0. Also defined was *FB\_VERSION* which we can be cut over to eventually.

The list of implementations in *utl.c* was cut over from InterBase/xxx to Firebird/xxx. The implementation numbers in *common.h* and *ibase.h* were changed to match the list of implementations in *utl.c*

### Enhancements to *isc\_database\_info* items set

*Funded by Jason Wharton, CPS (IB Objects)*

In the API, four request buffer items have been added to the *isc\_database\_info* items structure, for calling it to retrieve transaction statistics without needing to initialize and call the Services API, viz.

#### Item

65 *isc\_info\_oldest\_transaction*

```
66  isc_info_oldest_active
67  isc_info_oldest_snapshot
68  isc_info_next_transaction
```

All items return integers. Please refer to the API Guide for examples of calling `isc_database_info()`. The example found there fits perfectly, since it retrieves integer values and our new items are integers, too.

Replace the items declared in the example with:

```
char db_items[] = {
isc_info_oldest_transaction,
isc_info_oldest_active,
isc_info_oldest_snapshot,
isc_info_next_transaction,
isc_info_end};
```

The example has a loop with a `switch()`. Inside the loop, change the "case" to use our new values. There should be 4 cases plus the default one instead of the 2 cases + default in the example.

## **External Functions (UDFs)**

### **In the "standard" library, `ib_udf.dll`, `ib_udf.so`**

#### **SUBSTR( *<string expr>*, *<pos1>*, *<pos2>* )**

Returns a string consisting of the characters from *<pos1>* to *<pos2>* inclusively. If *<pos2>* is past the end of the string, the function will return all characters from *<pos1>* to the end of the string. NOTE that this behavior is different from that of the external function SUBSTR in Borland and previous Firebird versions of `ib_udf.dll` UDF library, which returns NULL when *<pos2>* is past the end of the string.

```
UPDATE ATABLE
SET COLUMNB = SUBSTR(COLUMNB, 4, 32765)
WHERE...
```

#### **SUBSTRLEN( *<string expr>*, *<pos>*, *<length>* )**

Returns a string of size *<length>* starting at *<pos>*. The length of the string will be the lesser of *<length>* or the number of characters from *<pos>* to the end of the input string.

```
UPDATE ATABLE
SET COLUMNB = SUBSTRLEN(COLUMNB, 4, 99)
WHERE...
```

#### **ascii\_char()**

Claudio Valderrama fixed an old bug in the declaration for `ascii_char` supplied in `ib_udf.sql`. It has been corrected so that it returns a one-character C string instead of (erroneously) an InterBase CHAR(1) type:

```
DECLARE EXTERNAL FUNCTION ascii_char
INTEGER
```

```
RETURNS CSTRING(1) FREE_IT
ENTRY_POINT 'IB_UDF_ascii_char' MODULE_NAME 'ib_udf';
```

## **In the new Firebird UDF library, FBUDF.dll**

*Development by Claudio Valderrama C., funded by Craig L. Leonardi*

Distributed with this release is FBUDF.dll, the first Firebird UDF library using BY DESCRIPTOR syntax to pass arguments, which provides more versatility. In declaring the functions in SQL, notice the differences between the multiple declarations that map to the same function. See, for example, that INULLIF() is mapped to by both INULLIF() and I64NULLIF().

Author Claudio Valderrama comments that the library is still in beta and warns that it will probably exhibit some bugs. So far it has not been compiled on any other platform but Windows. Bug reports and comments are welcome.

The source and the DDL for declarations are in the Firebird CVS tree. To find them, select 'Browse the CVS tree' from <http://sourceforge.net/projects/firebird>, click on 'Browse CVS Repository' and then select Developers | Latest sources | interbase | extlib | fbudf.\*.

### **NVL() functions for both exact precision ('invl') and string ('snvl') parameters**

These functions attempt to mimic the NVL function of Oracle, to output an actual value when the column has a NULL value. They take two arguments, the first being the expression being tested for NULL, the second the value to output if the first argument is NULL. NVL will return the first argument if it's not null and the second argument if the first one is null. If both are null, you get null.

The pair of parameters should be compatible, either two numeric values (smallint, int, int64) or two string values (char, varchar, cstring). The engine does not honor the parameter types when using the technique exercised by FBUDF, so mixing a numeric and a string as arguments will yield wrong results.

### **NULLIF() for string ('snullif'), integer ('inullif') and INT64 ('i64nullif') parameters**

NULLIF should take two arguments, returning NULL if they are equivalent, or the result of the first expression if they are not equivalent. Because of a shortcoming in the engine which prevents NULL being returned from a UDF, each of these three functions returns a zero-equivalent. This non-standard behaviour makes it not useful for "casting" certain values as NULL in order to have aggregate functions ignore nulls.

NOTE that the function call to both the integer and int64 functions is the same ('inullif').

**Day-of-Week functions** - one returning a short string ('SDOW'), the other a longer one ('DOW'), from a timestamp input. The return strings can be localized.

Several functions to **add segments of time to a timestamp** - 'addDay', 'AddWeek', etc.

**A RIGHT() function** (like RString in BASIC) to return the rightmost n characters from an input string.

**A GetExactTimestamp() function** returning the system timestamp with milliseconds precision.

**Truncate() and i64truncate()** truncate 32-bit and 64-bit integers respectively, taking scaled (exact-precision) numerics of any range (up to 9 in Dialect 1 or up to 19 in Dialect 3) and returning the whole-number portion. They do not work with float or double types.



```
truncate(14.76) returns 14
round(14.22) returns 14
```

**Round()** and **i64Round()** take 32-bit and 64-bit integers respectively, accepting scaled (exact-precision) numerics of any range (up to 9 in Dialect 1 or up to 19 in Dialect 3) and returning the nearest whole number. You cannot specify the number of decimal places.

```
round(14.76) returns 15
round(14.22) returns 14
```

These functions are compiled to use math rounding, i.e., always towards higher values starting at n.5. This differs from functions found in Borland tools that round to the ABSOLUTE higher value (they are symmetric).

```
round(1.5) => 2
round(-1.5) => -1
```

**String2blob()** converts a **char** or **varchar** type to a **blob**. It is like the function that exists in freeUdfLib, but is much simpler internally.

## ***New Configuration Parameters***

### Changes affecting the configuration files ibconfig/iscconfig

#### **Added -NONAGLE switch for Linux ports**

Disabling the TCP/IP Nagle Algorithm typically improves speed on slow networks.

The Nagle TCP/IP algorithm was designed to avoid problems with small packets, called tinygrams, on slow networks. The algorithm says that a TCP/IP connection can have only one outstanding small segment that has not yet been acknowledged. The definition of small varies but usually it is defined as less than the segment size which on ethernet is about 1500 bytes.

By default, the socket library will use an internal algorithm known as Nagle's algorithm for buffering bytes on write before actually sending the data in order to minimise actual physical writes.

The presence of the new switch on Linux allows developers to determine, for themselves, the possible pro's and con's of using this alternative packet handling approach.

The preset isc\_config/ibconfig is to use the Nagle algorithm:

```
#TCP_NO_NAGLE          0
```

To disable it you would uncomment this line and change it to

```
TCP_NO_NAGLE          1
```

#### **CPU\_AFFINITY Configuration Parameter**

With Firebird SuperServer on Windows, there is a problem with Windows continually swapping the server process back and forth between processors on SMP machines. This ruins performance. Until now, to set ibserver's affinity to a single CPU, it was necessary to run the server as an application and to run a utility (IB\_Affinity.exe) on top of the running server program.

This new configuration parameter can be added to `ibconfig` to remove the need for any external program to change CPU affinity on an SMP Windows system. It will take effect when the service starts up.

The CPU affinity is a config file parameter `CPU_AFFINITY`, taking one integer, that is the CPU mask.

Example:

```
CPU_AFFINITY 1
only runs on the first CPU (CPU 0).
CPU_AFFINITY 2
only runs on the second CPU (CPU 1).
CPU_AFFINITY 3
runs on both first and second CPU.
```

NOTE :: this parameter has no effect in W9x, as it uses an NT API call. W9x flavors DO NOT take advantage of multiple processors.

## Expressly Define the Location(s) of External Files

The new configuration-level setting for locating EXTERNAL FILES is specific to platform. It is currently implemented only for Windows. Here's how it works:

In the `ibconfig` file make an entry for each (existing) physical directory location where you want Firebird to search for external files. Notice that you must enclose the path in double quotes in this environment:

```
EXTERNAL_FILE_DIRECTORY "H:\test"
EXTERNAL_FILE_DIRECTORY "H:\external"
```

There is no limit on the number of directories that will be searched. Firebird will treat a series of entries as a search list.

Then -

```
isql> create table foo1 external file 'foo1.txt' (afield(char1), crlf char(2));
```

Using our example entries, Firebird will look in both `h:\test` and `h:\external` to see if the file `foo1.txt` exists. If it does, the external file table will be created and will point to the file in the right directory (although the `rdb$external_file` field in `rdb$relations` will store just `foo1.txt`, with no directory details).

When the table `foo1` is accessed, the file will be picked up automatically in the proper external file directory.

If the file doesn't exist, it gets created by default in the default (current) directory, wherever you are when `isql` or the relevant utility is run. In this case, the directory name will be stripped from the external file definition in `rdb$relations` because you have already defined `EXTERNAL_FILE_DIRECTORY`. The assumption is that, if you are not specifying a path, then you are referring to an existing file in an external file directory that has been previously defined. It will pay you in this case, therefore, to run `isql` from the directory where you want the file to be found by the database engine.

If `EXTERNAL_FILE_DIRECTORY` isn't defined in the config file, the external file gets the full path treatment as per `ISC_expand_filename`. If you do

```
isql> create table fool external file 'h:\files\fool.txt' (  
    afield(char1),  
    crlf char(2));
```

i.e. specify the directory name for the external file. If the file exists it gets defined as normal or, if doesn't exist, gets created as normal, in the directory specified.

# INSTALLATION NOTES

## Install on Windows 32

### Contents

- [Before installation](#)
- [Installing on a system with InterBase](#)
- [Installation assumptions](#)
- [Uninstallation](#)
- [Other Notes](#)

### Before Installation

#### IMPORTANT!

This installation package will try to detect if an existing version of Firebird or InterBase is installed and/or running. You must either STOP the current server and/or remove the currently installed version before continuing.

#### Stopping the Server

- If it is running as a service stop it via 'Control Panel | Services'.
- If it is an application just close it.

#### Removing an existing server

It is recommended that you uninstall a previous version of Firebird or InterBase, but it is not a requirement. See the [Uninstallation](#) section below for more details of the Firebird uninstallation routine.

#### Installing on a system with InterBase®

Firebird 1.0 cannot be run at the same time as InterBase® (any version). You must uninstall InterBase® before installing Firebird. This may change with future releases.

It is recommended that you do NOT install Firebird over an existing InterBase® install. Install it to a new directory.

#### Security database (users and passwords)

- Use **gbak** to back up your old **isc4.gdb** security database.
- Restore it under a new name.
- Stop the Firebird server.
- Rename the database to **isc4.gdb** and then restart the server.

If you have special settings in **ibconfig** you can copy and paste them into the new **ibconfig**. Don't forget that they won't take effect until you restart the server.

### Installation assumptions

- Admin rights are needed to install Firebird as a service. This requirement does not apply to Win95, Win98 or Win ME.

- If an existing, newer version of GDS32.DLL exists you will be prompted to overwrite it. It is recommended to answer YES if you are doing a server install.
- If an existing version of MSVCRT.DLL exists it is no longer updated. The installation will install it only if it does not exist on the target system.
- If certain configuration files exist in the installation directory they will be preserved. The files are
  - isc4.gdb
  - interbase.log
  - ibconfig

## Uninstallation

The Firebird uninstall routine preserves the following key files:

- isc4.gdb
- interbase.log
- ibconfig

No attempt is made to uninstall files that were not part of the original installation.

Shared files such as gds32.dll will be deleted if the share count indicates that no other application is using it.

The registry keys that were created will be removed.

## Other Notes

### Winsock2

Firebird requires WinSock2. All Win32 platforms should have this, except for Win95. A test for the Winsock2 library is made during install. If it is not found the install will fail. To find out how to go about upgrading, visit this link:

<http://support.microsoft.com/default.aspx?scid=kb;EN-US;q177719>

### Windows ME and XP

Windows ME and XP (Home and Professional editions) have a feature called System Restore, that causes auto-updating (backup caching?) of all files on the system having a ".gdb" suffix. The effect is to slow down InterBase/Firebird database access to a virtual standstill as the files are backed up every time an I/O operation occurs. (On XP there is no System Restore on the .NET Servers).

A file in the Windows directory of ME, c:\windows\system\filelist.xml, contains "protected file types". ".gdb" is named there. Charlie Caro originally recommended deleting the GDB extension from the "includes" section of this file. However, since then, it has been demonstrated that WinME might be rebuilding this list. In XP, it is not possible to edit filelist.xml at all.

On ME, the permanent workarounds suggested are one of:

- use FDB (Firebird DB) as the extension for your primary database files
- move the database to C:\My Documents, which is ignored by System Restore
- switch off System Restore entirely (consult Windows doc for instructions).

On Windows XP Home and Professional editions you can move your databases to a separate partition and set System Restore to exclude that volume.

Windows XP uses smart copy, so the overhead seen in Windows ME may be less of an issue on XP, for smaller files at least. For larger files (e.g. Firebird database files, natch!) there doesn't seem to be a better answer as long as you have ".gdb" files located in the general filesystem.

This leaves the security database isc4.gdb, which is considered writable by the code that should be simply validating a user's login, in order that isc4's header be updated for that transaction. Therefore, WinME probably makes a backup each time a user logs in.

We are trying to get an accurate problem description and a proven workaround to publish here. If you can help with the description and/or workaround, please post a message to the ib-support list or to the firebird-devel newsgroup interface at news://news.atkin.com

## Install on UNIX / Linux

(Mark O'Donohue)

The Firebird server comes in two forms, Classic which runs as a service, and SuperServer which runs as a background daemon. Although the future is likely to be SuperServer, for the user just starting out with Firebird the Classic server is likely to prove a better platform for initially experimenting with Firebird.

NOTES:

- 1) You will need to be root user to install Firebird.
- 2) For SuperServer to install correctly you will need to add localhost to your /etc/hosts.equiv file.
- 3) If you require database access from any remote machines, you will also need to add the remote machine names into the /etc/hosts.equiv file.

Super Server edition installs are as shown below, except that the install files have a SS tag rather than a CS tag.

### **For linux rpm install**

```
$rpm -Uvh FirebirdCS-1.0.0-nnn.i386.rpm
```

### **For linux .tar.gz install**

```
$tar -xzf FirebirdCS-1.0.0-nnn.tar.gz
$cd install
$./install.sh
```

\* OR FirebirdSS-1.0.0-nnn

### **What the Linux Install will do**

The Linux installations will

1. Attempt to stop any currently running server
2. If a previous installation of Firebird exists, then it and any associated files in /usr/lib /usr/include will be archived into the file /opt/interbase\_<datetimestamp>.tar.gz and will be subsequently deleted.
3. Install the software into the directory /opt/interbase and libraries into /usr/lib and header files into /usr/include
4. Automatically add gds\_db for port 3050 to /etc/services
5. Automatically add localhost.localdomain and HOSTNAME to /etc/host.equiv
6. SuperServer also installs a /etc/rc.d/init.d/firebird server start script. A new rcfirebird link is created in /usr/bin for the init.d script. This is preferred over direct execution of the Firebird initd script, because /usr/bin is on the search path for all users.
7. The /etc/rc.config Firebird entry is created on SuSE (SuSE specific configuration for service startup management).

Firebird should start automatically in runlevel 2, 3 and 5.

All Install packages are developed and tested for RH 7.x, MDK 7.x and SuSE 7.x and newer versions.

The .tar.gz packages do not support uninstallation.

**The Classic install** automatically sets up the xinetd entry if the /etc/xinetd.d directory is found, otherwise it will set up an inetd entry. Because some distributions locate xinetd in a different location than the /etc/xinetd.d directory, manual setup may be required in these conditions.

### Testing your Linux installation

To test local access for your installation:

```
$cd /opt/interbase/bin
$isql -user sysdba -password <password*>

>connect /opt/interbase/examples/employee.gdb;

>select * from sales;
>select rdb$relation_name from rdb$relations;
>help;

>quit;
```

To test remote access:

```
$cd /opt/interbase/bin
$isql -user sysdba -password <password*>

>connect '<hostname>:/opt/interbase/examples/employee.gdb';

>select * from sales;
>select rdb$relation_name from rdb$relations;
>help;

>quit;
```

\*If a password has been generated for you on installation, obtain it from the /opt/interbase/SYSDBA.file.

### Considerations for Linux

In addition to the standard install files the following three scripts are provided in the bin directory of this release:-

( Replace XX in the two scripts with CS for Firebird Classic and SS For Firebird SuperServer.)

XXchangeRunUser.sh - Create a new firebird unix user account and change the owner of the Firebird install and background tasks to run as to the firebird user.

XXrestoreRootRunUser.sh - Restore the owner of the Firebird files, and the owner user of the background tasks to the initial install default of root user.

It is STRONGLY recommended for a secure Firebird installation that the server processes do NOT be run as root.

Doing so however does place some restrictions on who can initially create Firebird databases and where they can be created.

changeDBAPassword.sh - Change the Firebird SYSDBA user password and, where necessary, change the init script /etc/rc.d/init.d/firebird with the new password as well.

//end of notes on Linux/Unix

## Install Firebird Classic & SuperServer on Solaris 2.7 Sparc

(Neil McCalden)

The release version uses 32-bit file access as previously, i.e. the 2Gb file size limit still applies. However, there is a 64-bit Superserver Release 1 version on the Downloads page of the web site and I will put up a v1.0 version as well. It is not in the main release area because of inadequate testing and feed back.

### **Basic install steps (Classic and Super Server):**

As root, extract the accompanying .tar in to the directory of your choice.

Create the links :-

```
ln -s /ExtractDirPath/interbase /usr/interbase
ln -s /usr/interbase /opt/interbase

cd /usr/interbase
./install
```

This will create links for header files, libraries and update the /etc/services and /etc/inetd.conf files.

### **Extra steps for Super Server**

Add localhost to /etc/hosts.equiv  
Create an interbas or firebird user and group  
Create script in /etc/init.d|rc3.d to start server  
- for an example see /usr/interbase/bin/firebird

If you have an earlier version of InterBase(tm) installed it is probably installed in /opt/interbase, you will need to remove the package or rename the directory as appropriate.

Note the examples files are not included in this release. They are available from the file downloads section on [firebird.sourceforge.net](http://firebird.sourceforge.net).

See README file or [www.firebirdsql.org](http://www.firebirdsql.org) for further Firebird information.

//end of notes on Solaris 2.7 Sparc



## Install Firebird Classic on MacOS X / Darwin

(John Bellardo)

Last Updated 20 February 2002

Firebird requires MacOS X 10.1 (Darwin kernel 5.2) or later. Download the Firebird build for Darwin. It is distributed as a .tar.gz archive. Most web browsers will invoke Stuffit Expander to extract the archive automatically. Make sure the OS X native version of Stuffit Expander is used. If the Classic version is used, you will have install problems.

An alternative to using Stuffit Expander to extract the archive is the following:

1. Open up terminal.
2. type "tar -zxf " and DO NOT hit return.
3. Go to the finder. Drag the FirebirdCS-1.0.0-Beta2-Darwin.tar.gz file you just downloaded and drop it into the terminal window.
4. Go back to the terminal window and press return.

To install the software you need to have administrator permissions. If you don't already have administrator permissions you can get them by:

1. go to System Preferences->Users.
2. double click your user name.
3. select the password tab.
4. Click the lock icon in the lower left corner of the window and get someone with administrator permissions to enter their password.
5. check the "Allow user to administer this computer" checkbox.

Once you have administrator permissions you can install FB by:

1. Open terminal (you can reuse an existing open terminal window if you want)
2. Type "cd " and DO NOT hit return.
3. Go to the Finder. Select the "firebird\_install" folder created from the uncompressed download. Drag this folder and drop it in the terminal window.
4. Go back to the terminal window and hit return.
5. Type "chmod a+x install" and hit return.
6. Type "./install" and hit return.
7. You may see an error to the effect of "no such process...". This can safely be ignored.

If you have problems installing or otherwise using Firebird don't suffer in silence. There are a number of good online forums that provide excellent technical support. In particular take a look at the "ib-support" group hosted at <http://groups.yahoo.com/>.

That should be it, Firebird is installed and ready to go.

The standard Firebird command line utilities are installed in /Library/Frameworks/Firebird.framework/Resources/bin. You should consider adding the directory to your path. There are no finished GUI utilities yet. All the usual Interbase/Firebird documentation applies to the MacOS X port.

For the curious users, the install script make the following changes to your computer:

1. Install the firebird framework in /Library/Frameworks
2. Install and load the system v semaphore kernel extension needed by CS. The extension is installed in /Library/StartupItems to ensure it gets loaded every time you reboot your computer.

3. Adds a firebird user to your computer. Wherever possible firebird servers and background processes are run as this user for increased security. If you install SS then the firebird user must have read/write access to all databases.
4. Adds the gds\_db entry to your /etc/services file and netinfo.
5. Installs the network database server as follows:
  - a. If installing the CS version, creates an entry in your inetd.conf file.
  - b. If installing the SS version, adds the firebird startup script to /Library/StartupItems and starts the server.

## Uninstalling

1. There is no uninstall script. To uninstall firebird do the following as root:
2. Delete /Library/Framework/Firebird.framework, /Library/StartupItems/FirebirdStartupItem, and /Library/StartupItems/SysV Semaphores.
3. comment out (or delete) the firebird entry in /etc/inetd.conf and HUP inetd (you can restart the computer in leu of HUP).
4. Delete the firebird user in System Preferences->Users (optional)
5. remove the gds\_db entry from /etc/services and netinfo (optional)
6. restart the computer. This unloads the SysV Semaphore kext. It is currently not possible to unload it without restarting the computer.

John Bellardo  
<bellardo@cs.ucsd.edu>

## Build or Install Firebird on FreeBSD

(Geoffrey Speicher, updated by Chris Knight)

Versions of FreeBSD prior to 4.4 will need to have the DES crypt libraries installed and set as the default crypt libraries. The easiest way to check this is:

```
# ls -l /usr/lib/libcrypt.so lrwxr-xr-x  1 root  wheel  14 Apr 24  2001
/usr/lib/libcrypt.so@ ->libdescrypt.so
```

If you have libcrypt.so after the ->, then you have the MD5 crypt libraries as default. You will need to run *sysinstall* to add the crypto distribution.

This can cause problems with other programs, so test this on a box that is not in a production environment. If you are unsure, installing Firebird on FreeBSD 4.4 or later is recommended, as the crypt library interface is managed automatically for you.

The recommended way is to build and install the port (as root):

```
# cd /usr/ports/databases/firebird
# make install
```

An alternative is to install the downloadable version of the package (as root) with pkg\_add:

```
# pkg_add http://prdownloads.sourceforge.net/firebird/firebird-1.0_xx.tgz
```

where xx is the version of FreeBSD the package has been built for.

E.g. for FreeBSD 4.5, you would run:

```
# pkg_add http://prdownloads.sourceforge.net/firebird/firebird-1.0_45.tgz
```

## Pure Java JCA-JDBC XA Capable Driver

The Firebird-Java team has implemented a pure Java driver for Firebird. This driver is based on both the new JCA standard for application server connections to enterprise information systems and the well known JDBC standard.

The JCA standard specifies an architecture in which an application server can cooperate with a driver so that the application server manages transactions, security, and resource pooling, and the driver supplies only the connection functionality. While similar to the JDBC 2 XADataSource idea, the JCA specification is considerably clearer on the division of responsibility between the application server and driver.

### **Implemented features**

- ❑ Most useful JDBC functionality ("useful" in the opinion of the developers).
- ❑ Complete jca spi support: may be used directly in jca-supporting application servers such as JBoss.
- ❑ XA transactions with true two-phase commit when used as a jca resource adapter in a managed environment (with a TransactionManager and jca deployment support).
- ❑ Includes optional internal connection pooling for standalone use and use in non-jca environments such as Tomcat 4.
- ❑ ObjectFactory implementation for use in environments with JNDI but no TransactionManager such as Tomcat 4.
- ❑ DataSource implementations with or without pooling.
- ❑ Driver implementation for use in legacy applications.
- ❑ Complete access to all Firebird database parameter block and transaction parameter block settings.
- ❑ Optional integrated logging through log4j.
- ❑ JMX mbean for database management (so far just database CREATE and DROP).

### **Installation**

In all cases the classes from the following packages must be available:

```
concurrent.jar  
connector.jar  
jaas.jar  
jta-spec1_0_1.jar  
log4j-core.jar
```

and either the jdbc 2 or 3 classes.

You can use the jmx management mbean either in a jmx agent (MBeanServer) or as a standalone class. So far it has been tested only in the jmxri, although since it is extremely simple it should have no problems in any jmx implementation.

For use in a managed environment as a jca resource adapter, deploy firebirdsql.rar according to the environments deployment mechanism. For instance, if using JBoss 3, put the file in the deploy directory. You will need to configure ConnectionFactories according to the requirements of the application server. An example configuration for JBoss 3 can be located through the JBoss online manual.

For all environments that do not support jca deployment, make the classes in firebirdsql.jar available to your application.

For use in a somewhat managed environment with JNDI but no jca support or transaction manager, use the FBDataSourceObjectFactory to bind a reference to a DataSource into jndi. Tomcat 4 is an example of this scenario. The JNDI implementation must support use of References/Referenceable. This will not work if the JNDI implementation only supports binding serialized objects.

For use in a standalone application with multiple connections that would benefit from connection pooling, use an instance of FBWrappingDataSource configured for pooling.

For use in a standalone application that only needs one connection, use either FBWrappingDataSource or FBDriver.

David Jencks, [d\\_jencks@users.sourceforge.net](mailto:d_jencks@users.sourceforge.net)

Support questions are best asked in the Firebird-Java email forum (mirrored to news://news.atkin.com)

[Firebird-Java@yahoogroups.com](mailto:Firebird-Java@yahoogroups.com)

## InterClient JDBC Driver

The Firebird team has fixed several bugs that were in the original Interclient 2 driver released by Borland. Firebird has built installation packages for Windows and Linux. These are available from the download areas.

We encourage you to try our pure Java driver (see above) to see if it meets your needs. Ask support questions on the [Firebird-Java@yahoogroups.com](mailto:Firebird-Java@yahoogroups.com) support and development list. When posting, please state clearly that you are asking about InterClient.

## **Further Information**

More information can be found about the Firebird database engine from:

<http://firebird.sourceforge.net>

or affiliated sites:

<http://firebirdsql.org>

<http://www.ibphoenix.com>

<http://www.cvalde.com>

If you are interested in being involved in Firebird development, or would like to raise a possible bug for discussion, please feel welcome to join our firebird-devel list. To subscribe, simply send an empty email message to:

[firebird-devel-request@lists.sourceforge.net](mailto:firebird-devel-request@lists.sourceforge.net)

with the word 'subscribe' in the Subject field.

**Please do not use the firebird-devel list for posting support questions.**

For technical support, please join the ib-support list by going to

<http://www.yahoogroups.com/groups/ib-support>

For InterClient and Java development and support, there is a specialized list:

<http://www.yahoogroups.com/groups/Firebird-Java>

The ib-support list handles technical problems with Firebird and InterBase(R). Please take your Delphi and other client development environment questions to the appropriate forum.

The open source community operates several other discussion lists on various aspects of Firebird development. For details, please refer to the Mail Lists and Newsgroups of the [Firebird community site](#).

The Firebird developers' list and the general community lists, along with some other lists of interest to Firebird and InterBase developers, are mirrored as newsgroups at

<news://news.atkin.com>

**Paid support** for Firebird can be arranged through IBPhoenix (contact addresses and numbers are at <http://www.ibphoenix.com>). Several members of the Firebird team are also available for support and consultancy. Please contact them directly.

**Database recovery services** for Firebird or InterBase databases can be handled by [IBPhoenix](#).

**Requests/offers for sponsored enhancements** to Firebird can be taken directly to the Firebird team by sending an email to [firebird-contact@lists.sourceforge.net](mailto:firebird-contact@lists.sourceforge.net).

**General discussion about FB enhancements** can be handled in the Firebird-priorities list (<http://www.yahoogroups.com/community/Firebird-priorities>).

IB-Architect (<http://www.yahoogroups.com/community/ib-architect>) is for **technical design discussions** ONLY. Support/conversion questions are definitely off-topic there.

## **Documentation**

The documentation for InterBase v 6.0 applies also to the current FireBird release. A beta version of InterBase(tm) 6 manuals is available in Adobe Acrobat format from

<http://www.ibphoenix.com/downloads/60All.zip>

A structured Documentation Index is maintained on the Firebird community site at

<http://firebird.sourceforge.net/index.php?op=doc>

This is work-in-progress and all additions are welcome - send a message to [firebird-docs@lists.sourceforge.net](mailto:firebird-docs@lists.sourceforge.net)

Some installation guidelines and other HowTos may be found in the documentation area which can be linked to from

<http://www.firebirdsql.org>

or more directly from

<http://sourceforge.net/projects/firebird>

The main repository for user and technical issues is the IBPhoenix site -

<http://www.ibphoenix.com>

Some additional documentation may be discovered by visiting the Borland techpubs area:

<http://www.borland.com/techpubs/interbase/>

## **Bugs Fixed Since Release of the Source Code**

### **SFID 492181.- Blob performance issues with JDBC**

JDBC access to BLOBs was reported to be very slow (about 50Kb/sec). It was observed that the problem could be fixed by increasing BLOB buffer size (`org.firebirdsql.jdbc.FBBlob.bufferlength`). For example, increasing it from 2Kb to 20Kb increased throughput by a factor of 9. Fixed.

### **SFID 527669.- Linux client library won't thread connections**

It was reported that the Linux client library (`gdslib.so`) did not support the ability to open/use database connections in individual application threads. Fixed for release 1. NOTE that `gdslib.so` is not yet thread-safe.

### **SFID 217138.- JOIN including a complex view kills the server**

Joining complex views whose underlying tables have indices in the fields being used for the join crashed the engine immediately. Fixed.

### **SFID 216733.- Too Many Generators Could Corrupt Database**

The number of generators you could have in IB was dependent on (page size - unknown overhead) / size of generator. You could create generators past this limit with no complaint, but these generators would return random data and corrupt the database if incremented.

Fixed.

SFID 225283. - **ORDER BY on a view makes values in fields NULL.**

The problem was fixed in Release 1.0 by Claudio Valderrama.

SFID 228135. - **NULL is returned as zero through a left join in a simple view.**

The problem was fixed in Release 1.0 by Claudio Valderrama.

SFID 518279. - **Left join causes a string truncation error.**

The problem was fixed in Release 1.0 by Claudio Valderrama.

SFID 514973. - **Sort/Temporary files > 2GB Fail**

The problem was fixed in Release 1.0 by Mike Grover.

SFID unknown. - **On Linux, gdslib.so client library is not thread-safe.**

The problem was fixed in Release 1.0.

SFID 518273. - **Numeric division on negative numbers in Dialect 3 mangles data.**

'The Klapperich Bug' - discovered February 2002 by Guido Klapperich - was inherited with the IB 6 source code. Confirmed, reported and fixed for Firebird 1 release by Claudio Valderrama.

SFID 213462. - **The 'Path String Bug' on Windows was fixed.**

InterBase had a nasty bug which was identified in IB 5 and ignored in the IB 6 Beta. It corrupts databases in a particularly horrible way.

The correct path in a connection string to a Windows server is

```
C:\patha\mydatabase.gdb
```

However, Windows will allow

```
C:patha\mydatabase.gdb
```

If two users connect, one using the first syntax and the other using the second, the server gets the wrong messages from Windows and the server thinks the two users are connecting to two different databases. The users' transactions are unaware of one another's work and data become deeply corrupted as a result.

With the fix, the first user's connection establishes an exclusive access lock on the database file, according to the path string through which she connected. Any other user attempting to connect with the alternative path string will see this error:

```
I/O error for file "C:patha\mydatabase.gdb"  
Error while trying to open file  
The process cannot access the file because it is being  
used by another process.
```

An RC1 anomaly, involving this fix and the ability to set Forced Writes on an open database, was fixed.

SFID 448062. - **CREATE DOMAIN ... CHECK (condition) left rdb\$fields.rdb\$validation\_source being CHECK (condition) but ALTER DOMAIN ... ADD CONSTRAINT CHECK (condition) leaves**

rdb\$fields.rdb\$validation\_source being CONSTRAINT CHECK (condition) that's inconsistent. Now, the first format is always stored; the extra CONSTRAINT word is always omitted.  
*Fix funded by Jason Wharton, CPS (IB Objects)*

#### SFID 227760. - **Zero length identifiers are now forbidden.**

The XSQLVAR struct used to communicate data to the client application doesn't convey information to distinguish between no ident (NULL) and blank ident. Therefore, blank field names (zero-length names) have been forbidden. Also, the server internally trims the trailing blanks, so a name like " " (only blanks) becomes a zero-length identifier and is forbidden, too. Same correction was done for both CREATE and ALTER syntax and for every different object type. In addition, several places where the engine couldn't handle embedded or leading blanks in dialect 3 identifiers (legit use) have been fixed.  
*Fix funded by Jason Wharton, CPS (IB Objects)*

#### SFID 428889. - **Column position was treated as zero-based but should be 1-based**

The syntax

```
alter table...alter column...position <n>;
```

is 1-based in Firebird. At a logical level, it is handled the same way as ORDER BY <n>, regardless of the fact that, internally, the engine uses the C convention and starts at zero.

Hence, to put some field in the first position, the command is:

```
alter table tbl
alter column c1n position 1;
```

An incompatibility now exists because Firebird has now been corrected to conform with both the standard and the IB6 documentation. To quote from EmbedSQL.pdf, chapter 5:

...

The ALTER TABLE ALTER command allows you to change the column position and name as well. For example, the following statement moves a column, EMP\_NO, from the third position to the second position in the EMPLOYEE table:

```
ALTER TABLE EMPLOYEE ALTER EMP_NO POSITION 2;
```

...

The example shown above, taken from the manual, works in Firebird. In IB it does nothing, since 2 is the third position for InterBase 6, whose syntax still has the bug of treating column position as if it were zero-based.

#### SFID 228526. - **Ambiguous JOIN statements are now rejected**

InterBase does not prevent you from submitting a statement like this:

```
SELECT A.FIELD A, B.FIELD B
FROM A JOIN B
ON FIELDX = FIELDY
WHERE FIELD A="99"
ORDER BY FIELD A
```

Such statements return unpredictable output sets. Now Firebird, in Dialect 3, will return an error if there are any unqualified column identifiers in join statements. In Dialect 1, it will return a warning but the ambiguous query will be allowed to proceed.

#### SFID 223133. - **Ambiguous self join produce bizarre results**

Also fixed.



**SFID 460261.- Blob API had problems with blanks embedded in names**

The following API calls would fail if a dialect 3 name with embedded blanks was presented:

*isc\_blob\_default\_desc*  
*isc\_blob\_lookup\_desc*  
*isc\_blob\_set\_desc*

The culprit, the function `get_name()` in `blob.e`, was fixed.

**SFID 436462.- Rows affected incorrect with BEFORE UPDATE trigger on views**

Rows Affected values now report on virtual operations, not physical operations. Beware that this works well in non-updatable views but updatable views with triggers will still increment the count by one for each physical operation.

**SFID 444463.- BEFORE triggers were firing after checks**

Before triggers (insert and update) were firing after CHECK constraints, which allowed to change fields to values that would fail table check constraints. This was wrong and could make data unrestoreable.

**SFID 229009.- CREATE VIEW not returning syntax error**

IB6 would allow CREATE VIEW command where number of view columns does not correspond to number of columns in select statement. According to SQL92 rules, such statement should return syntax error. Now it does.

**SFID 458888.- REFERENCES privileges were absurdly constrained/could cause crashes**

The bug reported was that a dependent table required REFERENCES privileges to be declared explicitly for its foreign key trigger to get permissions on the table it was referring to. This was fixed so that a table which owns an internal trigger gets implicit REFERENCES rights to the other table. The code that checks REFERENCES in "run-time" (at DML time) has been commented out and the response to the trigger's access request has been softened so that the conditions for the reported bug can not arise.

Analysis of this bug uncovered two more bugs, which have also been fixed:

- REFERENCES checking at "design-time" (DDL) was subject to being bypassed, so code was added to reinforce it even if the bypass should occur. Now it becomes mandatory for the creator of a foreign key to have REFERENCES rights on the master table or be the owner of the master table.

- fixes for the first bug also cured a bug that had been discussed but not logged. The REFERENCES checking bug was also indirectly responsible for a situation reported, where backing up and restoring a complex database would cause a crash immediately upon selecting from virtually any table. The analysis of the bug, which was complicated, uncovered a "deadly circle" of re-entrant REFERENCES checking.

**SFID 446237.- 'Column not found' error occurred where it should not**

Eliminated. It arose when a view was based on a table whose fields were dialect 3 identifiers with embedded blanks. Although this may be the only bug documented, it is one of the several bugs with embedded blanks that were fixed.

**SFID 229860.- Wrong error message**

The message ' DATE data type is now called TIMESTAMP' was being returned on unrelated errors, such as typos. Fixed.

#### SFID 460624.- **isql extraction of procedure parameters was broken in Dialect 3**

Fixed a bug in isql where it would not extract procedure parameters' names correctly in dialect 3, even if the names required double quotes and these were supplied. An incorrect parameter declaration in the routine that does the double quotes was also fixed.

#### SFID 451798.- **FIRST is applied before aggregation**

#### SFID 451810.- **SKIP is off by 1**

Both fixed.

#### SFID 412417.- **Error altering from CHAR to VARCHAR**

Altering CHAR to VARCHAR column was adding 2 bytes to field length. In addition, it was possible to alter a CHAR to VARCHAR of lesser size, causing possible string truncation error messages when selecting from the damaged table.

Fixed.

#### SFID 231998.- **space before CASTed numeric expression in dialect 1**

Eliminated.

#### SFID 212177.- **error with non-english column defaults**

GBAK would raise a transliteration error during restore with defaults stored in a database that was using a non-English character set.

#### SFID 221589.- **numeric fields and mathematical operations**

```
select field1 * field2 from mytable
or
select field1 * (1+field2/100) from mytable
```

where both fields were numeric (9,2) produced incorrect results.

#### SFID 450301.- **SUBSTRING did not work**

The new function was not working when used in a where-clause (e.g. with in or =) or in string-concatenation. Fixed.

#### SFID 223059.- **Updating VARCHAR did not clear old data**

When IB 6 updates the string stored in a VARCHAR, it does not zero rest of the string but concatenates the old value to the update value. Because VARCHARs contain length of the string, the client application will never notice any problem (i.e. it will always see correct result), but the gdb file can grow faster than expected (because such additional data can't be rle compressed), and database can get slower (because less useful data fits onto page).

Fixed in Firebird.

#### SFID 223512.- **DROP VIEW was causing the underlying table to be dropped**

It means tables and views share the same namespace but DROP VIEW only should work if there's a view with that name. If there's a table of the same name, the command should behave as if it doesn't exist. Fixed.

SFID 226456. - **SELECT/PLAN did not understand delimited SQL index names**

Fixed.

SFID 419964. - **buffer overflow in remote/interface.c li**

This was caused by the version string buffer being only 64 bytes long. It has now been increased.

SFID 453686. - **Unable to create a dialect 3 database with Firebird 1 betas**

This problem has been occurring using clients (e.g. IBConsole and IBExpert) that need to connect to a major version number greater than 5. It is fixed in Firebird RC 1.

SFID 233124. - **Connection lost during execution of bad code**

Eliminated.

SFID 425799. - **Renamed domain left behind dimensions**

With a domain being an array, renaming the domain caused rdb\$field\_dimensions to be left unchanged; hence the connection between a domain and its dimensions specification was broken.

Fixed.

SFID 450405. - **Tricky role could defeat basic SQL security**

It was a convoluted example: but the loophole has now been closed.

SFID 462800. - **Non-unique pair in RDB\$FORMATS**

Fixed.

SFID 447377. - **GDS error ...can't find TIP**

There is a bug in InterBase 5.6, 6.01, and early Firebird betas that causes the lookup of a transaction inventory page to fail if there are more than 32767 transaction pages. That makes the maximum safe transaction id for a database with:

1024 byte pages	131,596,287.
2048 byte pages	265,814,016.
4096 byte pages	534,249,472.
8192 byte pages	1,071,120,384.

Although those are large numbers, there was a particular database exceeded 131 million transactions in six months. Attempts to attach to the database failed with the error gds internal consistency check, can't find TIP.

Suggestions:

- 1) don't use a 1024 byte page size.
- 2) do check your next transaction number from time to time.
- 3) if you see the next transaction number approaching the limit, backup and restore the database.

SFID 229231. - **REVOKE is sensitive about the case of user names**

For example:

```
grant all on config to "admin";  
revoke all on config from "admin";
```

You would think this would result in admin having no privilege on config but a select on the RDB\$USER\_PRIVILEGE table shows the "admin" user still has full access! In order to remove the privileges you needed to do:

```
revoke all on config from "ADMIN"
```

In addition, we killed an ambiguity that existed in IB since v4:

```
grant update on tbl to <unknown>
```

Who's unknown, a role or a user? If a role exists, the role is granted the right. Otherwise, the user is granted the right. Even if you specify the USER keyword, it does nothing in IB. However, in Firebird, the above statement is still ambiguous because Firebird permits you to specify whether it is the ROLE or the USER to which the GRANT is to be made:

- to USER unknown will pick only a USER, even if a ROLE exists.
- to ROLE unknown will pick only a ROLE. Error if ROLE doesn't exist

There is still an open bug (SFID 223128) that allows SYSDBA to grant to non-existent roles.

A case logged by Ivan Prenosil, where isql couldn't copy with the "grant ... TO GROUP UNIX\_group" syntax was also fixed.

#### SFID 421260.- **Character length not filled for UDFs**

In the case of ODS 10, it was written only as a stub. Now completed.

#### SFID 227375.- **Grouping on derived fields processing NULL data kills InterBase**

This bug occurred when a SELECT on a view that calculated a derived column's value by subtracting one value from another attempted to GROUP BY this derived column. It now works properly in Firebird.

#### SFID 425949.- **Engine CRASH Error**

This statement

```
select count(*),adresy.rdb$db_key from adresy
```

can crash InterBase. adresy can be any table. Fix copied to Firebird from the Borland tree.

#### SFID 228467.- **Security bug with a hardcoded user with full rights to isc4.gdb**

This was the security vulnerability that was discovered to affect all versions of InterBase from Version 4.x forward. It was fixed in Firebird and also later by Borland in InterBase 6 source code and binary versions after January 2001.

#### SFID 229121.- **TEMP directory filling up**

Fixed.

#### SFID 213708 -**502 Declared cursor already exists**

Occurred in Microfocus programs connecting locally to IB6 Classic on AIX and remotely to IB6 Superserver on WINNT. Fixed.

**SFID 214298.- Select count(\*) expression anomaly when table was empty**

select count(\*) + 1 returned zero when table had no rows but returned correct result if count(\*) was greater than 0. Fixed.

**SFID 216579.- Generators in COMPUTED BY columns would return wrong results**

Using generators in COMPUTED BY columns would return wrong results and render the database unusable. Fixed.

**SFID 222476.- AVG and SUM returned empty field names in dialect 3**

Fixed. Same problem has also been fixed for CAST and UPPER.

**SFID 227717.- COBOL programs randomly return a -901 request sync. error**

COBOL programs randomly return a -901 request synchronization error. Alternative title: COBOL programs lose SQLCODE values during UPDATE:

```
UPDATE SET ... WHERE x=..
```

('Mass Update') error codes returned by the update are not returned to the program, instead the program will see a -901 request synchronization error.

The error was caused by bad code generated by GPRE. Fixed.

**SFID 212328.- IB Guardian leaked handles**

Fixed. However, it still creates more and more threads every time FB crashes until it is either shut down or its icon is double-clicked in the tray. This bug only happens with the Guardian running as an app because as a service, it does not put an icon in the tray.

**SFID 212263.- command line isql ignores -user / -password with -a or -x**

Fixed.

**SFID 421262.- ISQL reports UDF BLOB parameter BY VALUE**

The BLOB UDFs in ibudf have been fixed in the Firebird distribution.

**SFID 222563.- isql extracts wrong stored proc parameters with UNICODE**

ISQL was reading rdb\$field\_length instead of rdb\$character\_length for procedure's parameters. In the case of table's fields, the correct information is read and presented. The engine itself was doing the right thing.

Fixed.

**SFID 223516.- Missing types in rdb\$types**

Several, but not all proposed undocumented types were added.

**SFID 451944.- Trigger Activate/Deactivate increases meta counter**

Fixed.

## **Known Issues**

### **Pending**

#### **Allow FK indexes to be disabled**

An enhancement making it possible to set FOREIGN KEY indexes inactive, using ALTER INDEX, to resolve some performance issues related to maintaining indexes with low selectivity, has been deferred pending resolution of related effects.

SFID 446227. - **On Linux, gdslib.so client library is not thread-safe.**

A threadable gdslib.so was included in release 1.0 but the thread-safety issues are not resolved.

### **Core Engine**

#### **SFID**

479483 Bad treatment of FIRST/SKIP in subselect  
224810 DISTINCT propagates outside a VIEW  
442140 Grant for Roles on Views not working  
222376 Horrible plan with a lot of OR conditions  
223514 IB crashes with two procedures intermixed.  
217042 IB doesn't validate weird constructions  
419065 Join on different datatypes  
223058 Multi-hop server ability broken Confirmed Bug  
219525 No current record for fetch operation  
221921 ORDER BY has no effect  
213460 Registering Events w/certain configuration crashes IB Server  
233025 Server hangs when executing Stored Proc more than once  
221649 Unique index allowed on NULLABLE field  
211781 Win32: Server don't close thread handles  
233644 cannot specify PLAN in UPDATE statement

#### **DSQL**

#### **GBAK**

228431 gbak cannot restore backup made by IB v5

#### **GPRE**

416228 gpre generates invalid isc\_vtov calls  
223513 Ambiguity between tables and views in isql's SHOW commands.  
223126 Misplaced collation when extracting metadata with isql  
225219 isql -a: wrong order with domains based on table's fields  
450404 isql uppercases role in the command line

#### **InterClient**

227414 Sometimes Interbase/Interserver won't grant connections

#### **Security Issues**

229237 Blank passwords poorly supported  
229894 Client program can log on as any user.

222375 Grants overwrite previous rdb\$security\_classes entries