Firebird™ Version 1.5.1

Point Release Notes v.1.5.1

14 July 2004

IMPORTANT :: These are point release notes that augment the release notes for v.1.5. Amendments to the v.1.5 Release Notes appear at the end of this document.

Point release additions

Introducing NPTL builds for high Linuxen

(Alex Peshkoff)

Firebird Superserver has a link-time backward compatibility issue with the NPTL (Native POSIX Thread Library) that may cause it to be unstable on Linux distributions that enable the NPTL in the GNU C library, e.g. Red Hat 9, Mandrake 10, Fedora Core. The new NPTL builds of Superserver should solve these problems.

Services API is now supported on Classic

(Nickolay Samofatov)

Features of GSEC and GSTAT are now supported via the Services API in CS builds. It means that the entire Services API now works in both Superserver and Classic architectures.
Known issue: GSEC doesn't deliver error status vectors to the client side if forked from the CS process. Any error will prevent the security database from being changed; but the exception is decoded by the server and the appropriate message is delivered via the API communication buffer instead of the status vector. If this situation is not handled properly by the user program, it may cause application-specific errors. For example, IBExpert displays the message "unexpected output buffer value".

GSTAT can now connect to localhost

(Dmitry Yemanov)

GSTAT supported only the local connection string syntax: you could not specify "localhost:<path>" to retrieve the statistics for a local database. Since Win32 Classic does not yet support the local (IPC) protocol, it wasn't possible to use GSTAT with it. Now GSTAT is fixed to enable it. You may use either "localhost:" (for TCP/IP) or "\\.\" (for Named Pipes) to work around the Win32 Classic limitation.

Note that GSTAT still can not be run over remote databases. Because it has to open a database file directly to read the header page, it requires local access to the database file. This might change in a future version.

Character set NONE data now accepted "as is"

(J. Beesley, Nickolay Samofatov)

Changes were made in the engine to make the character set NONE more friendly about reading and writing data from and to fields (columns, variables) of another character set.

In Firebird 1.5.0, from a client connected with character set NONE, you could read data in two incompatible character sets—such as SJIS (Japanese) and WIN1251(Russian)—even though you could not read one of those character sets while connected from a client with the other character set. Data would be received "as is" and stored without raising an exception.

However, from this character set NONE client connection, an attempt to update any Russian or Japanese data columns using either parameterized queries or literal strings without introducer syntax would fail with transliteration errors; and subsequent queries on the stored "NONE" data would similarly fail.

In Firebird 1.5.1, both problems have been circumvented. Data received from the client in character set NONE are still stored "as is" but what is stored is an exact, binary copy of the received string. In the reverse case, when stored data are read into this client from columns with specific character sets, there will be no transliteration error. When the connection character set is NONE, no attempt is made in either case to resolve the string to well-formed characters, so neither the write nor the read will throw a transliteration error.

This opens the possibility for working with data from multiple character sets in a single database, as long as the connection character set is NONE. The client has full responsibility for submitting strings in the appropriate character set and converting strings returned by the engine, as needed. Abstraction layers that have to manage this can read the low byte of the sqlsubtype field in the XSQLVAR structure, which contains the character set identifier.

While character set NONE literals are accepted and implicitly stored in the character set of their context, the use of introducer syntax to coerce the character sets of literals is highly recommended when the application is handling literals in a mixture of character sets. This should avoid the string's being misinterpreted when the application shifts the context for literal usage to a different character set.

NOTE Coercion of the character set, using the introducer syntax or casting, is still required when handling heterogeneous character sets from a client context that is anything but NONE.

```
  _ISO8859_1 'ààààà'
```

or

```
  CAST (<string> as varchar(n) character set ISO8859_1)
```

Optional core dump on exceptions

(Nickolay Samofatov)

A debugging enhancement was added, to configure the server to abort a server process and produce a core dump when bugchecks or structured exceptions occur. The new parameter in firebird.conf is BugcheckAbort. It is off by default.

If turned on, this feature will produce a correct core dump on BUGCHECK or when an external function (UDF, BLOB filter, intl2 function) causes havoc. When BugcheckAbort is not enabled, structured exception handlers or a synchronous signal handler may mask the original cause of problem.

New collation added for Lithuanian language

(Jonas Jasas Jr)

Collation sequence LT_LT was added for the ISO8859_13 charset.

Small Win32 installation utility enhancement

(Olivier Mascia)

The Win32 service installer now adds a description string to the services configuration info.

Release 1.5 bugs fixed

Modification of /etc/init.d/firebird during Linux installation now works properly.
(Alex Peshkoff)

The wrong soname for libib_util.so (SF #750659) caused warnings in ldconfig and also meant that it could not be loaded automatically by external function libraries using calls to ib_util_malloc(). That made such libraries unusable, unless a PRELOAD environment setting was added for fbserver/fb_inet_server. Now fixed.
(Nickolay Samofatov)

Memory corruptions in the embedded DSQL API implementation were corrected.
(Nickolay Samofatov)

The following two optimizer bugs were fixed:
(Arno Brinkman)

1) IS NULL is now handled correctly when a view is used in an outer join.
2) MERGE is used instead of JOIN when indices are not applicable for a join.

'Extract' bug in ISQL was fixed: previously, procedure parameters would be extracted incorrectly.
(Dmitry Sibiryakov)

Mapping of COUNT(*) in a HAVING clause when used with IN, ANY/SOME, ALL has been corrected.
(Arno Brinkman)

NULLS FIRST ordering parameter now works in union queries (bug SF #918653).
(Nickolay Samofatov)

The algorithm for dependency tracking that was introduced into gbak in v.1.5, made restoring a database with many interdependent procedures too slow. It has been replaced by a faster algorithm.

(Nickolay Samofatov)

Fixed a memory leak in EXECUTE STATEMENT...INTO.
(Alex Peshkoff)

A serious memory leak was fixed in the Windows control panel applet. If the applet window was left open for several hours it would eventually trigger a low memory condition on the server.
(Paul Reeves)

Switching the server status via the control panel applet from certain configurations would lead to inconsistencies on Win XP and Win 2003. There were timing issues that were only apparent on these platforms. The problem was fixed.
(Paul Reeves)

A fresh implementation of the editline facility in isql was done to resolve a problem with isql crashing on some platforms.
(Nickolay Samofatov)

Fixed a bug in the QLI utility, whereby assigning a variable would generate an access violation.
(Vladimir Tsvigun)

On Windows, server startup was failing to look for the FIREBIRD environment variable when trying to locate the root directory. Fixed.
(Alex Peshkoff)

DOC for Embedded Server (README_embedded.txt) was updated slightly to reduce confusion about the client connection string format.
(Helen Borrie)

Old bugs fixed

"SF" bug items refer to the Firebird bugtracker on SourceForce. "QC" bugs refer to the InterBase section of the bug tracker on Borland's Quality Central.

Events bug fixed

Fixed a persistent issue with events, where a client using events would cause the server's CPU usage to go up to 100 percent when it terminated its connection. The problem exists in all Firebird versions, but became more acute in v. 1.5 because it has the default dummy packet interval at 0 and relies instead on TCP/IP keepalive packets. Now, the client library tries to wake up the server's port when detaching, allowing the server to scan for broken connections and close them sooner.
(Dmitry Yemanov)

gbak bugs fixed

The –n[o_validity], that should make gbak restore a database without restoring validity constraints, was ignoring NOT NULL constraints and restoring them regardless. Now it works as it ought to.
(Claudio Valderrama)

When a fresh database needs to be created by metadata-backup/restore, the user-defined generators should be set to zero. However, during a metadata-only backup, gbak would preserve the values of all generators, including non-system ones (bug SF #750659). Now, they are correctly reset.
(Claudio Valderrama)

Corrected a bug that prevented proper restoration of inter-dependent views.
(Arno Brinkman)

Memory problems fixed

A memory leak in the Services API.
(Dmitry Yemanov)

A memory leak in isc_database_info().
(Nickolay Samofatov)

Memory leaks occurred in PSQL modules when aggregation errors were handled in a WHEN block (bug QC #7496).
(Oleg Loa)

A stability problem with usage of previously freed memory in the client library was found and eliminated. It was present in InterBase and earlier Firebird versions but it re-surfaced in Firebird 1.5, due to the memory manager actively returning memory to the system (bug SF #919246). When using TCP/IP with Windows clients, application programs would hang because of an access violation in fbclient.dll, or fbclient.dll itself would hang. In some cases, further connections to the same database would be rejected.
(Nickolay Samofatov)

Generator bugs fixed

1) Attempting to update a generator in a read-only database would cause database detach to be prolonged
2) Such a situation would leave the database locked to other connections.

Fixed the broken CCH finalization sequence and a page lock leak.
(Nickolay Samofatov)

3) Generator pages were not flushed properly, causing generator value changes to be lost after a server failure, even in Forced Writes mode

(Oleg Loa, Vlad Horsun, Dmitry Yemanov)

Old tracking logic bug fixed

A nested query that contained a variable in the WHERE clause would be erroneously flagged as invariant (SF #627057 and #922602), e.g.

```
select max(d2.id) from demo d2
  where d2.id < :id
```

Request cloning logic was broken. Clones of procedures/triggers were not accounting for invariants and the requests opened by them at all. Thus, they were inheriting invariant values from previous executions and were not freeing resources (highly limited! - you can have only 1000 copies of a request open). This is why most recursive procedures didn't work at all and using procedures from multiple Superserver connections produced results that were inconsistent or timing-dependent.

Invariant dependency tracking was not working properly. The engine now keeps account of which variables an invariant depends on and clears the cached invariant value when values are being assigned to these variables.
(Nickolay Samofatov)

Win32 lock manager bugs solved

A number of problems with the win32 lock manager, that manifested themselves as low performance when several operations were waiting on locks simultaneously, were solved.
(Nickolay Samofatov)

Language bugs fixed

Expression evaluation was not supported in LEFT JOIN (SF bug #784121).
(Claudio Valderrama)

Server would crash when views and selectable procedures were intermixed.
(Dmitry Yemanov)

Fixed an obscure bug where the server would crash if LIKE ESCAPE <symbol> was asked to operate on NULL. LIKE syntax is:

```
<string> LIKE <string> [ESCAPE <one-char-string>]
```

A request such as

```
select * from rdb$database
  where 'a' like 'a' escape cast(null as char(1))
```

would crash the engine. It now complies with the SQL standard and returns an empty result set from such a query.
(Nickolay Samofatov)

UDF bugs fixed

Length of string types containing binary data (OCTETS) was being determined incorrectly in UDFs.
(Fred Polizo Jr)

Known Issues

PLANs for selectable procedures having multiple FOR loops are reported in the wrong order.
Example (from QMDB test):

```
create procedure proc1
returns (a integer)
as
begin
  for select a from table1 into :a do
    suspend;
  for select b from table2 into :a do
    suspend;
end ^^
...
select * from proc1;
```

1.5.1 produces the PLAN:

```
PLAN (TABLE2 NATURAL) (TABLE1 NATURAL)
```

instead of

```
PLAN (TABLE1 NATURAL) (TABLE2 NATURAL)
```

Amendments to v.1.5 Release Notes

Pages 23-24, New context variables - **INSERTING/UPDATING/DELETING**

```
IF (INSERTING OR DELETING) THEN
  NEW.ID = GEN_ID(G_GENERATOR_1, 1);
```

Explanation: although it's possible to reference the NEW context in the INSERT OR DELETE trigger, the above assignment will cause an error (cannot assign to read-only column) for the DELETE action.

Page 25, **LEAVE/BREAK**

(i)

```
WHILE (<condition>) DO
BEGIN
  <statements>;
  IF (<conditions>) THEN
    LEAVE;
  <statements>;
END
```

The condition that branches to a LEAVE statement must be inside a block that is controlled by a looping construct (i.e., WHILE or FOR SELECT...INTO...DO).

It is emphasised that LEAVE will not terminate other types of BEGIN...END block.

Page 27,

Added in 1.0

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

This is not correct. Sandor's collation provides a specific dictionary sort, but it is not case-insensitive.

From p. 54, some **typos in the Linux installation notes**, with corrections provided by "zinfabbe" (Tom ?):

```
isql -user sysdba -password <password*>
```

should be

```
./isql -user sysdba -password <password*>
```

```
SQL>connect localhost:employee.fdb /* this is an aliased path */
```

should be

```
SQL>connect localhost:employee.fdb; /* this is an aliased path */
```

```
$./createDBAlias.sh test.fdb /var/firebird/test.fdb
(Usage is: createDBAlias.sh <dbname> <pathtodb>)
...
```

As an alternative (for step 2) the steps in the /createDBAlias.sh script can be performed manually ...

should be

```
$./createAliasDB.sh test.fdb /var/firebird/test.fdb
 (Usage is: createAliasDB.sh <dbname> <pathtodb>)
```

As an alternative (for step 2) the steps in the createAliasDB.sh/ script can be performed manually ..

In 3 places :

```
$/opt/firebird/isql -u sysdba -p <password*>
```

should be

```
$/opt/firebird/bin/isql -u sysdba -p <password*>
```