Known compatibility issues

D. Yemanov

This chapter is intended as a set of alerts to those who are migrating Firebird 1.0 or 1.5 databases to Firebird 2.0. It should be studied before attempting to install any servers.

The FIREBIRD variable

FIREBIRD is an optional environment variable that provides a system-level pointer to the root directory of the Firebird installation. If it exists, it is available everywhere in the scope for which the variable was defined.

The FIREBIRD variable is *NOT* removed by scripted uninstalls and it is not updated by the installer scripts. If you leave it defined to point to the root directory of a v.1.5.x installation, there will be situations where the Firebird engine, command-line tools, cron scripts, batch files, installers, etc., will not work as expected.

If the Windows installer program finds a value for **%FIREBIRD%** it will make that path the default location that it offers, instead of c:\Program Files\Firebird\Firebird_2_0.

Unless you are very clear about the effects of having a wrong value in this variable, you should remove or update it before you begin installing Firebird 2.0. After doing so, you should also check that the old value is no longer visible in the workspace where you are installing Firebird–use the SET FIREBIRD command in a Windows shell or printenv FIREBIRD in a POSIX shell.

Security in Firebird 2 (all platforms)

Be aware of the following changes that introduce incompatibilities with how your existing applications interface with Firebird's security:

Direct connections to the security database are no longer allowed

Apart from the enhancement this offers to server security, it also isolates the mechanisms of authentication from the implementation.

- User accounts can now be configured only by using the Services API or the gsec utility.
- For backing up the security database, the Services API is now the only route. You can employ the -se[rvice] hostname:service_mgr switch when invoking the gbak utility for this purpose.

Non-SYSDBA users no longer can see other users' accounts in the security database

A non-privileged user can retrieve or modify only its own account and it can change its own password.

Remote attachments to the server without a login and password are now prohibited

- For attachments to Superserver, even root trying to connect locally without localhost: in the

database file string, will be rejected by the remote interface if a correct login is not supplied.
- Embedded access without login/password works fine. On Windows, authentication is bypassed. On POSIX, the Unix user name is used to validate access to database files.

The security database is renamed to security2.fdb

If you upgrade an existing installation, be sure to upgrade the security database using the provided script in order to keep your existing user logins.

Before you begin the necessary alterations to commission an existing security database on the Firebird 2.0 server, you should create a gbak backup of your old security.fdb (from v.1.5) or isc4.gdb (from v.1.0) using the old server's version of gbak and then restore it using the Firebird 2.0 gbak.

Important: You must make sure that you restore the security database to have a page size of at least 4 Kb. The new security2.fdb will not work with a smaller page size.

Warning: A simple 'cp security.fdb security2.fdb' will make it impossible to attach to the firebird server !

For more details see the notes in the chapter on security, New security features. Also read the file security_database.txt in the upgrade directory beneath the root directory of your installation.

back to top of page

SQL migration issues

DDL

Views made updatable via triggers no longer perform direct table operations

In former versions, a naturally updatable view with triggers passed the DML operation to the underlying table and executed the triggers as well. The result was that, if you followed the official documentation and used triggers to perform a table update (inserted to, updated or deleted from the underlying table), the operation was done twice: once executing the view's trigger code and again executing the table's trigger code. This situation caused performance problems or exceptions, particularly if blobs were involved.

Now, if you define triggers for a naturally updatable view, it becomes effectively like a non-updatable view that has triggers to make it updatable, in that a DML request has to be defined on the view to make the operation on the underlying table happen, viz:

1. If the view's triggers define a DML operation on the underlying table, the operation in question is executed once and the table triggers will operate on the outcome of the view's triggers.
2. If the view's triggers do not define any DML request on the underlying table then no DML operation will take place in that table.

Important: Some existing code may depend on the assumption that requesting a DML operation on an updatable view with triggers defined would cause the said operation to occur automatically, as it does for an updatable view with no triggers. For example, this “feature” might have been used as a quick

way to write records to a log table en route to the “real” update. Now, it will be necessary to adjust your view trigger code in order to make the update happen at all.

New reserved words (keywords)

A number of new reserved keywords are introduced. The full list is available in the chapter New reserved words and changes and also in Firebird's CVS tree in /doc/sql.extentions/README.keywords. You must ensure that your DSQL statements and procedure/trigger sources do not contain those keywords as identifiers.

Note: In a Dialect 3 database, such identifiers can be redefined using the same words, as long as the identifiers are enclosed in double-quotes. In a Dialect 1 database there is no way to retain them: they must be redefined with new, legal words.

CHECK constraint change

Formerly, CHECK constraints were not SQL standard-compliant in regard to the handling of NULL. For example, CHECK (DEPTNO IN (10, 20, 30)) should allow NULL in the DEPTNO column but it did not.

In Firebird 2.0, if you need to make NULL invalid in a CHECK constraint, you must do so explicitly by extending the constraint. Using the example above:

```
CHECK (DEPTNO IN (10, 20, 30) AND DEPTNO IS NOT NULL)
```

See also:

DDL - Data Definition Language

back to top of page

DML

Changed ambiguity rules in SQL

A. Brinkman

In summary, the changes are:

1. When an alias is present for a table, that alias, and not the table identifier, must be used to qualify columns; or no alias is used. Use of an alias makes it invalid to use the table identifier to qualify a column.
2. Columns can now be used without qualifiers in a higher scope level. The current scope level is checked first and ambiguous field checking is done at scope level.

Examples

a) 1. When an alias is present it must be used or no alias at all must be used.

This query was allowed in FB1.5 and earlier versions:

```
SELECT
```

```
   RDB$RELATIONS.RDB$RELATION_NAME
FROM RDB$RELATIONS R
```

Now, the engine will correctly report an error that the field RDB$RELATIONS.RDB $RELATION_NAME could not be found.

Use this (preferred):

```
SELECT
   R.RDB$RELATION_NAME
FROM RDB$RELATIONS R
```

or this statement:

```
SELECT
   RDB$RELATION_NAME
FROM
   RDB$RELATIONS R
```

a) 2. The next statement will now use the appropriate FieldID correctly from the subquery and from the updating table:

```
UPDATE TableA
   SET
      FieldA = (SELECT SUM(A.FieldB) FROM TableA A
      WHERE A.FieldID = TableA.FieldID)
```

Note: Although it is possible in Firebird to provide an alias in an update statement, many other database vendors do not support it. These SQL statement syntaxes provide better interchangeability with other SQL database products.

a) 3. This example ran incorrectly in Firebird 1.5 and earlier:

```
SELECT
   RDB$RELATIONS.RDB$RELATION_NAME,
   R2.RDB$RELATION_NAME
FROM RDB$RELATIONS
JOIN RDB$RELATIONS R2 ON
   (R2.RDB$RELATION_NAME = RDB$RELATIONS.RDB$RELATION_NAME)
```

If RDB$RELATIONS contained 90 rows, it would return 90 * 90 = 8100 rows, but in Firebird 2.0 it will correctly return 90 rows.

b) 1. This would fail in Firebird 1.5, but is possible in Firebird 2.0:

```
SELECT
   (SELECT RDB$RELATION_NAME FROM RDB$DATABASE)
FROM RDB$RELATIONS
```

b) 2. Ambiguity checking in subqueries

This would run on Firebird 1.5 without reporting an ambiguity, but will report it in Firebird 2.0:

```
SELECT
   (SELECT FIRST 1 RDB$RELATION_NAME
   FROM RDB$RELATIONS R1
   JOIN RDB$RELATIONS R2 ON
      (R2.RDB$RELATION_NAME = R1.RDB$RELATION_NAME))
```

Multiple hits to same column now illegal

It is no longer allowed to make multiple “hits” on the same column in an INSERT or UPDATE statement.

Thus, a statement like

```
INSERT INTO T(A, B, A) ...
```

or

```
UPDATE T SET A = x, B = y, A = z
```

will be rejected in Firebird 2.n, even though it was tolerated in InterBase and previous Firebird versions.

Query plans

Stricter validation of user-specified plans

User-specified plans are validated more strictly than they were formerly. If you encounter an exception related to plans, e.g. Table T is not referenced in plan, it will be necessary to inspect your procedure and trigger sources and adjust the plans to make them semantically correct.

Important: Such errors could also show up during the restoreRestore Database process when you are migrating databases to the new version. It will be necessary to correct these conditions in original database before you attempt to perform a backup/restore cycle.

Plan must refer to all tables in query

Using a plan without a reference to all tables in query is now illegal and will cause an exception. Some previous versions would accept plans with missing references, but it was a bug.

See also:

DML - Data Manipulation Language

back to top of page

PSQL

Restrictions on assignment to context variables in triggers

- Assignments to the OLD context variables are now prohibited for every kind of trigger.
- Assignments to NEW context variables in AFTER-triggers are also prohibited.

Tip: If you get an unexpected error Cannot update a read-only column then violation of one of these restrictions will be the source of the exception.

Reference to current of <cursor> outside scope of loop

In Firebird 1.5 and earlier, referring to current of <cursor> outside the scope of the cursor loop was accepted by the PSQL parser, allowing the likelihood of run-time occurring as a result. Now, it will be rejected in the procedure or trigger definition.

NULLS are now “lowest” for SORTS

NULL is now treated as the lowest possible value for ordering purposes and sets ordered on nullable criteria are sorted accordingly. Thus: .

- for ascending sorts NULLs are placed at the beginning of the result set,
- for descending sorts NULLs are placed at the end of the result set.

Important: In former versions, NULLs were always at the end. If you have client code or PSQL definitions that rely on the legacy NULLs placement, it will be necessary to use the NULLS LAST option in your ORDER BY clauses for ascending sorts.

CURRENT_TIMESTAMP now returns milliseconds by default

The context variable CURRENT_TIMESTAMP now returns milliseconds by default, while it truncated sub-seconds back to seconds in former versions. If you need to continue receiving the truncated value, you will now need to specify the required accuracy explicitly, i.e. specify CURRENT_TIMESTAMP(0).

ORDER BY <ordinal-number> now causes SELECT * expansion

When columns are referred to by the ordinal number (degree) in an ORDER BY clause, when the output list uses SELECT * FROM ... syntax, the column list will be expanded and taken into account when determining which column the number refers to.

This means that, now, SELECT T1.*, T2.COL FROM T1, T2 ORDER BY 2 sorts on the second column of table T1, while the previous versions sorted on T2.COL.

Tip: This change makes it possible to specify queries like SELECT * FROM TAB ORDER BY 5.

back to top of page

Configuration parameters

Configuration parameter DeadThreadsCollection is deprecated

The parameter DeadThreadsCollection for Superserver in firebird.conf is deprecated and will be ignored if set. Firebird version 2 efficiently cleans up dead threads straight away.

Command-line tools

Change to gbak -R semantics

An important change has been done to prevent accidental database overwrites as the result of users mistakenly treating -R as an abbreviation for restore. gbak -R was formerly a shortcut for -REPLACE_DATABASE. Now the -R switch no longer restores a database by overwriting an existing one, but instead reports an error.

If you actually want the former behaviour, you have two alternatives:

- Specify the full syntax gbak -REPLACE_DATABASE. There is a new shortcut for the -REPLACE_DATABASE switch: gbak -REP

or

- Use the new command -R[ECREATE_DATABASE] OVERWRITE. The -R shortcut now represents the -R[ECREATE_DATABASE] switch and the OVERWRITE keyword must be present in either the full or the abbreviated form.

Warning: If you use the full syntax, you are expected to know what this restore mode actually means and have some recovery strategy available if the backup subsequently turns out to be unrestorable.

See also:

- gbak
- gbak backup/porting/restore utility

back to top of page

Performance

The following changes should be noted as possible sources of performance loss:

Existence predicates NOT IN and ALL may be slow

Firebird and, before that, InterBase, have produced incorrect results for the logical existence predicates ALL and NOT IN for many years. That problem has been corrected in Firebird 2.0, but the

change means that indexes on the inner tables cannot be used and performance may be slow compared to the same query's performance in V.1.5. “Inner tables” are the tables used in the subquery argument inside an ALL or NOT IN expression.

Note: NOT EXISTS is approximately equivalent to NOT IN and will allow Firebird to use indexes.

Superserver garbage collection changes

Formerly, Superserver performed only background garbage collection. By contrast, Classic performs “cooperative” GC, where multiple connections share the performance hit of GC.

Superserver's default behaviour for GC is now to combine cooperative and background modes. The new default behaviour generally guarantees better overall performance as the garbage collection is performed online, curtailing the growth of version chains under high load.

It means that some queries may be slower to start to return data if the volume of old record versions in the affected tables is especially high. ODS10 and lower databases, having ineffective garbage collection on indices, will be particularly prone to this problem.

The GCPolicy parameter in firebird.conf allows the former behaviour to be reinstated if you have databases exhibiting this problem.

back to top of page

Firebird API

Note the following changes affecting the API:

isc_interpret is deprecated

isc_interpret() is deprecated as dangerous. Use fb_interpret() instead.

Events callback routine declaration corrected

The new prototype for isc_callback reflects the actual callback signature. Formerly, it was:

```
typedef void (* isc_callback) ();
ISC_STATUS isc_que_events(
   ISC_STATUS *, isc_db_handle *, ISC_LONG *, short,
   char *, isc_callback, void *);
```

In the Firebird 2.0 API it is:

```
typedef void (*ISC_EVENT_CALLBACK)
  (void*, ISC_USHORT, const ISC_UCHAR*);
ISC_STATUS isc_que_events(
   ISC_STATUS*, isc_db_handle*, ISC_LONG*, short,
  const ISC_SCHAR*, ISC_EVENT_CALLBACK, void*);
```

It may cause a compile-time incompatibility, as older event handling programs cannot be compiled if they use a bit different signature for a callback routine (e.g., void* instead of const char* as the last parameter).

back to top of page

Windows-specific issues

For installing, configuring and connecting to Windows servers, be aware of the following issues:

Windows local connection protocol with XNet

The transport internals for the local protocol have been reimplemented (XNET instead of IPServer). With regard to the local protocol, the new client library is therefore incompatible with older servers and older client libraries are incompatible with the Firebird 2 servers.

If you need to use the local protocol, please ensure your server and client binaries have exactly the same version numbers.

Client impersonation no longer works

WNET (a.k.a. NetBEUI, Named Pipes) protocol no longer performs client impersonation. For more information, refer to Change to WNET protcol in the chapter about new features.

Interactive option added to instsvc.exe

D. Yemanov

The optional switch -i[nteractive] has been implemented in instsvc.exe to enable an interactive mode for LocalSystem services.

For v.1.5, it was required (as *Allow service to interact with desktop*) to run the local IPC protocol, as it used a windows message to connect the server. In v.2.0, it is no longer necessary and the server itself does not need this option.

However, some custom UDFs may use the Win32 messaging facilities and this option allows them to work as expected.

Note: instsvc.exe is a command-line utility for installing and uninstalling the Firebird service. It does not apply to Windows systems that do not have the ability to run services (Win9x, WinME).

For detailed usage instructions, refer to the document README.instsvc in the doc directory of your Firebird installation.