FrontBase 8.2.22

FrontBase version 8.2.22 is maintenance release and includes bug fixes additions and improve long term stability.

FrontBase

Conversion to and from any type timestamp values now retains the precision.

FBBRSBackup

Did crash when -k and an illegal configuration file was used

Did crash when -r on a configuration with a backup archive.

FrontBase 8.2.21

FrontBase version 8.2.21 is maintenance release.

FrontBase

Some cases of variable assignment on the statement level would not assign the complete value.

Some triggers caches was not cleared when the trigger was dropped.

Optimize database has been improved and now release unused space in the underlying .fb file when it is a regular file.

The new_uid(length) function now pads with random data.

sql92

The SHOW SIZE command now calculates the index and blob sizes.

The command line options -version and --version, prints the version number and the options -vv, -vversion and --vversion prints the version number and version timestamp.

The %c format specifier did not quote the ' characters inside characters literals.

FBCAccess

Two functions fbcdcFBCAccessVersion and fbcdcFBCAccessVerboseVersion provides version and verbose version information for the FBCCAccess library.

FrontBase 8.2.20

FrontBase version 8.2.20 is maintenance release and adds support for Apple Silicon and Big Sur.

FrontBase 8.2.19

FrontBase version 8.2.19 is maintenance release and includes a bug fixes and additions.

FrontBase

The transaction log was in some cases incorrect if a SET UNIQUE was executed alone in a transaction causing the replication an clustering to fail.

Setting timezones with unknown timezone names names was not reported.

FrontBase 8.2.18

FrontBase version 8.2.18 is maintenance release and includes a bug fixes and additions.

Release

This version of FrontBase is also released for MacOS BigSur with native support for the arm64 architecture.

FrontBase

The error messages issued when an attempt to roll a transaction log for an encrypted database forward and the given keys does not match have been improved.

FBReplicator

The replicator would, crash when a client database during startup requested a transaction which was not available from the master.

sql92

The exit code is set if any error occurs during execution.

The output format template have been enhanced with %q which expand one double quote to two.

A summary of the SET FORMAT <format>; output format statement.

<format> = COLUMN | ROW | INSERT | SUPPRESS | CSV | '<template'

The template string may contain printf style format specifiers:

```
 <format>  :==     %{<column>}{.<width>}<flag>    |
                   %*{<width>}<flag>             |
```

```
                    %{<width>}[<template>]          |
                    %%
<column>  :==       positive integer                |
                    "<label>"
<width>   :==       positive integer
<flag>    :==       c | l | T | S | n | # | r | B | b | [<m>]q
<m>       :==       <character>
```

If omitted, column or the width are interpreted as 0,

For a flags expands to

- c : column value
- l : column label
- T : table name
- S : schema name
- n : column name
- p : the following is used a a printf format string
- # : the row number in current result set
- r : he result set number
- B : the content of the lobs for a row;
- b : he column values without any type decoration and no quoting
- q : the column value without any type decoration and " as ""
- mq : the column values without any type decoration and quote m as mm where m is any character

The expanded string is padded with spaces to the width.
The flag B expands to the standard sql92 blob definition for all blobs in the select

B: The standard blob definition

%{<width}[<template>l{<template>] iterates over all columns and expands the template for each. If the <column>, ie. label or number, is not specified inside the iterator the current column is referenced. The first iteration expands using the the first template and the following with the second.

Printf formats implemented (%p)

- w : column datatype width
- s : column datatype scale
- f : column datatype from
- t : column datatype to
- d : column datatype string
- * : expands to all separated with comma

Escapes: \n new line, \r carridge return, \t tab, \ddd to the digital unsigned byte values, \c to the character.

A ' (quote) inside a templage must be written as '' (two quotes);

The syntactic ambiguity %"q" which is interpreted as a label q and not a quote of ". Instead use either %q or %."q.

Note:: %b is the same as %''q
%q is the same as %"q
%q\q can be used to for a q after a %q expansion
%*<flag> is the same as %[%<flag>l,<%flag>].
%% expands as %
''%"q'' is the same as %c for strings

FrontBase 8.2.17

==========

FrontBase version 8.2.17 is maintenance release and includes a bug fixes and additions.

FrontBase

The "STOP DATABASE ...; command would always terminate the connection, even in the case of errors. The connection is now retained in the case of errors.

Conditional return statements in persistens store modules would in some cases not return but continue execution of subsequent statements.

FBCAccess

Getlogin may not return a value which caused the creation of a session in connection when connection to a URL. If neither the getlogin nor the environmental variable USER returns a values the string unknown is used as the login.

If a database URL does not provide a schema frontbase:// is assumed.

FrontBase 8.2.16

==========

FrontBase version 8.2.16 is maintenance release and includes a few bug fixes and additions.

FrontBase

A client connection can now receive notification of any insert/update/delete statement committed by the server. The FrontBase notification schema is describe in detail in a separate document. You can find the document in the documentation section on the web-site www.frontbase.com.

The variables declare by a DECLARE statement lives until they are explicitly

dropped with a DROP VARIABLES; statement or when the session disconnects. In previous releases the variables life was restricted to the statement list of one server round trip. The FrontBase server option option -ddvs will enable the pre 8.2.16 semantics.

Transaction logging of schema changes were in rare cases incorrect.

Prepared statements with schema changes could, in rare cases, cause a crash.

Semantic checks of statements in the body of persistent store modules are postponed until the execution time if the statements may change the schema. In that case a warning is issued. This allows conditional execution of schema changing statements.

sql92

The statement

SHOW NOTIFIVATION [FOREVER];

will wait for and show one notification. If forever is provided the command will repeat until it is interrupted.

FrontBase 8.2.15

==========

FrontBase version 8.2.15 is maintenance release and includes a few bug fixes and additions.

FrontBase

The options -clients=no will make the access to the transaction log readonly.

The statement SET WRITE SQL TRUE GLOBAL did not turn on SQL logging for existing client connections**.**

Cast of a time or time with time zone values now include the specified precision.

Prepared statements with create table statements would in rare cases cause a crash.

FBCAccess

The function fbcdsBuildTimeStringP have been added in FBCDateSupport, it includes the precision of the second part.

sql92

The command SET FORMAT INSERT; sets the format to

'%Binsert into \"%S\".\"%T\" (%[\"%l\"l,\"%l\"]) values (%*c);'

The output with be sql insert statements.

Any type column values was not printet correctly when the format was controlled by a template.

Values of time and time with time zone values now displayed with the specified precision.

FrontBase 8.2.14

==========

FrontBase version 8.2.14 is maintenance release and includes a few bug fixes and additions.

FrontBase

The UID generation now set the bytes longer that 12 to random bytes instead of zeros.

The support for UID mapping during when importing data with INSERT INTO ... FROM INPUT have been improved.

Allow more that 999 columns in when importing with

```
INSERT ... INTO ... INPUT ...
```

Creating collations with the NO PAD attribute is now supported.

FBCAccess

Two function have been added to FBCDatabaseConnection to control the timeout used establishing a connection to a database. The time could be considerable in cases where the target machine was not responsive.

```
void   fbcdcSetConnectTimeout ( double seconds);
double fbcdcConnectTimeout    ( void );
```

FBCAccess

The local timezone name id now derived correctly on Linux hosts. Consequently, sql92 set the local time zone as expected.

FBAccess

Two class methods have been added to the FBDatabaseConnection class in order to control the timeout when establishing a database connection.

```
+ (void)setConnectTimeout:(double)seconds;
+ (double)connectTimeout;
```

Four instance methods have been added the FBExecHandler class in order to controll timeouts when reading from and writing to the FBExec.

```
- (void)setReadTimeout:(double)seconds;
- (void)setWriteTimeout:(double)seconds;
- (double)readTimeout;
- (double)writeTimeout;
```

In a normal environment the FBExec should response immediately so timeout values of one second is adequate.

An instance method returning the hostname of an FBExecHandler have been added.

```
- (NSString*)hostname;
```

FrontBaseManager
The FrontBaseManager stability have been improved.

It is now possible to control timeout used when communicating with the FBExec and when establishing a connection to a database. In the cases where the target host was unresponsive the standard timeout is considerable.

The timeout can be set in the preference pane. the standard values of one second should be adequate most operational environments.

FrontBase 8.2.13
==========
FrontBase version 8.2.13 is maintenance release and includes a few bug fixes and additions.

FrontBase

An error was introduced in 8.2.12, it it now again possible to delete blobs.

FrontBase 8.2.12
==========
FrontBase version 8.2.12 is maintenance release and includes a few bug fixes and additions.

FrontBase

An option, -logSQLFilename=<template>, have been added. When a sql log file is created, either at startup, by turning on sql logging, or when switching to a new sql log file, the template is used to derive the name of the sql log file. The template is expanded with strftime function and If the template starts with "./" , on Windows ".\," the default database log directory is pre-pended, and as the final step the filename is translated according to the patterns in the FrontBase configuration file.

The SQL LOG FILE in the the "extract database;" now show the correct path name.

The Pid.pid file now contains the pid of the running server, ant not the format string.

SQL92
The %c format specifier did not print column values of type DATE

Show size reports the size of blobs correctly for databases that are created and run with 8.2.12.

The -C "Commands" have been added. sql92 with execute the commands and exit.

The "set format csv;" causes sql92 to format result sets as comma separated values.

FrontBase Manager

The stability of the FrontBase manager running on Mac OS 10.13 have been improved.

FrontBase 8.2.11
==========
FrontBase version 8.2.11 is maintenance release and includes a few bug fixes and additions.

FrontBase
FrontBase server was not able to derive the local time zone when running on High Sierra.

Support for the time zone specification format have been added.

SQL92
The predefined output format INSERT have been implemented. The statement

set format insert;

vil format the output as insert statement;

The %B format for any type blobs did not produce the correct output.

FrontBase 8.2.10

==========

FrontBase version 8.2.10 is maintenance release and includes a few bug fixes and additions.

FrontBase

Delete database did not work on Mac OS X 10.13

FrontBaseManager

The FrontBaseManager on Mac OS X was not able to open database connection when running on Mac OS X 10.13

FrontBase 8.2.9

==========

FrontBase version 8.2.9 is maintenance release and includes a few bug fixes and additions.

FrontBase

The OPTIMIZE DATABASE and CALL INFORMATION_SCHEMA. DEFRAGMENT(0) has been improved.

Commit and rollback statements in nested statement have never been support and they now produce an error.

A couple memory leaks have been fixed.

The memory performance of stored procedures and function calls have been improved.

The results of a function calls are now used for lookup in indexes during evaluation of predicates,

FBKeyGenerator

The key generator did crash when the cipher provided was not know.

FBCAccess

The stop database function (`fbccdcStopDatabase) can` now stops FrontBase 5 databases. That function is used in the sql92 command line tool and in the

FrontBaseManager.

utf8
Now detects and skips the UTF8 encoding marker: ef bb bf

Developer libraries

The Mac OS X developer librariy have moved to

/Libraries/FrontBase/lib/MaxOSX/libFrontBase.a

and is build with Mac OS X 10.6 as deployment target

and the iOS developer library have been moved to

/Libraries/FrontBase/lib/iOS/libFrontBase.a

and is build with iOS 5.0 as deployment target.

FrontBase 8.2.8

==========

FrontBase version 8.2.8 is maintenance release and includes a few bug fixes and additions.

FrontBase
Updates on large tables could, in rare cases, cause false optimistic locking conflict

Malformed UTF8 could in some cases cause a crash. The user is now warned when the input is not wellformed.

The ascii del character is now accepted as a legal character.

The last row in the data files create by the WRITE statement was repeated. The import would typically fail because of constraint errors.

The OUPUT specification used in the WRITE all statements have been augmented with the options RINDEX=<boolean-expr> and CLIENTFILENAMES=<boolean-expr>. The internal row indexes are written to the export files when RINDEX evaluates to true true, and is omitted when RINDEX evaluates to false. The default is not to write the row indexes, thus maintaining compatibility with earlier versions of FrontBase. If CLIENTFILENAMES evaluates to true, a subsequent import will read the data

files from the filesystem of the client, and there will not be necessary to copy the data files files to the target system. In general adding a '<' character to the filename in the INPUT specification will cause the file to be read an the client file system.

If a file names begins with a '>" in connection with a the WRITE BACKUP TO <ffilename> statement the backup will be written to the local filesystem.

Declared variables is now handled correctly in connection with replication.

Return the number of bit for the running server in EXTRACT DATABASE.

FBCAccess
The error code and the error count now reflect the result after a fetch.

The syntax of the connection url have been augmented if the schema part is omitted the url is interpreted as a frontbase:// url.

```
<URL>        :==  [<schema>://]<user>][<host>][:<database>][/<path>]  |
                  <database>[@<host>]
```

In the case of ambiguity the form <database>[@<host>] is preferred.

sql92
The new %b format specifier acts as %c except that it prints without the sql literal type sugar.

Only report warnings generated during fetch once.

If sql92 is invited with the -b option then BEGIN .. END; statement nesting is ignored.

All isolation levels and locking disciplines are now listet correctly withe the SHOW USAGE; command.

The "SET COMMIT TRUE;" warning is not printet when -s, silent, option is provided.

If it is not possible to connect to and url specified with the -d option sql92 now exit.

FBExec
The FBExec now identified started replication processes/services correctly cross platform.

FBReplicator
The registration of the replication process was not working between Windows and the IX'es.

The initialization of a new client have been simplified in is thus more robust.

A progress report is included when a newly created client is initializing the schema.

Windows installation

If you experience the error message:

"A required privilege is not held by the client."

Then try the following:

1. Select Start > All Programs > Accessories > Run.

2 Open SECPOL.MSC.

3. Select Security Settings > Local Policies > Security Options.

4. Select “User Account Control: Run all administrators in Admin Approval Mode” and disable it.

5. Restart the computer.

FrontBase 8.2.7
==========
FrontBase version 8.2.7 is maintenance release and includes a few bug fixes and additions.

Windows installation

As part of the installation on Window all FrontBase related services are stopped and removed, in order to ensure that all the executables can be installed.

After the installation the FBExec is started and so are all databases that are registered for autostart, all other databases replicators must be started manually again, either with the service manager, the sql92 command line tool or by the executable itself.

FBExec (Windows)

The FBExec now read the content of the FrontBase.exe symbolic link correctly fixing the protecting causing the start of database to return the 0x5 error.

The FBExec now reports the error text as well as numeric error codes.

The FBExec now supports the options -stopall and -removeall. The -stopall option causes the FBExec to stop all FrontBase related services. The -removeall option causes the FBExec to stop and remove all FrontBase related services.

FrontBase, FBExec and FBReplicator (Windows)

The service manager interface now support -start and -stop.

fbversion

The tool did crash when the /Library/FrontBase/Version directory did not exist.

FrontBase
FrontBase now supports encryption with AES encryption algorithm:

AES128	key length 128 bit
AES192	key length 192 bit
AES256	key length 256 bit
AES128NF	key length 128 bit no feed back

Alle the algorithms are run in a stream mode with clear text feed back, except the AES128NF with does not feature any feed back.

The set encryption statement is enhance with a TO option

```
SET ENCRYPTION DES [TO '/tmp/foo.key'];
```

The TO option causes the server to write the generated key to the specified file.

The default for the RINDEX option to the WRITE ALL OUTPUT(); statement have been changed to False, such that the files exported from FrontBase 8 can be imported by earlier versions of FrontBase.

FrontBaseManager

The FrontBaseManager displays timestamp values in the local timezone.

FrontBase 8.2.6
==========
FrontBase version 8.2.6 is maintenance release and includes a few bug fixes.

FrontBase

Evaluation of predicates in a multicolumn index would return an incorrect result when one of the n-1 columns predicates was is null, the operator was < or <= and the row was inserted or updated in the same transaction as the predicates was evaluated.

Creation of tables with a row size greater than 8k bytes caused a the server to crash.

Windows installation

The distribution on Windows include a tool for selecting either the 32-bit version or the 64-bit version.

FBSelect [-v] [32 | 64]

Without arguments the tool select the 64-bit if the local system supports 64-bits otherwise it select the 32-bit version. With the argument 32 the 32-bit version is selected and with 64 the 64-bit version is selected if possible. The option -v make the program chatty.

If you experience problems during installation make sure that there is no instances of FrontBase, FBReplicator or FBExec running on the machine.

Also make sure that you run the installation as an administrator. If you do experience problems allow the FrontBase installation image to run as administrator, do the following

1. Make an alias pointing to the self expanding image Frontbase-<version-number>.exe,

2. Select the properties->shortcut->advanced.

3. Check the Run as administrator check box .

4. Ok and Ok.

Double click the shortcut to run the installer.

FrontBase 8.2.5

==========

FrontBase version 8.2.5 is maintenance release and includes a few bug fixes.

FBCAccess

The conversion to the output character have been fixed.

The string returned by fbcmdMessage is now converted to the output character set.

The function fbcrhConvertToOutputCharSet have been removed as it was redundant.

FrontBase 8.2.4
==========

FrontBase version 8.2.4 is release on Mac OS X, Windows and Linux. The Mac OS X release include support for El Capitan.

FrontBase 8.2.3
==========

FrontBase version 8.2.3 is major release of FrontBase that ensures that transaction logs are serialized. Earlier versions of FrontBase have, in rare cases,and with exotic transaction settings, generated transaction logs which were not serialized.

FrontBase

FrontBase 8 is not binary compatible with previous version of FrontBase, consequently an existing database must be exported and then imported with FrontBase 8. A more detailed description of the export/import process is found at the end of this note.

However, FrontBase 8 is binary compatible across platforms, and 32, 64 bits versions. So you can copy a FrontBase 8 database from a 32-bit IOS system to a 64-bit windows system and expect it to work.

In order to ensure that the transaction log is serialized correctly some of the statement in the transaction log have been augmented with the internal row index, and thus earlier version of FrontBase cannot interpret the transaction logs created by FrontBase 8. FrontBase 8 can interpret transaction logs created by earlier versions of FrontBase. Replication clients should be upgraded before replication masters, and all cluster member should be upgraded at the same at the same time.

The internal row index serves as a primary key and is never reused. The row index is maintained across replication clients and cluster members, and can be used in predicate and in select lists.

The restriction that delete statements embedded in stored procedures and

function should have a primary key has been lifted; the row index function as the primary key. It is also possible to delete and update rows based on the index in a replication or cluster environment.

FBAccess

FBAccess version 8 can connect to earlier version of FrontBase and earlier versions of FBAccess can connect to FrontBase 8 databases.

FBCAccess

FBCAccess version 8 can connect to earlier version of FrontBase and earlier versions of FBCAccess can connect to FrontBase 8 databases.

FrontBase 7.2.21

FrontBase version 7.2.21 is maintenance release and includes a number of bug fixes and additions.

FrontBase

In extremely rare cases selects with predicates using > operators would return too few rows.

FBAccess

Fetching next row from a connection based on a file URL could cause a crash.

FrontBase 7.2.20

FrontBase version 7.2.20 is maintenance release and includes a number of bug fixes and additions.

FrontBase

The pseudo procedure updateSearch crashed the server when the updated row did not exist.

The handling of Unicode surrogates was not correct, an example is telephone

receiver: 📞

A pseudo procedure, "information_schema"."clear_table_statistics", for clearing the table statistics have added. The procedure does not have any arguments.

The server does not report an error when the peer have disappeared in connections with disconnects.

I rare high load cases using repeatable read isolation level and pessimistic locking discipline causes conflicts when none was present

The data read to create a backup is not inserted in the RDD cache.

Improved statistics collection of the RDD cache performance

FrontBase 7.2.19

==========

FrontBase version 7.2.19 is maintenance release and includes a number of bug fixes and additions.

FrontBase

The index cache statistics was incorrect reported by extract table;.

The option file was written before the .fb directory was created when a new file:// database was bootstrapped, and was thus lost.

Temporary tables dirtied the transactions and cause unnecessary file synchronizations.

Long running agent connections would leak memory.

sql92

The command line tool, sql92, did not format values of floating point types and fixed point types with the precision and scale specified in at the definition.

FBCAccess

It is now possible to listen for network connection for an application which use and embedded FrontBase server enabling your application to accept connections from remote clients.

The API is exposed in the FBCListen.h header file. A small example

```
FBCListen* fbclListen
 ( const char*  url,
   const char*  hostName,
   unsigned int port,
   unsigned int secure,
   unsigned int registerWithExec,
   const char*  registerAs
 )

FBCListen* listen = fbclListen*("file://localhost/foo.fb",
                                "192.168.55.163",
                                19273,
                                0,
                                1,
                                "off" );
```

The database is stored in foo.fb. Listen for incoming connections on the interface with the ip address "192.168.55.163", you can specify a host name, NULL will allow allow connection on any interface. Do not enable secure connections. Register with the local FBExec withe the name "off".

FrontBase 7.2.18

==========

FrontBase version 7.2.18 is maintenance release and includes a number of bug fixes and additions.

FrontBase

In cases where the transaction log have been damaged by external tools, like an incomplete copy, the server might crash during startup.

FBCAccess

==========

Memory was leaked when a result set was not completely exhausted upon release

The version 7 FBCAccess headers defines the symbol

#define FBVersion 7

so it is possible to determine if if you are using version 7.

The FBCAccess headers have been prepared to be included in c++ programs such that the c++ compiler does not mangle the c names.

Mac OS X Installation

The installation of multiple versions was not included in 7.2.17 for Mac OS X

FrontBase 7.2.17
==========

FrontBase

Satisfies predicates could is rare high load situations cause the server to crash.

Access to tables with rows with a static size created that 8K could cause the server to crash.

Long words (> 1280 characters) in Look See indexes could crash the server.

The -init=<file-name> option is not written to the options file, and are not included in the evaluation of "without any options".

In same rare cases the server would crash when satisfies predicates was used in prepared statements.

FBAccess

The type FBBlobHandle is now a subclass of NSString instead of a alias. Consequently, the blob handles objects may be identified by their class.

FrontBaseManager

It s now possible to download blobs to files with the FrontBaseManager.

FBCAccess

The function fbcdcReadBLOBToFile is now producing the expected result.

SQL92

The sql92 command "`WRITE BLOB <blob-handle> FILE <file-name>;`" is now producing the expected result.

FrontBase 7.2.16
==========

FrontBase version 7.2.16 is maintenance release and includes a number of bug fixes and additions, and have been ported to RedHat Linux.

FrontBase

The sql statements

```
prepare; execute 0;
```

return a syntax error, instead of crashing the server.

Writing a backup to a file local to the client system, (write backup to '>path-name') crashed the client.

The memory footprint for schema operations like add/drop/update column have been improved.

In rare situations, select top(a,b) did return incorrect number of rows.

Database names ending with '/' is accepted.

FrontBase options

If a FrontBase database is started without any options, the options is read from

<name>.fb/Options.opt

in the database directory, if that files does not exist then the default

etc/FrontBase.options

is read.

If FrontBase is started with options the active options are written to the options file in the database directory.

The term "without any options" is little more elaborate. FrontBase has a number of options which does not make sense when a database is started for normal operations, they are not included in the calculation of "without any options". Those options are

```
-create[=<g-bytes>]
-restore[=<file-name>]
-gtc=<signed number>
-rfp
```

-rollforwardprogress
-rollforward=<transaction number>
-transaction=<transaction number>
-daemon

The option -config print the active options and the filename translations defined in the configuration file to stderr. The server will exit immediately.

FBExec

The create and start database commands does not return until the database server have been started is registered and ready to accept user connections.

The database option files is not saved by the FBExec, but is save when FrontBase server is started.

FrontBase 7.2.15

==========

FrontBase version 7.2.15 is maintenance release and includes a number of bug fixes and additions.

FrontBase

Conversion from floating point literal to integer types may yield incorrect results if the literal was terminated in two or more zeros in the fractional part.

FrontBase 7.2.14

==========

FrontBase version 7.2.14 is maintenance release and includes a number of bug fixes and additions.

An explicit collation used for an escaped like involving any types is now selected correctly.

Explicit per roundtrip auto commits is now handled correctly.

Any type tiny integers was not compared as a numeric comparison.

The SQL log file and the FrontBase log files were not controlled by the etc/ FrontBase.config.

Example

If the FrontBase.config contain the patters

```
^\(.*\)/Databases/\(.*\)\.fb/FrontBase.fb$  = \1/Databases/\2.fb
^\(.*\)/Databases/\(.*\)\.fb/Options.opt$    = \1/Databases/\2.options
^\(.*\)/Databases/\(.*\)\.fb/Backups.bck$  = \1/Backups/\2
^\(.*\)/Databases/\(.*\)\.fb/TLogs.tlog$     = \1/TransactionLogs/\2
^\(.*\)/Databases/\(.*\)\.fb/Logs.log$          = \1/Databases/\2.logs
^\(.*\)/Databases/\(.*\)\.fb/Logs.log/FrontBase.log$= \1/Databases/
\2.fb.log
^\(.*\)/Databases/\(.*\)\.fb/Logs.log/sql.log$ = \1/Databases/\2.fb.sql
```

the files will be created with structure similar to earlier FrontBase installations.

The option -config causes the server to print the file name translations caused by the configuration file to stderr.

The IN (…) predicate involving indexed ANY TYPE columns was not evaluated correctly.

The default value for DNS translation of client ip addresses to domain names in connection with the EXTRACT USAGE; statement is now controlled by the option

```
-dnslookup=yeslno
```

Translation of client ip-addresses to domain names may be time consuming whenDNS configuration is incomplete.

Windows
The tcp keep alive timer is now controlled with the two options

```
-tcpkeepalivetime=<seconds>
-tcpkeepaliveinterval=<seconds>
```

and if both values are non-zero tcp keep alive timers will be enabled for all client connections.

Files overwritten during execution of WRITE ALL is now truncated to length 0 before they are overwritten.

FBCAccess

The function

int fbcdIsCreated (const char* url);

have been added to FBCDatabases.h. The function returns 1 if the database denoted by the url exists.

FrontBase 7.2.13

==========

FrontBase version 7.2.13 is maintenance release and includes a number of stability improvements and bug fixes and optimizations.

FrontBase

It is now possible to specify collations to the string comparison and like operators involving any type values.

Comparison of any type numeric types produce the results you would expect.

The evaluation of predicates involving sub-select without outer column references have been optimized.

A select top (0,1) ... return one row as expected.

Import of BLOBS and CLOBS with a size greater that 128M did fail.

Any type values are now imported correctly.

FrontBase 7.2.12

==========

FrontBase version 7.2.12 is maintenance release and includes a number of stability improvements and bug fixes.

FrontBase

Comparison of any type character types always used a case insensitive collation, the comparison have been changed to behave as the standard character type, which uses the default case sensitive collation.

FrontBase 7.2.11

==========

FrontBase version 7.2.11 is maintenance release and includes a number of stability improvements and bug fixes.

FrontBase

A transaction which did create tables would prevent creation of new connections until the transaction was either rolled back or committed.

The precision of the timestamps in the sql log have been improved.

Memory was leaked in the rare situation where result sets were not exhausted before the meta data was released.

Explicit collations was not respected by the like operator when the match expression was any type.

Comparison of numeric any type types did produce incorrect and unexpected results.

FrontBase 7.2.10

==========

FrontBase version 7.2.10 is maintenance release and includes a number of stability improvements and bug fixes.

FrontBase

The server might crash, if an exception did cause a transaction to roll back, and that transaction did have result sets which have not been exhausted.

The statement EXTRACT LICENSE is re-introduced to maintain compatibility with existing drivers.

In rare, high load situations, the the server would crash when checking a users privileges.

in rare high load situations, the first blob manipulations after the server was started could cause a crash.

FrontBase Manager

User names are case insensitive, but the FrontBaseManagers test for _system user was case sensitive. That would leave buttons disabled in cases where the action requires the current user to be the _system user.

FrontBase 7.2.9

==========

FrontBase version 7.2.9 is maintenance release and includes a number of improvements and bug fixes.

FrontBase

FrontBase version 7.2.9 run without a license.

The option -ignorepasswords[=yes|no] has been added. When FrontBase is started with -ignorepasswords=no all password checks are ignored.

FBCAccess

The function fbcmdRowCount returned the row count of the last batch received, it now correctly returns, if known, the total number of rows in the result set.

The new function fbcmdFetchedRows returns the nuber of rows that haves been fetched so far.

FBAccess

The method [FBMetaData rowCount] returned the row count of the last batch received, it now correctly returns, if it is known. the total number of rows in the result set.

The new method [FBMetaData fetchedRows] returns how many rows that haves been fetched so far.

FrontBase 7.2.8

==========

FrontBase version 7.2.8 is maintenance release and includes a number of improvements and bug fixes.

FrontBase

SQL statements of the form values(set unique = <number> for <table> for replication; was not written to the transaction log.

The lookup of user names have been optimized.

The index based evaluation of range predicates involving any type values did produce empty results.

The fractional part of a timestamp value was not always transaction logged.

Any type objects with the actual type of character and bit would not always be transaction logged correctly.

FBCAccess

In high load situations the SET TIME ZONE sql statement may fail due to lack of resources on the server, the failure is reported back to the client as expected. However, the create session does set the time zone to the default time zone for the client. In case of failure, that failure was not reported correctly.

The function fbcmdConnected have been added for convinience.

FrontBase 7.2.7

FrontBase version 7.2.7 is maintenance release and includes a number of improvements and bug fixes.

FrontBase

The substring string function in rare cases not produce the correct result when the original string contained non ascii characters.

The sql statement values (select unique foo); was not transaction logged correctly, which could leaving the sequence counter out of sync when a database status was changed from replication client to be replication master.

The construct insert into foo...; delete from foo; insert into foo ...; could crash the server.

Insert a bunch of row and and delete all of them in one transaction would cause the size of the .fb file to grow beyond any reasonably size.

The command line option -preload[=yes|no] control the preload of table caches when the server is started.

FBAccess

Memory was leaked when fetching rows on a database connection established across the network.

FrontBase 7.2.6

FrontBase version 7.2.5 is maintenance release and includes a number of improvements and bug fixes.

FrontBase

The last rollforward message was always written to stdout during startup, it should be controlled by the -rollforwardprogress option.

Correlation names would incorrectly hide the table name in nested subselects, like
select * from t0 where exists (select * where from t0 t1 where t1.c0 ='x' and t0.c1= items.original_id).
t0.c1 was incorrectly hidden.

FrontBase 7.2.5

==========

FrontBase version 7.2.5 is maintenance release and includes a number of improvements and bug fixes.

FrontBase

The first start, and only the first, of a database after the installation of a license, would crash when the database was created without a license.

sql92

The sql92 command create database would case sql92 to crash when file url schema was used.

FrontBase 7.2.4

==========

FrontBase version 7.2.4 is maintenance release and includes a number of improvements and bug fixes.

FrontBase

Select unique for negative values was not transaction logged correctly.

FrontBase 7.2.3

==========

FrontBase version 7.2.3 is maintenance release and includes a number of improvements and bug fixes.

FrontBase versions

It is now possible to have several binary incompatible versions of FrontBase installed on a machine, and it is possible to select the default the installations with the command line tool fbversion.

When you attempt to install FrontBase 7 on a machine with an earlier version of FrontBase installed you will be asked if you want to retain the old installation. If you choose to do so the existing installation and will be moved to the /Library/FrontBaseVersions/<major-version-number>, and the FrontBase 7 installation will be installed in /Library/FrontBaseVersions/7. To select an installation the

following softlinks

/Library/FrontBase
/Applications/FrontBaseManager.app
/Applications/FBUnicodeManager.app
/Library/Frameworks/Frameworks/FBCAccess.framework
/Library/Frameworks/Frameworks/FBAccess.framework
/Library/LaunchDaemons/com.frontbase.fbexec.plist

will point to the relevant directories in the /Library/FrontBaseVersions structure.

The tool fbversion

fbversion
fbversion <index>
fbversion <version>

Select the active FrontBase version among the installed versions

The first form lists the the possible choices and the selected
The second form selects the index'th versioned
The third form select the named version

The tool does require authentication. If you do not want to provide you password all the time you can always make the tool setuid. (chmod 04555 /usr/sbin/fbversion).

Installation

The FrontBase installation have been updated to reflect that changes introduced with Xcode 4.3 on Mac OS X. The developer libraries for the various SDK's are now installed in the FrontBase installation sub-directory lib. The libraries are name to reflect the SDK:

libFBCAccess-MacOSX10.7.a
libFBCAccess-MacOSX10.6.a
libFBAccess-MacOSX10.7.a
libFBAccess-MacOSX10.6.a
libFrontBase-iOS5.1.a

The libFBCAccess-MacOSX10.7.a and libFBCAccess-MacOSX10.6.a supports the architectures i386 and x86_64, where as libFrontBase-iOS5.1.a supports armv7 and i386. You can link with the same library for both iOS device and iOS simulator.

The headers for libraries are identical and they are installed in the FrontBase installation sub-directory include.

The most robust way to incorporate the libraries in you applications is to copy the library and the headers into your Xcode project, which allows you to copy Xcode project without having to install FrontBase on you current development machine.

The libraries are complete in the sense that they you include all the code you need to embed FrontBase into your applications, and the iOS 5.1 library does include both the C and the objC API's.

FBAccess

The values returned for columns of time types was not formatted correctly.

FrontBase

If a database is created with a specified size (-create=<number-of-GB) the server attempts to allocate all the disk space as consecutive blocks of the disk, if that is not possible a warning is printed.

Storing any type values with a size of 28 and 29 would cause the server to crash.

Any type values was in rare cases not transaction logged correctly.

Enable use of indexes based on any type columns types to be used in predicate evaluation.

FrontBaseManager

Selecting the show content is schema objects pane caused an exception, such that the content was not displayed.

FrontBase 7.2.2

==========

FrontBase version 7.2.2 is maintenance release and include a number of improvements.

Database file structure

A database is a directory with the structure

<name>.fb/FrontBase.fb

<name>.fb/TLog.tlog
<name>.fb/Backup.bck
<name>.fb/Log.log
<name>.fb/Log.log/FrontBase.log
<name>.fb/Log.log/sql.log
<name>.fb/FrontBase.options

If you feel the need for changing location of the individual files in the database directory you can do so by providing a FrontBase.config file in the installation etc directory. The FrontBase.config file contain lines with the form

<reg-expr> = <expansion>

The regular expression is a basic regular expression, see re_format. If the reg-expr or the expansion contains spaces, tabs and such, enclose in ".

Before any of the files in a database directory is accessed the filename is translated according to the config file, the first regular expression that match the file name will be used and the expanded name is the result. If no match is found the name is used unaltered. When a name is expanded the \1 .. \9 is expanded as corresponding parenthesized subexpression.

When databases are deleted the wrapper directory is deleted, including all contents, completely when the FrontBase.fb file have not been relocated.

Example

If the FrontBase.config contain the patters

```
^\(.*\)/Databases/\(.*\)\.fb/FrontBase.fb$   = \1/Databases/\2.fb
^\(.*\)/Databases/\(.*\)\.fb/Options.opt$    = \1/Databases/\2.options
^\(.*\)/Databases/\(.*\)\.fb/Backups.bck$    = \1/Backups/\2
^\(.*\)/Databases/\(.*\)\.fb/TLogs.tlog$     = \1/TransactionLogs/\2
^\(.*\)/Databases/\(.*\)\.fb/Logs.log/\(.*\)$  = \1/Databases/\1.fb.log
^\(.*\)/Databases/\(.*\)\.fb/Logs.log/sql.log/\(.*\)$  = \1/Databases/\1.fb.sql
```

the files will be created with structure similar to earlier FrontBase installations.

FrontBase Manager

The FrontBase manager has new file menu point Open URL.

FrontBase

Fix a race condition in connection with 64-bit memory access on 32-bit buses.

Select count(*) in connection with join expression have been optimized.

Righ truncations in connection with static size bit values could crash the server.

Certain, rare, combination of table values expressions and join operators would crash the server.

Encryption

The following new SQL statements have been introduced to support encryption of the whole database on the fly:

```
SET ENCRYPTION [DES | XOR | CCSC] ;
SET ENCRYPTION [DES | XOR | CCSC] '<textual-key>';
SET ENCRYPTION [DES | XOR | CCSC] PIN '<pin-code>';
DROP ENCRYPTION;
```

The statements change the encryption of the complete database. The first form generates a random key and change the database to be encrypted with that key and the specified algorithm. The key in textual for is returned ad the message component of the meta data. The second form used the specified key. The third form derive a key from the <pin-code> and change the encryption to the algorithm and the derived key. The algorithm deriving the key is a slow one way function, the calculation on a 3G HZ cpu is approximately 1s, protecting the pin against brute force attacks.

Before the encryption is changed a backup with the old key is created. The encryption of the transaction log and sql log are changed to use the new key.

In order to start the database is is necessary to provide the correct key, you can do that with the -key=<filename-option>, if the key is not specified the server will attempt to read the key from stdin. If stdin a tty device the server will prompt for the key. You can provide either a textual key or a pin code.

In order to generate a textual key you can else use the command line tool FBGenerateKey.

Please note that the file specified with the -key option is only used during startup, thus you can keep the keep on a USB stick or similar removable devices.

The forth form removes the encryption of the database.

The command does require that you are authenticated as the "_system" user,

Transaction grouping

FrontBase is can to group small transactions such that the file system is not synched when a transaction is committed, thus improving the write performance considerably. You can control the number of transactions in a group with the option -syncafter=<number>. The cost is potential slower startup, and slightly larger memory footprint. Numbers in the range form 200 to 2000 have been used with success. Transactions which may rollback does break the optimization an causes an sync before the group is filled, as does very large serializable pessimistic locked transactions.

You can also set the group size with the sql

```
set fsync after <number> commit;
```

if you are authenticated as "_system".

You can inspect the performance of the mechanism with the sql

```
select * from information_schema.commit_statistics;
```

which with give you the following counters

commitGroupSize:	The group size.
commitGroupCount:	The number of transactions left in the current group
rollbackTransactions:	The number of transaction which could be rolled back
commitTransactions:	The number of transactions which was known to commit
degradedTransactions:	The number of transactions which changed
commitCount:	The number of commits
rollbackCount:	The number of rollbacks
flushCount:	The number of syncs

You want small flush counts compared to the commit count, you do not want any transactions to rollback in this context.

Filesystem caches and rdd

When using large -rdd caches it is advisable to suppress the native file system buffering, large is compared to the amount of RAM an the machine, you can to

that with the option -fscache=[yes|no]. The effect is that the data is not double buffered, and FrontBase will not end up using all the RAM on the machine for file system buffers,which is not use for anything and makes the system less predictable.

Partition capacity

The size of a partition is per default unlimited (0x7fffffffffff) 4k blocks which will do for most applications. If the size is unlimited FrontBase will attempt to use as little disk space as possible, but if if you set the limit FrontBase will use all the disk space available, which will allow FrontBase to improve write performance significantly if free space is available.

You can set the size of the initial partition with the -create=[<number-of-Gs>] option, and you can set the partition capacity of the system partition with the statement

```
alter partition "system" size <number-of-4k-blocks>;
```

and you can inspect the partitions with the command

```
select * from information_schema.partitions;
```

rdd

If you have RAM to hold the whole database do so, reading from disk causes waits. The option -loadrdd will upon start cause FrontBase to fill the rdd. The load is throttled so it does not interfere with normal operation.

When the amount of pending output exceeds 10% of the rdd writing is delayed in order not fill the red with dirty data. The disk system is falling behind so filling the cache with dirty data will just makes it worse.

In order to improve write performance data data is not written until amount of dirty data exceeds 1 MB, or inactivity have existed for 60s.

FrontBase 7.1.1

==========

FrontBase 7.1 is , as the version number suggests, not binary compatible with previous version of FrontBase.

FrontBase Distribution 6.1.10

==========

FrontBase version 6.1.10 is maintenance release.

FrontBase

Statistic counters for the raw device driver is now 64-bit.

Large fixed size any_type values was truncated.

The accumulated size of blobs and clobs columns in a table was not updated correctly and was limited to 4G.

Satisfies predicates was in rare situations rejected with error code 512: "SATISFIES predicate must be either the only expression or part of a top-level AND expression"

The identification of BLOBS and CLOBS was in rare cases duplicated on systems with realtime clocks with poor resolution.

The SIGPIPE signal could cause the server to terminate prematurely.

Frameworks and developer libraries

Large BLOBS and CLOBS, mega-bytes, could cause address faults when the connection to the database server was encrypted.

Connections to the FBExec and to the FrontBase server did leak mach ports.

sql92

The show size command is now using 64-bit counters.

FrontBase Distribution 6.1.9

FrontBase version 6.1.9 is maintenance release.

FBAccess framework

The use of md.fetchhandle would cause a cyclic retaindependency between the md and the md.connection, making it impossible to release the connection. The cyclic dependency have been removed.

FrontBase Distribution 6.1.8

FrontBase version 6.1.8 is maintenance release.

The naming of the components of in the database wrapper have been changed to the more copy friendly names: FrontBase.fb, TLog.tlog, Log.log, Backups.bck.

FrontBase Distribution 6.1.7

==========

FrontBase version 6.1.7 is maintenance release with focus on bug fixes.

Any file type extension is removed from the filename for the database wrapper.

FrontBase Distribution 6.1.6

==========

FrontBase version 6.1.6 is maintenance release with focus on bug fixes.

The like operator did not return correct result in the case of a pattern begining with
'%' followed by one character.

FrontBase Distribution 6.1.5

==========

FrontBase version 6.1.5 is maintenance release with focus on bug fixes.

FrontBase

Certain like operators with escape specifications would not return always all matching rows.

FrontBase Distribution 6.1.4

==========

FrontBase version 6.1.4 is maintenance release with focus on bug fixes.

Improve the error messages for failing constraint checks to include the failing column values when the column type is bit or bit varying.

Allow cast of CLOBs to CHARs.

Reading BLOBs and CLOBs before they were committed returned incorrect values and would cause incorrect values to be stored when the BLOBs/CLOBs were actually committed.

Reading blobs without having an open transaction could cause the server to crash.

The unique number generator was exported with the default values, when export was done using versioned transactions, the correct unique number is now exported.

Several clients creating blobs simultaneously could cause FrontBase to crash.

FBCAccess/FBAccess

The libraries debug symbols have been stripped for the debug symbols.

FrontBase Distribution 6.1.3

FrontBase version 6.1.3 is maintenance release with focus on bug fixes.

FBAccess

An access violation was caused, in not too friendly network environments, by broken
connection to the FBExec, has been fixed.

iOS 4.3

Support for iOS 4.3 has been added.

FrontBase Distribution 6.1.2

FrontBase version 6.1.2 is maintenance release with focus on bug fixes.

FrontBase

A number of problems with reading blobs which caused the FrontBase to stop have been fixed.

FrontBase Distribution 6.1.1

The FrontBase 6 Distribution is a major update of FrontBase, the most noticeable news is that FrontBase supports iOS.

On the iOS FrontBase is embedded, there is no need for a server and network connections, all the FrontBase code is included in the application, and the database connection is made directly to a file. Connections to databases on remote servers is of course also supported.

The embedded FrontBase version is also available on Mac OS X, as part of the client frameworks and libraries.

The connection methods in the client libraries have been complemented with

URL connections, connection to an embedded database or to a remote database server is simply a matter of using the right URL.

FrontBase 6 database file format is incompatible with earlier versions of FrontBase, to migrate: export the database to text form and import is into a newly create FrontBase 6 database.

FrontBase

All versions of FrontBase is binary compatible, so you can copy a embedded iOS database to you Mac and start it with the 64 bit server version.

Installation file structure

The file structure have been revamped in FrontBase, static and dynamic data have been separated in order to accommodate the iOS system.

The Mac OS X server installation directory has the structure

/Library/FrontBase/bin
/Library/FrontBase/Library
/Library/FrontBase/Collations
/Library/FrontBase/Translations
/Library/FrontBase/etc
/Library/FrontBase/etc/Options.opt
/Library/FrontBase/etc/FrontBase.config
/Library/FrontBase/Databases

In the iOS version, the files from the Library, Collations, and Translations directory are all included in the library, the developer only needs to reference the library and will not need to provide those files as resources in the application. However, if the need arise, either to add translations or collations or modify the bootstrap files, they should be copied as app wrapper resources with the path:

Library
Collations
Translations
LicenseString
etc

In the Mac OS X embedded installation, the FBCAccess framework the files are included in the resources directory

Resources/Collations

Resources/Translations
Resources/Library
Resources/LicenseString
Resources/etc

Database file structure

A database is a directory with the structure

<name>.fb/FrontBase.fb
<name>.fb/TLog.tlog
<name>.fb/Backups.bck
<name>.fb/Logs.log
<name>.fb/Logs.log/FrontBase.log
<name>.fb/Logs.log/FrontBase.sql
<name>.fb/Options.opt
  <name>.fb/Pid.pid
  <name>.fb/Cluster.plist

Default options

If a FrontBase database is started without any options, the options is read from

<name>.fb/Options.opt

in the database directory, if that files does not exist then the default

etc/FrontBase.options

is read.

If FrontBase is started with options the active options are written to the options file in the database directory.

Connection URLs

The functions and methods for connecting to a FrontBase have been supplemented with url based connections, there is a need to distinguish between connections to embedded databases or to a database running on a remote server.

The URL syntax:

The following url syntax is accepted

<URL> :== <schema>://[<user>][<host>][:<database>][/<path>] |
<database>[@<host>]

<schema> :== frontbase | file
<user> :== <name>[:[<database-password>:]<password>]@
<database> :== <name> | <port>
<path> :== <name> | <path>/<name>
<port-> :== ... number in the range from 1 to 65565
<name> :== ... anything not containing /

The passwords defaults to the empty password, and user to _system, and hostname to localhost. The database name defaults to the path name and vice-versa. The only meaningful host name when using the file schema is localhost. If the file schema is used then the path name has preferences and if the frontbase schema is used the database name has preference

Examples:

frontbase://john:secret:moresescret@db1.frontbase.com/m0

Connect to the FrontBase server m0 on the host db1.frontbase.com

Various forms for identical URLs

file:///foo.fb	same as file://localhost/foo.fb
file://:foo	same as file://localhost/foo.fb
frontbase://i8.local/foo.fb	same as frontbase://i8.local:foo.fb
foo@i8.local	same as frontbase://i8.local:foo.fb

Frameworks

The FBCAccess and FBAccess frameworks are now built with and install name @rpath/FBAccess.framework/Versions/A/FBAccess which enables applications to include the frameworks in the app. Set the LD_RUNPATH_SEARCH_PATHS in the application to include to include the directory where you placed the frameworks. A typical example would be @executable_path/../Frameworks.

iOS

The iOS libraries did contain a double definition of FBCColumnMetaData, which have been removed.

FBAccess

Methods for creating blobs from the contents of files and creating files with the contents of blobs have been implemented

The methods are part of FBDatabaseConnection class:

```
- (FBMetaData*)writeBLOBStreaming:(NSString*)path;
- (FBMetaData*)readBLOB:(FBBlobHandle*)handle into:(NSString*)path;
```

The use od NSHost for obtaining the host name of current host have been replaced with BSD gethostname to avoid dependencies to DNS configurations,

The method signature of the hash method in the FBAutoStartInfo class habe been updated to match its super class.

The readCLOB method is now return the object instead of nil.

The was a trailing ; in an include in FBDatabaseConnection.h, it has been removed.

FBCAccess

Functions for creating blobs from the contents of files and creating files from the contents of blobs have been implemented.

The functions is question are defined the FBCDatabaseConnection.h header file:

```
FBCMetaData*  fbcdcWriteBLOBFromFile
                (FBCDatabaseConnection*  self,
                 const char*             fileName
                );
FBCMetaData*  fbcdcReadBLOBToFile
                ( FBCDatabaseConnection*  self,
                  FBCBlobHandle*          blobHandle,
                  const char*             fileName
                );
```

The FBCAccess is build for 32 and 64 bits.

The error message from the connect functions and methods have been corrected.

The fbcdcDisconnectAll function, and friends, does not cause a crash anymore.

sql92

Commands for creating blobs from the contents of files and creating files with the contents of blobs have been added:

```
DEFINE BLOB @'<name>' FILE <file-name>;
```

DEFINE CHAR @'<name>' FILE <file-name>;
WRITE BLOB @'<blob-handle> FILE <file-name>;

The -d option now supports the URL syntax.

Acids can be specified directly in the stop and interrupt agent commands, in which case it is not necessary to execute a SHOW USAGE: command before the STOP/INTERRUPT command.

FrontBaseManager

The FrontBaseManager now connect correctly to databases on remote hosts.

The FrontBaseManager is now more robust against restarting FBExec's.

==========

FrontBase Distribution 5.2.16

==========

FrontBase version 5.2.16 is maintenance release and includes a number of bug fixes and additions.

FBCAccess

FBCAccess did a small amount of memory for each unique time zone that was used.

FrontBase Distribution 5.2.15

==========

FrontBase version 5.2.15 is maintenance release and includes a number of bug fixes and additions.

sql92

The command line tool, sql92, did not format values of floating point types and fixed point types with the precision and scale give at the definition.

The result set of selects causing warnings was not displayed.

FrontBase Distribution 5.2.14

==========

FrontBase version 5.2.14 is maintenance release and includes a number of bug fixes and additions.

FrontBase

In rare high load cases satisfies predicates could cause server crashes.

FrontBase Distribution 5.2.13

==========

FrontBase version 5.2.13 is maintenance release and includes a number of bug fixes and additions.

FrontBase

Satisfies predicates in connection with prepared statements will in rare cases crash the server.

Full precision of timestamps are written to the transaction log.

FrontBase Distribution 5.2.12

==========

FrontBase version 5.2.12 is maintenance release and includes a number of bug fixes and additions.

Improve memory performance for massive serialized rows update.

In rare situations, select top(a,b) did return incorrect number of rows.

FrontBase Distribution 5.2.11

==========

FrontBase version 5.2.11 is maintenance release and includes a number of bug fixes and additions.

Disallow the illegal statement PREPARE;

FrontBase Distribution 5.2.11

==========

FrontBase version 5.2.11 is maintenance release and includes a number of bug fixes and additions.

FrontBase

The sql statements

```
prepare; execute 0;
```

return a syntax error, instead of crashing the server.

FBExec

The create and start database commands does not return until the database server have been started is registered and ready to accept user connections.

FrontBase Distribution 5.2.11

==========

FrontBase version 5.2.11 is maintenance release with focus on bug fixes.

FrontBase

The schema was not set when triggers was created in a write all schema file.

The server DNS lookup in connection with the EXTRACT USAGE; command can be disabled by the option

-dnslookup=yes|no

Conversion from floating point literal to integer types may yield incorrect results if the literal was terminated in two or more zeros in the fractional part.

Windows installations

The tcp keep alive timer is now controlled with the two options

-tcpkeepalivetime=<seconds>
-tcpkeepaliveinterval=<seconds>

and if both values are non-zero tcp keep alive timers will be enabled for all client connections.

Files overwritten during execution of WRITE ALL is now truncated to length 0 before they are overwritten.

FrontBase Distribution 5.2.10

==========

FrontBase version 5.2.10 is maintenance release with focus on bug fixes.

FrontBase

select top (0,1) … return 0 or 1 row.

A number of performance improvements for sql involving sub-selects not having

outer column references used in where predicated.

The schema was not set when triggers was created in a write all schema file.

Linux installation

The /etc/init.d/FrontBase startup script now stop and start the service when restart is requested.

FrontBase Distribution 5.2.9

==========

FrontBase version 5.2.9 is maintenance release with focus on bug fixes.

FrontBase

The first blob manipulations after the server had started up could in rare high load situations cause a crash.

In rare high load situations the the server would crash when checking a users privileges.

The server might crash, if an exception did cause a transaction to roll back, and that transaction did have result sets which have not been exhausted.

The statement EXTRACT LICENSE is re-introduced for compatibility with existing drivers.

FrontBase Distribution 5.2.8

==========

FrontBase version 5.2.8 run without a license.

FrontBase

The option -ignorepasswords[=yes|no] has been added. When FrontBase is started with -ignorepasswords=no all password checks are ignored.

FrontBase Distribution 5.2.7

==========

FrontBase version 5.2.7 is maintenance release with focus on bug fixes.

FBCAccess

In high load situations the SET TIME ZONE sql statement may fail due to lack of resources on the server, the failure is reported back to the client as expected. However, the create session does set the time zone to the default time zone for

the client. In case of failure, that failure was not reported correctly.

FrontBase

SQL statements of the form values(set unique = <number> for <table> for replication; was not written to the transaction log.

FrontBase Distribution 5.2.6

FrontBase version 5.2.6 is maintenance release with focus on bug fixes.

FrontBase

The substring string function could cause the server to crash when the substring did not include a FOR clause and the string did contain several non ascii characters.

The sql statement values (select unique foo); was not transaction logged correctly, which could leaving the sequence counter out of sync when a database status was changed from replication client to be replication master.

The construct insert into foo...; delete from foo; insert into foo ...; could crash the server.

Insert a bunch of row and and delete all of them in one transaction would cause the size of the .fb file to grow beyond any reasonably size.

FrontBase Distribution 5.2.5

FrontBase version 5.2.5 is maintenance release with focus on bug fixes.

FrontBase

The command line option -preload[=yes|no] control the preload of table caches when the server is started.

The substring string function would in rare situations return an extra character when the original string contained non ascii characters.

Correlation names would incorrectly hide the table name in nested sub-selects, like select * from t0 where exists (select * where from t0 t1 where t1.c0 ='x' and t0.c1= items.original_id). t0.c1 was incorrectly hidden.

Select unique for negative values was not transaction logged correctly.

FrontBase Distribution 5.2.4

FrontBase version 5.2.4 is maintenance release with focus on bug fixes.

FrontBase

Fix a race condition in connection with 64-bit memory access on 32-bit buses.

FrontBase Distribution 5.2.3

FrontBase

Select count(*) in connection with join expression have been optimized.

Right truncations in connection with static size bit values could crash the server.

FrontBase Distribution 5.2.2

FrontBase version 5.2.2 is maintenance release with focus on bug fixes.

Query optimization of joins with more than 13 table have been improved, again.

Transaction logs was not continued when the server was started.

FrontBase Distribution 5.2.1m

FrontBase version 5.2.1l is maintenance release with focus on bug fixes.

FrontBase

Query optimization of joins with more than 13 table have been improved.

Certain, rare, combination of table values expressions and join operators would crash the server.

FrontBase Distribution 5.2.1l

FrontBase version 5.2.1l is maintenance release with focus on bug fixes.

FrontBase

Statistic counters for the raw device driver is now 64-bit.

Large fixed size any_type values was truncated.

The accumulated size of blobs and clobs columns in a table was not updated correctly and was limited to 4G.

Satisfies predicates was in rare situations rejected with error code 512: "SATISFIES predicate must be either the only expression or part of a top-level AND expression"

The identification of BLOBS and CLOBS was in rare cases duplicated on systems with realtime clocks with poor resolution.

The SIGPIPE signal could cause the server to terminate prematurely.

sql92

The show size command is now using 64-bit counters.

Frameworks and developer libraries

Large BLOBS and CLOBS, mega-bytes, could cause address faults when the connection to the database server was encrypted.

64-bits applications would not connect correctly to the database server when the connection was encrypted.

FrontBase Distribution 5.2.1j

FrontBase version 5.2.1j is maintenance release with focus on bug fixes.

FrontBase

The termination of FrontBase could is rare cases hang.

FrontBase Distribution 5.2.1i

FrontBase version 5.2.1i is maintenance release with focus on bug fixes.

FrontBase

The predicate of the form INDEX = <integer-expression> is now considered illegal when the server is started with the -rmaster, -rclient, or -rcluster options.

Rollbacks of transactions which have inserted blobs or clobs could cause the server to crash.

The FrontBase Distribution 5.2.1h

FrontBase version 5.2.1h is maintenance release with focus on bug fixes.

FrontBase
The servers robustness against improbably situations in connection with drop tables with implicitly created indexes which have been deleted.

Some extremely rare cases of reuse of common sub-selects cause the server to crash.

Some cases of joins with 12+ tables would cause the query optimizer use too much time to determine an evaluation order, the optimizer have been optimized.

FrontBase Distribution 5.2.1g
==========
FrontBase version 5.2.1g is maintenance release with focus on bug fixes.

FrontBase
The like operator did not return correct result in the case of a pattern beginning with
'%' followed by one character.

FrontBase Distribution 5.2.1f
==========
FrontBase version 5.2.1f is maintenance release with focus on bug fixes.

FrontBase
Certain like operators with escape specifications would not return always all matching rows.

FrontBase Distribution 5.2.1e
==========
FrontBase version 5.2.1e is maintenance release with focus on bug fixes.

FrontBase

Improve the error messages for failing constraint checks to include the failing column values when the column type is bit or bit varying.

Allow cast of CLOBs to CHARs.

Reading BLOBs and CLOBs before they were committed returned incorrect values and would cause incorrect values to be stored when the BLOBs/CLOBs were actually committed.

FrontBase Distribution 5.2.1d

==========

FrontBase version 5.2.1d is maintenance release with focus on bug fixes.

FrontBase

Reading blobs without having an open transaction could cause the server to crash.

FrontBase Distribution 5.2.1c

==========

FrontBase version 5.2.1c is maintenance release with focus on bug fixes.

FrontBase

Blobs were not validated correctly when the were created with an older version of FrontBase, the validation id not correct.

The unique number generator was exported with the default values, when export was done using versioned transactions, the correct unique number is now exported.

Several clients creating blobs simultaneously could cause FrontBase to crash, the crash have been fixed.

FrontBase Distribution 5.2.1b

==========

FrontBase version 5.2.1a is maintenance release with focus on bug fixes.

FrontBase

The option -loadrdd have been added to the FrontBase server instruction the server to to fill up the rdd cache, during startup. The option is intended for, but not restricted to, situations where the size of the rdd is in the same ball park as the size of the database file.

FrontBase Distribution 5.2.1a

==========

FrontBase version 5.2.1a is maintenance release with focus on bug fixes.

FrontBase, FBAccess, FBCAaccess, and sql92

The secure communication did not work corectly for the 64-bit versions.

FrontBase Distribution 5.2.1
==========

FrontBase version 5.2.1 is maintenance release with focus on bug fixes.

FrontBase
==========

Some combinations of _ and escapes in connection with like patterns was not interpreted correctly.

Users names containing spaces are now accepted,

Some like patterns with %'s would cause the server to crash.

FrontBase Distribution 5.2.0
==========

The 5.2.0 distribution can read 5.1 databases but will upgrade the database, once
upgraded the database is not compatible with the 5.1 releases.

FrontBase
==========
BLOB handles and CLOB handles can now be used across transactions.

LookSee is now working for all users.

FrontBase Distribution 5.1.3i
==========

FrontBase version 5.1.3i is maintenance release with focus on bug fixes.

FrontBase
==========

The evaluation of like expressions with escaped wildcard characters was yielding incorrect results when the escaped wild card character was not preceded by a wildcard character.

FrontBase Distribution 5.1.3h
==========

FrontBase version 5.1.3h is maintenance release with focus on bug fixes.

FrontBase
==========

When the port number specified with -port option was greater than 65535 it not registered correctly when with the FBExec, and the database was unreachable.

FrontBase Distribution 5.1.3g
==========

FrontBase version 5.1.3g is maintenance release with focus on bug fixes.

FrontBase
==========

The error message issued when a server is started for a database that are already
running have been improved.

Dropping or rolling back a newly created look see index would in rare cases crash
sthe server.

sql92
==========

The session name is now set correctly when using the -d option.

FrontBase Distribution 5.1.3f
==========

FrontBase version 5.1.3f is maintenance release with focus on bug fixes.

FrontBase
==========

The transaction number was exported incorrectly when exporting using the isolation
level versioned.

Collection of statistics on input output activity was not entirely correct on all platforms

FrontBase Distribution 5.1.3c

FrontBase version 5.1.3c is maintenance release with focus on bug fixes.

FrontBase

Triggers did in rare cases report incorrect semantic errors when fired.

FrontBase Distribution 5.1.3b

FrontBase version 5.1.3b is maintenance release with focus on bug fixes.

FrontBase

Simple delete operations on look see indexes could in rare cases cause inconsistencies in database disk storage.

The optimize database command could in rare cases cause inconsistencies in the
database file storage.

Change a disk zone for a table or an index could in rare cases cause inconsistencies in the database disk storage.

sql92

The sql92 command line tool now supports the -d <database-spec> option which cause sql92 to connect to the specified database. Use the following syntax for the
database specification:

```
<database-spec>  :==  <databasename>
                      <hostname>:<databasename>
                      <username>@<host-
name>:<databasename>
                      <username>@databasename
```

The default user name is _system and the default host name is localhost.

FrontBase Distribution 5.1.3a

FrontBase version 5.1.3a on bug fixes and optimizations.

FrontBase
==========

In rare circumstances was very small transactions not recored in the transaction log while a bacup was running.

The execution time of LookSee queries involving or operator on large tables have been improved.

FBCAccess library
==========

Circa dates and any types are now formatted correctly.

FrontBase Distribution 5.1.2
==========

Installation
==========

FrontBase version 5.1.2 correct an upgrade problem which left an already installed FrontBase exeutable intact, instead of linking it to either FrontBase32 or FrontBase64.

sql92
==========

Sql92 now interpret the "set default schema ..." correctly.

When a user is auto created the corresponding default schema is created with the same casing as the user name.

FrontBase Distribution 5.1.1
==========

FrontBase version 5 is a major upgrade which is NOT binary compatible with FrontBase 4. In order to migrate databases from FrontBase 4 to FrontBase 5 is is
necessary to perform an export of the data to the architecture neutral format with FrontBase 4, create a new FrontBase 5 database, and import the data again.

FrontBase 5 includes a number of new features of which the most important is

the full 64 bit mode, which greatly enhance the caching capabilities of the server. Consequently it is possible to specify rdd cache larger than 1800 M.

FrontBase 5 supports 2**60 rows per table.

FrontBase 5 offers improved performance when dealing with many small write transactions.

The FBExec is now started by launchd the replacement for startup items.

sql92 command line tool
==========

sql92 support an option for specifying a database on the command

-d <user-name>@<host-name>:<database-name>
-d <database-name>
-d <host-name>:<database-name>

The second form uses localhost and _system, the third form use _system as defaults

FrontBase Distribution 4.2.9
==========

This is a maintenance release with focus on bug fixes and optimizations.

FrontBase 4.2.9

The agent connection executing a WRITE BACKUP is now throttled such that the disk is not
monopolized, which could starve other agent connections.

The query optimizer has been improved wrt. dealing with IN predicates more efficiently.

The amount of RAM used for when evaluating stored procedures has been reduced.

Incorrect collation used in connection with a LIKE operator on the information_schema.users table.

Initially deferred constraints was not always checked in connection with inserts.

sql92 command line tool

The new sql92 command

THROTTLE <float>[s|%];

If s is specified the command will wait float seconds, if % is specified the command will wait such
that the overall load will be float percent. The command is intended to regulate the load on a server
during massive imports.

The set format format string have been enhanced such that blobs may be expanded with the %B flag
and the general printf format strings can be specified with %p<printf-format-string>. There is no checks
of the compatibility of the column values and the printf format string, so you may get unexpected results.

It also possible to refer to column by their label.

The complete SET FORMAT '<format>';:

```
  <format> :== %{<column>}{.<width>}<flag>
               %*{<width>}<flag>
               %{<width}[<template>|{<template>]
                %%
<column>   :== positive integer
               "<label>"
<width>    :== positive integer
<flag>     :==  c l T S n # r p
```

If the column or the width is omitted they are interpreted as 0.

The flags expands to

c : column value
l : column label
T : table name
S : scehma name
n : column name
: the row number in current result set
r : the result set number
B : expands to the standard sql92 blob definition for all blobs in

one row

for the column. The expanded string is padded with spaces to the width.

%{<width}[<template>|{<template>] iterates over all columns and expands the template for each. If the <column> is not spefied inside the iterator the current column is referenced. The first column is expanded using the first template and the following columns by expanding the second.

The %*<flag> from is a shorthand for %[%<flag>|,<%flag>].

The %% expand to %.

Other expansions:

\n	new line,
\r	carridge return,
\t	tab,
\ddd	to the decimal unsigned byte values,
\c	to the character.

JDBC 2.5.2

- Changed method FBJPreparedStatement.executeSetObject (Now it supports the Types.BOOLEAN sql type).
- Fixed bug in FBJPreparedStatement.executeSetObject (Exception when using the Types.DATE sql type).
- Added method FBJMetaData.isSelect (Returns true if statement is SELECT or UNIQUE).
- Added constructor FBJURL (Will initialize with java.util.Properties object).
- Changed method FBJDatabaseMetaData.getExportedKeys (Simplified CASE statement).
- Changed method FBJDataSource.getConnection (Prepare FBJURL with java.sql.Properties).
- Changed method FBJConnectionPoolDataSource.getPooledConnection (Prepare FBJURL with java.sql.Properties).
- Changed class FBJConnectionPoolDataSource (Imherits from FBJDataSource).
- Added support for enabling client side prepared statements in the connection url.

JDBC 2.5.1

- Added constructor FBJReplicationHandler (Will attempt to start the replicator if it is not running).

- Added method FBJReplicationHandler.isClient (Will check if a database is already a replicator client).
- Added method FBJReplicationHandler.resfresh (Updates the client information).
- Fixed bug in FBJClob (Handling of large Clobs incorrect).

Changes in FrontBase JDBC 2.5.0

- Added FBJReplicationHandler class (This class makes it possible to administer a replication setup through the driver).
- Fixed bug in FBJCallableStatement.execute (If procedure is a function the last result set will get returned).
- Fixed bug in FBJCallableStatement.parseSQL (Procedure name is trimmed due to bug in WebObjects).
- Added method FBJConnection.getDatabaseCreationVersion (Returns version of database that used to create database).
- Added method FBJConnection.getTransactionInfo (Returns the current transaction info).
- Fixed bug in FBJConnection.getDatabaseCreationVersion (Incorrect sort ordering).
- Fixed bug in FBJConnection.cancelFetch (Fetches were sometime canceled using invalid fetch handles).
- Added method FBJExecHandler.getLicenseInfo (Returns info on what is licensed and what is not).
- Changed method FBJExecHandler.close (Changed access modifier to public).
- Added method FBJErrorMessageTemplate.getSQLState (Returns a 5 digit string describing the exception).
- Changes FBJErrorMetaData constructors (Templates now loaded in instance initializer block).
- Changed method FBJStatement.executeQuery(Now it will accept updates but just return null).
- Added method FBJDatabaseMetaData.getCurrentSchema (Returns the current schema).
- Fixed bug in FBJMetaData.hasErrors (Checks on multiple metadata sets not performed correctly).
- Fixed bug in FBJResultSet constructor (Batch not fetched when handling EXTRACT statements).
- Fixed bug in FBJPreparedStatement.setBinaryStream (Saved data as BLOB but the API dictates
  the this data should be saved as BINARY data).
- Added method feature in FBJPreparedStatement.setObject (InputStream data can be saved as BLOB if Types.BLOB is specified).

FrontBase Distribution 4.2.8d

==========

sql92

The replicator command "CREATE CLIENT" is now building a new client database, which is did not in 4.2.8

The '>' letter have been removed from the sql92 output.

FrontBase Distribution 4.2.8
==========

INSERT, DELETE and UPDATE statements executed in a stored procedure called from a fired trigger were incorrectly transaction logged.

Certain BIT(96) to CHAR(24) CASTs could crash the server.

Incorrect optimization of a sub-select with outer column references could cause an infinite loop.

An INSERT privilege check related to a flat-file could crash the server.

"ORDER BY <negative number>" could crash the server.

Extremely (>255 characters) long procedure names could crash the server.

Specifying a cache setting when creating a LookSee index would cause the server to deadlock.

The replication mechanism failed when a blob handle was referenced more than once in a transaction.

SELECT -(SELECT SUM(c0) FROM t0) FROM t0; caused the server to crash.

Implementation of ON UPDATE CASCADE clause in a FOREIGN KEY constraint was incorrect in certain nested cases.

Interval day-time literals was incorrectly transaction logged.

Occurrence of transaction conflict in the replication management was incorrectly handled.

sql92

The new sql92 command

PROTECT [COUNT <positive-integer>] [DELAY <float-seconds>];

can be used to protect as sequence of sql statements stating with the PROTECT statement and ending with and including a COMMIT or a ROLLBACK statement against transaction conflicts. If a transaction conflict is detected during the execution
of the sequence of statements, the whole sequence is executed again after DELAY
seconds. That process is repeated at most COUNT times. The default for DELAY is
10 and and the default for COUNT is 3.

FrontBase Distribution 4.2.7
==========

FrontBase 4.2.7

Constraints with ON DELETE SET NULL could cause excessive null constraint violations.

Deleting from updateable view could delete too many rows.

The resulting type of null fixed string in a row value construction was 0, causing a data exception to be raised.

The EXTRACT MONTH did incorrectly include timezone information.

SELECT UNIQUE used within stored procedures was not transaction logged correctly.

Updates of tables when using versioned isolation level was not reported.

PH: The deletion of all rows in a table have been optimized.

PH: A looksee index may now caontain more than one row. Collations. Syntax.

SELECT COUNT(*) have been optimize when used in joins.

sql92

It is now possible to throttle the sql92 load on the server with the command:

THROTTLE <float>[%ls];

If s is specified the command will wait the specified seconds. If x% is specified it will
wait (100-x)% of the real time elapsed since the last execution of the throttle command.

The numbering of agent connections in the show usage is now unique.

FrontBase Distribution 4.2.5

This is a maintenance release with focus on bug fixes and optimizations.

FrontBase 4.2.5

The long term stability have been improved, and performance on multi processors
have been improved.

FBExec 4.2.4

The long term stability have been improved.

FBWeb Manager

The web manager is not included in the distribution anymore. It has been replaced
by the FBJmanager which is available from our download page.

FrontBase Distribution 4.2.4

This is a maintenance release with focus on bug fixes and optimizations.

FrontBase 4.2.4

A bug that under high load by multiple active clients could cause a crash has been fixed.

FBExec 4.2.4

The long term stability have been improved.

Sql92

Timestamp literals is now displayed with the proper seconds precision.

FrontBase Distribution 4.2.3

==========

FrontBase 4.2.3

This is a maintenance release with focus on bug fixes and optimizations.

A few bug related to architecture specific alignment have been fixed, that in rare cases would cause a crash.

FrontBase Distribution 4.2.2

==========

FrontBase 4.2.2

This is a maintenance release with focus on bug fixes and optimizations.

The expression evaluation precedence is now determined correctly for expression including row value constructors.

The query optimizer now evaluates expressions involving UNION and OR operators correctly.

The flat file importer will always terminate.

FrontBaseManager

The FrontBase manager will only display the release the first time it is launched.

FrontBase Distribution 4.2.1

==========

FrontBase 4.2.1

FrontBase 4.2 includes an improved query optimizer. Most queries should execute as fast in earlier releases, but for some multi-table queries with joins, the previous query optimizer made some less than optimal choices which could result in slower than expected execution times. The new query optimizer should

make the correct choices in all cases.

Various minor bugs, mostly related to an earlier SELECT COUNT(*) optimization, have been fixed. 4.2 also contains the beginnings on an implementation of TRIGGERs which is planned enabled with 4.3.

FrontBaseJManager 0.505 (Beta)

The java based FrontBase manager. On Mac OS X its wrapped as an application but the jar runs on Windows and Linux as
well.

Requirements

J2RE 1.4 or later.
FrontBase JDBC driver (included).

Installation

Make sure the frontbasejdbc.jar (JDBC driver) file can be located in your Java CLASSPATH or is in the Java extensions folder. If you are using an existing version of the JDBC driver then please update it with the version that ships with this release of FrontBaseJManager.

To use the backup management pane with databases running on Mac OS X you must replace the current version of the FBExec with the version contained within this release. To do this, stop the running FBExec process, replace the file in the bin folder of your installation and then restart the process.

The FrontBaseJManager.jar file can be double clicked to start the application. You can also start the application from the command line by using,

java -jar FrontBaseJManager.jar

Documentation

Currently there is no documentation but the most of the functionality follows from that provided in FrontBaseManager.

Backup Management Pane

Currently this functionality is only supported by databases on Mac OS X running the version of FBExec which is available as part of this release. To update the FBExec stop the running version and replace the file in the bin folder of your

installation.

From this pane you can only list backups contained in the default location, i.e. the Backups folder of the FrontBase installation. Only backups created with the default filename in the form B_YYYY_MM_DD-HH_MM_SS will be listed. When a database is selected then it's backups are displayed in order of the most recent. You can create, restore and delete backups from this panel. Also the list will mark backups which have expired either because they are older than the backup expiry limit or older than a specified date. You can then delete all expired backups. These expiry settings are configurable from the preferences panel. By default no backups expire.

Bugs and Comments

Please report any new bugs or comments to colin@frontbase.com.

Certification

Mac OS X 10.4.3	J2RE 1.4.2_09/1.5.0_02
Win2000	J2RE 1.5.0_05
WinXP	J2RE 1.5.0_05
RedHat 9 (GNOME)	J2RE 1.5.0_05

Bug fixes in 0.505

Auto refresh in extract usage doesn't terminate when connection frame is closed.

Single quotes in table content update and insert row are not escaped.

Enhancements in 0.504

Added support for extract/describe commands.
Double click on table column definition will open edit pane.
Finished menu shortcuts.
Added printing support.
Added menu bars to all frames on non-Mac platforms.
Added fetch limiting and batching to result and content panel.
Added Backup Management panel and expiry options to Preferences.

Bug fixes in 0.504

There is a pending transaction which needs removing before a table is first edited.

Create Full Text Index should only list char columns.
Create Full Text Index misaligns with long column names.
Insert Table Row should place edit focus on first value.
Create Check constraint. Remove scrollpane on text area.
Double click Upload CLOB after already selected value will toggle entry to NULL. (Windows)
Collation pane does not hide/close when collation created.
Creating a procedure relists functions in objects pane not procedures.
Procedures listed on schema object pane does not disabled Open Content button.
Create view panel should only enable CREATE button once VIEW is specified.
Can't clear invalid options from Start Database Advanced.
Usage Pane. Refresh clears refresh box.
Drop table is not prompting to alter transaction mode.
Null binary columns are returned as <null>.
Connecting on port should display database/catalog name in connection pane title.
Procedure panel does not resize when containing window is resized.
Scroll bar on error dialog is not drawing when line count is 10.
Enter on running database now acts as double click and prompts for connect.
View definition columns are not displayed in order of definition but alphabetically.

Enhancements in 0.503

Improved look and feel support on Linux.

Bug fixes in 0.503

Can't connect direct to port.
Can't open database unless database name selected.
Connecting to an unmonitored database caused dummy entries in monitored databases.
License pane in connection frame will not redisplay.
Usage pane does not list null connections.
Usage entries are in the wrong order.
Create Full Text Index display problems on XP.

Known Bugs

Some tables do not support moving columns.
Table rendering has problems with last column not being expandable.
Uploading CLOBs is restricted by available memory.

Enter autostart management for the first time. Sometimes asks to save changes, even when none occur.
Horizontal scroll bars on some tables are redundant.
New Schema pane; components are misaligned (Windows).
Specify hours, minutes, seconds on usage pane times.
Specify MB on BLOB/CLOB sizes.
Result set binary/LOB values are not dereferenced as in the content pane.
Start database/advanced/stop should be disabled for an unknown database.
New database should display any errors.
Preference Panel is sometimes not added and removed from windows menu.
Open, New and Restore database are only active from main monitor window.
On non-mac platforms Null and CLOB values should be highlighted in white when a row is selected.
Transaction duration is always NONE on usage pane.
The viewport containing the privilege table for views, procedures and functions should match the tables background (Windows).
New view, procedure and function panel should be larger, resizeable and omit horizontal scrollbar.

FrontBase PlugIn 2.6.3

Fixed bug in framework (FrontBasePlugIn object corrupted).

FrontBase Distribution 4.1.16
==========

This is a maintenance release with focus on bug fixes and optimizations.

The memory management system have been adapted to enhance performance on multiprocessor system.

Certain not too complicated SELECT COUNT(*) statements with count(*) have been optimized.

The comparison of NaN's was not ordering, causing index lookups, in some cases, to return incorrect results.

The server now logs connection details when it receives an unknown command from a client.

Stored procedures was not always written correctly to the schema file during a WRITE ALL.

Use of CONVERT now generates a completion condition instead of causing the server to loop or crash.

Certain attempts at index optimizing failed when "<column> = <literal>" was combined with a join (<column> = <column>).

Altering the data type of a column could in rare cases render indexes unusable.

LEFT is now allowed as an identifier

In contrast to comparison predicates, the LIKE predicate should not use the collation pad specifications, the collation is typically specifying space padding.

If e.g. UPDATE statements was used with the ExecuteQuery method in JDBC (and the similar mechanism in WO 4.5), the protocol between the client and server would be broken leading to "strange" results and exceptions on the client side.

FrontBaseManager 4.4.5

Some special characters were not allowed in the sql view.

FrontBaseManager 4.4.4

When creating named constraints, the schema name is now added to the constraint name.

FrontBaseManager 4.4.3

A number of windows used when editing a table definition have been made resizable.

FrontBaseManager 4.4.2

When importing data, the list of tables available was not updated between imports. This has been corrected.

FrontBaseManager 4.4.1

When clicking the License option in the connection pane, no information was displayed. This has been corrected.

FrontBaseManager 4.4

FrontBaseManager now saves size and position of opened windows.

FrontBase PlugIn 2.6.2

JDBC driver update

FrontBase Distribution 4.1.15
==========

This is a maintenance release fixing one compatibility problem related to the latest version of the JDBC driver and the FrontBase server. Upon a fetch the JDBC driver is test the flag telling that the result set is exhausted, but the flag was not always set correctly in the first batch.

FrontBase Distribution 4.1.14
==========

This is a maintenance release with focus on bug fixes.

The current distribution for Mac OS X has built using 10.3.5.

FrontBase 4.1.14

FETCH NEXT FROM <cursor>; caused an infinite loop in the server.

LookSee now works properly on UTF8 (Unicode) encoded source strings.

Certain combinations of multi-column indexes, literal values and join columns could cause incorrect result sets.

INSERT INTO <table>(<column>) ...; caused the server to crash if the <column> didn't exist in the <table>.

Very long CHECK constraints (<expr> AND <expr> AND ...) would cause crashes or memory thrashing if checking
the constraint led to an error.

PREPARE ALTER TABLE <table> SET INDEX PRESERVE TIME; EXECUTE 0; caused the server to crash or
yield "weird" errors.

Variables weren't always used for index lookups in certain queries.

INSERT INTO T0 SELECT ... FROM T0 WHERE ...; has been optimized so table scans are avoided in more cases.

Incorrect maintenance of an implicit result set cache could lead to incorrect result sets.

Certain complex JOIN and RIGHT OUTER JOIN queries could crash the server.

sql92

Sql92 now supports cluster management with the following commands:

Set Default Database
SET DEFAULT DATABASE <database-name> [ON | @ | HOST <host-name>]
[DATABASE PASSWORD '<password>']
[PASSWORD '<password>']
[HOST PASSWORD '<password>'];
Identify the specified database as the "default" database. The default database identifies the master database for replication commands, and a (n arbitrary) cluster member for clustering commands. Any necessary password may be specified.

Show Default Database
SHOW DEFAULT DATABASE;
Show the current default database.

Create Cluster
CREATE CLUSTER [OPTIONS <option> ...];
Create a cluster consisting of the default database (only). If the default database does not
exist, a new database is created. Then a server for the default database is started, supplementing
the -rcluster option with the specified options, if any.

Start Cluster
START CLUSTER;
Attempt to start servers for all members of the cluster identified by the default database.
The servers will be started in the proper order.

Stop Cluster
STOP CLUSTER;
Attempt to stop any running servers for all members of the cluster identified by the default
database.

Show Cluster

SHOW CLUSTER [DESCRIPTOR | ALL];
Display properties of the cluster identified by the default database.
DESCRIPTOR, if specified,
directs certain static information about the cluster to be displayed whereas ALL, if specified,
requests further dynamic information to be displayed.

Set Cluster Descriptor
SET CLUSTER DESCRIPTOR;
Distribute the cluster descriptor associated with the cluster member identified by the default
database to all members identified by said cluster descriptor. After this operation, all cluster
descriptors of all members of a cluster should be consistent (provided that all members were
accessible), and that is exactly the purpose of this command.

Add Member
ADD MEMBER <database-name> [ON | @ | HOST <host-name>] [OPTIONS <option> ...];
Add the specified database as a member to the cluster identified by the default database. If
the specified database does not exist, a new database is created, and it must be possible to
bring the new member up-to-date from existing members of the cluster. A server for the
new member is started, supplementing the -rcluster option with the specified options, if any.

Remove Member
REMOVE MEMBER <database-name> [ON | @ | HOST <host-name>];
Remove the specified database from the cluster identified by the default database.

Start Member
START MEMBER <database-name> [ON | @ | HOST <host-name>] [OPTIONS <option> ...];
Attempt to start a server for the specified cluster member, supplementing the -rcluster option
with the specified options, if any.

Stop Member
STOP MEMBER <database-name> [ON | @ | HOST <host-name>];
Attempt to stop a running server for the specified cluster member.

Show Member

SHOW MEMBER <database-name> [ON | @ | HOST <host-name>]
[DESCRIPTOR | ALL];
Display properties of the specified cluster member. DESCRIPTOR, if specified, directs certain
static information about the cluster to be displayed whereas ALL, if specified, requests further
dynamic information to be displayed.

Set Member
SET MEMBER <database-name> [ON | @ | HOST <host-name>] MAJORITY
TRUE | FALSE;
SET MEMBER <database-name> [ON | @ | HOST <host-name>] READONLY
TRUE | FALSE;
Set the specified property to the specified value for the specified cluster member. The explicit
setting of the readonly property to FALSE opens for the possibility to create inconsistencies
between cluster members, and should only be done by someone knowledgeable.

FrontBase PlugIn 2.6.1

Fixed bug in FrontBasePlugIn.createTableStatementsForEntityGroup (Bug in SQL generation).

JDBC 2.4.7

Fixed bug in FBJConnection.getCatalog (Certain result set not released properly).

Added method FBJConnection.getSessionTimeZone (Returns the current session time zone).

Added method FBJDatabaseMetadata.getDatabaseCreationVersion (Returns the version of the server which was used to create the database).

Changed method FBJDatabaseMetadata.getDriverSubVersion (Changed access modifier to public).

Fixed bug in FBJStatement.execute (Certain result set not released properly).

Fixed bug in FBJExecHandler.startDatabaseWithOptions (options not written to .options file).

Fixed bug in FBJExecHandler.getOptionsForDatabaseNamed (incorrect result

when .options file empty).

Fixed bug in FBJMetaData.isFetchDone (caused unnecessary cancelFetch to be called).

Changed FBJConnection.stopDatabase (Added specific exception handling).

Fixed bug in FBJResultSet.FBJResultSet (fetchBatch called on Extract statements).

FrontBaseManager 4.3

It is now possible to manage clusters.

Replication management has been improved.

From the SQL Interpreter panel the 'Execute File' now recognizes files with extensions .sql, .txt and .text.

The panel to stop a database has been simplified.

The panel to connect to a database by double-clicking at it has been simplified.

You can now remove a reachable host from the host list in the monitor panel.

When specifying -sdisk as a startup or create option you must specify the path to the key file. This field has bee added to the panel.

t is now possible to see the startup options for the selected database by clicking the 'Show Options' in the Database menu.

It is now possible to create and delete domains and use them in column definitions.

Sometimes when editing schema objects and an error was encountered, no further changes could be made until you made a new connection to the database. This has been currected.

When viewing table content and adding a where clause, you can now use the ESC button to refetch objects.

When editing table definitions the order of the columns is the order retrieved from the database server. Now you can sort the columns by their name by clicking the header of the 'Name' column.

When updating table definitions you would have manually end the editing for the

update to take effect. This has been changed.

The sql for retrieving privilege information has been optimized.

he two columns: 'Transaction #' and 'Calls' in the usage pane can now be resized.

FrontBase Distribution 4.1.6
==========

This is a maintenance release with focus on bug fixes.

LookSee has been enhanced to support COLLATIONs properly. Further documentation will follow, for urgent needs, please e-mail to support@frontbase.com.

Using the keyword AT as an identifier could cause the server to crash.

Outer column references in an IN predicate could lead to incorrect result sets.

Place holders (?) would in certain cases lead to incorrect error messages when a prepared
statement was executed ("Missing value for place holder").

In certain situations a query running in REPEATABLE READ, OPTIMISTIC could become
very slow if other connections were accessing the same tables, as referenced in the query,
simultaneously with the query.

CASTing character strings to CLOB could cause the server to stop.

<column> LIKE <expr> COLLATE <collation> would incorrectly use an index on <column>.

SET SESSION AUTHORIZATION <user> + ROLLBACK caused an incorrect state in the server.

HAVING <column> <op> <expr> didn't work properly.

VALUES(CAST (TIMESTAMP '2002-01-01 23:59:59-07:00' AS DATE)); returned DATE '2002-01-02'.

CAST(<column> AS FLOAT) * 86400 the calculated FLOAT value wasn't properly truncated wrt. mantissa.

DROPping a user with granted privileges didn't drop the privileges.

In extremely rare cases the evaluation of a range predicate would give an incorrect
result.

DELETE ... WHERE INDEX IN (SELECT ...) was incorrectly transformed to ... INDEX = (SELECT ...)

SELECT DISTINCT C0*2.54/300 ... caused incorrect "Data Exception - Numeric value out of range" messages.

In certain cases, GROUP BY select statements would return a row count of 1 as part of the metadata for the select.

A connection using the isolation levels, read committed and repeatable read would
during table scans cause unnecessary dead lock situations causing the transaction in question to fail.

In certain situations the restored databases containing blobs would leave the blobs
inaccessible. The problem with the restore have been solved, and the server will not leave the blobs inaccessible even if the database was restored by a previous version.

Privileges for for dropped users were inherited by a newly created user.

Restore of a backup with blobs would set the size of the blob incorrectly, causing the blob to be inaccessible.

JDBC 2.4.7

Fixed bug in FBJConnection.getCatalog (Certain result set not released properly).

Added method FBJConnection.getSessionTimeZone (Returns the current session time zone).

Added method FBJDatabaseMetadata.getDatabaseCreationVersion (Returns the version of the
server which was used to create the database).

Changed method FBJDatabaseMetadata.getDriverSubVersion (Changed access

modifier to public).

JDBC 2.4.6

Fixed bug in FBJStatement.execute (Certain result set not released properly).

JDBC 2.4.5

Fixed bug in FBJExecHandler.startDatabaseWithOptions (options not written to .options file).
Fixed bug in FBJExecHandler.getOptionsForDatabaseNamed (incorrect result when .options file empty).

JDBC 2.4.4

Fixed bug in FBJMetaData.isFetchDone (caused unnecessary cancelFetch to be called).

JDBC 2.4.3

Changed FBJConnection.stopDatabase (Added specific exception handling).
Fixed bug in FBJResultSet.FBJResultSet (fetchBatch called on Extract statements).

FBWebManager

The FBWebManager have been overhauled.

FrontBaseManager 4.1

Sometimes when editing schema objects and an error was encountered, no further changes could be made until you made a new connection to the database. This has been corrected.

When viewing table content and adding a where clause, you can now use the ESC button to refetch objects.

When editing table definitions the order of the columns is the order retrieved from the database server. Now you can sort the columns by their name by clicking the header of the 'Name' column.

When updating table definitions you would have manually end the editing for the update to take effect. This has been changed.

The sql for retrieving privilege information has been optimized.

The two columns: 'Transaction #' and 'Calls' in the usage pane can now be resized.

FrontBase Distribution 4.1.2
==========

This is a maintenance release with includes fixes for a few bugs.

FrontBase 4.1.2 - Bug fixes

Updating columns with normalized strings, enabled with
ALTER TABLE <table-name> SET COLUMN MATCH (<column-name-list);
caused the server to crash.

The column normalization was not included in schema created by WRITE ALL.

FrontBase Distribution 4.1.1
==========

Introduction

FrontBase 4.1.1 represents the newest major upgrade to FrontBase and continues the established trend with significant new features being added in a mature fashion. Due to time constraints, there is no new version of the FrontBaseManager included as well as no detailed documentation of the new features. We are working on remedying this situation, but various customers have kindly asked us to release an official 4 build now.

*Please observe that there is **NO** binary compatibility between FrontBase 3.x and FrontBase 4.x databases (see below).*

All your existing applications that works with FrontBase 3.x, should work unmodified with FrontBase 4.x, i.e. there are no client side library changes.

The most significant new or enhanced features in FrontBase 4.1.1 are:

o 64-bit I/O system. The maximum database size is now 1,000,000 peta-bytes.
A side-effect of the new I/O system is a feature called FrontBase Advanced
Device Management ("table space" is a term another vendor is using), that
allows you to distribute the various parts of a database across multiple

hard drives and controllers.

o LookSee. Our full-text search feature has undergone a dramatic change from
3.x. LookSee is now not only much faster, but also far more scalable.

o Compressed backups. During generation of a backup, the resulting file
can be compressed, offering a significant space reduction:

```
WRITE BACKUP [TO '<path>] [COMPRESSED];
```

A restore can naturally be done directly from a compressed backup.

o Flat-file export. The WRITE ALL flat-file export is now up to 5 times faster than in 3.x.

o Transaction log. The transaction log and associated mechanisms, e.g. rollforward,
has been made more robust and flexible. Transaction logs are also encrypted (if the
database itself is encrypted).

Some of the new features that will be introduced with later versions of 4.x are:

o TRIGGERs. The full SQL 4 syntax and semantics of TRIGGER are currently being
implemented. An expected release time will be early Q2 2004.

o Hierarchcial SELECTs. The full Oracle syntax and semantics for hierarchical
SELECTs are currently being implemented. An expected release time will be
sometime Q2 2004.

Installation and upgrade

If you are installing FrontBase 4.x on a system with no previous FrontBase installation or if you don't care about existing FrontBase 3.x databases, read no further. Simply install FrontBase 4.x and start using it.

If you are currently using FrontBase 3.x and wish to migrate existing databases, you will need to export each database into a flat-file version using the WRITE

ALL statement (connect as _SYSTEM):

```
    WRITE ALL OUTPUT(DIR = '<path>', TYPE = 'FrontBase',
CONTENT = TRUE);
```

It could be a good idea to generate a backup prior to the WRITE ALL:

```
    WRITE BACKUP;
```

Install FrontBase 4.x. For each database you want to migrate, you will need to follow these steps using the sql92 command line tool:

1) Create a new database with the same name (or a different name per your liking). It may be necessary to specify the -create option to create a new database unconditionally.

2) Connect as _system and execute this command:

   ```
   script <path>/schema.sql;
   ```

Once the script command has completed, the database should now be in 4.x format. If you have any problems upgrading to FrontBase 4.x or migrating databases, feel free to send an e-mail to support@frontbase.com.

The commands to sql92 are:

```
create database <database-name> options -create;
connect to <database-name> user _system;
script <path>/schema.sql;
```