FrontBase 8 Linux

1 Introduction

FrontBase is a high performance, scalable, SQL 92 compliant relational database server created in the for universal deployment.

On Linux FrontBase can run as a service available to clients across the network. FrontBase can be embedded into the applications, providing the full FrontBase functionality to the application without the network complexity. Once embedded into you application it you can expose the database functionality to users on the network.

2 The FrontBase installation

2.1 The FrontBase server

The components needed for a FrontBase server installation are installed in the FrontBase installation directory, /usr/local/FrontBase, the executables in the bin subdirectory, the auxiliary files in the Library, the Translations, and the Collations subdirectories.

2.1.1 /usr/local/FrontBase/bin/FBExec

The FBExec is a service background process that is started when the computer is restarted. The FBExec provides status, management and access information to the client applications. The FBExec is started by the launch daemon by a launch daemon plist, com.frontbase.fbexec.plist, which is installed in /Library/LaunchDaemons. The plist does contain absolute references to the FBExec executable, so the plist must be updated if the FBExec is moved.

2.1.2 /usr/local/FrontBase/bin/FrontBase

FrontBase is the database server executable. There is a FrontBase process running for each running database. A FrontBase process is typically started by various management tools, but can also be started from the command line. Clients communicates with the FrontBase server across a TCP/IP connection established to a listen port created by the server.

2.1.3 /usr/local/FrontBase/bin/FBReplicator

The replication daemon distributing transactions from a replication master, to its replication clients.

2.1.4 /usr/local/FrontBase/bin/FBTLogs

A command line tool for summarizing the transaction log of a database.

2.1.5 /usr/local/FrontBase/bin/FBTLog

A command line tool for displaying parts the transaction log of a database as SQL statements.

2.1.6 /usr/local/FrontBase/bin/FBInfoCenter

A command line tool which prints out details about your FrontBase installation and your machine.

2.1.7 /usr/local/FrontBase/bin/FBBackup

A command line tool for creating a backup of a FrontBase database without the need to start the server.

2.1.8 /usr/local/FrontBase/bin/FBRestore

A command line tool for restoring a backup of a FrontBase database.

2.1.9 /usr/local/FrontBase/bin/FBListContent

A command line tool for verifying the the content of a FrontBase backup.

2.1.10 /usr/local/FrontBase/Translations

Translation are unicode character to unicode character mappings. Translanslation are created and maintained by the FBUnicodeManager. By default two translations are installed, ToLower.trans and ToUpper.trans.

2.1.11 /usr/local/FrontBase/Collations

Collations specifies a named ordering of unicode character. A collation is create and maintained by the FBUnicodeManager. By default the collations CaseInSensitive.coll1 is installed.

2.1.12 /usr/local/FrontBase/Library

Contains bootstrap files used when a database is created, and error message definitions..

2.2 The FBCAccess developer library

The FBAccess developer library is C client API but in the form of a static library. The developer library is installed in /usr/local/FrontBase/lib directory and corresponding headers are installed in the /usr/local/FrontBase/include directory. The developer library is also build as as a shared object file.

2.6 The sql92 command line tool

A command line tool for managing and accessing databases. sql92 will let you create, start stop, delete databases, connect to databases, and execute SQL statements.

The sql92 command line tool is installed in the base installation bin directory, the executable is installed in /usr/local/FrontBase/bin/sql92 and auxiliary files describing 8 bit character set are installed in /usr/local/FrontBase/Library/*.ucm.

3 The innards of a FrontBase database

A FrontBase database is a wrapper containing all dynamic data pertaining to a FrontBase database. The extension is .fb by default but can be chosen freely.

3.1 The database file

The database file contains the current state of the database, and is named FrontBase.fb. You cannot copy the database file when a server or an application is accessing it, the server and applications will access and update the data in in random order, thus making it impossible for sequential copy to guarantee correctness.

3.2 The tlog directory

FrontBase may be requested to maintain a transaction log, which records all changes made to the database in the for of SQL statement. The transaction log is a directory and is named TLog.tlog. The transaction log provides the necessary redundancy such that database may be recovered even in case of catastrophic failure. The transactions in the transaction log is also read by the FBReplicator when it push changes to its clients.

3.3 The bck directory

FrontBase may be requested to produce a backup, and the backups are per default written to the backup directory. The backup directory is named Backups.bck

3.4 The log directory

The log directory contain logs files, and is names Log.log

3.5 The options file

If present contains the options to be used when the database is started without any explicit options, the options files is named Options.opt. The content of the file is updated when the FrontBase server is started with options. You can edit the file as it is a simple text file. If this options does not exist an attempt to read the options form the / usr/local/FrontBase/etc/Options.opt file.

4 Download and installation

The latest version of FrontBase can always be downloaded from the www.frontbase.com.

If you have downloaded the .rpm install run rpm -i to install.

If you have dowloaded the .tar install, unpack the installation (tar xf) and run the install.sh command procedure. The .tar install works for most unix flavor. You are wellcome to peek.

Please note that the FBExec will be started the computer is restarted. The FBExec is also started as part of the installation process, i.e. there is no need to restart the computer after installation.

If the /usr/lib/systemd directory exist the FBExec will installed a systemd service otherwise the System V init script will be used.quit

5 Post installation

Once you have installed FrontBase it is a good idea to include /usr/local/FrontBase/bin in the search path of the various accounts that will use the FrontBase tools. This step is not necessary, but it can make your command line tool life easier.

It is always a good idea to verify that the FBExec process is running by doing (from a terminal window):

```
ps axc l grep FBExec
```

which should produce output like the following example:

```
374 ? S 0:00 FBExec
```

If for some reason the FBExec isn't running, try to launch it from the command line:

```
/usr/local/FrontBase/bin/FBExec &;
```

In case of problems, you can e-mail mailto:support@frontbase.com and ask for directions.

6 sql92 - super brief introduction

sql92 is an interactive command line tool which allows you to manage and access all databases on the local machine and available on the network. sql92 is launched by issuing the following shell command:

```
/usr/local/FrontBase/bin/sql92
```

To create and start a database:

```
sql92> create database firstdb;
```

To connect to a database:

```
sql92> connect to <database-name> [on <host-name>] [user <user-name>];
```

If you omit the host name, the current host name is used, and if the user name is omitted, the _system.

Always terminate sql92 commands with a ';', the command is not send to the server before the ';' is entered.

Quit sql92 by entering quit<enter> or by pressing Ctrl-D.

The sql92 help command:

```
sql92> help;
```

will print a summary of the available sql92 specific commands.

7 Getting started

Small examples on how to get started

7.1 URLs

When a connection to a database is established the database and connection method is identified by a URL with the syntax;

```
<URL>       :==  <schema>://[<usert>][<host>][:<database>][/<path]  |
                 <database>@<host>
<user>      :==  <user>[:[<database-password>:]<user-password>]@
<schema>    :==  frontbase | file
<database>  :==  <name> | <port>
<path>      :==  <name> | <path>/<name>
<port>      :==  ...  number in the range from 1 to 65565
<name>      :==  ...  anything which is not a number and does not contain /
```

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

7.3 sql92 command line tool

First open a terminal window and launch the sql92 tool:

```
/usr/local/FrontBase/bin/sql92
```

Enter the following sql92 commands (the semicolons are important):

```
create database firstdb;
connect to firstdb;
values(server_name);
disconnect current;
stop database;
delete database firstdb;
```

Congratulations! You have just created a database, connected to it, created a new user, committed a transaction, and deleted a database. It is as simple as that!

7.4 sql92 command line tool, embedded server

First open a terminal window and launch the sql92 tool:

```
/usr/local/FrontBase/bin/sql92
```

Enter the following sql92 commands (the semicolons are important):

```
connect to file://localhost/./seconddb;
values(server_name);
disconnect current;
quit;
```

Congratulations! You have just created a database with the server embedded in the sql92 tool. The database is created in current directory, you verify that with the command

```
ls seconddb.fb
```

8. Transactions - please read

A transaction is a grouping of a sequence of SQL statements into a unit; execution of the unit as a whole either succeeds completely, or, if one of the commands fails or is a ROLLBACK command, the database is left unchanged.

Transactions are initiated automatically by the server when needed. Most SQL statements need a transaction. A COMMIT or a ROLLBACK command ends the current transaction.

Most of the FrontBase tools use, as default transaction settings, SERIALIZABLE, WRITE, PESSIMISTIC, which has the consequence that access to referenced tables is effectively serialized.

The user may change the defaults according to the SQL 92 specification.

When you are working with the FrontBase tools, you should always remember to COMMIT or ROLLBACK transactions in order not to block access to the database.

If you try to access a table and it seems that your application or tool doesn't do anything, it is most likely waiting for some other session to COMMIT or ROLLBACK an active transaction. Neither your application/tool nor the FrontBase server is hanging!

FrontBase offers a feature called AUTO COMMIT, i.e. a way to tell the server that it should automatically commit each statement that is successfully executed. The sql92 command line tool uses, as default, this feature.

You can control the AUTO COMMIT feature by executing the following statements:

```
SET COMMIT FALSE; -- Turns off AUTO COMMIT
SET COMMIT TRUE;  -- Turns on  AUTO COMMIT
```

10 Documentation

The documentation for the FrontBase database server can be downloaded from the http://www.frontbase.com and click the documentation tab.

Please take the time to peruse this documentation, so you at least have a good feeling for what's in there.