

Installing the ODBC drivers for Firebird

: 13.01.2022

This article is part of a series that includes SQLite, Postgresql, Microsoft SQLServer, Oracle RDBMS, HSQLDB, MariaDB, MongoDB, and Excel. The goal is to set up a self-standing environment for testing an ODBC extension for gawk presented here *to be completed*. Refer to [SQLite](#) for installing the required ODBC Driver Manager.

The test system is a debian v11 (bullseye).

As root, install the Firebird RDBMS and its ODBC drivers from the official repositories:

```
1# apt install firebird3.0-server
```

A systemd service was set up and launched:

```
1
2 # systemctl status firebird3.0.service
3 • firebird3.0.service - Firebird Database Server ( SuperServer )
4 Loaded: loaded (/lib/systemd/system/firebird3.0.service; enabled; vendor preset:
5 enabled)
6 Active: active (running) since Mon 2021-10-18 15:00:25 CEST; 50min ago
7 Process: 546 ExecStart=/usr/sbin/fbguard -daemon -forever (code=exited,
8 status=0/SUCCESS)
9 Main PID: 576 (fbguard)
10 Tasks: 4 (limit: 4659)
11 Memory: 10.1M
12 CPU: 75ms
13 CGroup: /system.slice/firebird3.0.service
14 └─576 /usr/sbin/fbguard -daemon -forever
15   └─577 /usr/sbin/firebird
16 Oct 18 15:00:23 debian systemd[1]: Starting Firebird Database Server ( SuperServer )...
17 Oct 18 15:00:25 debian systemd[1]: Started Firebird Database Server ( SuperServer ).
```

The service runs as the newly created firebird account:

```
1# ps -ef | grep firebird
2firebird 28053 1 0 15:15 ? 00:00:00 /usr/sbin/fbguard -daemon -forever
```

```
3firebird    28054    28053    0 15:15 ?          00:00:00 /usr/sbin/firebird
```

Create the symlink below:

```
1# ln -s /usr/lib/x86_64-linux-gnu/libfbclient.so.3.0.7 /usr/lib/x86_64-linux-  
1gnu/libgds.so
```

Get and install the binaries ODBC drivers for Firebird:

```
1# wget https://sourceforge.net/projects/firebird/files/firebird-ODBC-driver/2.0.5-  
1Release/OdbcFb-LIB-2.0.5.156.amd64.gz/download  
2# gunzip OdbcFb-LIB-2.0.5.156.amd64.gz  
3# tar xvf OdbcFb-LIB-2.0.5.156.amd64  
4# cp libOdbcFb.so /usr/lib/x86_64-linux-gnu/odbc/.
```

Edit the odbcinst.ini file:

```
1 # vi /etc/odbcinst.ini  
2 ...  
3 [Firebird]  
4 Description=InterBase/Firebird ODBC  
5 Driver=/usr/lib/x86_64-linux-gnu/odbc/libOdbcFb.so  
6 Setup=/usr/lib/x86_64-linux-gnu/odbc/libOdbcFb.so  
7 Threading=1  
8 FileUsage=1  
9 CPOutput=  
9 CPReuse=  
10
```

Check the ODBC driver file /etc/odbcinst.ini:

```
1 # odbcinst -q -d  
2 ...  
3 [Firebird]  
4 # odbcinst -q -d -n Firebird  
5 [Firebird]  
6 Description=InterBase/Firebird ODBC  
7 Driver=/usr/lib/x86_64-linux-gnu/odbc/libOdbcFb.so  
8 Setup=/usr/lib/x86_64-linux-gnu/odbc/libOdbcFb.so  
9 Threading=1  
9 FileUsage=1
```

```
10CPTimeout=
```

```
11CPReuse=
```

```
12
```

```
13
```

As debian, create the sampled database using the native administrative tool isql-fb:

```
1$ cd $workdir
```

```
2$ mkdir firebird
```

```
3$ cd firebird
```

```
4$ isql-fb -user SYSDBA -password 'SYSDBA'
```

Use CONNECT or CREATE DATABASE to specify a database

```
1SQL> create database "/home/debian/odbc4gawk/firebird/sampled.db" page_size 8192 user  
SYSDBA password 'SYSDBA';
```

```
2SQL> commit;
```

```
3SQL> quit;
```

As there are no sample scripts specifically for Firebird, use [the postgresql's ones](#) to create the tables but first replace the "SERIAL" keyword in the "CREATE TABLE" statements with the equivalent firebird's syntax "INT GENERATED BY DEFAULT AS IDENTITY". Use your favorite text editor or even sed for this. Save the above file as create_tables_postgresl.sql. With the vi editor, this could be done as shown below:

```
1$ vi create_tables_postgresl.sql
```

```
2:1,$s/ SERIAL / INT GENERATED BY DEFAULT AS IDENTITY /g
```

```
3:w create_tables_firebird.sql
```

```
4:q
```

Launch again isql-fb with a connection to the newly created database and copy/paste the statements from create_tables_firebird.sql as they are just a few of them:

```
1$ isql-fb /home/debian/odbc4gawk/firebird/sampled.db -u sysdba -p sysdba
```

```
2Database: /home/debian/odbc4gawk/firebird/sampled.db, User: SYSDBA
```

```
3SQL> show tables;
```

There are no tables in this database

Paste the create table statements here.

Also, copy/paste the statements from [here](#) and append a final commit; to confirm the INSERT. Finally, check the data:

```
1 SELECT
```

```
c.country_name,
```

```

2 c.country_id,
3 l.country_id,
4 l.street_address,
5 l.city
6 FROM
7 countries c
8 LEFT JOIN locations l ON l.country_id = c.country_id
9 WHERE
10 c.country_id IN ('US', 'UK', 'CN');
11
12 Output:
13
14 COUNTRY_NAME                COUNTRY_ID COUNTRY_ID
15 STREET_ADDRESS              CITY
16 =====
17
18 China
19 CN
20
21 United Kingdom
22 St London UK UK 8204 Arthur
23
24 United Kingdom
25 Oxford Science Park Oxford UK UK Magdalen Centre, The
26
27 United States of America
28 Rd Southlake US US 2014 Jabberwocky
29
30 United States of America
31 Blvd South San Francisco US US 2011 Interiors
32
33 United States of America
34 Rd Seattle US US 2004 Charade
35
36 SQL>

```

Configure an ODBC DSN by editing the user's DSN file:

```

1 $ vi ~/.odbc.ini
2 ...
3 [myfddb]
4 Description=Firebird
5 Driver=Firebird
6 Dbname=/home/debian/odbc4gawk/firebird/sampled.db
7 User=SYSDBA
8 Password=SYSDBA

```

```
9 Role=
10 CharacterSet=UTF8
11 ReadOnly=No
12 NoWait=No
```

Check it:

```
1
2 $ odbcinst -q -s -n
3 ...
4 [myfbdb]
5 ...
6 $ odbcinst -q -s -n myfbdb
7 [myfbdb]
8 Description=Firebird
9 Driver=Firebird
10 Dbdname=/home/debian/odbc4gawk/firebird/sampledbs.fb
11 User=SYSDBA
12 Password=SYSDBA
13 Role=
14 CharacterSet=UTF8
15 ReadOnly=No
16 NoWait=No
17
```

Try a connection to the firebird db via ODBC:

```
1 $ isql -v myfbdb sysdba sysdba
2 +-----+
3 | Connected!                                |
4 |                                           |
5 | sql-statement                            |
6 | help [tablename]                         |
7 | quit                                     |
8 |                                           |
```

9 +-----+

10 SQL> SELECT c.country_name, c.country_id, l.country_id, l.street_address, l.city FROM
countries c LEFT JOIN locations l ON l.country_id = c.country_id WHERE c.country_id IN
11 ('US', 'UK', 'CN');

12 Output:

13 +-----+-----+-----+-----+
-----+
14 | country_name | country_id | country_id |
street_address | city |
15 +-----+-----+-----+-----+
-----+
16 | China | CN |
17 | | |
18 | United Kingdom | UK | UK | 8204 Arthur
St | London |
19 | United Kingdom | UK | UK | Magdalen Centre,
20 The Oxford Science Park | Oxford |
21 | United States of America | US | US | 2014 Jabberwocky
Rd | Southlake |
22 | United States of America | US | US | 2011 Interiors
Blvd | South San Francisco |
23 | United States of America | US | US | 2004 Charade
Rd | Seattle |
+-----+-----+-----+-----+
-----+

SQLRowCount returns 6

6 rows fetched

Test the DSN with pyodbc:

1 \$ python3

2 Python 3.9.2 (default, Feb 28 2021, 17:03:44)

3 [GCC 10.2.1 20210110] on linux

4 Type "help", "copyright", "credits" or "license" for more information.

5 >>> import pyodbc

6 >>> con = pyodbc.connect('DSN=myfbdb;UID=SYSDBA;PWD=SYSDBA;DBNAME=sampled;')

7 Segmentation fault

Although isql has no problem with them, it looks like the official Firebird ODBC drivers don't work well with pyodbc. The same behavior was noticed later with the gawk the extension (see article here ...). For this reason, we tried the commercial drivers from Devart available [here](#). After having installed them by following the clear and simple instructions and having created the DSN myfbdb_devart using those

drivers, pyodbc could work fine.

After installation of those drivers, check the DSN definition:

```
1 $ odbcinst -q -s -n myfbdb_devart
2 [myfbdb_devart]
3 Description=Firebird
4 Driver=Devart ODBC Driver for Firebird
5 Database=/media/sf_customers/dbi/odbc4gawk/sampledб.fb
6 Port=3050
7 User=SYSDBA
8 Password=SYSDBA
9 Role=
10 CharacterSet=UTF8
11 ReadOnly=No
12 NoWait=No
```

And the drivers:

```
1$ odbcinst -q -d -n "Devart ODBC Driver for Firebird"
2[Devart ODBC Driver for Firebird]
3Driver=/usr/local/lib/libdevartodbcfirebird.so
```

Retry the pyodbc module:

```
1 $ python3
2 Python 3.9.2 (default, Feb 28 2021, 17:03:44)
3 [GCC 10.2.1 20210110] on linux
4 Type "help", "copyright", "credits" or "license" for more information.
5 import pyodbc
6 cnxn = pyodbc.connect('DSN=myfbdb_devart;UID=SYSDBA;PWD=SYSDBA;DBNAME=sampledб;')
7 cursor = cnxn.cursor()
8 cursor.execute("""SELECT
9 ...          c.country_name,
10 ...          c.country_id,
11 ...          l.country_id,
12 ...          l.street_address,
```

```

12...         l.city
13... FROM
14...         countries c
15... LEFT JOIN locations l ON l.country_id = c.country_id
16... WHERE
17...         c.country_id IN ('US', 'UK', 'CN')""")
18    row = cursor.fetchone()
19    while row:
20        print (row)
21        row = cursor.fetchone()
22...
23Output:
24('China', 'CN', None, None, None)
25('United Kingdom', 'UK', 'UK', '8204 Arthur St', 'London')
26('United Kingdom', 'UK', 'UK', 'Magdalen Centre, The Oxford Science Park', 'Oxford')
27('United States of America', 'US', 'US', '2014 Jabberwocky Rd', 'Southlake')
28('United States of America', 'US', 'US', '2011 Interiors Blvd', 'South San Francisco')
29('United States of America', 'US', 'US', '2004 Charade Rd', 'Seattle')
30
31

```

The Devart's drivers work fine. However, let's try to recompile the official ODBC drivers for Firebird from their source code available [here](#). To compile them, follow the following steps:

```

1$ apt install unixodbc-dev
2$ apt install firebird-dev

```

Once downloaded and untarred, move the directory OdbcJdbc/Builds/Gcc.lin and compile the drivers:

```

1$ export FBINCDIR=/usr/include/firebird
2$ export FBLIBDIR=/usr/lib/x86_64-linux-gnu
3$ make -e -d -f makefile.linux

```

The compiled libraries are put in OdbcJdbc/Builds/Gcc.lin/Release_x86_64. As root, install the shared library libOdbcFb.so in its usual sub-directory and make it readable for everyone:


```
1# cp OdbcJdbc/Builds/Gcc.lin/Release_x86_64/libOdbcFb.so /usr/lib/x86_64-linux-  
gnu/odbc/.  
1# chmod a+r /usr/lib/x86_64-linux-gnu/odbc/libOdbcFb.so
```

With those recompiled drivers, isql still works and pyodbc still fails but only while executing the SELECT statement. As shown later, the gawk extension has no problem anymore: it works with the drivers from Devart as well as the recompiled drivers; thus, we still have the option of free drivers.

Despite pyodbc, Firebird is now fully accessible from any ODBC application under the debian account.

Instructions for the other data sources can be accessed through the following links:

[SQLite](#)

[HSQLDB](#)

[MariaDB](#)

[PostgreSQL](#)

[Oracle](#)

[Microsoft SQLServer for Linux](#)

[MongoDB](#)

[Excel](#)

Post Views: 5,864

ORACLE

by

Oracle Team