

Lazarus » Forum » Programming » General » Using SQLite database in command line program on RPi4B?



 **Author**

devEric69

Hero Member



Posts: 646

Topic: Using SQLite database in command line program on RPi4B?

Re: Using SQLite database in command line program on RPi4B?

« **Reply #15 on:** November 21, 2019, 07:43:54 pm »


Just for information and brainstorming, I saw that some use " ssh -Y " (from a remote Linux machine having X-ming) or PuTTY with the " X11 option " (from a remote Windows machine having cygwin).

Documentation says " ssh -Y " would only allow you to launch one remote UI application at a time, so as not to abuse graphic resources on a { Raspberry } server.

If this would allow the use of an ultra light remote desktop with a Pi, then it could be possible to do a Lazarus UI application (TForm + TFrames alClient with TSchrollBox alClient and controls inside) to, from time to time, query \ see data stored in the database, transfer files, etc.

2 cents.

« *Last Edit:* November 21, 2019, 07:58:58 pm by devEric69 »

 Logged

use: Linux 64 bits (Ubuntu 20.04 LTS).

Lazarus version: 2.0.4 (svn revision: 62502M) compiled with fpc 3.0.4 - fpDebug \ Dwarf3.

BosseB

Sr. Member



Posts: 372

Re: Using SQLite database in command line program on RPi4B?

« **Reply #16 on:** November 21, 2019, 08:38:15 pm »

Quote from: Zvoni on November 21, 2019, 06:19:33 pm

the "locate sqlite"-command?

What I get is:

Code: [Select]

```
$ locate libsqlite3.so
-bash: locate: command not found
```

Same if I try sqlite3 on command line:

Code: [Select]

```
$ sqlite3
-bash: sqlite3: command not found
```

So I did:

Code: [Select]

```
$ sudo apt install sqlite3
$ which sqlite3
/usr/bin/sqlite3
sudo find / -name libsqlite3.so
find: '/tmp/.vnc-1000/run/gvfs': Permission denied
```

AFAICU from [the Lazarus SQLitewiki](#) I have to specify the location of the libsqlite3.so in my Lazarus/Fpc code when I open the database:

Code: Pascal [Select]

[+]

```
1. | SQLiteLibraryName:= './sqlite3.so';
```

This is when the so file has been placed in the current dir.
But it is hard when there is no such thing on the system...
I must have missed something important here.

 Logged

--

Bo Berglund
Sweden

BosseB

Sr. Member



Posts: 372

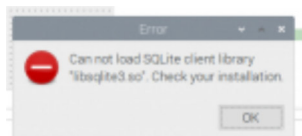
 **Re: Using SQLite database in command line program on RPi4B?**


« **Reply #17 on:** November 21, 2019, 09:34:38 pm »

I found a component on the Lazarus tab SQLdb named TSQLite3Connection, if I use that maybe it will be able itself to dig up where to call sqlite3?
I am trying to find working examples (a hello world program for sqlite) to analyze after first checking that it actually works in my RPi4.

LATER:

I tried to create a new GUI program on the RPi4 and added the TSQLite3Connection component to the form.
Then I set its databasename to /path/to/monitor.sqlite and then checked the Connected checkbox in object inspector. This error message is what turned up as a pop-up error (cannot load libsqlite3.so).
So apparently it still needs this library file, which does not exist on the disk of my RPi4 even after installing sqlite3 itself...



 SQLite_Lazarus_Error.png (12.37 kB, 391x187 - viewed 177 times.)

« *Last Edit:* November 21, 2019, 10:11:52 pm by BosseB »

 Logged

--

Bo Berglund
Sweden

Zvoni

Hero Member



Posts: 1276

 **Re: Using SQLite database in command line program on RPi4B?**


« **Reply #18 on:** November 21, 2019, 10:08:47 pm »

sudo apt-get install mlocate

The "libsqlite3.so" file is located in the "usr/lib" directory.

Then "locate libsqlite3.so"

Sorry

 Logged

One System to rule them all, One Code to find them,
One IDE to bring them all, and to the Framework bind them,
in the Land of Redmond, where the Windows lie

Code is like a joke: If you have to explain it, it's bad

BosseB

Sr. Member



Posts: 372

 **Re: Using SQLite database in command line program on RPi4B?**


« **Reply #19 on:** November 21, 2019, 10:15:22 pm »

Didn't work:

Code: [Select]

```
$ sudo apt install mlocate
-- stuff happens ---
$ locate libsqlite3.so
locate: can not stat () `/var/lib/mlocate/mlocate.db': No such
file or directory
```

What exactly does locate do that find is not doing?

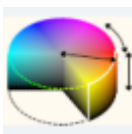
 Logged

--

Bo Berglund
Sweden

winni

Hero Member



Posts: 3126

 **Re: Using SQLite database in command line program on RPi4B?**

« **Reply #20 on:** November 21, 2019, 10:23:12 pm »

Hi!

See

Code: Bash [Select]

[+]

```
1. | man locate
```

Meanwhile start a

Code: Bash [Select]

[+]

```
1. | find / -name libsqlite3.so
```

That will take some time.

Winni

fmc

New Member



Posts: 37


Re: Using SQLite database in command line program on RPi4B?

 « **Reply #21 on:** November 21, 2019, 10:48:59 pm »

After installing mlocate, issue this command at the prompt: **updatedb**

 « *Last Edit: November 21, 2019, 10:50:42 pm by fmc* »

Win X Pro / Lazarus 2.0.6 / FPC 3.0.4

BosseB

Sr. Member



Posts: 372


Re: Using SQLite database in command line program on RPi4B?

 « **Reply #22 on:** November 21, 2019, 11:37:55 pm »

Quote from: fmc on November 21, 2019, 10:48:59 pm

After installing mlocate, issue this command at the prompt: **updatedb**

So I did that and here is the sequence I used:

Code: [Select]

```
$ sudo apt install mlocate
$ locate libsqlite3.so
locate: can not stat () `/var/lib/mlocate/mlocate.db': No such
file or director
$ sudo updatedb
$ locate libsqlite3.so
/usr/lib/arm-linux-gnueabi/libsqlite3.so.0
/usr/lib/arm-linux-gnueabi/libsqlite3.so.0.8.6
```

So now I have 2 files found but with the wrong extension...

Further investigation shows that the one ending in 0.8.6 is the actual file and it is symlinked to the other.

Am I supposed to symlink it to my own project directory as libsqlite3.so?

I tried to symlink like that to my project dir but I get the exact same error message when I try to set the connection object to Connected=true....

--

Bo Berglund
Sweden

BosseB

Sr. Member



Posts: 372


Re: Using SQLite database in command line program on RPi4B?

 « **Reply #23 on:** November 22, 2019, 12:39:09 am »

FINALLY!

After reading dozens of threads and how-tos I finally found one place where I

got into a solution:

[Getting started with SQLite under Linux](#)

The previously missing link for me is the following:

Code: [Select]

```
sudo apt install sqlite3 libsqlite3-dev
```

I did not know of the last part before...

So I checked what it did using the locate command (after updatedeb) and it found a so file at the same place where the others were. Strangely what this file was is a symlink from the exact same file I symlinked over to the project dir earlier.

So it does not help having it next to the executable, but apparently if it is /usr/lib/arm-linux-gnueabi/libsqlite3.so

Go figure.... Or maybe this install sets up something else as well, but what?

In any case now that I check the connected property inside Lazarus IDE there is no longer an error and the empty database file gets created.

Now it is way too late, I have to get some sleep....

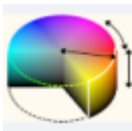
 Logged

--

Bo Berglund
Sweden

winni

Hero Member



Posts: 3126

 **Re: Using SQLite database in command line program on RPi4B?**

« **Reply #24 on:** November 22, 2019, 01:28:15 am »

Hi!

Keep that in mind for the future. It is often (but not always) that you need the development package for some reasons. So look for the packages with *dev* or *devel* - helps you in some strange situations, because the developers don't mention it - and perhaps they don't know that the devel-package is needed!

Winni

 Logged

MarkMLI

Hero Member



Posts: 4801

 **Re: Using SQLite database in command line program on RPi4B?**

« **Reply #25 on:** November 22, 2019, 09:39:22 am »

That's not the first time I've seen a Debian/Raspbian -dev package set up a symlink that common sense would suggest would always be needed.

I believe that was why <https://wiki.freepascal.org/TSQLDBLibraryLoader> was added to the LCL.

MarkMLI

 Logged

BosseB

Sr. Member



Posts: 372



Re: Using SQLite database in command line program on RPi4B?

« **Reply #26 on:** November 22, 2019, 09:55:35 am »

Thanks Mark!

I did not know about the Library Loader, but found it on the palette now. Since I am basically working on a command line program I have made a test app (no functionality) just to see what works or not in the GUI. However, it has a default LibraryName selectable for SQLite (libsqlite3.so) with no_path, so it seems like it does not find it by itself. One has to provide the full path in the configuration, I guess this means I have to make this a configuration item, right? So it needs to go into the conf file and loaded on application initialization, right? And do I have to "connect" it to the SQLiteConnection component somehow? Or is it enough to use the LoadLibrary method after setting the ConnectionType and LibraryName properties?

Then I have to translate this into non-GUI code too.

« *Last Edit:* November 22, 2019, 10:29:10 am by BosseB »

Logged

--

Bo Berglund
Sweden

MarkMLI

Hero Member



Posts: 4801



Re: Using SQLite database in command line program on RPi4B?

« **Reply #27 on:** November 22, 2019, 10:38:01 am »

In fairness, it's reasonable to expect a developer's machine to have the relevant -dev packages on it. Unfortunately, in the case of Debian/Raspbian the lack of the symlink set up by installation of the -dev package impacts usability of end-user systems which normally wouldn't (and possibly shouldn't) have developer packages installed.

My recollection is that the underlying SQL connection components are thin wrappers around classes in the FCL, but I can't remember whether I've used the FCL directly other than for minimal bug report code.

I tend to be oriented towards PostgreSQL here, when I was shopping around for a suitable database back in the early 00s it was the only accessible one that had comprehensive transaction support which was something I needed badly. Having all libraries in the right place was not an issue with Slackware and Windows ODBC/BDE which were what I was using at the time, I suspect that Debian initially set things up properly and when the missing symlink became an issue I simply started creating a symlink and (later) installing the -dev package as routine... something which I'm not entirely happy doing on a non-development machine.

My recollection is that the FCL component gained a search path as an

alternative way of fixing the same problem. Now you /could/ look for the .so library in a sequence of "usual suspect" directories, but that implies that application-level code is making the decision that xxx-1.2.3.so is a suitable implementation of the API defined for xxx.so, which is something that should really be done by the package maintainer at the distro level. So in short, this is an distro-level problem which for some reason Debian et al. are refusing to fix, and FPC/Lazarus allow the path to be set at the application level as a not-entirely-satisfactory way around the problem.

I agree that the library path is probably something you want in your configuration file, with the obvious caveat that it will make the configuration version-specific. What I've got here is a .ini file defining how apps are to connect to the database, after which they get everything else from scripts and scheduler tables stored in the database itself: and that .ini file has not needed maintaining for a long time.

I'd suggest that the best solution would be a hybrid sequence:

- * Look for an explicit path in the .ini file. If there isn't one then:
- * Attempt to make the connection without an explicit path.
- * Look for well-known (tested) library names that the distro /should/ have symlinked.
- * If this fails, e.g. because your "well known" sequence goes up to xxx-1.2.3.so but the library is now at xxx-1.2.4.so then put it as an explicit parameter in the .ini file until the next software release, then remove it.

I've got a bit of example code here but I don't think it would tell you anything you don't already know, the only bit which might be useful is that the comments suggest that the FCL gained the path parameter > 2.6.0 and the LCL gained a wrapper component at some point after 1.0.

MarkMLI

« *Last Edit: November 22, 2019, 11:28:06 am by MarkMLI* »

 Logged

MT+86 & Turbo Pascal v1 on CCP/M-86, multitasking with LAN & graphics in 128Kb.
Pet hate: people who boast about the size and sophistication of their computer.
GitHub repositories: <https://github.com/MarkMLI?tab=repositories>

BosseB

Sr. Member



Posts: 372



Re: Using SQLite database in command line program on RPi4B?

« **Reply #28 on:** November 22, 2019, 02:22:35 pm »

Thanks again! Very helpful.

Today I have been looking for a SQLite database browser and [what I found](#) seemed to be OK, except after installation on my Windows 7 x64 Pro laptop it complains about missing libraries (they look like Microsoft ones). So I uninstalled it and will now try to get it onto the RPi4 via apt. I like to be able to inspect the database directly...

 Logged

--

Bo Berglund

MarkMLI

Hero Member



Posts: 4801



Re: Using SQLite database in command line program on RPi4B?

« **Reply #29 on:** November 22, 2019, 02:33:38 pm »

Quote from: BosseB on November 22, 2019, 02:22:35 pm

I like to be able to inspect the database directly...

I agree. I believe it can be done and that it's fairly common for e.g. Firefox developers to do that during debugging.

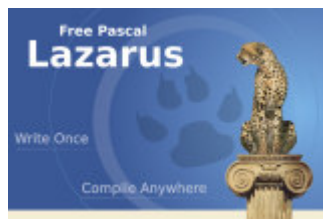
Many thanks to whoever it was that pointed out that multiple applications could potentially access (or at least read) the files representing an SQLite database. With the usual caveat about the extent to which file and record locking was grafted onto unix as an afterthought.

MarkMLI

Logged

MT+86 & Turbo Pascal v1 on CCP/M-86, multitasking with LAN & graphics in 128Kb.
Pet hate: people who boast about the size and sophistication of their computer.
GitHub repositories: <https://github.com/MarkMLI?tab=repositories>

Jump to: => General ▾ go



Free Pascal

- [Website](#)
- [Downloads](#)
- [Wiki](#)
- [Bugtracker](#)
- [Mailing List](#)

Lazarus

- [Website](#)
- [Downloads \(Laz+FPC\)](#)
- [Packages \(OPM\)](#)
- [FAQ](#)
- [Wiki](#)
- [Bugtracker](#)
- [CCR Bugs](#)

Recent

Find a MM/DD/YY date in a... by **Bart**
[Today at 11:49:10 am]

repeat until as a single ... by **MarkMLI**
[Today at 11:46:07 am]

Strange behavior of compi... by **wp**
[Today at 11:35:17 am]

PCHAR Children's Question by **wp**
[Today at 11:14:06 am]

pi PICO controlling ws281... by **jb007**
[Today at 10:42:46 am]

- IRC channel
- GIT
- Mailing List
- Other languages

Foundation

- Website

Useful Wiki Links

- Project Roadmap
- Getting the Source
- Screenshots
- How to use the forum

Donate €

Donate \$

About donations (wiki)

PChar and UTF-8. Weird st... by **MarkMLI**

[**Today** at 10:00:31 am]

OpenBSD error: Illegal ch... by **simonm**

[**Today** at 08:39:25 am]

RawByteString Questions by **bytebites**

[**Today** at 08:31:32 am]

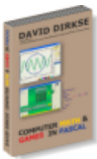
install fail Lazarus 2.2.... by **Handoko**

[**Today** at 05:12:19 am]

Tail recursion vs tail ca... by **raoul**

[**Today** at 02:14:28 am]

Bookstore



Computer Math and Games in Pascal
(preview)



Lazarus Handbook

Search



Search

[Advanced search](#)

