



GDataBase for SQLite

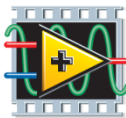
USER GUIDE

GDataBase



> For SQLite

COMPATIBLE WITH



NATIONAL INSTRUMENTS

LabVIEW™

Table of contents

1.	Foreword	2
1.1	Overview	2
1.2	Features	2
1.2.1	SQLite	2
1.2.2	Event logger	2
1.3	Requirement	3
2.	Function palette	4
2.1	SQLite Functions	4
2.1.1	SQLite palette	4
2.1.1.a	Model	4
2.1.1.b	Open Database	5
2.1.1.c	Close database	5
2.1.1.d	Execute SQL query	5
2.1.1.e	Begin transaction	6
2.1.1.f	Commit transaction	6
2.1.1.g	Rollback transaction	6
2.1.2	Advanced palette	7
2.1.2.a	New statement	7
2.1.2.b	Execute statement	7
2.1.2.c	Clear statement	7
2.2	Event Logger functions	7
2.2.1	Event Logger palette	8
2.2.1.a	Model	8
2.2.1.b	Open Event Logger	8
2.2.1.c	Close Event Logger	9
2.2.1.d	Write	9
2.2.1.e	Read last events	9
2.2.1.f	Read events by date	10
2.2.1.g	Viewer	10
3.	Activation	11
4.	Support	12

1. Foreword

1.1 Overview

- *GDataBase for SQLite* toolkit is an add-on to NI LabVIEW that adds capabilities to connect SQLite databases from any LabVIEW program.
- With *GDataBase for SQLite* you can easily implement an event logger in your NI LabVIEW application



SQLite uses a relational database engine that can be accessed using the SQL language. As opposed to similar systems SQLite does not use the standard client/server format, but stores it's information in a file. SQLite is in the public domain and can thus be used in any application, proprietary or open.

1.2 Features

1.2.1 SQLite

GDataBase for SQLite toolkit supports the following features:

- Open connection to database
- Execute SQL queries
- Manage queries with transaction and statements

1.2.2 Event logger

GDataBase for SQLite toolkit provides an Event logger based on SQLite which provides the following features:

- Create an event logger
- Write events with different types
- Display an event logger viewer with advanced filtering capabilities

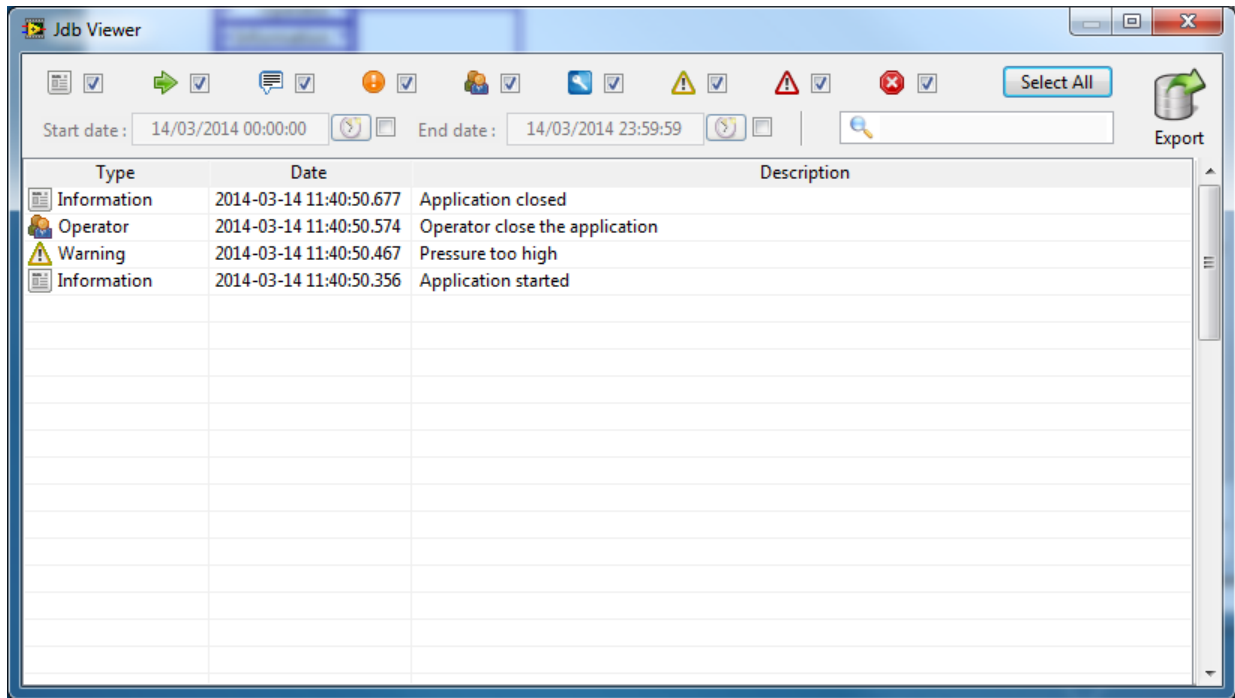


Figure 1: Event logger viewer

1.3 Requirement

- LabVIEW version : 2010 or later
- Operating System : Windows XP and newer
- VI Package Manager from JKI - Community Edition For installation process

2. Function palette

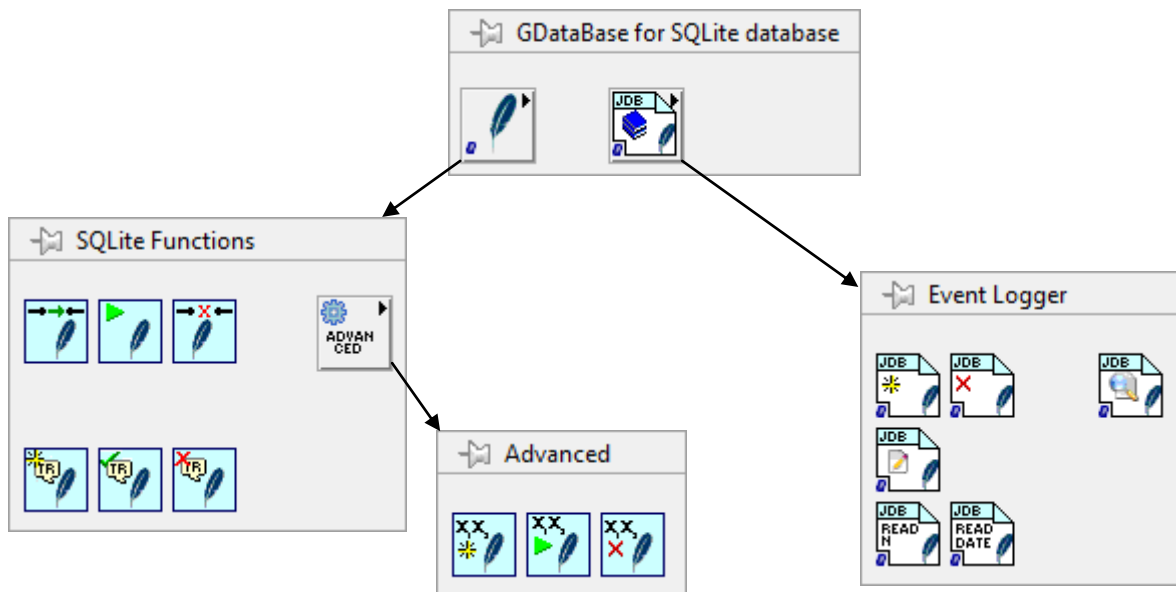


Figure 2: GDatabase for SQLite palettes

2.1 SQLite Functions

2.1.1 SQLite palette

2.1.1.a Model



Figure 3: Model

This palette contains Vis which follows this model.

- **DatabaseRefnum**: reference number to the database connection.
- **error in**: describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use exception control to treat what is normally an error as no error or to treat a warning as an error. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **DatabaseRefnum dup**: reference number to the database connection.
- **error out**: contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out front panel indicator and select Explain Error from the shortcut menu for more information about the error.

2.1.1.b Open Database

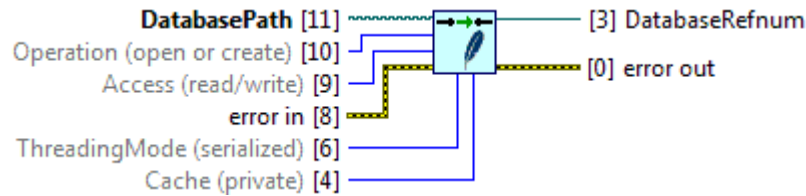


Figure 4: OpenDatabaseConnection.vi

This VI opens an SQLite database file whose name is given by the DatabasePath parameter.

- **DatabasePath**: absolute path to the database's file to open.
- **Operation**: operation to perform. The default is open or creates.
- **Access**: specifies how you plan to access the database. The default is read/write.
- **Threading mode**: specifies the database connection opens in the single-thread, multi-thread or serialized threading mode. The default is serialized. For more information, see the web page "<http://www.sqlite.org/threadsafe.html>"
- **Cache**: specifies if the database connection can (shared) or cannot (private) participate in shared cache mode. The default is private. For more information, see the web page "<http://www.sqlite.org/sharedcache.html>"

2.1.1.c Close database

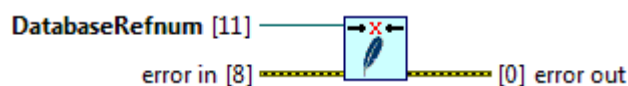


Figure 5: CloseDatabaseConnection.vi

This VI closes a previously opened database.

2.1.1.d Execute SQL query



Figure 6: ExecuteSqlQuery.vi

This VI execute SQL query given bySqlQuery parameter

- **SqlQuery**: SQL query to evaluate.
- **Data[]**: Query result (can be empty).

2.1.1.e Begin transaction

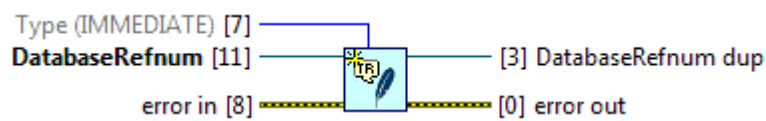


Figure 7: BeginTransaction.vi

This VI create a new transaction. Such transactions usually persist until the next COMMIT or ROLLBACK commands. But a transaction will also ROLLBACK if the database is closed or if an error occurs and the ROLLBACK conflict resolution algorithm is specified. See the documentation on theON CONFLICT clause for additional information about the ROLLBACK conflict resolution algorithm.

- **Type**: specifies the type of the new transaction. The default is DEFERRED. For more information, see Details section.

2.1.1.f Commit transaction



Figure 8: CommitTransaction.vi

This VI commits all modifications that occurred on the database since the last call of the command BEGIN.

2.1.1.g Rollback transaction



Figure 9: RollbackTransaction.vi

2.1.2 Advanced palette

2.1.2.a New statement

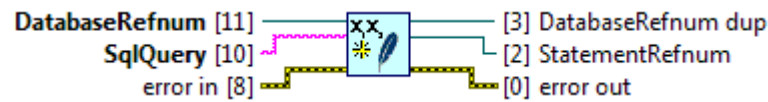


Figure 10: NewStatement.vi

This VI prepares a new statement to be evaluated by Execute statement function.

- **SqlQuery**: SQL query to evaluate where literals may be replaced by a parameter that matches one of following templates:
 - ?
 - ?NNN
 - :VV

In the templates above, NNN represents an integer literal, and VV represents an alphanumeric identifier. The values of these parameters can be set using Execute statement function.

- **StatementRefnum**: reference number to the new statement.

2.1.2.b Execute statement



Figure 11: ExecuteStatement.vi

This VI executes SQL query given by StatementRefnum using Parameters input as value.

- **Parameters**: can be an array of string or variant.
- **Data[]**: retrieved can be an array of string or variant.

2.1.2.c Clear statement



Figure 12: ClearStatement.vi

This VI release statement given by Statement refnum parameter.

- **StatementRefnum**: reference number to the new statement.

2.2 Event Logger functions

2.2.1 Event Logger palette

2.2.1.a Model

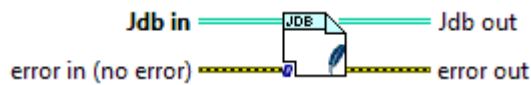


Figure 13: Model

This palette contains Vis which follows this model.

- **Jdb in**: reference to a logger events file.
- **error in**: describes error conditions that occur before this VI runs. The default is no error. If an error occurred before this VI runs, the VI passes the error in value to error out. This VI runs normally only if no error occurred before this VI runs.
- **Jdb out**: reference to a logger events file.
- **error out**: contains error information. If error in indicates that an error occurred before this VI ran, error out contains the same error information. Otherwise, it describes the error status that this VI produces. Right-click the error out front panel indicator and select Explain Error from the shortcut menu for more information about the error.

2.2.1.b Open Event Logger

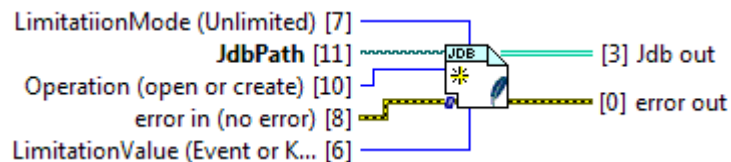


Figure 14: Open.vi

This VI attempts to open a connection with an Event logger. This VI must complete successfully before you can execute any other API VI.

- **JdbPath**: path of the SQLite file to open or create.
- **Operation**: mode of file's open :
 - open or create: if the SQLite file exists, the function open it. if the SQLite file don't exist, the function create and open it successively.
 - open: if the SQLite file exists, the function open it. if the SQLite file don't exist, the function returns an error.
 - The default value is "open or create" mode.
- **LimitationMode**: define the limitation mode of the events logger :
 - Unlimited: No limitations are defined.
 - Event: logger is limited the number of saved events.
 - Bytes: logger is limited the size of the SQLite file.

- The default value is "Unlimited" mode.
- **LimitationValue**: define the limitation value of the events logger :
 - in Unlimited mode: not used.
 - in Event mode: number of saved events to limit.
 - in Bytes mode: bytes' number of the SQLite file to limit.
 - The default value is 0.

2.2.1.c Close Event Logger

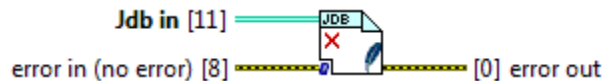


Figure 15: Close.vi

This VI attempts to close a connection with an Event logger.

2.2.1.d Write

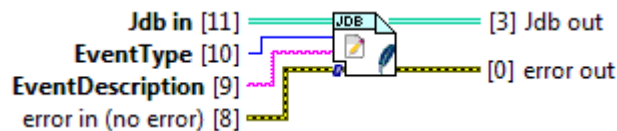


Figure 16: Write.vi

This VI attempts to write an event into an Event logger.

- **EventType**: type of event to write.
- **EventDescription**: events description to write.

2.2.1.e Read last events

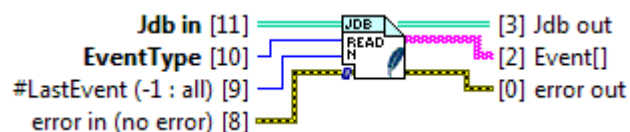


Figure 17: ReadLastEvents.vi

This VI attempts to read last events in an Event logger.

- **EventType**: type of events to read.
- **#LastEvent**: number of last events to return. if "-1" value is set, all events of the logger are returned. The default value is "-1".
- **Event[]**: 2D array of events corresponding of the inputs configuration. First column contains the event's type, second column contains the event's date (YYYY-MM-DD hh:mm:ss.xxx) and the third column contains the event's description.

2.2.1.f Read events by date

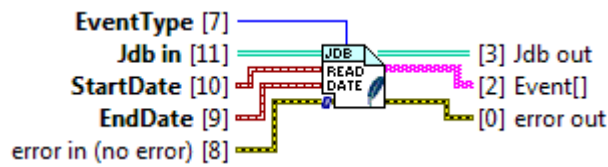


Figure 18: ReadEventsByDate.vi

This VI attempts to read events between two dates in an Event logger.

- **EventType**: type of events to read.
- **StartDate**: minimal date of reading events.
- **EndDate**: maximal date of reading events.
- **Event[]**: 2D array of events corresponding of the inputs configuration. First column contains the event's type, second column contains the event's date (YYYY-MM-DD hh:mm:ss.xxx) and the third column contains the event's description.

2.2.1.g Viewer



Figure 19: Viewer.vi

This VI attempts a dialog to consult events of a Event logger.

3. Activation

After the download and installation of GDataBase for SQLite toolkit, an activation window will pop up at LabVIEW launching. You can also go to help > Activate Add-ons, you'll get the following window. Follow the steps of the add-ons activation.

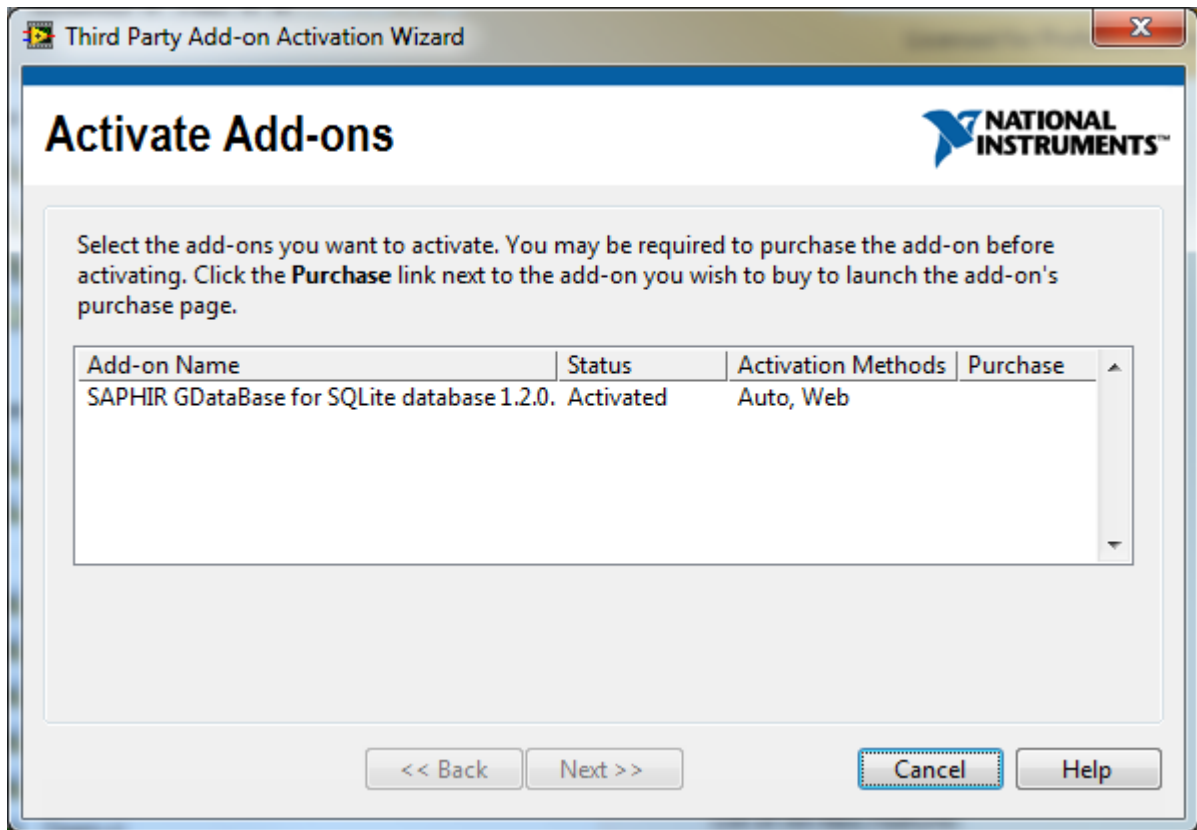


Figure 20: Third Party Add-on



You can try *GDataBase for SQLite* during 30 days for free. After this period toolkit's VIs will become broken. To activate the toolkit after this period, simply go to Help menu and select Activate Add-ons...

4. Support

License purchase provides you 90 days of e-mail support during activation and initial use of the toolkit. Contact us at support@saphir.fr.

Anyway you can access community support using “Online Support & Resources” menu. It opens the following SAPHIR community page:
<http://decibel.ni.com/content/groups/saphir-toolkit>

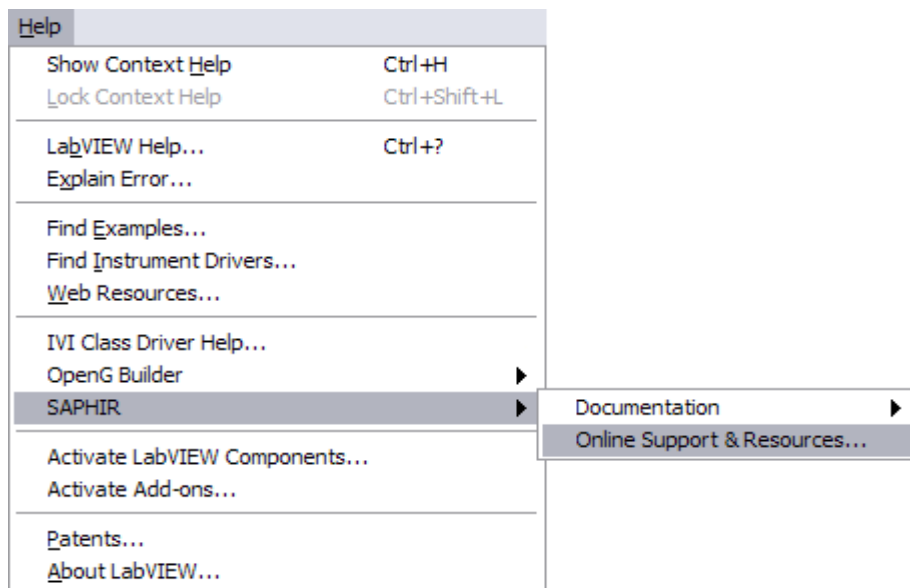


Figure 21: Online support & resources

All the support, discussions and other information related to the product are gathered here.

INDEX

Figure 1: Event logger viewer	3
Figure 2: GDatabase for SQLite palettes.....	4
Figure 3: Model	4
Figure 4: OpenDatabaseConnection.vi	5
Figure 5: CloseDatabaseConnection.vi	5
Figure 6: ExecuteSqlQuery.vi	6
Figure 7: BeginTransaction.vi	6
Figure 8: CommitTransaction.vi	6
Figure 9: RollbackTransaction.vi	6
Figure 10: NewStatement.vi.....	7
Figure 11: ExecuteStatement.vi	7
Figure 12: ClearStatement.vi	7
Figure 13: Model	8
Figure 14: Open.vi	8
Figure 15: Close.vi	9
Figure 16: Write.vi	9
Figure 17: ReadLastEvents.vi.....	9
Figure 18: ReadEventsByDate.vi	10
Figure 19: Viewer.vi.....	10
Figure 20: Third Party Add-on.....	11
Figure 21: Online support & resources.....	12

Other add-ons that could be helpful

GModBus



> Over TCP

GModBus



> Over Serial Line

GDataBase



> For MySQL™

VIBox



> Probes

VIBox



> XControls

 contact@saphir.fr
 +33 (0)4 38 92 15 50

 www.saphir.fr