SQLite Client Examples – Connect To Server – Line 1 in Examples

Before you begin learning how to create your own client application using the SQLite Server, make sure you first started by installing the SQL-Lite Server. After that, you must do the following if you have not done this already:

1. **Stop the SQLite Server, and end the application as the option.**
2. **Copy the SQL-Lite.db file in the client example folder to 'C:\ProgramData\SQLite\SQLite-ServerPlatform\Data' replacing the database.**
3. **Launch the SQLite Server and Login (User: Server | Password: 123456)**
4. **Enable encryption and use the following encryption key for this example: 0238682577625686**

Now you can start on example 1 – Connect to Server:

Email all support questions to: michaelbio516@aol.com

Description:

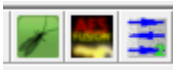
An example that will show you how to connect to the SQLite Server using the proper authentication method.

All details are listed in the source file. For these examples, you must have encryption on with the key as: 0238682577625686

Important Notes:

The custom query is intended to only process SQL Select Statements. Including insert/update/delete statements wasn't designed or tested to perform other SQL syntaxes besides select statements. Other examples will show you how to write or delete data properly.

3 Required Objects:



Lacewing Relay Client – Communicates to server, transmits SQL data.

String Parser (Server Authentication): - Connecting to your server properly.

AES-256 Encryption Object: All examples use the encryption method.

SQLite Client Examples – Query Database Custom – Line 2 in Examples

Email all support questions to: michaelbio516@aol.com

Description:

An example that will show you how to query a database using the custom syntax. The custom syntax allows you to include a complete SQL select statement in the syntax and has no limits. Requires knowledge of the syntax language of SQL.

All details are listed in the source file.

Important Notes:

The custom query is intended to only process SQL Select Statements. Including insert/update/delete statements wasn't designed or tested to perform other SQL syntaxes besides select statements. Other examples will show you how to write or delete data properly.

4 Required Objects:

Lacewing Relay Client – Communicates to server, transmits SQL data.

String Parser (Server Authentication): - Connecting to your server properly.

String Parser (For SQL Parsing): - Use to parse received lacewing data from server. Uses these characters as delimiters:

Client to Server Delimiter: §

Server to Client Delimiter: º

AES-256 Encryption Object: All examples use the encryption method.

Lacewing Sub channel For Select Statements: 10

When performing an SQL Select statement, all communication is done on lacewing subchannel 10

Understanding the Syntax:

If you look at the data in the Custom Query Loop, this is what is being sent to lacewing using this syntax, to the server:

```
Encrypt$( "AESFusion object", "direct"+"§"+Self_Name$( "Lacewing Relay Client" )+"§Select * From TicketSystem where date like
'%2015%'"+"§custom§null§null§null§null§+null§"+Edittext$( "Edit Box" )+"§null")
```

Delimiter 1 - 'direct' is just a command to let the server know that an authenticated request is being made by a client. Without using 'direct', the server would ignore the incoming client data.

Delimiter 2 - The Lacewing Client Name - Your lacewing client name has to be passed to the server, so the server knows who the send the data back to. This allows the server to respond back to the correct user.

Delimiter 3 - The SQL Query

Delimter 4 - The Data Type. In this example, I set 'custom' to let the server know, that I want to use my own custom query. Other synthax's allow you to make an SQL statements without entering standard SQL Select statements.

Delimiter 10 (EditBox In This Example) - The starting Row to begin from in your query. You can receive up to 50 records at a time. This always allows fast transactions to be sent to the client. Also, you wouldn't need want to send 100,000 records back to the client. This allows the user to select what row they want to start from. So if you have 500 records, but you want to view from row 250, it will show you record 250 to 300 when you execute the statement.

SQLite Client Examples – Query Database Basic - Line 3 in Examples

Email all support questions to: michaelbio516@aol.com

Description:

An example that will show you how to query a database using the basic syntax. The basic syntax allows you to perform limited SQL select statements without understanding the language knowledge of SQL.

All details are listed in the source file.

Important Notes:

The custom query is intended to only process SQL Select Statements. Including insert/update/delete statements wasn't designed or tested to perform other SQL syntaxes besides select statements. Other examples will show you how to write or delete data properly.

4 Required Objects:

Lacewing Relay Client – Communicates to server, transmits SQL data.

String Parser (Server Authentication): - Connecting to your server properly.

String Parser (For SQL Parsing): - Use to parse received lacewing data from server. Uses these characters as delimiters:

Client to Server Delimiter: §

Server to Client Delimiter: º

AES-256 Encryption Object: All examples use the encryption method.

Lacewing Sub channel For Select Statements: 10

When performing an SQL Select statement, all communication is done on lacewing subchannel 10

Understanding the Syntax:

If you look at the data in the 'Single Basic Query Loop', this is what is being sent to lacewing using this syntax, to the server:

Single Search:

```
Encrypt$( "AESFusion object", "direct"+"§"+Self_Name$( "Lacewing Relay Client" )+"§exact§single§TicketSystem§date§2016-01-01§null§null§"+Edittext$( "Edit Box" )+"§null")
```

If you look at the data in the 'Multiple Basic Query Loop', this is what is being sent to lacewing using this syntax, to the server:

Multiple Search:

```
Encrypt$( "AESFusion object", "direct"+"§"+Self_Name$( "Lacewing Relay Client" )+"§both wild§multiple§TicketSystem§date§2016-01-01§subject§offline§"+Edittext$( "Edit Box" )+"§and")
```

If you look at the data in the 'Range Basic Query Loop', this is what is being sent to lacewing using this syntax, to the server:

Range Search:

```
Encrypt$( "AESFusion object", "direct"+"§"+Self_Name$( "Lacewing Relay Client" )+"§range§between§TicketSystem§date§2016-01-01§date§2016-01-02§"+Edittext$( "Edit Box" )+"§and")
```

SQLite Client Examples – Query Database Basic – Continued…

Continued…

Delimiter 1 - 'direct' is just a command to let the server know that an authenticated request is being made by a client. Without using 'direct', the server would ignore the incoming client data.

Delimiter 2 - The Lacewing Client Name - Your lacewing client name has to be passed to the server, so the server knows who the send the data back to. This allows the server to respond back to the correct user.

Delimiter 3 – SQL Wild Cards Terms:

exact = Uses exact keyword

no wild = Uses a 'like' condition in sql

start wild= Uses a 'like' condition in sql with start wild card condition

end wild = Uses a 'like' condition in sql with an end wild card condition

both wild = Uses a 'like' condition in sql with a start and end wild card condition

range = Use if delimiter 4 is between / not between.

Delimter 4 – single / multiple / between / not between. Basic uses only 4 required parameters to perform a SQL select statement

single = Searching only a single column

multiple = Searching two columns

between = using a range condition

not between = using a range condition

Delimter 5 – The SQL table you want to connect to

Delimiter 6 – 1st column you want to search in

Delimiter 7 – The keyword/value you will be searching for

Delimiter 8 – 2nd column you want to search in **(Only If multiple/between/not between is set for delimiter 4 or else enter: null)**

Delimiter 9 – 2nd keyword/value you will search **(Only If multiple/between/not between is set for delimiter 4 or else enter: null)**

Delimiter 10 (EditBox In This Example)- The starting Row to begin from in your query. You can receive up to 50 records at a time. This always allows fast transactions to be sent to the client. Also, you wouldn't need want to send 100,000 records back to the client. This allows the user to select what row they want to start from. So if you have 500 records, but you want to view from row 250, it will show you record 250 to 300 when you execute the statement.

Delimiter 11 – And/Or **(Only If multiple/between/not between is set for delimiter 4 or else enter: null)**

and = Will perform an SQL query for and require that both keywords match condition.

or = Will perform an SQL query for and require that one of the keywords match condition.

SQLite Client Examples – Using a Dropdown – Line 4 in Examples

Email all support questions to: michaelbio516@aol.com

Description:

Uses the same protocol as example line 2 and 3, but this allows you to use a combo box dropdown menu to populate records for easy loading.

4 Required Objects:

Lacewing Relay Client – Communicates to server, transmits SQL data.

String Parser (Server Authentication): - Connecting to your server properly.

String Parser (For SQL Parsing): - Use to parse received lacewing data from server. Uses these characters as delimiters:

Client to Server Delimiter: §

Server to Client Delimiter: º

AES-256 Encryption Object: All examples use the encryption method.

SQLite Client Examples – Writing to Database – Line 5 in Examples

Email all support questions to: michaelbio516@aol.com

Description:

Allows a user to write information into a database using a simple syntax. Refer to the example file.

2 Required Objects:

Lacewing Relay Client – Communicates to server, transmits SQL data.

AES-256 Encryption Object: All examples use the encryption method.

Client to Server Delimiter: §

Simple Write Only To Database: USE LACEWING SUB CHANNEL: 11

No SQL Knowledge required

It doesn't matter if you want to insert or update a record. The SQLite Server automatically determines if a records exists or not, and when to use the insert or update clause in SQL. However, you must update a single field per line of code in your client application.

If you are adding a new record, the main column must be added as the first line in your MMF code, in order for it to insert than update any fields you are adding.

Understanding the Syntax:

Let's say we want to add account number '13720' into the database

rowid	Account	Customer
1	13720	test

Once you are connected to the SQLite server by lacewing, you can write to the database:

// Line 1: Account Number First

```
Encrypt$( "AESFusion object", "WriteExample§"+EBLine$( "Record Save", "Account", 0)+"§"+"Account§"+EBLine$( "Record Save", "Account", 0)+"§Account")
```

// Line 2: Customer Name Second

```
Encrypt$( "AESFusion object", "WriteExample§"+EBLine$( "Record Save", "Account", 0)+"§"+"Customer§"+EBLine$( "Record Save", "Customer", 0)+"§Account")
```

Client to Server Delimiter: §

Delimiter 1 – The table you want to write to. (WriteExample is the name of the table)

Delimiter 2 – The field data for the primary column. In this example, it is a unique customer account number. Example: 13720

Delimiter 3 – The column you want to write into.

Delimiter 4 – The field data for the column you want to write into.

Delimiter 5 – The primary column name. (Account is the primary column, and generally the first column is the primary)

See Next page below for the continuation of this example…

SQLite Client Examples – Writing to Database – Line 5 in Examples – cont…

Continued…

Deleting Or Writing To Database: USE LACEWING SUB CHANNEL: 21

SQL Knowledge required

Note: It is recommended you use lacewing sub channel 21 to execute delete statements only. However, you can run any SQL query. If you run select statements, you will not receive them to your client application. Always use sub channel 10 and refer to the examples file in this document for Select statements,

Understanding the Syntax:

How to delete a record:

// Line 1: Simply run a delete statement from SQL

```
Encrypt$( "AESFusion object", "Delete From WriteExample Where Account="+EBLine$( "Record Save", "Account", 0))
```

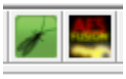
SQLite Client Examples – Using Timestamp command – Line 6 in Examples

Email all support questions to: michaelbio516@aol.com

Description:

Allows a user to write information into a database using a simple syntax with server timestamp commands. Refer to the example file.

2 Required Objects:



Lacewing Relay Client – Communicates to server, transmits SQL data.

AES-256 Encryption Object: All examples use the encryption method.

Client to Server Delimiter: §

Please refer to the Line 5 Example in this documentation to understand writing to a database. This uses the same protocol only that we are including a server command. With timestamp commands, you have the ability to use the server to create a timestamp requested by the client. This is useful if you want to track when a user logged in.

Create Date: Using %CREATEDATE% in a field you want to write into, the server will timestamp into the following format:

2016-01-04

```
Encrypt$( "AESFusion object", "WriteExample2§"+EBLine$( "Record Save", "Account",
0)+"§"+"DateCreated§"+"%CREATEDATE%"+"§Account")
```

Create Time: Using %CREATETIME% in a field you want to write into, the server will timestamp into the following format:

03:04:37 AM

```
Encrypt$( "AESFusion object", "WriteExample2§"+EBLine$( "Record Save", "Account",
0)+"§"+"TimeCreated§"+"%CREATETIME%"+"§Account")
```

Custom Clock: Using %CUSTOMCLOCK% as the first condition, and than using the timestamp extension syntax as the second condition, you can create a custom timestamp in a field you want to write into:

Example: %CUSTOMCLOCK%%c

07/26/17 16:43:46

```
Encrypt$( "AESFusion object", "WriteExample2§"+EBLine$( "Record Save", "Account",
0)+"§"+"TimeCreated§"+"%CUSTOMCLOCK%c"+"§Account")
```

SQLite Client Examples – Binary Requests from Server – Line 7 in Examples

Email all support questions to: michaelbio516@aol.com

Description:

Allows a user to send and receive files to and from server. Uses can include storing and accessing files remotely.

6 Required Objects:

Lacewing Relay Client – Communicates to server, transmits SQL data.

AES-256 Encryption Object: All examples use the encryption method.

Client to Server Delimiter: §

File Object: Required for loading files

File/Folder Object: Required for deleting files.

Binary Quickload: Required for file stacking with lacewing.

String Parser (For SQL Parsing): - Use to parse received lacewing data from server. Uses these characters as delimiters:

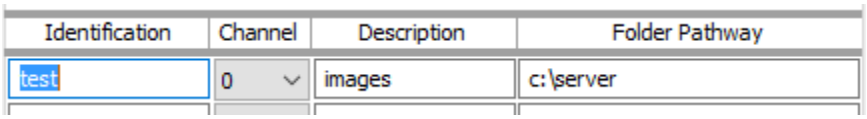
Client to Server Delimiter: §

Lacewing Sub channel For Binary Manager Requests: 31

When sending requests for file transfers, all communication is done on lacewing subchannel 31

Before running the example, please do the following:

1. Log into the SQL-Lite Server, on the application menu, go to 'Server Actions' and select ' Binary Manager'. Duplicate the same settings exactly in the first or any available fields list. To learn more about the Binary Manager, read the SQL-Lite Server Documentation.



2. Create the folder path: c:\server
3. Create the sub folder path: c:\server\Test User
4. Copy the file from the 'Client Example' Folder: Copy MyProfile.jpg into c:\server\Test User
5. Copy the file from the 'Client Example' Folder: Copy database.db into c:\

See Next Page…

SQLite Client Examples – Binary Requests from Server – Line 7 in Examples

Continued…

Understanding the syntax when sending from lacewing on sub channel 31.

Elements =

---

1. ID - Identification of binary manager

2. Sending or Receiving File: %SEND% or %GET%

3. File Name with extension included.

---

4. Pathway Options:

A. **%VAR%**

For Get & Send: Uses the server root pathway of the identifier, but also makes a request to a sub folder using a variable. Variable is included in element 5.

Additionally, (Send Only): Creates new folder on server if folder doesn't exist. Variable is included in element 5.

B. **%FIXED%**

Use the root server pathway of identifier. On Send, ignore element 5

---

5. Variable Only (From Element 4: **%VAR%)**

Can be any name that is a sub folder in the server root identifier.

Additional Commands:

%CLIENT% - Uses own **lacewing name** to access sub folder within the server root identifier.

%DATE% - Uses server date (**Year_Month_Day**) to access sub folder within the server root identifier.

NOTE: If not using element 5 (is using element 4: **%FIXED%**), use **%NULL%** or else element 6 will become 5

---

6. Getting Transfer Type: Works on Get only - %**FULL%** = Sends or Receives File as whole. **%CHUNK%** Sends or Receives File in chunks. For smaller files, full is ok, for larger files always use chunks.

See Next Page…

SQLite Client Examples – Binary Requests from Server – Line 7 in Examples

Continued…

Example Syntax Uses:

1. You want to request a file from a variable path:

Let's say I set the following entry in my binary manager:

Identification	Channel	Description	Folder Pathway
test	0	PDF Files	C:\SQL-Lite\Storage

Let's say in 'C:\SQL-Lite\Storage' I created another sub folder called 'PDFs' and I want to access a file called 'Test.PDF'

This is the syntax to be used with the lacewing object to request this file on sub channel 31:

```
Encrypt$( "AESFusion object", "test§%GET%§Test.PDF§%VAR%§PDFs§%CHUNK%")
```

Delimiter 1: test – I am telling the server to access files under the 'test' identifier. I am basically telling the server to access files in the root path of 'C:\SQL-Lite\Storage' I set in my binary manager.

Delimiter 2: I am using a **%GET%** command to state that I want to request a file.

Delimiter 3: The file I want to get from the server. In this case it is 'Test.PDF' that I want.

Delimiter 4: Since I have many different sub folders under 'C:\SQL-Lite\Storage' I want to use a **%VAR%** command to specify the sub-folder I want.

Delimiter 5: The variable name. In this case, it is 'PDFs'

Delimiter 6: How I am requesting this file. In most cases, you are going to use **%CHUNK%** unless the file is under 1 megabyte, which than **%FULL%**

Example Syntax Uses:

1. You want to request a file from a fixed path:

Let's say I set the following entry in my binary manager:

Identification	Channel	Description	Folder Pathway
test	0	Images	C:\SQL-Lite\Storage\Images

Let's say I want to access a certain image file called 'Test.jpg' under 'C:\SQL-Lite\Storage\Images'

This is the syntax to be used with the lacewing object to request this file on sub channel 31:

```
Encrypt$( "AESFusion object", "test§%GET%§Test.jpg§%FIXED%§NULL§%FULL%")
```

Delimiter 1: test – I am telling the server to access files under the 'test' identifier. I am basically telling the server to access files in the root path of 'C:\SQL-Lite\Storage\Images' I set in my binary manager.

Delimiter 2: I am using a **%GET%** command to state that I want to request a file.

Delimiter 3: The file I want to get from the server. In this case it is 'Test.jpg' that I want.

Delimiter 4: Since I don't need to access any sub folders,I want to use a **%FIXED%** command to use the root path ('C:\SQL-Lite\Storage\Images').

Delimiter 5: Because I am not using a variable, I will need to use **%NULL%** or else the syntax won't be sent correctly to the server.

Delimiter 6: Since I know this file is under 1MB, I am using the **%FULL%** command.

See The Next Page…

SQLite Client Examples – Binary Requests from Server – Line 7 in Examples

Continued…

Example Syntax Uses:

Let's say you want to send a file called 'Song.mpg' on my local machine to the server.

This is the syntax to be used with the lacewing object to request this file on sub channel 31:

```
Encrypt$( "AESFusion object", "test§%SEND%§C:\Song.mp3§%VAR%§MyMusic§")
```

Delimiter 1: test – I am telling the server to access files under the 'test' identifier. I am basically telling the server to access files in the root path of 'C:\SQL-Lite\Storage\' I set in my binary manager.

Delimiter 2: I am using a **%SEND%** command to state that I want to send a file.

Delimiter 3: The file I want to send to the server. In this case it is the full pathway 'C:\Song.mp3' I am uploading.

Delimiter 4: Since I want to send it to a sub folder, I want to use a **%VAR%**

Delimiter 5: Because I am using a variable, I have set 'MyMusic' as the folder I am saving too. If MyMusic was never created, the server will create it for me.

Example Syntax Uses:

Let's say you want to send a file called 'Song.mpg' on my local machine to the server.

This is the syntax to be used with the lacewing object to request this file on sub channel 31:

```
Encrypt$( "AESFusion object", "test§%SEND%§C:\Song.mp3§%FIXED%§%NULL%§")
```

Delimiter 1: test – I am telling the server to access files under the 'test' identifier. I am basically telling the server to access files in the root path of 'C:\SQL-Lite\Storage\' I set in my binary manager.

Delimiter 2: I am using a **%SEND%** command to state that I want to send a file.

Delimiter 3: The file I want to send to the server. In this case it is the full pathway 'C:\Song.mp3' I am uploading.

Delimiter 4: Since I do not want to send it to a sub folder, I want to use a **%FIXED%**

Delimiter 5: No variable, enter **%NULL%**

SQLite Client Examples –Line 8 & Line 9

Review the example source file since the functions are easy to follow.