SQLite Server - About

Description:

The SQLite Server application allows a SQLite database to be hosted over a network connection, allowing multiple clients to concurrently write & read to the SQLite database. The server application handles a queue of requests which prevents file locking, which would normally occur if you were to share a SQLite database by multiple users. The clients connect by a TCP/IP lacewing socket connection. For Support email michaelbio516@aol.com

Operating System Requirements:

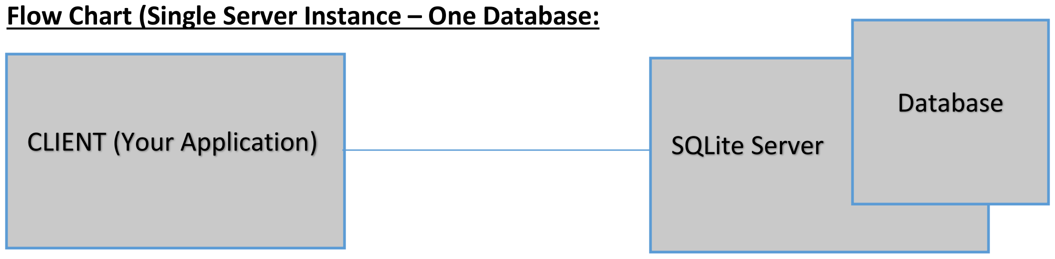
Windows XP, Vista, 7, 8, 10, Server 2008 & 2012. Must have administrator access to run.

At least 256Mb Ram

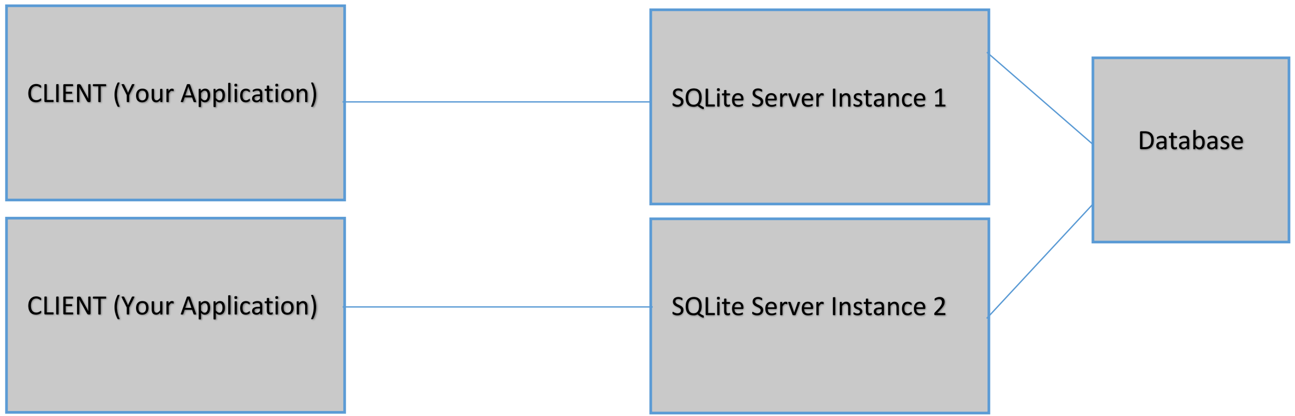
Features:

- Requests made to the server are queued, allowing many clients to write/read from database. Tested on 1000+ clients.
- 256-bit Encryption. The option to allow encrypted message transactions.
- Whitelisting & blacklisting IP addresses from accessing the server.
- Log All Client Data transmission.
- Single database sharing across multiple server processes. Multiple servers can queue across a single database allowing load balancing to be handled across multiple servers.
- Multi database sharing across multiple server processes. Run several databases from multiple server processes on a single machine.
- Memory or File database mode.
- Adobe flash supported mode.
- Clients can download and upload large files concurrently.
- Easy syntax for SQL select statements and Insert/Update transactions.
- Log of all communication and SQL transactions
- Backup database option.

Flow Chart (Single Server Instance – One Database:



Flow Chart (Multiple Server Instance – One Database:



SQLite Server – Installation

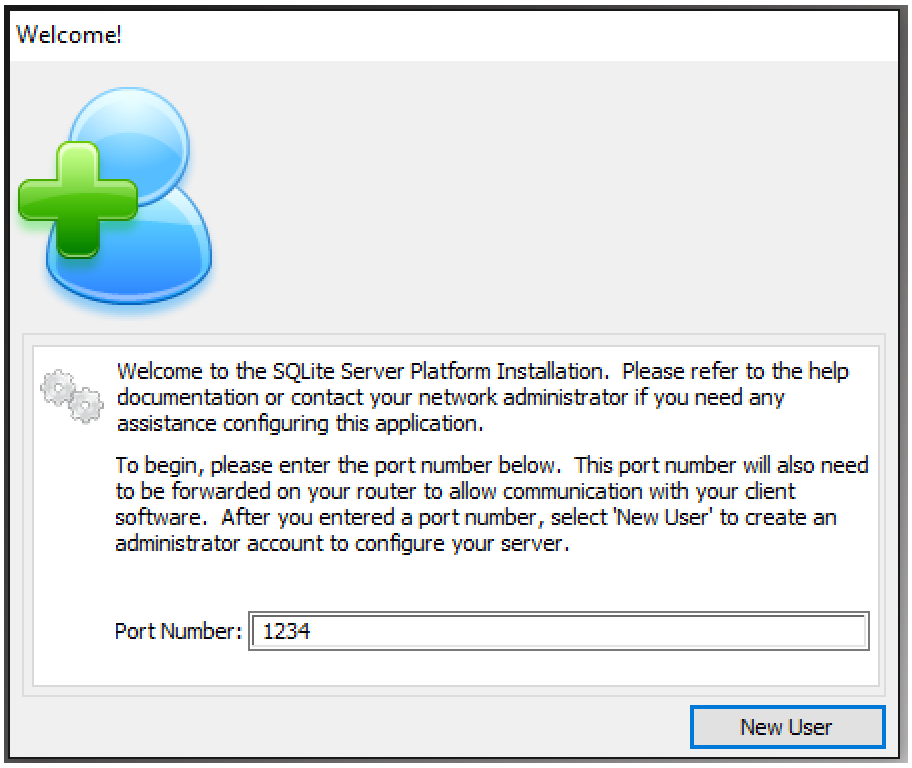
Note:

If you will be sharing this application over the internet you will have to port forward this application both tcp & udp.

Install:

Run the application. Allow access in your firewall.

1. Select the port you want to use for this application for communication. This can be changed after install.



Welcome!

Welcome to the SQLite Server Platform Installation. Please refer to the help documentation or contact your network administrator if you need any assistance configuring this application.

To begin, please enter the port number below. This port number will also need to be forwarded on your router to allow communication with your client software. After you entered a port number, select 'New User' to create an administrator account to configure your server.

Port Number: 1234

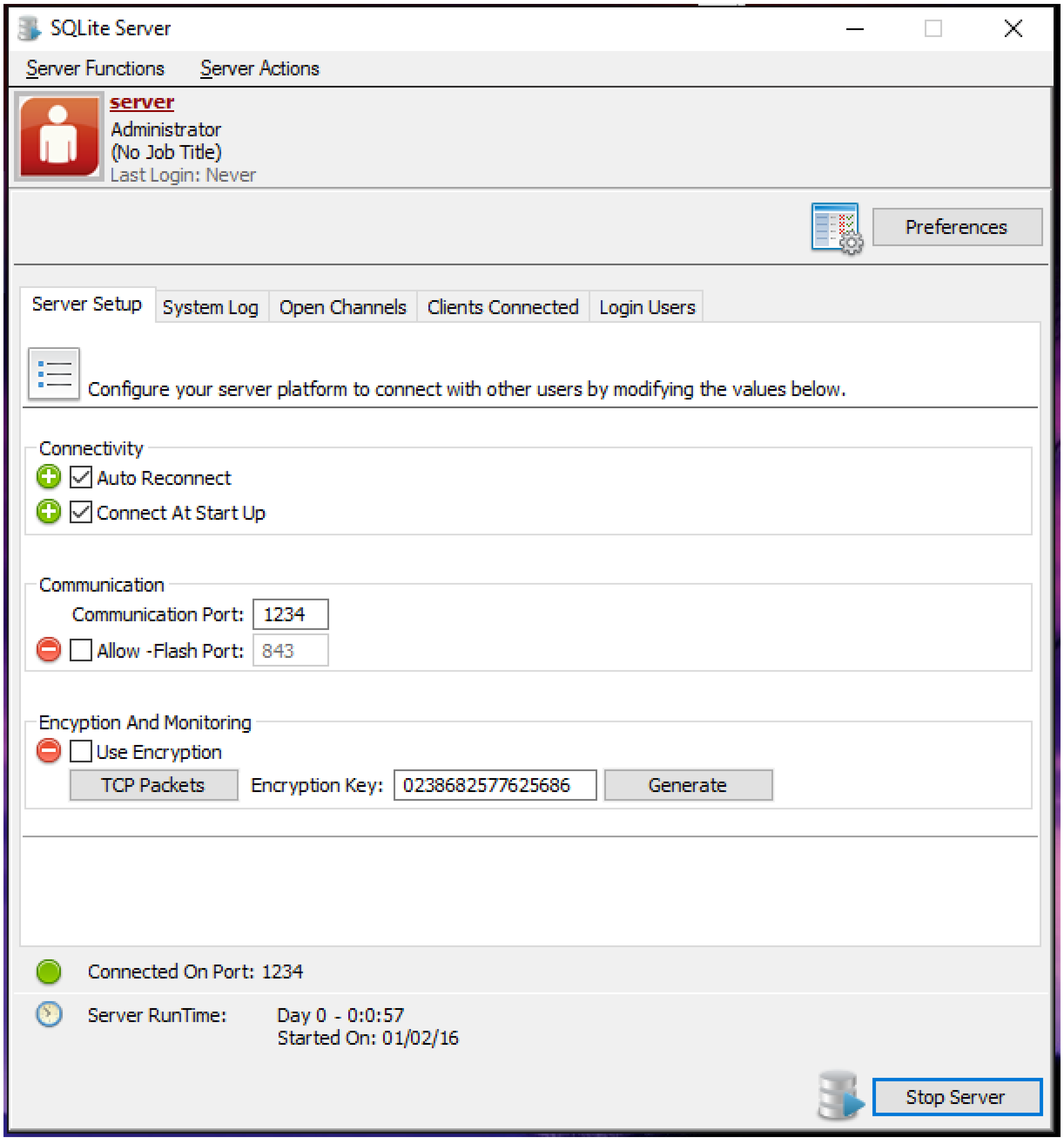
2. Create the administrator account. The default user name is **server**
3. Enter a password, all the fields are optional. Select **save**
4. Let the application reboot.
5. When you login it is recommended you change the database mode from a memory database to a **File Database.** Go to **Preferences,** select the **Database** tab, and enable **File Database.** To reflect the change made, restart the server by going to the application menu, and select **Restart** under Server Functions.

You can now use the application.

License Activation:

The server can be tested as a trial, but will automatically stop hosting and terminate every 4 hours, and allows a max of 5 lacewing user connections concurrently. For full features without these limits, request a license to michaelbio516@aol.com. A license is $25.00 for one computer and can only be installed once. Exceptions can be made for computer crashes upon request.

SQLite Server – Server Setup



server
Administrator
(No Job Title)
Last Login: Never

Configure your server platform to connect with other users by modifying the values below.

Connectivity

☑ Auto Reconnect

☑ Connect At Start Up

Communication

Communication Port: 1234

☐ Allow -Flash Port: 843

Encyption And Monitoring

☐ Use Encryption

Encryption Key: 0238682577625686

Connected On Port: 1234

Server RunTime: Day 0 - 0:0:57
Started On: 01/02/16

- **Auto Reconnect:** Will automatically connect to server if host failure occurs.
- **Connect At Start Up:** Will host server when application starts.
- **Communication Port:** The port being used to allow client communication.
- **Allow – Flash Port:** For hosting Adobe Flash for client flash communication.
- **Use Encryption:** For securing message transactions using 256-bit encryption.
- **TCP Packets:** Can show message transactions on each lacewing sub channel.
- **Sound on User Connection:** Plays a sound in a user connects (On Join Request)
- **Sound on User Disconnection:** Plays a sound if a user disconnects. (On Leave Request)
- **Log Communication Data:** Logs all communication data for clients from sub channel 1

SQLite Server – Preferences

General:

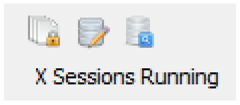
- **Hide Application at Start Up:** When the application launches, it will remain in the system tray. When clicked, the application will appear.
- **Sign out when closed:** When application window is closed, current logged in user on the server will log off.
- **System Login Bypass:** Requires no login to manage server properties.
- **Don't Log User Connection Changes in The System Log:** When lacewing clients connect, do not keep a log history of connections.
- **Make User Login & Search Queries Priority:** The server will handle user logins and SQL select statements requested by clients the first most priority to handle among other server tasks occurring concurrently.

Performance:

- **Processing Power:** Controlled by the frame rate of the application. Higher processing results in faster handling of queued requests. However, the CPU usage will increase.
- **Application Priority:** Windows processing control over running applications.
- **Maximize Processing Power For Any Queued Database Writes:** If a request for writing to the database occurs, the server application will use the highest processing power to write as fast as it can. However, the CPU usage will increase temporarily until the write queue has been completed.
- **Maximize Processing During SQL Queries:** If a request for reading to the database occurs, the server application will use the highest processing power to read as fast as it can. However, the CPU usage will increase temporarily until the read queue has been completed.
- **Maximize Processing For All Other Queued Data:** If a request for non-database tasks occurs, the server application will use the highest processing power to handles these tasks as fast as it can. However, the CPU usage will increase temporarily until the tasks has been completed.
- **Use Session Queuing:** Session Queuing allows multiple server applications to communicate together, while accessing the same database without locking failures. This minimizes the risk of locking. In the event of a lock during heavy transactions requests, the last transaction to the database will always be executed, so no loss will ever occur. If you do not allow session queuing in this scenario, you will run into major locking issues. Rarely, will you receive any locking issues with this function enabled.

  Enable If:
  A. You are using the backup database option.
  B. You are running multiple server applications, which are accessing the same database.
  C. You are running a single server application, but have many users connecting to the server and are writing to the database often.

  Disable If:
  D. You are not using the backup database option.
  E. You are not running multiple server applications.
  F. Or, you are running multiple server applications, but not on the same database.
  G. You are running a single server application, but have little users connecting to the server and not writing very often.



Icon 1 indicates a session file is in use. Icon 2 indicates a write session is started. Icon 3 indicates a read session is started.

Performance (Continued):

- **Write Delay:** If session queuing is enabled, this is the release for when other server applications can have access to begin writing again. For instance, when a server application receives a request from a client for an insert/update/delete record in the database, the server receiving the request will become the **master**. All other servers will wait until that server has finished writing data. The delay is set allowing the master server at the time, to wait for additional requests longer, before allowing the other server applications to process requests.
- **Read Delay:** If session queuing is enabled, this is the release for when other server applications can have access to begin reading again. For instance, when a server application receives a request from a client for a select statement in the database, the server receiving the request will become the **master**. All other servers will wait until that server has finished reading data. The delay is set allowing the master server at the time, to wait for additional requests longer, before allowing the other server applications to process requests.
- **Use Write Limiting:** If session queuing is enabled, this prevents a server from taking too much time to process all the writing requests. Let's say a server receives 500 requests to write to the database, the other server applications would have to wait for the master server to finish all 500 requests before the other servers could start their requests. This sets a limit so multiple servers can evenly write their requests without waiting too long.
- **Use Read Limiting:** If session queuing is enabled, this prevents a server from taking too much time to process all the read requests. Let's say a server receives 500 requests to read to the database, the other server applications would have to wait for the master server to finish all 500 requests before the other servers could start their requests. This sets a limit so multiple servers can evenly read their requests without waiting too long.

Maintenance Mode:

- If maintenance mode is enabled, this allows each sub channel being used in the server application to be restricted or allowed for control purposes. It is not recommended to leave this mode enabled unless you are conducting specific tests or need to temporarily disable a sub channel for a specific purpose.

SQL Queries:

- **Log All SQL Queries in the System Log:** Records all SQL select statements to the system log files.
- **Log Output To In The System Log:** Records all data initiated from a SQL select query, and sent back from the client. This is not recommended to be left on other than testing purposes, since requests will take longer to process.
- **Make SQL Reading a Priority over Writing (Session Queuing Must Be Enabled):** SQL statements received from lacewing sub channel 10 will be processed first. If Insert/Update statements are in the queue, those will be processed once the SQL statement is completed.
- **Response Delay:** The delay set between to time a select query has been processed and sent back to the client. If the delay is set below 0.50 seconds, your client application may not properly receive data during a request.

Security:

- **Blacklisting Mode:** Will reject a specific connection from a lacewing socket. Local or Public IP addresses can be entered. Example: 192.168.1.200
- **Whitelisting Mode:** Will allow connections only from a specific IP address from a lacewing socket, all other connections that are not authorized will be rejected. Local or Public IP addresses can be entered. Example: 192.168.1.200
- **No Security Mode:** Will allow any connection from a lacewing socket.

Shutdown:

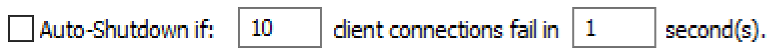
- **Auto Shutdown:** Will perform a shutdown request at a specified time. Example Format: 12:00 AM
- **On Specific Days:** Only perform a shutdown request on specific days.

- **Auto Shutdown on Client Failures:** This option should be enabled if you have hundreds of clients connected to monitor dropout failures due to network instability. Its purpose is to determine if a major dropout in client connections occurred, and to initiate a shutdown request.

SQLite Server – Preferences

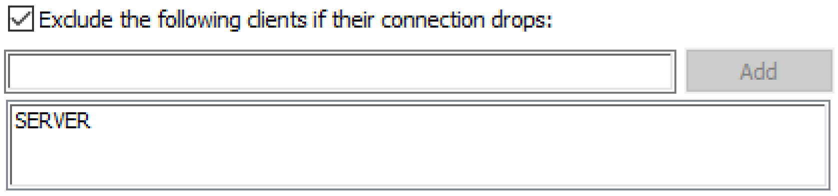
Shutdown (Continued):

- **Include Server Errors As A Client Connection Failure:**

  ☐ Auto-Shutdown if: 10 client connections fail in 1 second(s).

  If the above is enabled, include a server socket error as a connection failure
- **Exclude the following clients if their connection drops:**

  ☑ Exclude the following clients if their connection drops:

  Add

  SERVER

  Enter client names in the list to be ignored as a connection dropout. Enter a name in the field, and select add. To delete a name added, highlight the name in the list and select delete on your keyboard to remove.

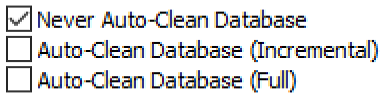
Database:

- **Database Folder:** The folder pathway to the database file.
- **Database File Name:** The file name of the database.

Note: *If you want to use an existing SQLite database you created, please do the following:*

A. Exit the server application.
B. Make sure you have SQLite database ready in the proper folder you want it to reside in.
C. Create the following table in your database (Copy this and execute statement in your database manager):
CREATE TABLE SystemUsers (Name VARCHAR(100),Password VARCHAR(100),Roles VARCHAR(100),Rights VARCHAR(100),Title VARCHAR(100),First VARCHAR(100),Last VARCHAR(100),Street VARCHAR(250),City VARCHAR(250),Zip VARCHAR(50),State VARCHAR(50),phone VARCHAR(100),phone2 VARCHAR(100),cellphone VARCHAR(100),cellprovider VARCHAR(100),Email VARCHAR(100),URLImage VARCHAR(2500),lastlogin VARCHAR(100),Locked VARCHAR(10))
D. Under the Name field enter: Server
E. In the default SQL-Lite.db file created in the server application when you first installed the application, copy the hashed password from the password field, into your database under the SystemUsers table created for the row that has the name: Server.
F. Launch the server application, and change the database folder and database file name to your existing database.
G. Exit the server application.
H. Launch the server application.

- **Read Only**: Enable if you want to use this database for read-only access. You will not be able to write any data in this mode.
- **Memory Database**: Will run the database in system memory. Not recommended to be used in most circumstances.
- **File Database**: Directly run the database from a file. Recommended.
- **Database Clean & Backup:**

  ☑ Never Auto-Clean Database
  ☐ Auto-Clean Database (Incremental)
  ☐ Auto-Clean Database (Full)

SQLite Server – Lacewing Sub Channels

The following sub channels with lacewing are reserved. Please use other sub channels, outside this range. (35 to 254)

SubChannel 0 = Internal Use

SubChannel 1 = Log Messages on all lacewing communication

SubChannel 2 = Internal Use

SubChannel 3 = Sends system user information to server to add as a record

SubChannel 4 = New System User Wants To Join. Get sign/on credentials for user, receive roles, rights.

SubChannel 5 = System User Logged In Communication – Creates Channel

SubChannel 6 = Multiple Users On Network

SubChannel 7 = Internal Use

SubChannel 8 = Internal Use

SubChannel 9 = Maintenance & Master Communication. Let's Client's know if the server is not running or maintenance is enabled

SubChannel 10 = SQL Search Queries To Server and Relayed Back

SubChannel 11 = Allows Any Table to receive insert/update record

SubChannel 12 = Internal Use

SubChannel 13 = Internal Use

SubChannel 14 = Internal Use

SubChannel 15 = Internal Use

SubChannel 16 = Internal Use

SubChannel 17 = Internal Use

SubChannel 18 = Internal Use

SubChannel 19 = Internal Use

SubChannel 20 = Internal Use

SubChannel 21 = Delete Any Records

SubChannel 22 = Internal Use

SubChannel 23 = Internal Use

SubChannel 24 = Internal Use

SubChannel 25 = Internal Use

SubChannel 26 = Internal Use

SubChannel 27 = Internal Use

SubChannel 28 = Internal Use

SubChannel 29 = Send Log RTF Data

SubChannel 30 = Command Requests. Peer or channel provides a command, receives a response from server.

SubChannel 31 = Binary Manager Delegation

SubChannel 32 = Internal Use

SubChannel 33 = Internal Use

SubChannel 34 = Internal Use

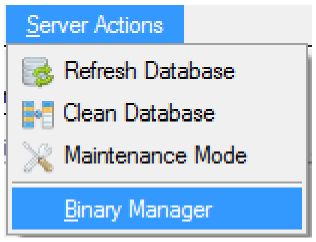
SubChannel 35 = Internal Use

SubChannel 36 = Internal Use

SubChannel 255 = Internal Use

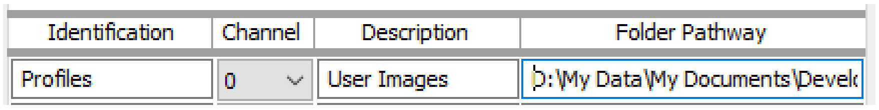
SQLite Server – Binary Manager

The SQLite server allows you the ability to do file transfers by simple uploading (small files) or stacking (large files).



<u>How It Works:</u>

When your client application requests to upload or download a file, a call command is requested to the server. The command contains a syntax to allow you to retrieve files. The Binary Manager allows you to use directories to store file data controlled within the manager. When a request is made from your client application, it is communicating with the binary manager to determine where to save or retrieve files.



Identification	Channel	Description	Folder Pathway
Profiles	0	User Images	D:\My Data\My Documents\Develo

Identification: Unique Identifier that is included in your client application to access a directory and save or retrieve data.

Channel: Sub Channel used to transfer or receive file.

Description: Description of files

Folder Pathway: Pathway to files (Example: C:\MyData\ProfileImages)

Stack Size: File size chunk set to client (bytes) (16384 is the default and recommended value)

Please refer to the example file included on how to develop a client application.

SQLite Server – Running Multiple Servers

To run another server, follow the steps:

1. Create another directory where your SQL-Lite Server will run. You can copy the files from the original directory of your main server.

2. Edit the following files in that new directory.
   A. LoadServerSet.txt: Create a new path for the server config file (Example: C:\server2.dlg)
   B. LoadPrefSet.txt
   C. LoadBinarySet.txt

3. Launch the file and begin your configuration.

SQLite Server – Allowing the Flash Port.

Follow the steps:

1. Enable - Allow flash port. Stop the server.

2. Edit the following files in that new directory.
   D. FlashPlayerPolicy.xml: Change the port of what is running on the server. Not the flash port, the host port.

3. Copy the crossdomain.xml to your web server.

4. Run the SQL Lite Server.