

Table of Contents

Creating a DB link by using BioStar 2 SQLite	1
--	---

Creating a DB link by using BioStar 2 SQLite

APPLIES TO: BioStar 2.2.2 or older

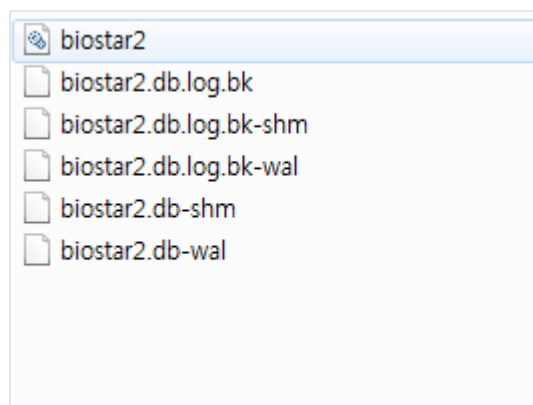
If you are going to install BioStar 2 to the customer site that has a large number of users or its own ERP and payroll system, one of the questions you will most likely encounter is how they can make a connection with their existing systems to BioStar 2. The quick simple answer for this question is to make a database link between the systems, because it's one of the easiest and common ways.

BioStar 2 uses SQLite which is known as simple and light database type that can be embedded in an application, similar to library files. So, it does not need to install independent DBMS(Database Management Server) and manage a DB instance.

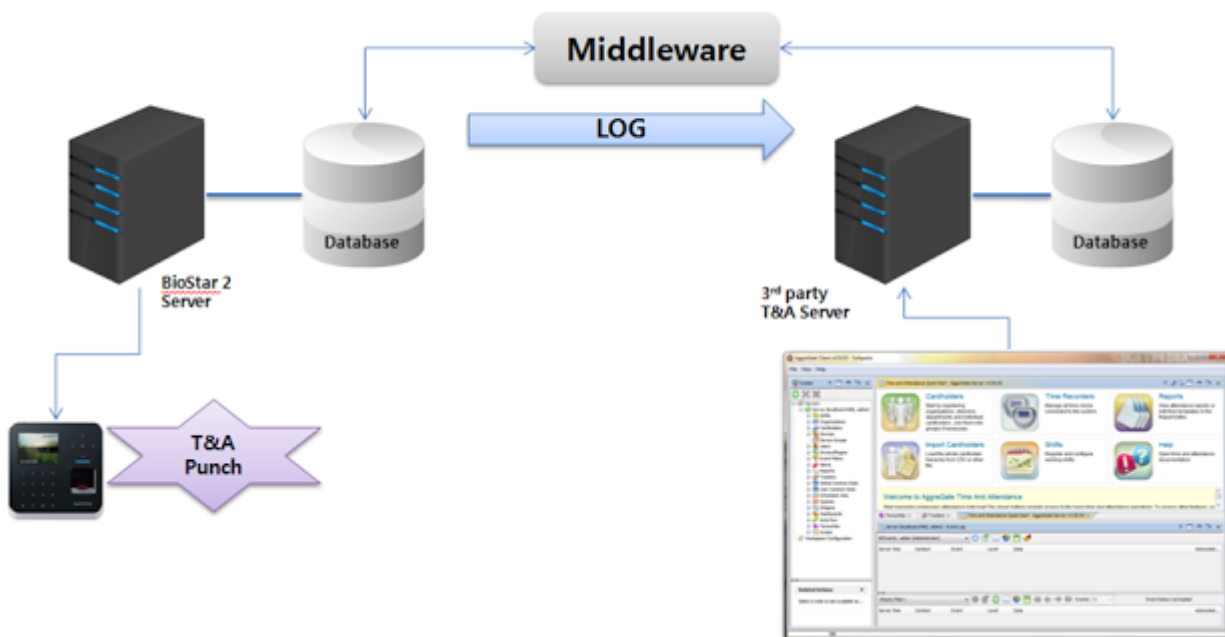
If you have an experience with BioStar 1.x, you can remember that the DBMS installation itself is heavily dependent on the system environment. So, the initial installation itself could be a burden to the installers who are not very familiar with database management.

On the contrary, SQLite is quite simple embedded type of database. It does not need to control a connection between server and database, and it also does not need to download SQL patch files to troubleshoot when DBMS installation is not finished properly.

Then, how can it be connected and linked with another DBMS as we do with BioStar 1.x?



Fortunately, SQLite also has similar architectures and inherited relational database structures. Therefore, it's possible to make a DB link middleware program, and connect it to another database.



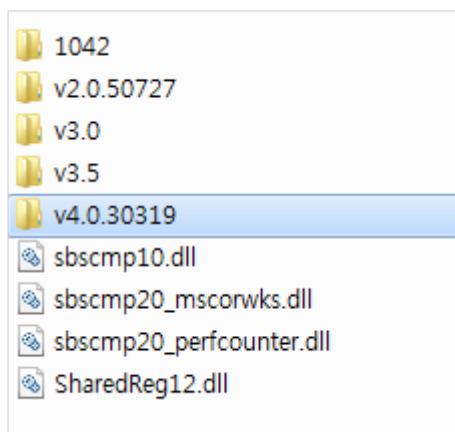
In order to make a DB link application with SQLite, it's necessary to take the below preparation steps. I've used C# to carry out a relatively simple test. So, please note that all the following reference images are based on the C# programming.

1. Check your programming environment and install SQLite library package according to it from the below SQLite download page.

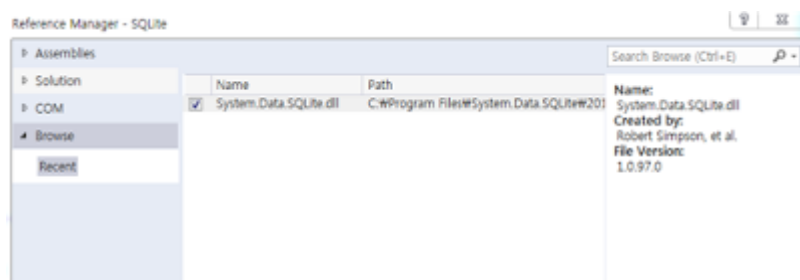
<http://system.data.sqlite.org/index.html/doc/trunk/www/downloads.wiki>

Also, please check your .Net framework version and make sure to install the right version. You can check your .Net framework version by referring to your local PC folder where Microsoft.Net framework files are located.

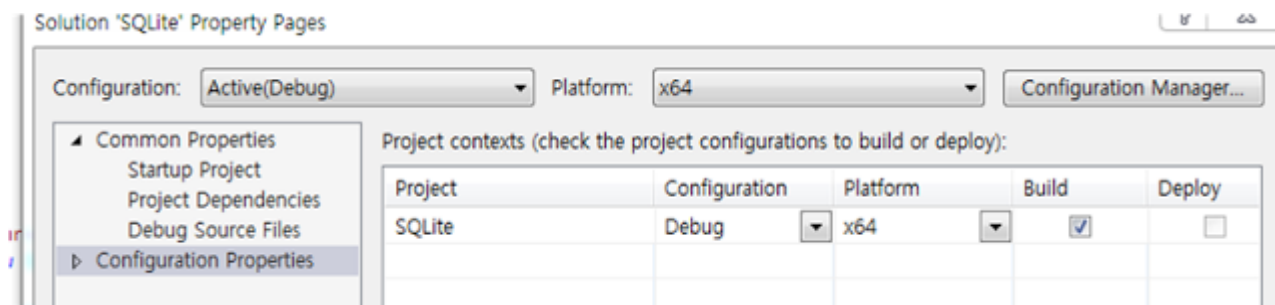
(e.g.) C:\Windows\Microsoft.NET\Framework64



2. Locate the package file to the right location, and add it to the project 'References'



3. Make sure to use the right Platform to run compiler.



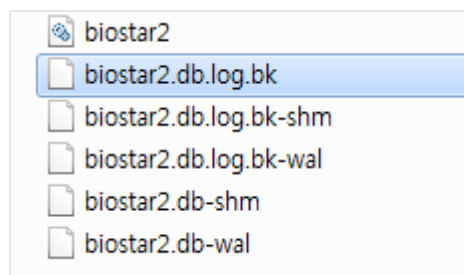
Now, we are ready to make a programming to implement the middleware application. In order to make a DB connection to the SQLite, it's necessary to refer to the DB table information. From BioStar 2.1 version, BioStar.db.log.bk file is used for the log event recording. Please refer to the latest table description.

Database Table Description

Software Version 2.1

What's different from BioStar 2.0.1

- The database now consists of two separate database files, **biostar2.db** and **biostar2.db.log.bk**. Event logs are stored in **biostar2.db.log.bk** and the rest are still in **biostar2.db**. For this reason, log tables such as *T_LGTBIDX*, *T_LGyyyymm*, *T_ALMEVTyyyymm* are now located under **biostar2.db.log.bk** instead.



And, with the db table, the log table is changed to be split to store log files monthly basis from BioStar 2.1v.

T_LGyyyyymm : Log Information

Column	Data Type	Key	Description
EVTLGUID	D_UNQID - INTEGER	PK	Unique event log ID
SRVDT	D_DT - NONE		Date/time saved in the server (in UNIX time)
DEVDT	D_DEVDT - INTEGER		Date/time when the event happens (Device – in UNIX time)
ISVLDATH	D_VALYORN - TEXT		TBD
DEVUID	D_UNQID - INTEGER		Device ID
IMGLGUID	D_UNQID - INTEGER		TBD
USRUID	D_UNQID - INTEGER		'Unique user ID' : when authenticated by the corresponding user 'Card number' : when card authentication failed with the corresponding card '0' = when access denied or authentication failed by unknown user or card
EVT	INTEGER		Type of event
SUBEVT	INTEGER		Type of event (additional)
CRDSL	TEXT		Card slot
TNAKEY	TEXT		TBD

(Log table)

```

CREATE TABLE T_LGIMGDAT201504 (
  IMGLGUID INTEGER NOT NULL,
  IMGDT NONE,
  IMGTP TEXT
  CHECK (IMGTP IN ('JPG', 'PNG', 'BMP', 'WAV', 'MP3', 'WMA', 'PDF', 'TXT')),
  CONSTRAINT Unique_backup_T_LGIMGDAT201504_Identifier PRIMARY KEY (IMGLGUID)
)

CREATE TABLE T_LGIMGDAT201505 (
  IMGLGUID INTEGER NOT NULL,
  IMGDT NONE,
  IMGTP TEXT
  CHECK (IMGTP IN ('JPG', 'PNG', 'BMP', 'WAV', 'MP3', 'WMA', 'PDF', 'TXT')),
  CONSTRAINT Unique_backup_T_LGIMGDAT201505_Identifier PRIMARY KEY (IMGLGUID)
)

CREATE TABLE T_LGIMGDAT201506 (
  IMGLGUID INTEGER NOT NULL,
  IMGDT NONE,
  IMGTP TEXT
  CHECK (IMGTP IN ('JPG', 'PNG', 'BMP', 'WAV', 'MP3', 'WMA', 'PDF', 'TXT')),
  CONSTRAINT Unique_backup_T_LGIMGDAT201506_Identifier PRIMARY KEY (IMGLGUID)
)

CREATE TABLE T_LGIMGDAT201507 (
  IMGLGUID INTEGER NOT NULL,
  IMGDT NONE,
  IMGTP TEXT
  CHECK (IMGTP IN ('JPG', 'PNG', 'BMP', 'WAV', 'MP3', 'WMA', 'PDF', 'TXT')),
  CONSTRAINT Unique_backup_T_LGIMGDAT201507_Identifier PRIMARY KEY (IMGLGUID)
)

```

(View from SQLite)

I've made a DB connection to the latest DB table and used SQLiteCommand to use execute reader command.

```
string strConn = @"Data Source=C:\Program Files (x86)\BioStar 2\db\biostar2.db.log.bk";
SQLiteConnection conn = new SQLiteConnection(strConn);
conn.Open();
string strSQL = "SELECT * FROM T_LG201507 where EVTLGUID";
SQLiteCommand cmd = new SQLiteCommand(strSQL, conn);
SQLiteDataReader rd = cmd.ExecuteReader();
```

On the other side, I've made a DB connection to the SQL server, and made commands to insert the BioStar 2 DB log data to the SQL DB table.

```
SqlConnection sqlcon = new SqlConnection();

sqlcon.ConnectionString = "Server=localhost\\sqlexpress; Database=TEST; Trusted_Connection=Yes";
sqlcon.Open();

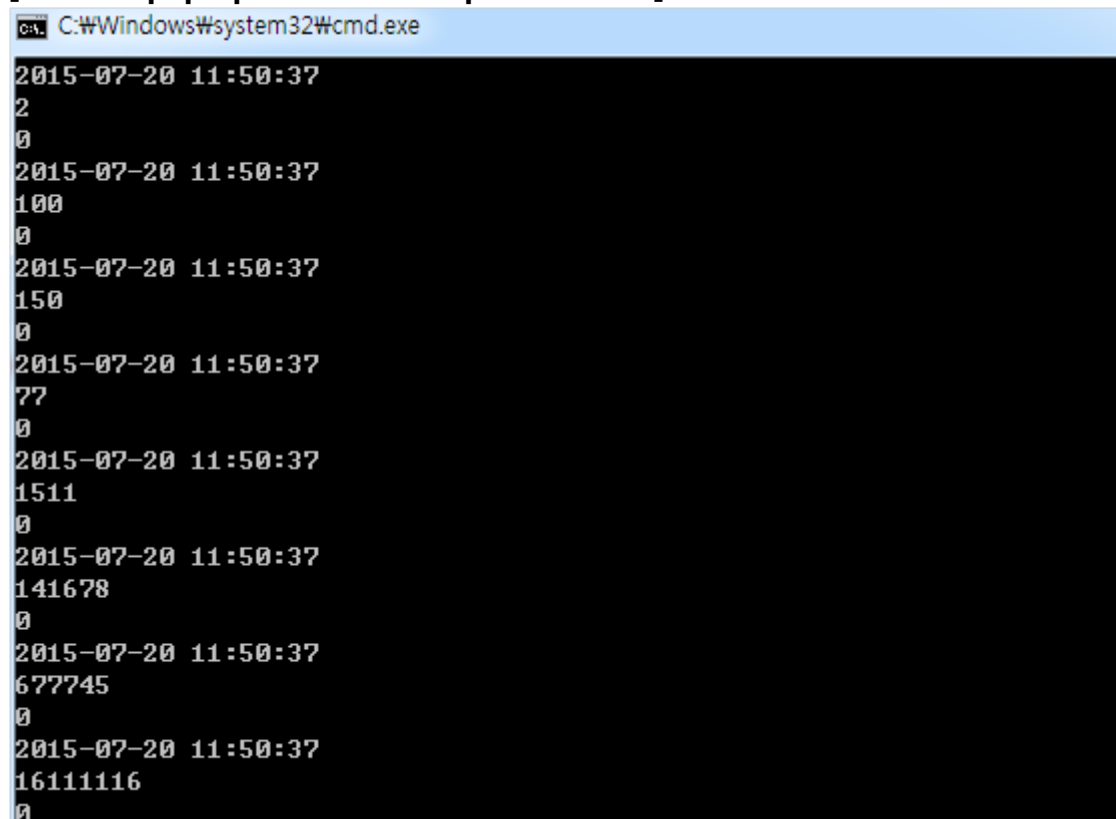
SqlCommand scmd = new SqlCommand();
scmd.Connection = sqlcon;

//scmd.CommandText = "Select datetime From dbo.DBLINK where datetime >= 'rd[1]'";
if (rd[7] == "")
    scmd.CommandText = "INSERT INTO dbo.DBLINK (datetime, userID, TNAevent) VALUES('" + rd[1] + "', NULL, " + rd[10] + ")";

else
    scmd.CommandText = "INSERT INTO dbo.DBLINK (datetime, userID, TNAevent) VALUES('" + rd[1] + "', " + rd[7] + ", " + rd[10] + ")";
scmd.ExecuteNonQuery();
```

And lastly, I've gave it Thread. Sleep with While loop. So, it can make a retry at a certain time interval.

[Console pop-up to show the update status]



```
C:\Windows\system32\cmd.exe
2015-07-20 11:50:37
2
0
2015-07-20 11:50:37
100
0
2015-07-20 11:50:37
150
0
2015-07-20 11:50:37
77
0
2015-07-20 11:50:37
1511
0
2015-07-20 11:50:37
141678
0
2015-07-20 11:50:37
677745
0
2015-07-20 11:50:37
16111116
0
```

[The connected SQL Table for the data base input]

	datetime	userID	TNAevent
472	2015-07-20 11:50:37	NULL	0
473	2015-07-20 11:50:37	NULL	0
474	2015-07-20 11:50:37	1	0
475	2015-07-20 11:50:37	2	0
476	2015-07-20 11:50:37	100	0
477	2015-07-20 11:50:37	150	0
478	2015-07-20 11:50:37	77	0
479	2015-07-20 11:50:37	1511	0
480	2015-07-20 11:50:37	141678	0
481	2015-07-20 11:50:37	677745	0
482	2015-07-20 11:50:37	16111116	0
483	2015-07-20 11:50:37	5446542	0
484	2015-07-20 11:50:37	200510	0
485	2015-07-20 11:50:37	NULL	0

Additionally, I need to make supplement logics to make the application more precisely inserting the data to the db table without overlapped event logs. I believe you can create this and the rest part and tune it for more specific use according to the project environment.

From:

<http://kb.supremainc.com/knowledge/> -

Permanent link:

http://kb.supremainc.com/knowledge/doku.php?id=en:creating_a_db_link_by_using_biostar_2_sqlite

Last update: **2016/10/06 14:09**