**Setting up the Webb-site Database in MySQL with Windows Server**

**Background**

In our setup, we have a Master-Slave (or “source-replica” in new parlance) pair of Windows servers – the master (in HK), an old Windows 10 PC, collects the data (both manual and automated) and the slave, on the Webb-site server (in USA), currently running Windows Server 2016, mirrors the data.

You could just run a single server, but then you would need to take care of backing that up, whereas we can just restore the slave from the master backups if anything goes wrong, and if the master crashes, the slave still operates with static data.

I use [MySQL Workbench](https://dev.mysql.com/downloads/workbench/) to administer both servers.

I am currently running MySQL 8.0.37, just because it works and it’s a pain to upgrade, but you could try the latest “LTS” (Long-Term Support) version 8.4.5 which should be compatible. Don’t try the “Innovation” series – that moves fast and breaks things, rather like watching a series of SpaceX Starship launches. A trial user tried the latest “9.3 Innovation” and found bugs disrupting our code. So be safe. [MySQL Community Server downloads](https://dev.mysql.com/downloads/mysql/) are at the link.

Of course, all of this may be different if you are running a Linux-based server or writing an interface with a different front-end such as PHP or ASP.NET. I used VBScript in Classic ASP because that’s all we had at the beginning and because it does everything I need. I built a library of handy functions in dbpub/ functions1.asp which enhance the language.

**MySQL parameters**

The schemas are large, and you will need plenty of RAM, otherwise you’ll get severe performance issues swapping data to and from disk – particularly if it is HDD rather than SSD. Our Master machine has 32GB of RAM.

We use only InnoDB engine, not myISAM.

You will also need certain settings in your mySQL.ini file for our database to run correctly:

div\_precision\_increment = 8

(needed for accurate total returns adjustments)

character-set-server = utf8mb4

(needed to handle multibyte Chinese characters)

collation-server = utf8mb4\_unicode\_ci

Some of our tables have full-text indices (for example, organisations name1 and people name1, name2). For these to work correctly, we need to alter the minimum token size so that short Chinese names like Au are indexed properly:

innodb\_ft\_min\_token\_size = 2

innodb\_ft\_enable\_stopword = 0

And for the MySQL roles granted to users, we need:

activate\_all\_roles\_on\_login

As guidance only, our performance options include:

innodb\_flush\_log\_at\_trx\_commit = 0

innodb\_log\_buffer\_size=1M

innodb\_buffer\_pool\_size = 20G (set this as high as possible, up to 80% of your machine’s RAM)

innodb\_redo\_log\_capacity = 2G

innodb\_thread\_concurrency=17

innodb\_autoextend\_increment=64

innodb\_buffer\_pool\_instances=8

innodb\_concurrency\_tickets=5000

innodb\_open\_files=300

innodb\_file\_per\_table=1

sort\_buffer\_size=256K

**Restoring the schema**

Some schema tales have Triggers which alter data as it is entered or loaded. Others have timestamp columns which automatically update when a row is changed.

To stop MySQL from overwriting timestamps and running triggers, I have split the backups of the Enigma schema into:

* Structure
* Data
* Triggers

You should restore them in that order, so that the Triggers are loaded last and don’t alter any of the existing data.

*Definer problem*

The schemas contain Stored Procedures, Functions and Views, most of which were “defined” by user ‘David’ (me), not by “root”, as that superuser can only be used on the server directly.

So before you load the schema, you need to create a user “David” in MySQL, otherwise you’ll get unknown user errors. In Workbench, create the user and go to “Global Privileges” and tick everything. That includes “Create View”, “Create Routine” etc.

After restoration you can of course create your own super-user and give them similar powers, and change the definers on each function, view or stored procedure – a lot of work though? I haven’t tried it.

**Additional schema on the Webb-site server**

There are 2 schema which reside only on the Webb-site server, not on the master. These schema deal with user-accounts, login, user stock-lists, user-opinion polling and throttling scrapers with captchas. For privacy reasons I am only dumping the structure and not the data of these.

They are:

* Mailvote – see “Notes on the Webb-site login system” in the “Mailvote schema and login system” folder. You will need this schema to set and retrieve variables in the mailvote.keys system retried by function getKey in functions1.asp. Those include the login details for your mail server, site Captcha keys (see iplog below) and the domain of your MasterHost (or the server if you are using a single-server setup).
* Iplog – see the “Notes on iplog schema” in the “iplog schema” folder. This schema stores the 4-byte IP addresses, latest hit and number of hits of users when visiting certain pages in which the “botchk” function is used.

**Additional schema on the Master server**

There is a schema called “private” with a single table “keys” which contains values needed by the various VB.net programs, including connection data for sending error e-mails when data collection fails, keys for the UK API, and an important keystring “wskey” which is used to encrypt/decrypt MySQL passwords in the wsroles table for editing the database. See the “Webb-site editing system” folder.

We have included a structure dump for the schema, and a “master private keys.csv” which contains the names and descriptions of each key (but not the values, of course).

**DSNs**

You will need to set up Data Source Names (**DSNs**) to connect to the database. In Windows, there are two types – User DSNs on each PC for use by Microsoft Access (or other Windows applications) and System DSNs used by IIS to run the website.

In our DSNs, we use MySQL ODBC 8.4.0 Unicode Driver, the last in the 8.x series. If you try later versions, then your mileage may vary, and things may break. Over the years, various bugs particularly in the handling of multibyte Unicode characters (including Chinese) caused us problems. The download page [is here](https://downloads.mysql.com/archives/c-odbc/).

*Webb-site public server (slave/replica)*

On the Webb-site server you will need four 64-bit “System DSNs” and a MySQL user called “Web”. GRANT Web the privileges shown in the “Web grants.csv” file in the Slave folder.

Set up the DSNs using Control Panel/ Administrative Tools/ ODBC Data Source Administrator (64-bit).

enigmaMySQL – this retrieves data from the enigma schema for the site.

* TCP/IP Sever: blank, or localhost
* User: Web
* Database: enigma

CCASSserver – this retrieves data from the ccass schema for the site.

* Details same as CCASSserver, except for Database: enigma

conAuto – this is used by the online editing pages, to connect to the master server and make certain changes without disturbing the userID fields which are normally set by triggers:

* TCP/IP server: the domain of your master (or localhost if you are not using replication).
* User: auto
* Database: enigma
* In the Cursors/Results tab, tick “Return matched rows instead of affected rows”. You will need this setting with any DSN which makes changes to the data.

mailvote – this is used by the login system, polling system and iplog.

* TCP/IP server: blank or localhost
* User: Web
* Database: mailvote
* Cursor/results: tick “Return matched rows instead of affected rows”

*Editor PCs on the LAN*

Each user on a PC will need user DSNs in order to use our Access front end application (enigma.accdb). Our Access application needs following user DSNs:

CCASS – to fetch/edit data from the CCASS schema

* User: <the local user’s MySQL account – in our case David or Cynthia>
* Database: ccass
* TCIP/IP server: <the LAN IP of your master server, or its Windows name>
* Cursor/results: tick “Return matched rows instead of affected rows”

EnigmaMySQL – to fetch/edit data from the enigma schema

* Details same as CCASS, except for Database: enigma

*Master server*

For our various VB.Net programs to edit the database, we need the following User DSNs:

CCASS

* User: David (or your master user)
* TCP/IP server: blank or localhost
* Database: ccass
* Cursor/results: tick “Return matched rows instead of affected rows”

enigmaMySQL

* Details same as CCASS, except for Database: enigma

**Master server users**

You will need to set up a MySQL user called “auto”. GRANT this the privileges shown in the “auto grants.csv” file.

For the editing pages on Webb-site, you will also need a bunch of other MySQL roles which are explained in a separate file “Notes on Webb-site editing system” in the “Webb-site editing system” folder. The GRANTs for each role are shown in CSV files in the “Master grants” subfolder.

**ASP settings**

In Windows Internet Information Services (IIS), you’ll need the ASP module and certain parameters:

* Enable Parent Paths: **True** (that’s because a lot of our includes use “../<directory>” when referencing between ccass and dbpub directories, for example.
* For the development server but not the public server, I find the following very useful under “Compilation/ Debugging Properties”: “Send Errors to Browser: **True”**

DMW

2025-04-30