



IT Consulting
Custom Applications
Database Development
Networking

Servizi Integrati Professionali s.a.s.
di Crampar Maria Angela & C.
Via Don Luigi Sturzo, 14 52100 Arezzo – Italy
Tel. +39-0575-324209 Fax +39-0575-354967
Email: info@sintpro.com Web: www.sintpro.com
Partita IVA: 0145311051

Servoy & MySQL Mini HowTo

This is not meant to be a full tutorial on installing and administering MySQL but a short guide to get you started to make Servoy and MySQL live happily together.

Why MySQL?

Why using MySQL when Servoy ships with the excellent DB server from Sybase bundled?

- 1) Because it's possible. (my favorite one!);
- 2) Because you or your customer already have a MySQL database running;
- 3) Because you need to access the database from applications other than Servoy and the bundled Sybase license does not allow to do this;
- 4) Because you want to access the same data that is published through Apache/PHP on your own or your customer's website;
- 5) Because otherwise the Servoy Forum would be a boring place.

Anyways, spend five minutes of your spare time reading the MySQL License before deciding what DB Server you are going to build your Revolutionary App™ on... it is an unusual license and a lot of people are still uncertain if it's a free software or not. You decide.

Ingredients

- MySQL server and Client
- MySQL Connector/J
- MySQL Administrator
- A Grain Of Salt™

Versions

I recommend to use MySQL version 4.1.x with InnoDB tables to have transactions support and the 3.1.x version of the JDBC driver which solves a lot of problems connecting to Servoy.

Don't be eager to upgrade to the newly released versions! Start with the current stable one and keep an eye on the change log when new versions come out, upgrade only when you see a bug fix that may affect you or a new feature that you really need.

Installation of MySQL Server

I assume you can install the MySQL Server and Client binaries on your own following the installation instructions for your platform, otherwise google for a specific Tutorial or HowTo, there are plenty of those flying around the net so I won't write another one.

The storage engine

MySQL is a really fast DB server and it's very popular among web sites developers mostly because it uses a really fast storage engine called **MyISAM**.

MyISAM is great but has a big disadvantage: **it doesn't support transactions.**

Transactions are really a must have when you work with Servoy so you surely want to use the **InnoDB** storage engine at least for the tables where you plan to use transactions on and for the Servoy Repository tables if you plan to store them in the MySQL backend.

Configuration of MySQL Server

If you are connecting to a hosted MySQL server read the chapter called "How to Choose an hosting service" otherwise if you are not read on.



IT Consulting
Custom Applications
Database Development
Networking

Servizi Integrati Professionali s.a.s.
di Crampar Maria Angela & C.
Via Don Luigi Sturzo, 14 52100 Arezzo – Italy
Tel. +39-0575-324209 Fax +39-0575-354967
Email: info@sintpro.com Web: www.sintpro.com
Partita IVA: 0145311051

I'm not going to fully explain how to configure the DB Server but I will point you in the right direction with the settings specifically needed for a good Servoy-MySQL integration.

If you don't like working with the Command Line Interface, I strongly suggest to download the MySQL Administrator Utility for the configuration and maintenance tasks.

MySQL configuration is stored in a text file called *my.cnf*, usually if you have a typical server configuration the my.cnf file will be located in the */etc* directory (sorry Windows users...), you can find some template files in the *support-files* directory of your MySQL installation (*my-small.cnf*, *my-medium.cnf*, etc.), you can copy one of those files to the */etc* directory and rename it my.cnf.

The config file is divided into sections, the ones you really need to edit are:

```
# The MySQL Server
```

In this section are stored the main settings for the server, add or uncomment (remove the trailing #) the following lines:

```
default-storage-engine=innodb
```

this will set the InnoDB storage engine as the default engine and is quite useful if sometimes you create a table on the fly from within Servoy Developer. If not specified the storage engine defaults to MyISAM.

```
default-character-set=utf8
```

this line sets the default character set to UTF format, I advise to use UTF as default character set, change it only if you know what you're doing.

```
log-bin
```

this turns on the binary logging feature and is needed for InnoDB tables support and useful in case of disaster recovery.

```
innodb_data_home_dir = /usr/local/mysql/data/  
innodb_data_file_path = ibdata1:10M:autoextend  
innodb_log_group_home_dir = /usr/local/mysql/data/  
innodb_log_arch_dir = /usr/local/mysql/data/  
# You can set ..buffer_pool_size up to 50 - 80 %  
# of RAM but beware of setting memory usage too high  
innodb_buffer_pool_size = 16M  
innodb_additional_mem_pool_size = 2M  
# Set ..log_file_size to 25 % of buffer pool size  
innodb_log_file_size = 5M  
innodb_log_buffer_size = 8M  
innodb_flush_log_at_trx_commit = 1  
innodb_lock_wait_timeout = 50
```

These lines are used to configure the InnoDB storage engine, double check the paths and make sure they reflect your MySQL installation structure.

If you use the MySQL Administrator utility for editing the configuration file follow the same guidelines



IT Consulting
Custom Applications
Database Development
Networking

Servizi Integrati Professionali s.a.s.
di Crampar Maria Angela & C.
Via Don Luigi Sturzo, 14 52100 Arezzo – Italy
Tel. +39-0575-324209 Fax +39-0575-354967
Email: info@sintpro.com Web: www.sintpro.com
Partita IVA: 0145311051

using the graphical interface.

Access Security

MySQL security system is quite complicated for a novice user but using the GRANT command in MySQL client or your favorite query browser is quite easy to setup a basic security system for the Servoy Server access to the Database.

First thing firsts: **I strongly advice not to connect to MySQL from Servoy Server using the root user**, remember that:

- passwords travel across the LAN (or even the Net!) in clear text when Servoy authenticates with MySQL;
- the passwords for the data sources are stored by Servoy in the servoy.properties files in clear text;
- it's usually a good security measure to allow root access to MySQL only from the localhost so if you are running Servoy Server on a different machine you'll have to change root's permissions;
- it's easier to maintain and administer the DB server when you see what sessions belong to Servoy and what sessions belong to the root user;

So the easiest way to have a tighter security is to create a MySQL user (let's call it 'servoy') and connect only using this special user; this user will need to have privileges to access and manage the databases that you use as Servoy data sources and should be able to connect only from the machine where the Servoy Server application runs.

To create this user you simply have to run this query in MySQL Client or in your query browser:

```
GRANT ALL ON database_name.* TO 'servoy'@'Servoy_IP_Address' identified by 'password';
```

where:

database_name = the name of the database you're setting privileges for;

servoy = the username to use in Servoy Server;

Servoy_IP_Address = the IP Address or the name of the machine where the Servoy Server runs;

password = the password you want to assign to the user.

Example:

```
GRANT ALL ON my_really_big_database.* to 'servoy'@'localhost' identified by 'ZZwtp73Spp';
```

if your Servoy Server runs on the same machine as the MySQL backend, or:

```
GRANT ALL ON my_really_big_database.* to 'servoy'@'192.168.0.2' identified by 'ZZwtp73Spp';
```

if your Servoy Server runs on the machine with the 192.168.0.2 IP Address.

Repeat this procedure for all the database you need to grant the access to, including the Servoy Repository database if you store it on MySQL.

Remember that if you access the databases with Servoy Developer and Servoy Server from different machines you have to issue two grants commands, one for the first machine the other for the second machine, or you can use a wildcard as '192.168.0.%' to permit access from the whole 192.168.0.0/24



IT Consulting
Custom Applications
Database Development
Networking

Servizi Integrati Professionali s.a.s.
di Crampar Maria Angela & C.
Via Don Luigi Sturzo, 14 52100 Arezzo – Italy
Tel. +39-0575-324209 Fax +39-0575-354967
Email: info@sintpro.com Web: www.sintpro.com
Partita IVA: 0145311051

subnet. Be careful with wildcards in permissions!! Remember: RTFM!!

JDBC Driver

The MySQL driver shipping with Servoy has some serious bugs but Servoy is unable to ship the official driver for licensing reasons, so download the MySQL Connector/J that is marked stable from the MySQL official site.

Unpack the MySQL Connector/J and copy the file named **mysql-connector-java-x.x.x-bin.jar** to your **Servoy/drivers** folder, you can optionally delete the original MySQL driver shipped with Servoy to avoid using it by mistake.

Configuring Servoy DB Servers

Open the Server Config tab from Servoy Preferences and from the "Auto Setup" menu select "MySQL", in the Server Config window specify the various parameters:

- **Server Name:** give a name to this Server Connection;
- **User name:** the username you have decided the Servoy Server should use to connect to MySQL, in the previous example we have already used "servoy" ;
- **Password:** the password for the specified user, in the previous example we used "ZZwtp73Spp";
- **Database Server URL:** `jdbc:mysql://<mysql_server_ip_address>/<database_name>` , following the previous example and assuming that Servoy Server is running on a different machine from MySQL Server: `jdbc:mysql://192.168.0.2/my_really_big_database;`
- **Driver:** select the `com.mysql.jdbc.Driver` from the list; keep in mind that if you select the `org.gjt.mm.mysql.jdbc.Driver` you use the standard driver that ships with Servoy, don't blame me if you run into problems!
- **Catalog:** <none> ;
- **Schema:** <none> ;
- **Max Active:** leave the default value unless you know what you're doing;
- **Max Idle:** leave the default value unless you know what you're doing;
- **Validation Type:** query validation ;
- **Validation Query:** `select 1 ;`

A special note is needed for the last two settings: Servoy Server opens some JDBC connections to the Database Server according to the "Max Active" and "Max Idle" settings, every time a Client needs to exchange data with the DB Server Servoy Server chooses an already open connection from the connections pool but it needs to validate it to see if the connection is still alive and working as expected; the fastest way to validate the connection is the "Exception Validation", this technique is quite easy: try to use this connection and see if the DB Server complains, if it does not, the connection is working fine (Sorry Servoy Team, I needed to be brief...).

Unfortunately the MySQL JDBC Driver still has (at the time I'm writing this Tutorial) a nasty bug: idle connections are dropped after several hours of inactivity without any notification and every time Servoy Server tries to validate an expired connection bad things happen (I said I needed to be brief...) so you should instead use the "Query Validation" option; query validation is simple too: throw a fast and easy query to the backend (`select 1` is perfect) and see if you get a result, if the result is received the connection is validated otherwise another connection from the pool is tested until one is working fine or a new one is established.

The idle timeout time is quite large (8 hours) so if you are using Servoy Developer and you open it in



IT Consulting
Custom Applications
Database Development
Networking

Servizi Integrati Professionali s.a.s.
di Crampar Maria Angela & C.
Via Don Luigi Sturzo, 14 52100 Arezzo – Italy
Tel. +39-0575-324209 Fax +39-0575-354967
Email: info@sintpro.com Web: www.sintpro.com
Partita IVA: 0145311051

the morning and close it at night (as I do...) you shouldn't notice any problems with Exception Validation (unless you only pretend to develop but play Doom™ or Tetris all day long...), problems arise when you install a server for a customer and it remains idle all night long (my customers tend to sleep at night, I'm investigating this issue and will report in "Issues and Bugs" Forum ASAP), in the morning you will find errors in the logs and angry customers calls in your voice mail. Use "Query Validation" and use a fast query as "select 1", you will find unnoticeable performance penalty and your customers will be happy, in the meanwhile write an email to the development team of MySQL Connector/J and ask them to fix this funny™ bug.

The Repository

There is no problem to even keep Servoy Repository in the MySQL backend, just create a database (I called mine: `servoy_repository`, smart, isn't it?), set the appropriate permissions for the "servoy" user and be sure to set the default table engine to InnoDB so that all your Repository tables will support transactions, modify the Server Config preferences for your Repository connection and let Servoy recreate the Repository tables for you.

Backups

Someone "really smart" said a long time ago: "Better safe than Sorry" and I guess that if he registered a trademark on this four simple words now he could have been a really happy camper. I bet you get the point.

The easiest way to backup MySQL databases is to "dump" the data to a text SQL file, if you need to restore you simply run your SQL against an empty database and voilà! everything should be fine. You have two options to create a SQL dump:

- 1) Use the MySQL Administrator graphical utility;
- 2) Use the CLI `mysqldump` utility;

Option 1 is really easy, RTM.

Option 2 is more appealing because you can create a nice script to feed to `cron` and let him dump your data for you during the night or whatever.

A basic syntax of `mysqldump` keeping in mind the previously used examples is the following:

```
mysqldump --user=servoy --password=ZZwtp73Spp --create-options --extended-insert --quick --single-transaction --allow-keywords my_really_big_database | gzip > my_really_big_database_backup.sql.gz
```

This will backup the entire `my_really_big_database` database with all its tables, structure and data. You can have a look in the MySQL online Manual if you want to know what every single option means but notice two things:

- 1) `--single-transaction` option is really important if you are using InnoDB tables (and you should!) cause it prevents data corruption encapsulating the whole backup operation in a transaction thus preventing a related table to change after a main table is being backed up and vice versa;
- 2) the piping trough `gunzip` (`| gzip > my_really_big_database_backup.sql.gz`) shrinks the file size and wipes out the need for an after backup compression.



IT Consulting
Custom Applications
Database Development
Networking

Servizi Integrati Professionali s.a.s.
di Crampar Maria Angela & C.
Via Don Luigi Sturzo, 14 52100 Arezzo – Italy
Tel. +39-0575-324209 Fax +39-0575-354967
Email: info@sintpro.com Web: www.sintpro.com
Partita IVA: 0145311051

mysql_backup.sh

This is a shell script I wrote to automate database backups, it can be used in a Linux or MacOSX system. It's really simple, you can schedule it to be executed by CRON each day and it saves two copies of each selected database in a directory of your choice for each day of the week. For each database it saves a "structure only" backup (useful to move the db structure to another server) and a "structure and data" backup, it pipes the backup files through gunzip to save disk space and allows you to keep seven days of backups before overwriting the older one. You only need to edit the script and customize the options. Instructions inside.

It's freeware and you can download the latest version of the script from here:

http://www.sintpro.com/downloads/mysql_backup.sh

MySQL Tools

From the official MySQL site you can download several graphical utilities to manage and use MySQL, I strongly recommend the following two amongst the others:

- MySQL Administrator (MacOSX/Win/Linux): a GUI app to setup and manage a MySQL server, useful to manage databases, users privileges, tables and backups;
- MySQL Query Browser (MacOSX/Win/Linux): not the best Query Tool but quite useful;

Take also a look at these applications:

- Aqua Data Studio (Java): the best query tool out there (IMHO);
- phpMyAdmin (PHP based): easy interface to MySQL from your Web Server;

Conclusions

I hope this mini HowTo will help you to get Servoy and MySQL live together, if you need further assistance or help remember that "Google is your Friend".

If you want to report typos, ask questions or just say "Well Done! Keep the good stuff coming!" you can reach me at nicola@sintpro.com.

You can download the latest version of this guide from this link:

<http://www.sintpro.com/downloads/Servoy-MySQL-miniHowTo.pdf>

Useful Resources

MySQL Official Site: <http://www.mysql.com>

MySQL Reference Manual: <http://dev.mysql.com/doc>

MySQL Developers Site: <http://dev.mysql.com>

Servoy Official Site: <http://www.servoy.com>

Servoy magazine: <http://www.servoymagazine.com>

Servoy Developer Forum: <http://forum.servoy.com>

Last Revision: 01-Feb-2006

Nicola Gervasi

Servizi Integrati Professionali s.a.s.

nicola@sintpro.com

<http://www.sintpro.com>